Accessible eGovernment services – developing a parental benefit calculator with improved usability

by

Peter Le

Master's thesis
in
Information and Communication Technology

Agder University College
Faculty of Engineering and Science

Grimstad, 29 May 2007

Keywords: eGovernment, web usability and accessibility, Open Sources and Open Standards, Ruby on Rails, MySQL

Abstract

An online parental benefit calculator offered earlier by the Norwegian Labour and Welfare Organisation (NAV) gave only simple estimates on how long a mother or father could take the parental leave and did not support assessment of scenarios allowing for combination of the leave with a part-time job, the so-called “graded withdrawal”-scenarios. To work out such scenarios, the parents-to-be had either to get acquainted with the regulations and work out the various scenarios manually or contact the local social security office for assistance.

We present a prototype of a new version of the parental benefit calculator. The new version provides an enhanced functionality, supporting the evaluations of the different “graded withdrawal”-scenarios among others. Our prototype offers also a support for monitoring of the use of the calculator. The developed database will collect data, allowing NAV to analyze in detail how the online calculator is used. These data are likely to provide a valuable insight into whether or not the provided calculator services meet its objectives and how it may be improved. We also propose a new graphical user, providing a more intuitive presentation of the possible parental benefit plans. Additionally, it is developed using Open Sources Software and in adherence with Open Standards to guarantee easy of further development and maintenance.
The prototype has also been developed adhering to usability and accessibility guidelines. Ensuring high usability and full accessibility is vital, especially in case of public services. The goals of eGovernment are to provide better, more effective services.

The developmental work was conducted using RAD (Rapid Application Development) approach, which involves development of applications through iterative process of prototyping and implementation. The initial prototypes of the graphical user interface were sketched on the paper rather than coded. The paper-prototyping approach is useful for quick exploration and communication of various ideas. The subsequent prototype versions were tested using heuristic evaluations and against accessibility guidelines.

The purpose of the prototype is to let the government offer an accessible service on the Internet to the citizens which is more usable for everyone, even for people with disabilities. NAV will gain benefit of this by being able to offer a better public service on the Internet which will lead to a substantial decrease in face-to-face consultations that currently need to be conducted by the NAV officers.

Our prototype demonstrates how the eGovernment solution can be improved both in terms of its utility and its technical design. It also demonstrates how the solution is made more efficient when taking both the accessibility and usability aspects explicitly into account.

Preface

This thesis is an accomplishment as the final assignment of the master degree in Information and Communication Technology at Agder University College, Faculty of Engineering and Science located at Grooseveien in Grimstad.

The length of the study presented in this thesis is 5 months, from January to May 2007. This assignment has been carried out in cooperation with NAV Trygd Grimstad, and has been supervised by Mikael Snaprud and Agata Sawicka.
I would like to use this opportunity to thank them, because I am very grateful for their excellent guidance and support during the whole project period.

Grimstad, May 2007

____

Peter Le

Table of Contents

Figure List

Table List

1 Introduction

This chapter gives an overview of my thesis. The first section presents the background and the second section will describe the problem area. Section 1.3 describes shortly about the motivation for the research and its purpose, and in the next section a short summary of the results is presented. The last section presents the outline for this thesis.

1.1 Background

The background for this thesis are eGovernment[1] services deployed with high usability and accessibility, and developed in adherence with Open Sources and Open Standards. eGovernment refers to government's use of information and communication technology to exchange information and services with citizens, businesses and other arms of government.

Norway is among the most best in Europe regarding the development and use of ICT[2]. The country is "growing" on broadband access, the citizens' utilization of new technology and the use of public electronic services[3]. The government wishes to have the best public sector in the whole world, and the active use of ICT and development of new electronic services are important factors for reaching this goal[4]. The public electronic services shall be the proper channel towards the public sector, and therefore is the government demanding that the enterprises must collaborate and establish more and better electronic services with around-the-clock work.

Since the eGovernment services are offered to the public, it is important that these services are usable and accessibly by everybody. In accordance with the national Norwegian eNorge 2009 plan[5] and the pan-European 2010 initiative[6], public services should be made accessible to all.
The government will have the possibility to utilize the resources on an efficient way by the reuse of solutions for public electronic services. By using Open Sources and Open Standards for development of these services it will also become easier to maintain, further develop the services and let others utilize the solutions[7].

Although the growth of the new electronic services is an important factor for moving towards a digital society, the goals of eGovernment services have to be kept. The new services have to be better and more efficient, and therefore the development of these services has to be in adherence to not only Open Sources and Open Standards, but also to usability and accessibility guidelines.

1.2 Problem Area

The assignment is given by NAV[8], the Norwegian Labour and Welfare Organisation, in collaboration with Agder University College. NAV was established on 1. July 2006 and was

[1] eGovernment, see http://www.egov4dev.org/egovdefn.htm
[2] ICT – Information and Communication Technology
[3] http://www.regjeringen.no/nb/dep/fad/pressesenter/pressemeldinger/2007/Norge-blant-de-beste-i-Europa-pa-IKT.html?id=462142
[4] http://www.regjeringen.no/nb/dep/fad/dep/Fornyings-_og_administrasjonsminister/taler_artikler/2007/IKT-gir-betre-offentleg-sektor.html?id=462722
[5] http://www.regjeringen.no/nb/dep/fad/Tema/IT-politikk__eNorge/eNorge-2009.html?id=439499
[6] http://ec.europa.eu/information_society/eeurope/i2010/index_en.htm
[7] http://www.regjeringen.no/nb/dep/fad/pressesenter/pressemeldinger/2007/MinSide-blir-fri-programvare.html?id=463325
[8] NAV, see http://www.nav.no

established as a result of the comprehensive welfare reform. The reform concerns 16000 employees and more than 2 million users. NAV is a merger of 3 former organizations, which are:

- The National Insurance organization
- The National Employment service
- The Social Welfare System

The goals of the merger were to get more people at work, a user friendly and oriented system, and one coordinated efficient and welfare administration. To achieve these goals it is necessary for the municipal and state organisations to find a proper way to interact locally.

NAV's services shall be offered as self-service where possible, and when these services are offered on the Internet and by telephone, they are more accessible for the users. This assignment concerns one of these public services.
In Norway the parental leave can be combined with a part-time job through the so-called "graded withdrawal" arrangements. The "graded-withdrawal" arrangement allows a parent to return to job part-time while still being on parental leave. An online parental benefit calculator offered earlier by NAV gave only simple estimates on how long a mother or father could take the parental leave and did not support assessment of scenarios allowing for combination of the leave with a part-time job. To work out such scenarios, the parents-to-be had either to get acquainted with the regulations and work out the various scenarios manually or contact the local social security office for assistance.

The assignment involves a prototype development of a new version of the parental benefit calculator. The new version must provide an enhanced functionality, supporting the evaluations of the different "graded withdrawal"-scenarios among others. It should also propose a new graphical user interface, providing a more intuitive presentation of the possible parental benefit plans. Additionally, it will be developed using Open Sources Software and in adherence with Open Standards to guarantee easy of further development and maintenance.
The purpose of the prototype is to let the government offer an accessible service on the Internet to the citizens. If the research turns out to be successful and the final solution is satisfying, this may lead to the fact that the prototype will be taken in use by NAV.
NAV will then gain benefit of this by being able to offer a public service on the Internet, and maybe the local social security offices will experience less influx by people that need assistance with combining the parental leave with a part-time job.

1.3 Research Motivation and Purpose

This research aims at studying eGovernment and working with Open Sources and Standards, and web usability and accessibility. It will illustrate how a public service offered by the government to the citizens, which means all kinds of target groups, can be designed and developed as a web application using Open Source tools and technologies, and still retain high quality and provide high usability and accessibility by using the results of evaluations and follow guidelines.

This research will give me the insight to the working life, and let me acquire knowledge about existing technologies and methods. We will also experience using Open Sources and Standards to develop a high quality web application with high web usability and accessibility for public use.

1.4 Short summary of results

The result of this project is a new prototype version of the parental benefit calculator. This prototype has enhanced functionality and an improved design. The design of the prototype has been supported by an initial evaluation interview, heuristic evaluations, structured interview with NAV, feedbacks from the caseworkers at NAV, and user testing with real users. The results from these can be found in chapter 5 on page 45. These interviews and evaluations also aided the specification of the requirements for the new prototype. The requirements can be found in chapter 5.4 on page 48.

From these requirements we designed the new prototype and implemented it. The design specification can be found in chapter 6 on page 52, and the implementation of our new prototype is located in chapter 7 on page 65. Our work has been iterative which means that our prototype has been designed, developed, and tested in several phases during the project period.

The prototype was developed based on the specified requirements and in conformance to WCAG[9]. We used accessibility checker tools for ensuring that the prototype is going to be accessible by everyone. The description of these tools can be found in chapter 4.4.2 on page 38. We corrected the most important accessibility problems in our prototype. The results from can be found in chapter 8.5 on page 102.

We also used these accessibility tools to check the current solution that is offered by NAV, located at http://tjenester.nav.no/fodselspenger/fpenger/fpenger1.do. We discovered that the web pages contained a lot of accessibility problems. The testing of NAV's current solution can be seen in chapter 8.6 on page 109.

Our prototype has also been tested with real users. The test users discovered new problems with the design and the functionality that we didn't discover in the design, development and testing phases. The results from the user testing can be found in chapter 8.2 on page 93.
We corrected these problems after the usability testing. We also used the accessibility checker tools to check our prototype for accessibility problems and corrected those problems.
Our parental benefit calculator prototype is located at http://128.39.61.95:3000/.

In addition to be an accessible solution, our prototype was also designed and developed to have improved usability.

- It is supporting calculations of the parental benefit in combination with a part-time job
- It can present the parental leave in a graphical calendar
- It is able to show the parental benefit for each week, month and year
- It has a database which can be used to monitor the use of the calculator
- It collects data that can be used to compile statistics
- It has the functionality for displaying the text in different languages
- It has the functionality for switching between different page layout styles
- It has the functionality for increasing and decreasing font size
- It is designed to support a login system
- It is designed to be easy to maintain and further develop

[9] WCAG – Web Content Accessibility Guidelines – For more information, visit http://www.w3.org/WAI/

1.5 Thesis Outline

Chapter 1 briefly introduce the reader to eGovernment services, web usability and accessibility together with Open Sources and Open Standards which are the domain of our work. It will also give a description of NAV and present the specific problem for this research.

Chapter 2 presents the information from our literature study. It summarizes and synthesizes the knowledge and ideas of other regarding eGovernment, web usability and accessibility and Open Sources and Open Standards. A summary at the end of this chapter will clearly illustrate how these topics are related to each other.

Chapter 3 presents the research problem. It will first present the current solution together with its problems, and then present the new solution with a solution for the discovered issues in the current solution. It will also present the scope of this research, which assumptions and limitations we made and some critical resources for this research.

Chapter 4 describes all the research approaches used in this research. It will describe how the different approaches are utilized and applied to the different phases for solving the research problem; evaluation and test phases, design phase and development phase. It will also present the various tools and technologies that are used, together with examples.

Chapter 5 presents the solution and results. It will describe how the results were obtained by the use of the different approaches, tools and technologies. The analytical work performed in this project will be presented in this chapter. It also presents all the requirements for the new solution of the parental benefit calculator; the functional requirements, the external interface requirements, the non-functional requirements and other requirements as well.

Chapter 6 presents the specification of the design of the parental benefit calculator prototype. It will describe the architecture of the solution, the functionality, the graphical user interface, the usage and behaviour of the solution, and the design of the database.

Chapter 7 is presenting the implementation of the design of solution. It will describe how the solution is implemented.

Chapter 8 describes the validation and testing of the implementation. It will test and validate that the implementation has been correctly done, and describe how the requirements for the solution are satisfied. This chapter will also present how the prototype is developed in conformance to accessibility guidelines for becoming an accessible service. In addition, all the other tests will also presented such as user testing with real users, heuristic evaluation of NAV’s calculator, testing in regards to W3C recommendations, and accessibility testing of NAV’s calculator.

Chapter 9 will discuss the results achieved and the solutions that were found for this research. This chapter will also discuss how the methodologies and technologies that were used, and what other alternatives that were seen. It also presents what we learned, how well the solution solves the problem in this research, what that could be done differently if we were to start again.

Chapter 10 is presenting the conclusion for this research. It summarizes the results and present what the contributions are of this research, and a proposal for future work.

2 Background/Literature Review

Following sections will discuss the research that has been done on the subject. This chapter presents works and refers to surveys carried out by others.

As mentioned earlier, the definition of eGovernment is the use of ICT to facilitate the activities of public sector organisations. Since eGovernment services are offered to the public and the target groups could be any kind users, it is important that such services must be both accessible and usable by everyone.
Web usability and accessibility are therefore important factors for an eGovernment service, since the aim of these services is that the public will use be able to use it to utilize them. But as with every single service, these services also need maintenance, upgrades or to be further developed for satisfying new needs. Open Sources and Open Standards can contribute to easy maintenance and further development, by making the eGovernment services' solutions public for everybody. Anyone in the society will have the opportunity to contribute in further developments of the services and new services will be more easily developed since existing solutions may be reused.

2.1 eGovernment

In Norway, the government is currently working with a plan called eNorge 2009 which has 3 target areas for the IT-politics according to (Meyer, 2005):

- The individual in the digitalized Norway
- Innovation and growth in Norwegian economic life
- One coordinated and user friendly public sector

The goal is to achieve a society where anybody can join. It should be made demands to manufacturers and suppliers regarding universal design of ICT tools.
ICT is making public services more efficient and is increasing the productivity in the country according to (St.meld. nr. 17, 2006-2007). It also states that digital expertise among the population, and good public electronic services are the main elements for Norway to still be the world's best country to live in. The government wishes to utilize the information technology to give the citizens and the businesses a simpler usual weekday. Examples of elements to achieve better public services are:

- Open Sources and Open Standards
- One common ICT architecture
- Cooperation between the private and public sectors to increase the level of expertise within the population

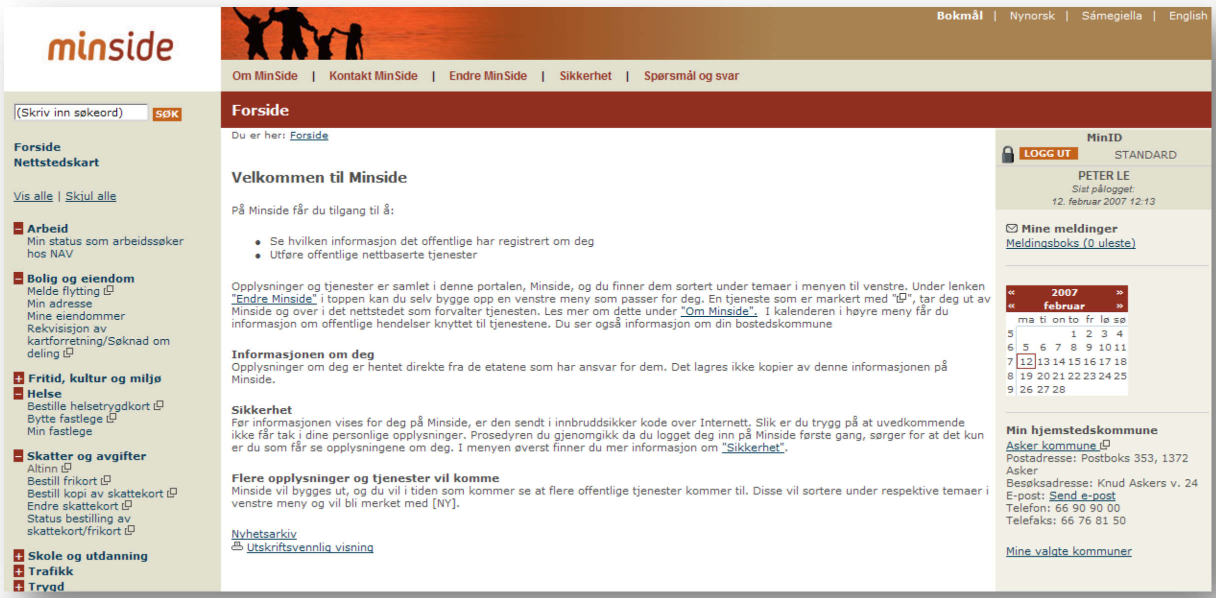
In order to provide an eGovernment for everybody in Norway, it has been established a site called "MinSide".

norge.no

minside

Figure 1: www.norge.no/minside

The web site “MinSide” is a personal customized page containing all the public service offerings and web based information.
The purpose of this web site is to give the citizens one common access point to all public electronic services, an opportunity to a simple dialogue with the public sector and an overview of what information the different agencies have on one individual. The citizens are able to get in touch with the public sector at anytime and anywhere, as long as they have an internet connection.


minside
Bokmål | Nynorsk | Sámegiella | English
Om MinSide | Kontakt MinSide | Endre MinSide | Sikkerhet | Spørsmål og svar
(Skriv inn søkeord)
SØK
Forside
Nettstedskart
Vis alle | Skjul alle

Arbeid
Min status som arbeidssøker hos NAV

Bolig og eiendom
Melde flytting
Min adresse
Mine eiendommer
Rekvisisjon av kartforretning/Søknad om deling

Fritid, kultur og miljø

Helse
Bestille helsetrygdkort
Bytte fastlege
Min fastlege

Skatter og avgifter
Altinn
Bestill frikort
Bestill kopi av skattekort
Endre skattekort
Status bestilling av skattekort/frikort

Skole og utdanning
Trafikk
Trygd

Forside

Du er her: Forside

Velkommen til Minside

På Minside får du tilgang til å:

- Se hvilken informasjon det offentlige har registrert om deg
- Utføre offentlige nettbaserte tjenester

Opplysninger og tjenester er samlet i denne portalen, Minside, og du finner dem sortert under temaer i menyen til venstre. Under lenken "Endre Minside" i toppen kan du selv bygge opp en venstre meny som passer for deg. En tjeneste som er markert med "⧉", tar deg ut av Minside og over i det nettstedet som forvalter tjenesten. Les mer om dette under "Om Minside". I kalenderen i høyre meny får du informasjon om offentlige hendelser knyttet til tjenestene. Du ser også informasjon om din bostedskommune

Informasjonen om deg
Opplysninger om deg er hentet direkte fra de etatene som har ansvar for dem. Det lagres ikke kopier av denne informasjonen på Minside.

Sikkerhet
Før informasjonen vises for deg på Minside, er den sendt i innbruddsikker kode over Internett. Slik er du trygg på at uvedkommende ikke får tak i dine personlige opplysninger. Prosedyren du gjenomgikk da du logget deg inn på Minside første gang, sørger for at det kun er du som får se opplysningene om deg. I menyen øverst finner du mer informasjon om "Sikkerhet".

Flere opplysninger og tjenester vil komme
Minside vil bygges ut, og du vil i tiden som kommer se at flere offentlige tjenester kommer til. Disse vil sortere under respektive temaer i venstre meny og vil bli merket med [NY].

Nyhetsarkiv
Utskriftsvennlig visning

MinID
LOGG UT STANDARD
PETER LE
Sist pålogget:
12. februar 2007 12:13

Mine meldinger
Meldingsboks (0 uleste)

« 2007 »
« februar »

	ma	ti	on	to	fr	lø	sø
5				1	2	3	4
6	5	6	7	8	9	10	11
7	12	13	14	15	16	17	18
8	19	20	21	22	23	24	25
9	26	27	28				

Min hjemstedskommune
Asker kommune
Postadresse: Postboks 353, 1372 Asker
Besøksadresse: Knud Askers v. 24
E-post: Send e-post
Telefon: 66 90 90 00
Telefaks: 66 76 81 50

Mine valgte kommuner

Figure 2: Screenshot of the “Minside” page

Figure 2 is illustrating which services a user may find at “Minside”. The electronic public services can be found on the left side of the page and can be taken in use by clicking on the specific service. The user will be redirected to the specific service in form of a pop window.
Examples of public electronic services a citizen can expect to find at www.norge.no/minside are:

- My status as job applicant at NAV
- Register relocation
- My estates
- Change fixed doctor
- Change tax card

- My student loan
- My vehicles
- Application to kindergartens

Only after 3 weeks after the launch of the portal the number of users has already passed 100.000. Not many Norwegian web sites can present that large amount of users in such a short period of time according to (Pressemelding Nr: 01/2007, 2007).

(Pressemelding Nr: 01/2007, 2007) also states that the challenge for the government is to offer more useful public services to the citizens. The goal is that all relevant electronic services from the government shall be made available on “MinSide”.

In order for these services to be useful to the citizens they have to be both usable and accessible. The services have to be development in a proper way so that everybody can use them, even people with disabilities.

2.2 Web Usability and Accessibility

2.2.1 General review

Since an eGovernment service is offered to the public, it is important that such a service must have quality which means that it must have high usability and accessibility.
The government emphasizes that Open Sources and Open Standards is the foundation for a more efficient interaction between the public sectors, so that they will be able to offer better public electronic services to the citizens according to (Pressemelding Nr. 41/2006, 2006).

Web usability and accessibility is about making a web site more usable and accessible for everyone. It is important that a web site must have these properties. Web accessibility ensures that everyone, even those with disabilities, is able to access a web site. No matter what browser a user uses, how much computer knowledge that user has or what internet connection speed the user has, the web site has to be accessible by those users. Web accessibility is about caring for the users’ needs and preferences so that they can find the information they are looking for at the web site.

A web site is not very useful if it is not accessible by everyone and even worse if it isn’t usable by everyone. Web usability ensures that everyone is able to navigate a web site as easy and flexible as possible.

Web usability and accessibility should concern every business that is offering services or products on the Internet. If their web site is poorly designed and the users can’t navigate through and use that web site, they will leave and maybe never visit the site again. A company will lose customers in quite a large scale over a longer period of time.

According to (Nielsen J. , Usability Engineering, 1993) usability is not just a single property of a user interface. It has multiple components, and those five components are:

1. Learn ability – The system should be easy to learn
2. Efficiency – The system should be efficient to use
3. Memory ability – The system should be easy to remember
4. Errors – The system should have a low rate of errors
5. Satisfaction – The system should be satisfying to use

The Web Accessibility Initiative (WAI[10]) is developing strategies and guidelines to help making the web accessible to people with disabilities. WAI has developed the Web Content Accessibility Guidelines (WCAG). These documents explain how the web content can be accessible to people with disabilities. They are intended for web content developers, authoring tool developers and accessibility evaluation tools developers etc.

These documents, the WCAG 1.0, have 14 guidelines that are general principles of accessible design. Each of them has one or more checkpoints which explain how a guideline is applied in a specific area. Each of these checkpoints is assigned a priority, ranging from 1 to 3.
With priority 1 a developer *must* satisfy the checkpoint. With priority 2 a developer *should* satisfy the checkpoint, and with priority 3 a developer *may* satisfy the checkpoint.

The documents have 3 conformance levels; A, AA and AAA. With conformance level A, all priority 1 checkpoints are satisfied. With conformance level AA, all priority 1 and 2 checkpoints are satisfied, and with conformance level AAA, all 3 priorities checkpoints are satisfied.
By following these guidelines when developing a public service to the citizens, the people with disabilities are also taken into account. The services will be in conformance with the guidelines and can be used by these people.

There has been developed several accessibility checker tools. These tools check web pages for accessibility problems and some of them correct the problems in conformance to the accessibility problems. Although they correct the accessibility problems, there are always problems that need human review in order to determine if a web site is fully accessible.

2.2.2 eGovernment evaluations

When an eGovernment service has been implemented it has to be evaluated to check if it is operating. The evaluations typically consist of discussions with users, user questionnaires and possibly observation of system use according to (Heeks, 2006, p. 256). The findings from these evaluations can then be compared to the system's requirements that were originally set.

A survey carried out by Lasse Bertzen[11] is consisting of evaluations of eGovernment web sites. The paper gave an overview of eGovernment and eDemocracy, the student assignments and some practical experience he had acquired. One of the lectures held was about evaluating eGovernment web sites. The lecture was about these web sites and how they could be evaluated. He stated that the guidelines that were used to evaluate eGovernment web sites were quite technical so that people without any experience in web programming would have some problems with evaluating.

Jyoti Chodrie, Gheorgita Ghinea and Vishanth Weearakkody have pulished a paper[12] concerning the evaluation of global eGovernment sites. In their research they used web diagnostic tools to evaluate web sites. The tools that they used include WebXact[13], Netmechanic[14], W3C's HTML validator[15] and vischeck[16]. By evaluating different web sites in different countries, this paper concluded that many governments are neglecting accessibility, quality and privacy criteria.

[10] WAI, http://www.w3.org/WAI/gettingstarted/Overview.html
[11] Lasse Bertzen's work can be found at this location, http://lasse.hive.no/EFO306/3SWSEGOV.pdf
[12] This paper can be found at http://www.ejeg.com/volume-2/volume2-issue2/v2-i2-choudrie-pp105-114.pdf
[13] WebXact, http://webxact.watchfire.com/
[14] Netmechanic, http://www.netmechanic.com/
[15] W3C's HTML validator , http://validator.w3.org/
[16] Vischeck, http://www.vischeck.com/

The WAB[17] Cluster has developed a methodology for the evaluation of web sites. This methodology is called the UWEM 1.0, which stands for Unified Evaluation Methodology, and is the result of a joint effort, by 23 European organisations in 3 European projects combined in a cluster for developing a UWEM. UWEM is based on the W3C Web Content Accessibility Guidelines 1.0 (WCAG).

2.3 Open Sources and Open Standards

Open Source is a development method for software according to (OpenSource.org, 2007). Its promises are better quality, higher reliability, more flexibility, lower costs and an end to vendor lock-in. The term Open Source doesn't just mean that people have free access to the source code. Open Source means that the source code for software is available under a license that allows other people to study, modify and improve the software and let them redistribute it if they wish to.

Open Standard is a specification, but it is more than just a specification according to (Perens). What make the standard open are the principles behind the standard and the practice of offering and operating the standard. Those principles are:

- Availability
- Maximizing End-User Choice
- No royalty or fees
- No discrimination
- Extension or subset
- Predatory practices

Everybody must be able to read Open Standards and implement it. The best practices are to let standards text and reference implementation to be available for free downloading from the Internet. Implementation of the standard must be possible by using any form of software license. Open standards must allow a wide range of implementations and of pricing from very expensive to zero-price. (sincerechoice.org) states that: "Intercommunication and file formats should follow standards that are sincerely open for all to implement, without royalty fees or discrimination.

With Open Standards people will have a number of choices between different products to read and edit files produced with a different product by other people. The government wishes that the citizens and the economic life shall be independent of software suppliers for communicating with the public enterprises according to (Pressemelding Nr.: 24/2007). The IT-minister in Norway has therefore sent out a proposal for public inquiry to new mandatory standards for document formats in public enterprises. Among the proposals that were sent out are the guidelines and demands for using open document formats for publishing on the internet, and for using characters set that handles different languages.

As mentioned earlier, the government in Norway emphasizes that Open Sources and Open Standards is the foundation for a more efficient interaction between the public sectors, so that they will be able to offer better public electronic services to the citizens. Therefore, the government is planning to publish the source code of MinSide.no according to (Pressemelding Nr.:17/2007). By doing this, the government hopes that this will contribute to common standards for exchanging and presenting electronic services in the public. It will also then be possible for other people to reuse parts of that solution in the development of other portals.

[17] WAB Cluster, Web Accessibility Benchmarking Cluster – for more information: http://www.wabcluster.org/

The government is also expecting that future projects will develop functionality that the MinSide-project can make use of.

2.4 Summary

Due to the increasing number of users on the Internet, it is important that the public sectors offer services of high quality. This is why this study should be carried out, to illustrate how one can use completely free existing development tools and technologies to develop a high quality web application so that it can be offered to the public in practice.

An eGovernment service, a free public service, will be most useful if it is usable and accessible by everybody. To ensure that such services will be easy to maintain and further developed, Open Sources and Open Standards can be utilized. Therefore the new prototype that shall be developed in this project must take this into account. The new prototype is meant to be offered as a free public service as the current solution offered by NAV today.

When the solution for these public services is available for the public, the solution can easily be reviewed and modified, and reused in other projects. These projects will maybe need some other functionality, and when these functionalities are developed, other projects can make use of them. This is resulting in that the services will gain higher quality over time, and that more functionality can be reused. Keeping this in mind our new prototype will be implemented with easy-to-understand comments in a good structured source code.

The accessibility guidelines developed by the WAI has to be followed when developing an eGovernment service in order for that service to be accessible by any kind of user groups. This is important to be able to include everybody in the society, even the people with disabilities such as colour blindness, aging-related conditions etc, and for ensuring that the public service is usable. The new prototype will be developed in conformance to these accessibility guidelines by utilizing accessibility checker tools for checking and correcting the problems. In this way, we will be able to ensure that our new prototype will become an accessible service.

3 Research Problem

This chapter describes the research problem in more detail. It presents the current parental benefit calculator that was offered by NAV until December 2006 and from April 2007 and towards, and what exactly the research problem is. It also presents the assumptions, limitations and the critical resources for this research.

3.1 Problem Statement

This research aims at improving an existing solution, the parental benefit calculator, both functional and graphical. In order to achieve an improved and expanded prototype, the current solution must be evaluated by making use of existing methodologies. Once the current solution has been evaluated and the weaknesses with the layout design and functionality have been addressed, a new prototype has to be developed using Open Sources and Open Standards, to show how the current solution can be improved. Since the new prototype is assumed to be offered to the public, it also has to become both accessible and usable by anyone.

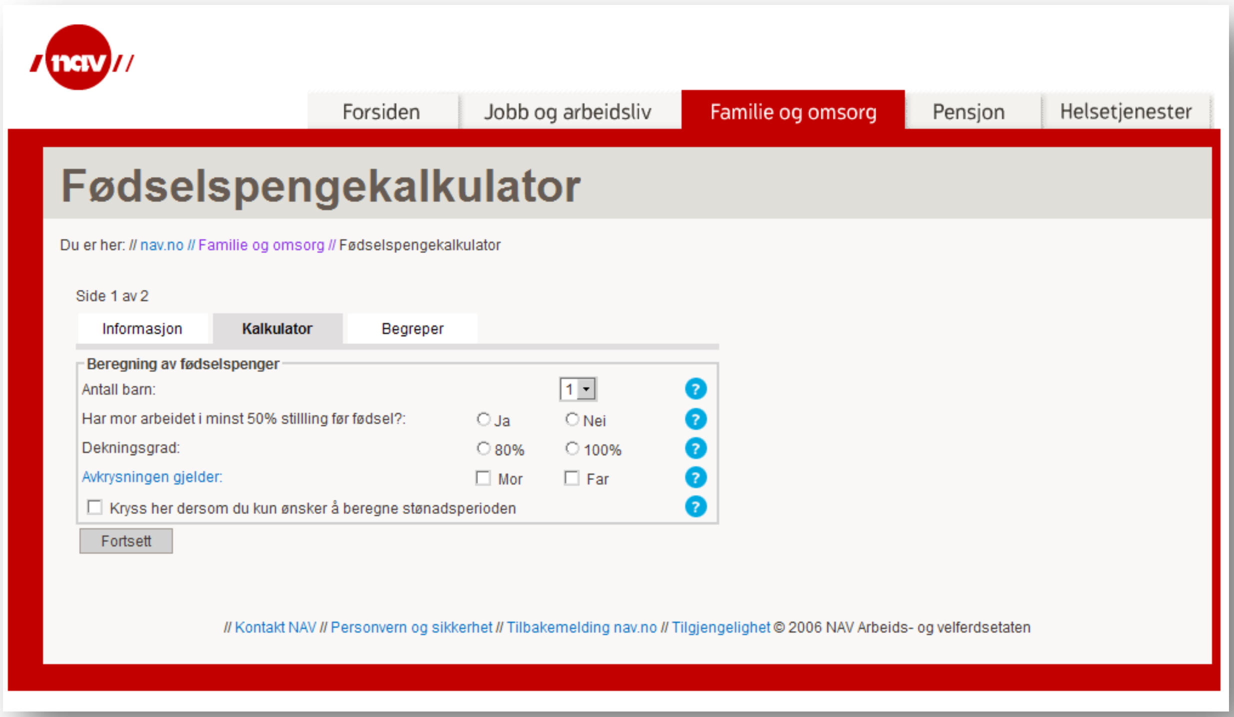
The two next following sections will describe our experience with the current solution of the parental benefit and a critical discussion of it, and what the new solution should hold.

3.1.1 Current solution

As mentioned earlier, in Norway it is possible to combine the parental leave with a part-time job through so-called “graded withdrawal”.
In 2006, NAV offered a parental benefit calculator at their website. That public online service was just giving simple estimates on how long a mother or father could take the parental leave and did not support assessment of scenarios allowing for combination of the leave with part-time job. To work out such scenarios, the parents to-be had either to get acquainted with the regulations and work out the various scenarios manually or contact the local social security office for assistance. In order to address the weaknesses of the current solution, an initial evaluation[18] of the parental benefit calculator was performed. This took place in the beginning of December 2006, and was performed with our supervisor.

Due to the changes of the rules of the parental leave in year 2007, NAV decided to remove the parental benefit calculator in the beginning of that year. The web site was updated immediately but the calculator was unavailable.
In April 2007 the parental benefit calculator came back online, but no differences regarding layout design and functionality, between the version from 2006 and 2007, were discovered after a new evaluation round. Since the update of the parental benefit calculator took approximately four months, we believe that the current parental benefit calculator was not designed for easy maintenance.

[18] The initial evaluation can be found at Appendix 11.2, page 70 and the summary in Chapter 5.2, page 33



Fødselspengekalkulator

Du er her: // nav.no // Familie og omsorg // Fødselspengekalkulator

Side 1 av 2

Informasjon | Kalkulator | Begreper

Beregning av fødselspenger

Antall barn:	1	
Har mor arbeidet i minst 50% stillling før fødsel?:	○ Ja	○ Nei
Dekningsgrad:	○ 80%	○ 100%
Avkrysningen gjelder:	☐ Mor	☐ Far

☐ Kryss her dersom du kun ønsker å beregne stønadsperioden

Fortsett

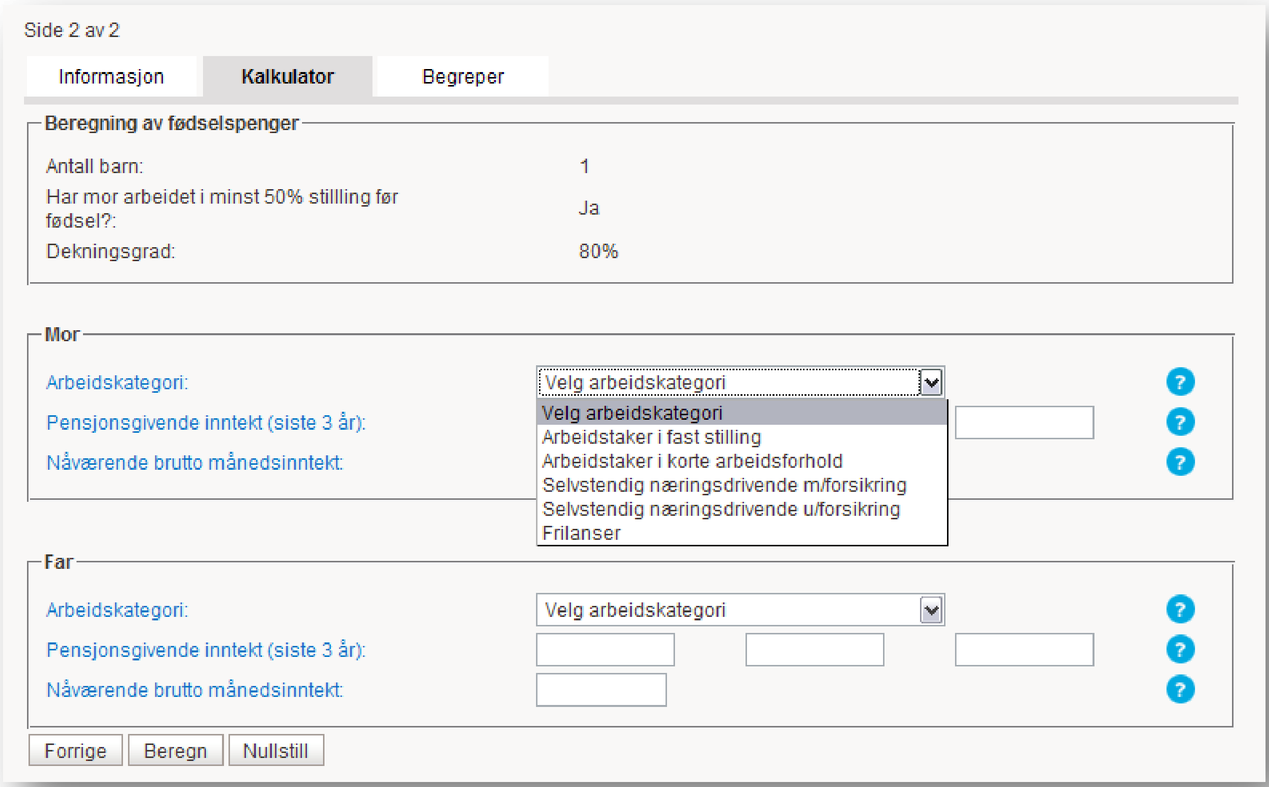
// Kontakt NAV // Personvern og sikkerhet // Tilbakemelding nav.no // Tilgjengelighet © 2006 NAV Arbeids- og velferdsetaten

Figure 3: Screenshot of the parental benefit calculator - user inputs

Figure 3 shows how the current parental benefit calculator looked like.
The calculator receives some inputs from the user in the first step, and it is mandatory that the user checks all of the required inputs. If the user needs more information about an input, he or she can click on the blue question mark icon. The “guiding” information will then be showed to the user on the right side.
The user is also able to get additional information regarding the parental leave by clicking on the 2 tabs, located beside the “Kalkulator” tab:

- “Informasjon”, provides additional information about the parental leave
- “Begreper”, provides explanations of the concepts

When the user has provided the calculator with all the necessary inputs, and clicked the “Fortsett” button, which stands for continue, he or she will be directed to a new input form. The user has to choose what occupation the mother or father has. The user will then have to input the yearly income for mother and/or father, depending on if the user chose calculation for mother and/or father on the first input form. The last information that the user has to enter is the pension able salary for the last three years.



Side 2 av 2

Informasjon
Kalkulator
Begreper

Beregning av fødselspenger

Antall barn:	1
Har mor arbeidet i minst 50% stillling før fødsel?:	Ja
Dekningsgrad:	80%

Mor

Arbeidskategori: Velg arbeidskategori

Pensjonsgivende inntekt (siste 3 år):

Nåværende brutto månedsinntekt:

Velg arbeidskategori
Arbeidstaker i fast stilling
Arbeidstaker i korte arbeidsforhold
Selvstendig næringsdrivende m/forsikring
Selvstendig næringsdrivende u/forsikring
Frilanser

Far

Arbeidskategori: Velg arbeidskategori

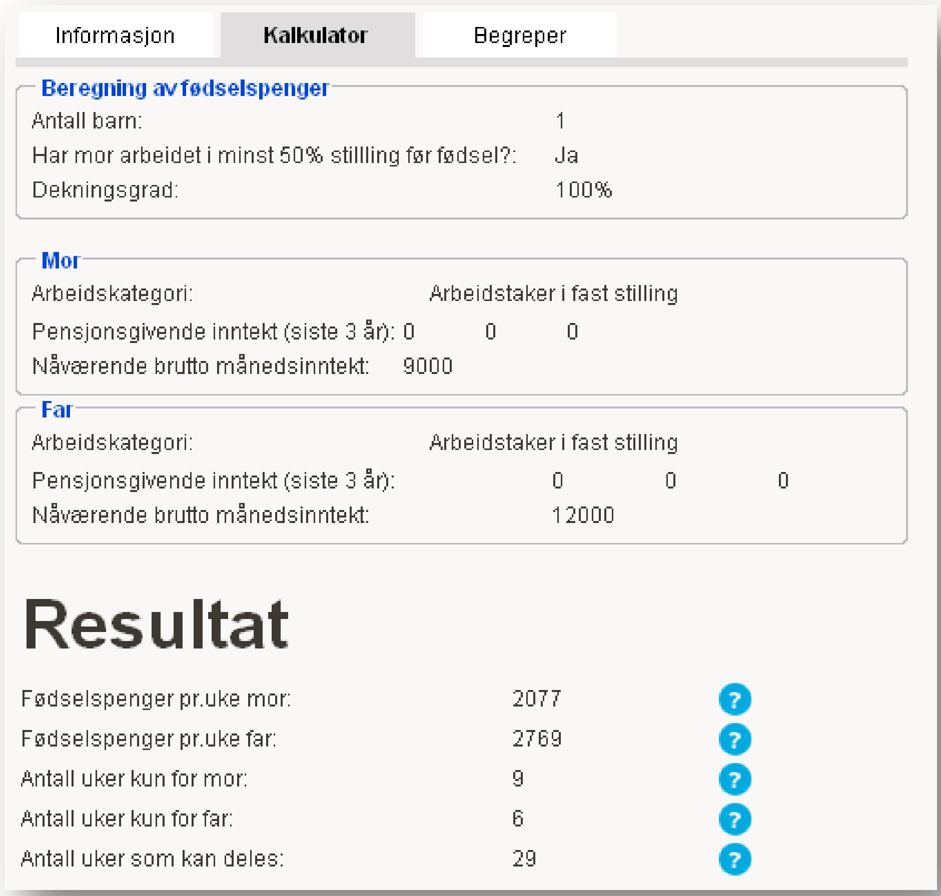
Pensjonsgivende inntekt (siste 3 år):

Nåværende brutto månedsinntekt:

Figure 4: Screenshot of the parental benefit calculator – user inputs page 2

- Employee in fulltime job
- Employee in part-time job
- Self-employee with insurance
- Self-employee without insurance
- Freelancer

After the completion of this form and a click on the “Fortsett” button again, the user will get the results presented in a new form.



Beregning av fødselspenger

Antall barn:	1
Har mor arbeidet i minst 50% stillling før fødsel?:	Ja
Dekningsgrad:	100%

Mor

Arbeidskategori:	Arbeidstaker i fast stilling		
Pensjonsgivende inntekt (siste 3 år):	0	0	0
Nåværende brutto månedsinntekt:	9000		

Far

Arbeidskategori:	Arbeidstaker i fast stilling		
Pensjonsgivende inntekt (siste 3 år):	0	0	0
Nåværende brutto månedsinntekt:	12000		

Resultat

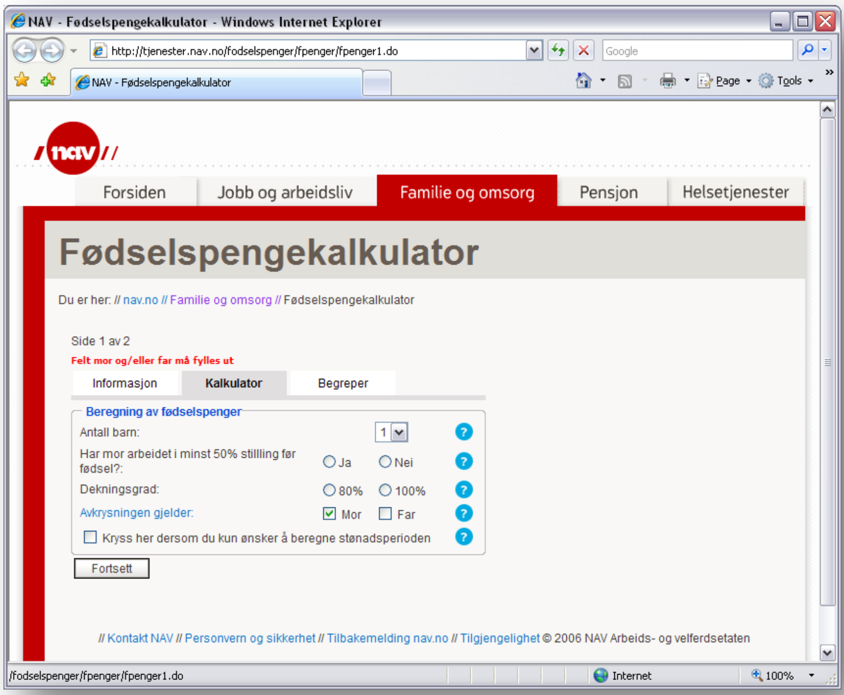
Fødselspenger pr.uke mor:	2077
Fødselspenger pr.uke far:	2769
Antall uker kun for mor:	9
Antall uker kun for far:	6
Antall uker som kan deles:	29

Figure 5: Screenshot of the parental benefit calculator - the results

The calculator will present what the user-inputs were on the top and the results below as illustrated in Figure 5. As the results the calculator presents how much parental benefit a father and/or mother gets per week, and how many weeks they can take the parental leave.

The whole calculation procedure of the parental benefit and leave was easy to perform.
The issues found in the current solution:

- At the first page, if a user forgets to choose check off for “Mor” or “Far” at the field “Avkrysningen gjelder”, the user is prompted with an error message telling the user what to correct. But if the user has forgotten to check off for the other fields, the calculation stops at page 2. The calculation of the parental leave and benefit is then impossible to finish. Figure 6 illustrates how this error can occur.


NAV - Fødselspengekalkulator - Windows Internet Explorer
http://tjenester.nav.no/fodselspenger/fpenger/fpenger1.do
Google
NAV - Fødselspengekalkulator
Page
Tools
nav
Forsiden
Jobb og arbeidsliv
Familie og omsorg
Pensjon
Helsetjenester
Internet
100%

Fødselspengekalkulator

Du er her: // nav.no // Familie og omsorg // Fødselspengekalkulator

Side 1 av 2
Felt mor og/eller far må fylles ut

Informasjon | Kalkulator | Begreper

Beregning av fødselspenger

Antall barn:	1	?
Har mor arbeidet i minst 50% stillling før fødsel?:	○ Ja ○ Nei	?
Dekningsgrad:	○ 80% ○ 100%	?
Avkrysningen gjelder:	☑ Mor ☐ Far	?
☐ Kryss her dersom du kun ønsker å beregne stønadsperioden		?

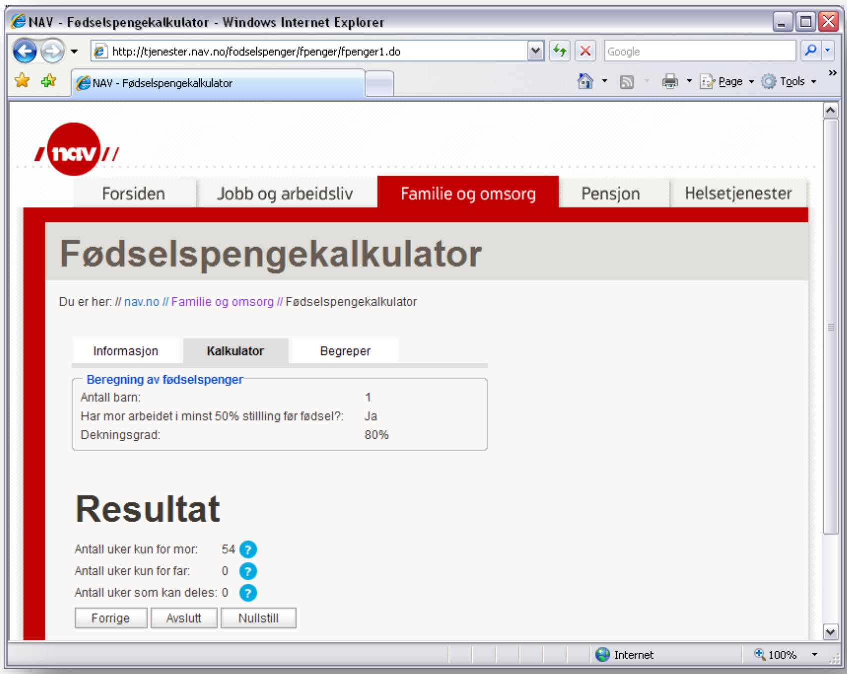
Fortsett

// Kontakt NAV // Personvern og sikkerhet // Tilbakemelding nav.no // Tilgjengelighet © 2006 NAV Arbeids- og velferdsetaten

/fodselspenger/fpenger/fpenger1.do

Figure 6: Screenshot of the parental benefit calculator – wrong number of user inputs page 1

- The checkbox for checking off “Kryss her dersom du kun ønsker å beregne stønadsperioden” is unusual designed. When a user checks off a checkbox, the user expect an expansion of functionality, not a decrease of functionality. If the user checks off the check box, he or she will only calculate the parental leave and not the parental benefit. This is illustrated in Figure 7.


NAV - Fødselspengekalkulator - Windows Internet Explorer
http://tjenester.nav.no/fodselspenger/fpenger/fpenger1.do
Google
NAV - Fødselspengekalkulator
Page
Tools
nav
Forsiden
Jobb og arbeidsliv
Familie og omsorg
Pensjon
Helsetjenester
Internet
100%

Fødselspengekalkulator

Du er her: // nav.no // Familie og omsorg // Fødselspengekalkulator

Informasjon | Kalkulator | Begreper

Beregning av fødselspenger

Antall barn:	1
Har mor arbeidet i minst 50% stillling før fødsel?:	Ja
Dekningsgrad:	80%

Resultat

Antall uker kun for mor:	54
Antall uker kun for far:	0
Antall uker som kan deles:	0

Forrige | Avslutt | Nullstill

Figure 7: Screenshot of the parental benefit calculator – check box usage

- The calculator has some unclear information.
- The tabulators on the page are hard to discover.
- The users doesn't have the possibility to change the font size or use the calculator with high contrasts
- Norwegian is the only supporting language for the calculation
- The design of the page layout is good
- The calculator is easy to use for calculations
- The calculator's performance is good
- The help texts on the page is good
- Too much information presented at once on the "Informasjon" tab.
- The calculator works good in different browsers
- The calculator is easy to locate
- The calculator only calculates the parental benefit and how many weeks leave a parent gets

As mentioned earlier in this section, the parental benefit calculator was offline for four months. We suspect that the calculator was offline due to maintenance and for updates because of the change of calculation rules. This indicates that the current solution isn't designed to be easily updated when the underlying calculation rules change.

3.1.2 New solution/prototype

In this project we develop a new prototype of the parental benefit calculator with improved usability, so it also will support assessment of scenarios allowing for combination of the leave with part-time jobs. In addition, we will correct the discovered issues in the current solution, redesign the layout and add additional functionality into the prototype. We will be using usability and accessibility guidelines for the designing, development and testing of the new prototype. In that way we will show how the parental benefit calculator can be improved and expanded. The prototype must also ensure that it will be easy to maintain and develop further and, in particular, that the underlying calculation rules may be easily updated when the regulations change.
The prototype will be built as a web-based application, using an Open Source Software, the Ruby programming language and in adherence with Open Standards, and the MySQL database tool will be applied. These tools are presented in chapter 4.4.2 on page 38.

This research focuses on how one can develop a high quality service offered on the Internet with high usability and accessibility using Open Sources and Open Standards. This consists of:

- Evaluation: evaluating the current solution to address its functionalities and weaknesses
- Requirement specification: gathering required data and stating requirements and specifications
- Design: designing and prototyping a new prototype with expanded functionality and improved usability
- Development: developing the new solution according to prototype
- Testing: test the new solution to see that it satisfies the initial requirements
- Evaluation: evaluate the new prototype according to eGovernment service quality

3.2 Research scope, limitations and critical resources

The new prototype, which will be an improved and expanded version of the current solution, will show how a public service can be designed and developed as a web application using absolute free tools, and still retain a high quality and be usable and accessible by all kinds of target

groups. The research goal is to achieve an eGovernment service, the parental benefit calculator prototype, with improved usability.
The purpose of the prototype is to let the government offer a good service on the Internet to the citizens. If this research turns out to be successful and the final solution is satisfying, this may lead to the fact that the prototype will be taken in use by NAV.
NAV will then gain benefit of this by being able to offer a public service on the Internet, and maybe the local social security offices will experience less influx by people that need assistance with combining the parental leave with a part-time job.
This research is also appropriate for further development, both in form of a publication on eGovernment or in form of a larger research project.

The prototype will be developed based on the results of the initial evaluations of the current parental benefit calculator. These evaluations are a part of the analytical work performed in this project. The results from these evaluations can be found in chapter 5 on page 45.

Different test persons must be obtained to perform the evaluations of the existing solution on. The results from the user testing with these persons are located in chapter 8.2 on page 93.

A new framework, programming language and database query language must be learned and managed. The description of these technologies can be seen in chapter 4.4.2 on page 38.

Information about the relevant regulations is expected to be found through NAV Trygd Grimstad by interviewing maternity leave caseworkers. The interview performed with the case workers can be found in Appendix 11.3 on page 121, and the results from the interview can be seen in chapter 5.3 on page 47.
An overview of the frequently asked questions that the parental benefit calculator should give answers on, the feedbacks for the new prototype of the calculator, and all the needed information about the existing solution will be provided by NAV Trygd Grimstad.

First of all, an initial evaluation of the current solution has to be done. This can be seen in Appendix 11.2 on page 120. This should be done before the implementation of the prototype. The results from this evaluation can be found in chapter 5.2 on page 45.

During the designing, development and testing of the prototype it is important that questions about the existing solution's functionalities have to be answered.
In addition, the overview over the frequently asked questions that have to be answered by the new solution must be obtained.

Another critical resource is the feedbacks on the prototype from NAV Trygd and other test persons. This could be the same test persons used in the start phase, but should be new test persons. These feedbacks have to be obtained as soon as possible when the new prototype is ready for testing and evaluations, so that the prototype can be modified and improved in time.

4 Research Approaches

This chapter will begin with presenting the solution strategy for how we intend to solve the problem in this research. It will also describe the methodologies that are going to be used and the tools that we are going to use to solve the research problem.

In this study we will evaluate the current solution, design, develop and test an improved new prototype which we also will evaluate. The outcome of this study is a public service offered by the public sector to the citizens. This means that the citizens will be interacting with a service on the Internet, and therefore it is necessary to look into Human-Computer Interaction (HCI[19]).

HCI is about designing computer systems for supporting productively and safety activities between humans and computers according to (Preece, 1994, p. 3). Its goals are to improve the interaction between the users and computers by producing usable and safe systems, as well as functional ones according to (Preece, 1994, p. 14). In other words, HCI is concerned with designing computer systems that match the users' needs.
(Preece, 1994) states that: *"HCI design should be user-centred, integrate knowledge from different disciplines and be highly iterative"*.
This means that one should involve users in the design phase as much as possible so that the users can take part in influencing the design, and test that the design does meet the users' requirements.

In order to solve the problem in this research, we intend to make use of the User-Centred design (UCD[20]) to evaluate to the existing solution, gather and specify the requirements, design and evaluate the prototype.

The solution strategy for this research is to evaluate the current solution, design, develop, test and then evaluate again as mentioned earlier in this chapter. The following sections will present methods and tools in each of the various development phases.

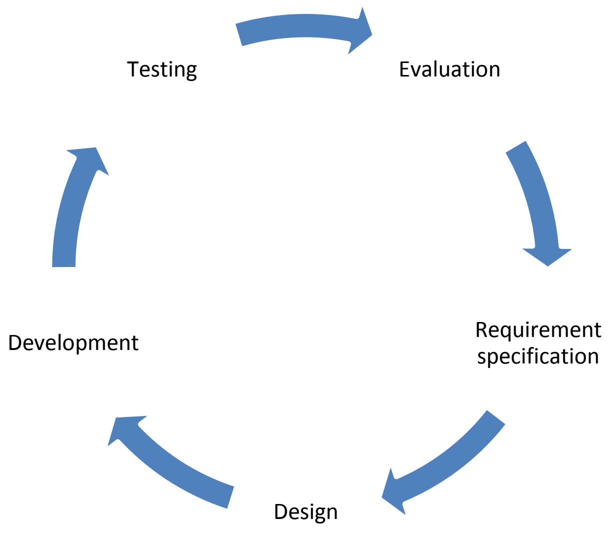


Figure 8: An iterative development process

[19] HCI – Human Computer Interaction - For more information visit http://sigchi.org/cdg/cdg2.html

[20] User-Centred Design - For more information visit http://www.webcredible.co.uk/user-friendly-resources/web-usability/user-centered-design.shtml

As illustrated in Figure 8, we will repeat the different phases in the development process several times during the project period.

4.1 Evaluations and Testing

This section will describe the methodologies that are going to be used in the evaluations and testing phases in this research.

In order to ensure that the parental benefit calculator prototype has high usability and full accessibility it is necessary to perform evaluations and testing.
Heuristic evaluations of the current solution of the calculator must be done in order to address what is good and bad about the user interface. Secondly, usability and accessibility guidelines will be used to ensure that the new version of the parental benefit calculator is both usable and accessible.
Next, usability testing of the new prototype will be performed so that the design can be changed before the delivery time of the final version of the prototype.

The following sections will first give a description of the approaches that will be used in this project and then describe what we will use them to and what the results became.

4.1.1 Heuristic evaluation

Heuristic evaluation is a method for quick, cheap and easy evaluation of a user interface design according to (useit.com). It is the most popular method of the usability inspection methods. Heuristic evaluation is done by looking at a user interface and systematically inspecting the user interface design for usability. Its goal is to find the usability problems in the user interface design so that they become a part of an iterative design process according to (Nielsen J. , 1993, p. 155). Heuristic evaluation is involving having some evaluators to examine a user interface and judge it with recognized usability principles.

Generally, heuristic evaluation is difficult to be performed by one single person, since this person will never be able to find all the usability problems in a user interface. According to (Nielsen J. , 1993, p. 155), different evaluators find different usability problems with a user interface. Therefore it is important to involve several evaluators when performing heuristic evaluations of a user interface design.

This method is performed by having the different evaluators inspect the same user interface design alone. When the evaluation has finished, the evaluators are allowed to communicate with each other and share their findings. This is important for ensuring independent and unbiased evaluation from each evaluator.

Jacob Nielsen's online writings on heuristic evaluation present a list of ten recommended heuristics for usable interface design[21]. These heuristics will be an important part for the final design of the parental benefit calculator, and are:

- Visibility of status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall

[21] See http://www.useit.com/papers/heuristic/heuristic_list.html

- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

The current parental benefit calculator will be evaluated in the early phases of designing the new prototype using heuristic evaluation. We will make use of heuristic evaluations before the initial evaluation of the current solution. We will then perform the initial evaluation in cooperation with our supervisor. The initial evaluation can be seen in Appendix 11.2 on page 120 and the results from this evaluation can be found in chapter 5.2 on page 45.
Heuristic evaluation has been used so that we could be able to identify the usability problems with NAV's current solution.

By addressing the weaknesses of the current solution, we will improve those discovered issues in the new prototype.

4.1.2 Usability

Since the parental benefit calculator is being offered to the public, where the user of the calculator could be anyone, it is important that the service has high usability. In order to ensure that the new prototype of the calculator has high usability, we need to perform usability testing on the new prototype after the development phase.

The goal of usability testing is to find out if what the expectations and what the needs of the users of a system are according to (Cascia). Testing with real users is the most fundamental usability method since it provides direct information about how people is using the computers and what their problems are with the user interface they are using according to (Nielsen J. , 1993, p. 165).

During the usability testing, the aim is for the developer to observe the test users using the user interface in a realistic situation as possible to discover the difficulties and areas of improvement.
If difficulties are discovered during usability testing, the developer should improve the design and test the interface again.
Developers usually focus on developing “cool” looking interfaces, and they often forget about the usability and functionality of the system. The primary objective for a developer should be more than appearance, so that they develop systems that satisfy people's needs.
Usability testing will be performed with several test users when the new prototype of the parental benefit calculator is about to be completed. The prototype will then be further improved before the dead-line for this project.

Before conducting the usability testing with real users, a plan has to be worked out and should address some of the following issues described by (Nielsen J. , 1993, p. 170). The issues that need to be addressed for the testing in this project are:

- The goal of the test: What to achieve?
- Where and when the test will take place?
- How long is the test session estimated to take?
- What computer support is needed for the test?
- What software is needed for the test?
- Who are the test users, and how to get hold of them?
- How many test users are needed?
- What tasks will the users perform?
- What user aids will be made available to the test users?
- How much help can the test users get from the researcher?

- What data is going to be collected, and how are they going to be analysed?

When the plan has been worked out the usability testing can take place when the test users are acquired. According to (Nielsen J. , 1993, p. 175) the test users should be as representative as possible of the intended users of the system. Since the users of the parental benefit calculator could be anybody, the test users for the new prototype will have different background and knowledge about computers.

The plan for the usability testing is presented in chapter 8.2 on page 93 together with a description of the different test users what were used for the testing. The problems that were discovered during the user testing are also presented in that chapter. We made use of the results and corrected the problems for improving our new prototype.

4.1.3 Accessibility

As mentioned earlier, the parental benefit calculator can be used by everybody. This also includes people with disabilities. This means that people with disabilities must also be able to perceive, understand, navigate and interact with parental benefit calculator as well. The use of services offered on the Internet is increasing, and therefore it is important to be able to give everybody the opportunity to participate in society, and let people with disabilities more actively participate.

Web Accessibility Initiative[22] (WAI) is developing guidelines and techniques that are describing accessibility solutions for both Web software and developers. These WAI guidelines are regarded as the international standard for web accessibility according to (W3 - WAI). WAI is also developing materials to help understand and implement web accessibility.

Creating a web site accessible can be both simple and complex. It depends on many factors such as the type of content, the size, the complexity of the site, and the development tools and environment according to (W3 - WAI). In order to easily implement accessibility features on the web site, the features have to be planned from the beginning of development or redesign of the web site. Fixing a website that is inaccessible can be quite a challenge, since the web site may not be “coded” properly with standard XHTML[23] markup.
The development of the new prototype of the parental benefit calculator will be using WAI guidelines and techniques to ensure that it becomes both a usable and accessible service. The guidelines will also be used early and throughout the development process for evaluation; addressing and identifying the accessibility problems. These problems are most easy to detect in the early phases.

There are several documents[24] offered by W3C which can be used for aiding the evaluation of a service for accessibility. These documents are presented in Table 1 together with their descriptions.

W3C document	**Description**
Preliminary Review of Web Sites for Accessibility	Approach to identify potential accessibility problems on a web site
Conformance Evaluation of Web Sites for Accessibility	Approach for determining if a web site is meeting accessibility standards such as WCAG
Evaluation Approaches for Specific Contexts	Approach for evaluating during a development

[22] For more information about WAI, see http://www.w3.org/WAI/
[23] For more information about this technology (XHTML), go to page 36
[24] These documents can be found at http://www.w3.org/WAI/intro/accessibility.php

	process, ongoing monitoring and dynamically generated web pages
Involving Users in Web Accessibility Evaluation	Guidance on how to include users with disabilities
Selecting Web Accessibility Evaluation Tools	Guidance on how to choose accessibility evaluation tools for the evaluation of web accessibility
Web Accessibility Evaluation Tools List Search	Search for and sort accessibility evaluation tools
Using Combined Expertise to Evaluate Web Accessibility	Describing the composition, training and operation of reviewer teams that evaluate accessibility
Template for Accessibility Evaluation Reports	Format for communicating results of a web accessibility evaluation

Table 1: W3C documents and their descriptions

Besides these documents there exist several accessibility evaluation tools[25]. These tools can help determine if a web site is accessible. By using these tools a developer can reduce time and effort required to carry out evaluations. When these tools are used throughout the design, implementation and maintenance they can assist the developers in preventing accessibility barriers, repairing the encountered barriers, and even improving the overall quality of web sites.

The tools that are going to be used in this research are:

- A-Checker
- A-Prompt
- WebEXACT
- AccessColor
- CSS Analyser

The tools will test and validate the web pages of the new prototype for conformance to the web content accessibility guidelines (WCAG[26]). The WCAG 1.0 will be used since it was approved in May 1999 and is the current stable and reference able version. Although these tools will be used to automatically review the web pages, we will also manually review the web pages for ensuring language clarification and the ease of navigation. These tools will <u>not</u> determine the accessibility of our prototype web pages, but will reduce the time for evaluating the pages.
The description of these tools is presented in chapter 4.4.2 on page 38.

A-checker, A-Prompt and WebXACT will be used to check our new prototype for accessibility problem using the WCAG 1.0. AccessColor and CSS Analyzer will be used to analyzing the CSS of the web pages for our prototype and for checking the colour contrast on the pages. This is represented in Table 2.

[25] The accessibility evaluation tools can be found at this location: http://www.w3.org/WAI/ER/tools/Overview.html
[26] The Accessibility guidelines can be found at http://www.w3.org/WAI/intro/wcag.php#version

Tools	Goal
A-Checker, A-Prompt and WebXACT	Check web pages of our prototype for accessibility problems and link them to the web content accessibility problems version 1.0 (WCAG 1.0).
AccessColor and CSS Analyzer	Check the internal and external CSS of the web pages of our prototype, and also the colour contrast on the web pages.

Table 2: The tools and their goals

4.2 Structured Interview

Structured interview[27] is going to be used for the evaluation phase in this project. In this phase, an initial evaluation of the current parental benefit calculator will be performed with structured interview, and an interview will also take place with a caseworker at NAV Trygd Grimstad. The structured interview can be found in chapter 11.3 on page 121.

Structured interview is an interview with pre-defined questions and responses according to (Usabiliy first). Before a structured interview starts, the specific goals and questions for the interview are detailed in advance. Since this provides guidelines, it therefore provides a structure according to (London South Bank University, 1997).

By using structured interview one can collect data for statistical surveys. The data are being collected by an interviewer, and not through a self-administered questionnaire. Most of the data are related, and this method is providing detailed information on a problem.
The questions in the survey questionnaire are read and asked exactly as they appear, and the choice of answers is often fixed or close-ended in advance. But structured interviews can also contain open-ended questions.

This approach will be used to aid the specification of the requirements, and will support the design of the new version of the parental benefit calculator.

4.3 Design

This section describes the methodology for designing the prototype in this research. We will first describe the two different kinds of prototyping, and the different types and techniques. Then we will describe which we will use and how we will use them.

Prototyping is a cheap and fast method to rapid iterative design of user interfaces, and is a form of design specification according to (Nielsen J. , Usability Engineering, 1993). It is an integral part of iterative user-centred design according to (Preece, 1994), and they say that there are two kinds of prototyping:

[27] See http://www.scism.sbu.ac.uk/inmandw/tutorials/kaqu/qu8.htm

- Computer-based prototyping
- Paper-based prototyping

The following sections will describe the two different kinds of prototyping, prototype types, and prototype techniques.

4.3.1 Computer-based prototyping

Computer-based prototyping is, as the name suggests, using computers to develop prototypes. This provides a version of the final system with limited features or functionality, but the users are able to interact with prototype. Computer-based prototyping makes use of user interface tools. Such tools can be any software that helps a developed to create user interfaces.

With computer-based prototyping designs can be rapidly prototyped and implemented. It will also be easier to make changes in the design process through user testing with several test users. Computer-based prototyping also allows having more realistic mock-ups for the user testing.
By using this technique it results in less code to be written by the developer for the final system, and the reliability of the user interface will be higher since the source code is automatically produced by the user interface tools.

There exist many different kinds of user interface tools, and some of them may be difficult to use because of poor usability or limited functionalities. This leads to that the developer has to change user interface tool later in the design phase, and has to learn how to use a completely new user interface tool.

In this project we will use computer-based prototyping for developing the prototype when the design of the prototype has been finalized. The result of the computer-based prototyping phase will be used to let test users test out the prototype. The prototype will then be further improved after the testing with test users.
We will use paper-based prototyping before we make use of the computer-based prototyping. The following section will describe the paper-based prototyping.

4.3.2 Paper-based prototyping

In the first design phase paper-based prototyping will be used for designing the new version of the parental benefit calculator.

With *paper-based prototyping* a developer doesn’t need to use much time and effort to develop a functional coded prototype. Instead, tools like scissors, papers, stickers and pen will be used to prototype. Our paper-based prototypes located in Appendix 11.6 on page 142 illustrate this.

(Kaufmann, 2003) says that: “Paper prototyping is a variation of usability testing where representative users perform realistic tasks by interacting with a paper version of the interface that is manipulated by a person ‘playing computer’, who doesn’t explain how the interface is intended to work”.
The most important factors of paper prototypes are:

- Team communication
- Usability testing
- Design testing
- Information architecture

By using paper prototyping this will help a development team to visualize and share ideas on how a user interface might look like. The user interface will be built up step by step and seen by every member of the team. Everybody in the team can participate and come up with ideas. Eventually the prototype can be used as a visual specification of the final system. The paper prototypes will be used during the meetings with the supervisors for obtaining feedback on the design.

Paper prototyping can also be used for usability testing with several test users. The users will perform realistic tasks while one member of the development team will respond to the users' actions by manipulating the paper prototype. Paper prototyping is very efficient since it is letting us user test early design ideas and fix the usability problems before the implementation phase so that we don't implement something that doesn't work according to (Nielsen J. , 2003).

Especially in Web Design, paper prototyping can be used to assist the designing of user interfaces. The prototype will be presented to the user and the user will be asked to identify the main navigation, clickable elements etc. This technique is also the recommended design testing technique in the CD[28] process.

The information architecture of a software or web site can be tested, when using paper prototyping. The test users will be asked where they would search some functionality or settings in software or topics on a web site.

This kind of prototyping provides great speed and flexibility, and is inexpensive. It also allows for easy modification, and usually results in good overall quality of the final system. Although this technique is good, it also has its downsides. First of all, paper prototyping doesn't produce any code and can affect the way users interact with the user interface. Some will become nervous because they fear users will think the paper prototype is unprofessional according to (Kaufmann, 2003).
According to (Snyder, 2004) paper prototyping is indeed low-tech, but this technique, a usability testing technique, can help a developer sidestep problems before writing the code.

4.3.3 Prototype Types

In this section we will describe the different types of prototypes. We had to become familiar with the different types in order to choose a type of prototype in this project.

The idea of prototyping is to develop something fast and cheap so that it can be tested on real users according to (Nielsen J. , 1993). In order to achieve this, the designer has to limit the number of features, or the level of functionality to the prototype compared to the final system.

There are several different types of prototypes according to (Preece, 1994). These are:

- Full prototype
- Horizontal and vertical prototypes
- High fidelity and low fidelity
- Chauffeured prototypes
- Wizard of Oz prototypes

A *full prototype* is a prototype with complete functionality of the full system although with lower performance.

[28] CD - Contextual Design is a user-centred design process developed by Hugh Beyer and Karen Holtzblatt. For more information visit http://sigchi.org/chi95/AP/t24.html

(Nielsen J. , Usability Engineering, 1993) says that: “Cutting down on the number of features is called *vertical prototyping”* and “Reducing the level of functionality is called *horizontal prototyping”*.

Although a vertical prototype doesn’t have all the features of a full system, still it can be tested with full functionality. With a vertical prototype of a parental benefit calculator, users are able to calculate how much a mother will get per month, but not for example have the possibility to choose between 80% or 100% coverage.

A horizontal prototype is different. This kind of prototype simulates the interface of a full system, but no real functionality. The users can navigate and execute all of the commands, but no real is done. The users can for example choose between 80% and 100% coverage, how many children the calculation should be calculated for, but no calculations will be done and the users won’t get any real data or any results back.

Designers also have the possibility to cut down both the number of features and level of functionality. With this combination the designers get an ultimate minimum prototype which only simulates some scenarios. These scenarios have two main uses according to (Nielsen J. , 1993).

1. They can be used during the design of a user interface to express and understand the way users eventually will interact with the final system
2. They can be used during an early evaluation of a user interface design to get feedback from the users without having to create running prototype

High fidelity prototypes are prototypes used through a medium, for example a video, where the prototype is very like the final interface.

Low fidelity prototypes is involving materials that are further away from the final interface, but tends to be more cheaper and faster than high fidelity prototypes.

Chauffeured prototyping is where the designer “drives” the system while a user watches. This is useful to get confirmation from the user if the design meets the user’s needs.

The last type of prototype is the *Wizard of Oz prototype.* This kind of prototypes is very similar to chauffeured prototypes, but here it also involves a third party. The user is unaware of that the designer is sitting at another screen to respond on the queries from the user when the user is interacting with a screen.
This kind of prototyping is useful in the early stages of development, because this gives the designer the understanding of the users and what their expectations and needs are.

By using these different kinds of prototypes in the different stage of design gives the designer a two-phase view of iterative design.
Prototypes are being developed in the beginning of a design phase to gather information, and test radically different alternatives with fast cycle times. Eventually the prototyping will end at some point with one single design proposal. This solution will then be iterated through implementation stages: design, code and test cycles.
At this moment, any radical changes to the design are much unlikely, since it will be too expensive to do big changes. This phase can be considered as a fine-tuning stage with slow cycle times.

We decided to develop a *full prototype* in this project. We wanted to have a prototype with full functionality in order to achieve a new of version of the parental benefit calculator with improved usability. We made use of the paper-based prototypes and the feedbacks on it, and developed a computer-based prototype which we also utilized for the user testing with real users. Our full prototype is located at http://128.39.61.95:3000.

4.3.4 Prototyping techniques/methods

In this section the different prototyping techniques will first be presented and then we will point out which one we made use of.
By prototyping a designer can try out several ideas, and it helps the designer to make decisions about:

- the look and feel of the user interface
- the necessary functionality of a system
- the user-help needs
- the sequences of operation

It also helps the designers to identify the weaknesses early, and save their time and expenses.
There have several different kinds of prototyping techniques or methods to obtain different kinds of information. These are:

- Requirements animation
- Rapid prototyping (Throw-it-away)
- Evolutionary prototyping
- Incremental prototyping

Requirements animation allows possible requirements to be demonstrated in a software prototype which can be accessed by the users.

Rapid prototyping is also a method to collect information on requirements and on the sufficiency of possible designs. This kind of prototypes gets thrown way in the end, but is an important resource during the project period.

Evolutionary prototyping is a compromise between production and prototyping, and is considered as the most extensive form of prototyping. The initial prototype is constructed, then evaluated and continues to involve into the final system.

Incremental prototyping can be used to avoid delays between specification and delivery, because it allows large systems to be installed in phases. After the agreement on core features between the customer and the supplier, the installation of a skeleton system can take place as soon as possible. Important requirements can be checked in the field enabling the possibility for changes to the core features.

The evolutionary prototyping technique is the one which will be used during this project for prototyping. As mentioned in the previous section, we will first create paper-based prototypes and make use of them so that we can get feedback on them. These prototypes will then involve into the computer-based prototype. We will then again get feedback on the computer-based prototype by performing user testing with real users. When the prototype has been involved in the different evaluations and improved by us, it will become the final version of our new prototype for the parental benefit calculator.

4.4 Development

4.4.1 Development Methodology

This section describes the methodology that is going to be used for the development phase in this research.

Rapid Application Development (RAD[29]) is a software development methodology that is focusing on developing applications quickly, which traditionally compromises usability, features and/or execution speed according to (Blue Ink , 2005). RAD involves iterative development, constructing prototypes and the use of computer-aided software engineering (CASE[30]) tools. The CASE tool that is going to be used is UML[31], which will assist in the development of the prototype.

According to (Blue Ink , 2005) the advantages and disadvantages with RAD are:

- Increased speed
- Increased quality
- Reduced scalability
- Reduced features

The applications are developed quickly and the time for delivery is decreased by using "Time Boxing" which means that features are pushed out to future versions of the application, in order to quickly complete a feature light version. "Time Boxing" is a time management technique common in software development where the plan is split up in several time periods.
CASE tools are used for converting requirements into code as quickly as possible. The applications developed will have increased quality, which means that the applications developed will meet the needs of users and the costs for maintenance are low. RAD is trying to deliver this quality by involving users early in the design phase.

By using RAD, the final solution may lack the scalability if a solution that was designed as a full application from the beginning. This is because of that RAD is focusing on development of a prototype which is iteratively developed into a full system. Due to the "Time Boxing" technique, RAD may result in producing applications that have fewer features than traditionally developed applications. The developer should as soon as possible agree with the customer/client on what that should to be delivered and when.

4.4.2 Development Technologies & Tools

In this section we will describe the different tools that are going to be used to design, develop and test the new prototype. These are presented in Table 3.

Technologies & Tools	**Description**
Eclipse	Eclipse is an open source community which focuses on building an open development platform according to (eclipse.org, 2007). The tool Eclipse will be used together with UML for creating UML Use Case diagrams for supporting the design of the parental benefit calculator prototype.

[29] RAD, Rapid Application Development, see http://www.blueink.biz/RapidApplicationDevelopment.aspx
[30] CASE, see http://www.cs.utexas.edu/users/almstrum/cs370/tlee/r1.htm
[31] UML, Unified Modelling Language, see http://www.uml.org

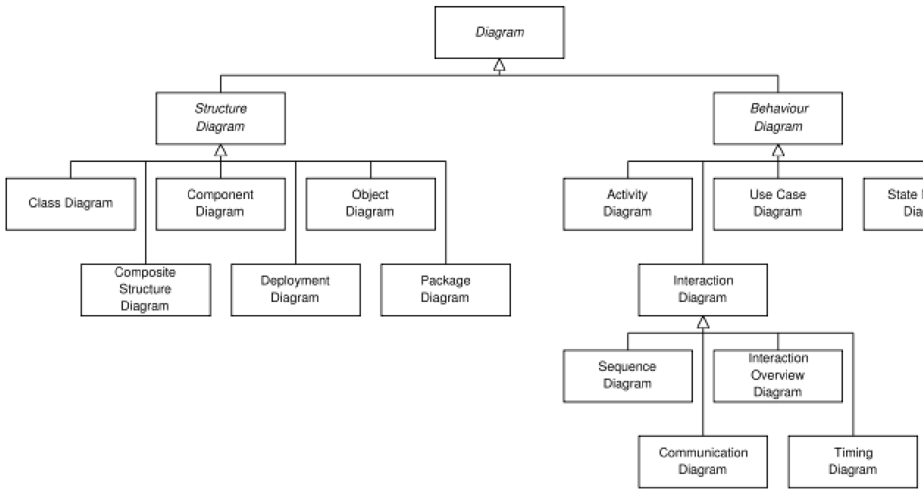
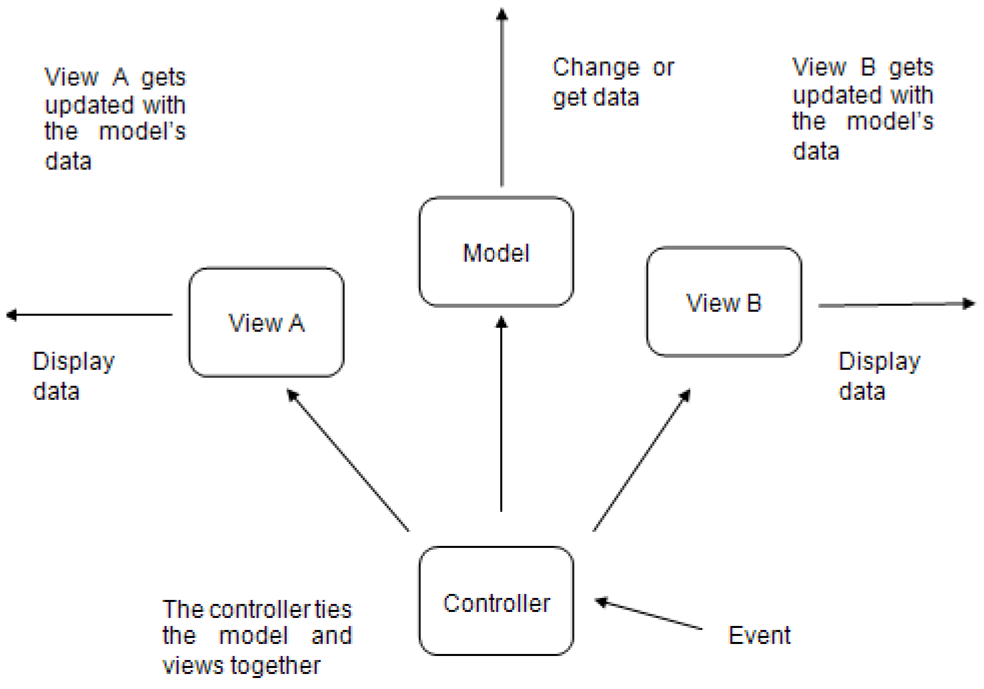
	 Figure 9: Eclipse logo *Downloaded from http://eclipsetutorial.forge.os4os.org/in2.htm on 07.05.2007*
UML	Unified Modelling Language or UML is a specification language for modelling objects. It is not used for only modelling application, behaviour and architecture, but also for modelling business processes and data structures according to (Object Management Group, 2007). In this project we will mainly use “Use Case diagrams” and “Sequence Diagrams” which are two of the thirteen types of diagrams in UML 2.0 according to (Agile modeling, 2007). In Figure 10 the diagrams are categorized hierarchically.  Diagram Structure Diagram Behaviour Diagram Class Diagram Component Diagram Object Diagram Composite Structure Diagram Deployment Diagram Package Diagram Activity Diagram Use Case Diagram State Machine Diagram Interaction Diagram Sequence Diagram Interaction Overview Diagram Communication Diagram Timing Diagram *Figure 10: The thirteen types of diagrams in UML 2.0 – Downloaded from (Wikipedia, UML).* Structure diagrams emphasize what a system being modelled must contain. Behaviour diagrams emphasize what should happen in a system being modelled. Interaction diagrams emphasize the control and data flow among the things in the system being modelled.
Ruby	Ruby is a dynamic and open source object-oriented programming language according to (Ruby - A Programmer's Best Friend).

	 Figure 11: Official Ruby Logo *Downloaded from www.ruby-lang.org/en/ on 23.01.2007* The programming language is influenced by other languages like Python, Perl, Smalltalk etc. It appeared to the public in 1995 and was designed by Yukihiro Matsumoto. Yukihiro Matsumoto used C to write Ruby, and designed it with Perl and Python capabilities in mind according to *(Stewart, 2001)*. In the interview with the creator of Ruby, he said that Ruby is designed to be human-oriented. It tries to push jobs back to the machines so that you can accomplish more tasks with less work. Ruby is an open source project, which means that it is completely free to use, copy, modify and distribute according to (Ruby - A Programmer's Best Friend). Ruby will be used as the programming language for the new prototype of the parental benefit calculator.
Ruby on Rails	Ruby on Rails is an open-source framework that is optimized for programmer happiness and sustainable productivity according to (Rails : Web development that doesn't hurt). David Heinemeier Hansson, a partner at the web design company 37signals, is one of the original authors of Ruby on Rails. *Figure 12: Ruby on Rails Logo* *Downloaded from http://www.rubyonrails.org/ on 23.01.2007* Ruby on Rails is often shortened to Rails or RoR, and was written in the programming language Ruby, and was released to the public in July 2004. Rails uses the MVC[32], Model-View-Controller, architecture for organizing the applications like most of the other contemporary web frameworks. Ruby on Rails provides out-of-the-box scaffolding, which quickly constructs most of the logic and view that are needed for a basic web site. It also provides a WEBrick[33] web server and other development tools. Ruby on Rails is striving for applications with less code. This means faster development time, less bugs and makes the source code easier to understand, enhance and maintain according to (Hibbs, 2005). In an interview with David Heinemeier Hansson from Ruby on Rails, he said

[32] Model-View-Controller is a design pattern used by applications that need to maintain multiple views of the same data. (eNode, 2002)

[33] WEBrick – a Ruby library program to build HTTP servers (WEBrick, 2004)

	that packages like Instant Rails for Windows make it easy for people to get started with Ruby on Rails. (Grimmer, 2006) He also said that Rails is about allowing beautiful code to solve problems most people have most of the time when developing web applications. The Instant Rails are installed on the computer by downloading the package from http://rubyforge.org/frs/?group_id=904. Next, the package has to be unpacked and save on the hard drive on the computer. RoR will be running when double clicking on the InstantRails.exe file; the Apache server and the MySQL server will be up and running.
Model-View-Controller	Model-View-Controller or MVC is a design pattern according to (SDN). This pattern is often used by applications that present a lot of data to the users. It separates the applications' data model, graphical user interface and control logic into three categories: model, view and controller. The model is used for the management of data and the view is used for displaying the data, while the controller takes care of the user interactions. This means that one can change the view without affecting the model and vice versa. Theoretically, the view and the model don't need to know anything about each other, since the controller can be used to tie them together according to (Ruby on rails: Understanding MVC). This results in that multiple views and controllers can make use of the same model. Figure 13 is illustrating this.  View A gets updated with the model's data Change or get data View B gets updated with the model's data Model View A View B Display data Display data The controller ties the model and views together Controller Event *Figure 13: The MVC architecture*
Javascript	Javascript is a scripting language developed by Netscape based on the concept of prototype-based programming according to (mozilla developer center, 2007). With Javascript developers are able to add dynamic content to their web sites. The language is an open language that anyone can use

	without having to purchase a license. Javascript works with browsers such as Internet Explorer, Mozilla, Firefox, Netscape and Opera and is the most popular scripting language on the internet according to (W3Schools: Javascript Intro) The prototype will be using Javascripts to get more additional functionality.
Ajax	Ajax which stands for Asynchronous Javascript and XML[34] is a technique for creating better, faster and more user-friendly web applications according to (W3Schools: Ajax Tutorial). With this technique an entire web page doesn't need to gets reloaded when a user requests a change, and will therefore increase a web page's interactivity, speed and usability according to (W3schools). Ajax will be used in the prototype, so that the whole pages don't reload each time a user wishes to see a help text for example when using the calculator. This technology will be used to update the results from the calculations, so that the data transfer to the server will be reduced.
MySQL	MySQL is an open-source, multi-threaded and multi-user relational database management system according to (University of Rhode Island). *Figure 14: MySQL Logo* *Downloaded from http://www.mysql.com/ on 24.01.2007* MySQL is developed and maintained by a Swedish company called MySQL AB. Most of the programming languages can be used to connect to a MySQL database; C++, Smalltalk, Java, Ruby etc. It is popular for web applications, since it is easy to use and free to use. It is also one of the most popular choices as a database backend for Ruby on Rails applications according to *(Grimmer, 2006)*. MySQL will be used as the query language in the prototype for getting the necessary data out from the database. Ruby on Rails also has predefined functions which made use of MySQL. The MySQL database will store all the data needed when using the parental benefit calculator prototype.
HeidiSQL	HeidiSQL is a free tool for working with a MySQL database. It has an easy-to-use interface and allows one to manage and browse databases and tables according to (HeidiSQL). This tool will be used to build MySQL database, but also for testing and validating the data in the database when working with the prototype. It facilitates the work and provides a great overview of the database design and data.

[34] XML – Extensible Markup Language (W3C: XHTML)

	Figure 15: HeidiSQL Logo *Downloaded from http://www.heidisql.com/ on 06.04.2007*
(X)HTML	HTML stands for HyperText Markup language according to (W3schools - HTML). It is used for the creation of web pages. HTML is used to structure information by denoting some text such as headlines, paragraphs, lists and so on, and to supplement that text with interactive forms, embedded images and other objects. HTML can also be used to describe, to some degree, the appearance and semantic of a document. It is also possible to include embedded scripting language code which can affect the behaviour of web browsers and other HTML processors. HTML is written in the form of labels, created by the signs < and >. XHTML or Extensible HyperText Markup language is the same language as HTML, but has a stricter syntax. Since XHTML must be syntactically correct, these XHTML documents are more easily automated processed and maintained. These documents are XML based, and are designed to work in conjunction with XML-based user agents according (W3C - XHTML). XHTML will be used for the development of the web pages for the parental benefit calculator. This technology provides the pages a structured source code for easy maintenance, accessibility and further development.
RHTML	RHTML is the same as XHTML, but can have embedded Ruby code in Ruby on Rails. One is able to embed Ruby code in a HTML file by writing <% “code” -%> in the HTML file. RHTML will be mainly used to control what content to show on the pages (dynamic content), and showing Ruby objects’ value on the web pages.
CSS	Cascading Style Sheets or CSS is a mechanism which is used to define the appearance of documents written in HTML, XHTML or XML. The principle is that the HTML, XHTML or XML documents should exclusively describe the structure and semantic, while the layout, colours and other style information should be described with the help of CSS according to (W3 CSS). By using CSS it enables web page designers to more easily change the layout, colours and other style information for several HTML documents, and it also makes the web pages easier to maintain. (Garshol, 1999). CSS will be used to style the web pages of the prototype. The prototype will have two main style sheets, one regular style sheet and one style sheet for displaying the page in high contrasts.
Apache HTTP Server	Apache HTTP Server is a web server for operating systems like UNIX and Microsoft Windows according to (Apache.org). It also states that Apache has been the most popular HTTP server on the Internet since April 1996. Apache is developed and maintained by an open community of developers and is therefore a free software. Apache server is the server used to deploy the new prototype on for hosting it.

	The Apache Software Foundation http://www.apache.org/ *Figure 16: Apache logo and address* *Downloaded from http://www.apache.org/ on 13.02.2007*
A-Checker	A-Checker[35] is a free online accessibility checker that tests web pages for conformance to different accessibility guidelines. The tool is supporting file formats like HTML and XHTML. It supports generating reports and step-by-step evaluations. This tool will be used for checking the web pages of the new prototype for accessibility problems.
A-Prompt	A-Prompt[36] is a free tool for accessibility evaluation and repairing. It supports file formats like HTML and XHTML. The tool is able to generate reports and step-by-step evaluations. It can also modify the source code of the supported file formats for repairing them. This tool will also be used to check the web pages of the new prototype for accessibility problems, but also for correcting the discovered problems.
WebXACT	WebXACT[37] is a free tool for testing the content of web pages of the prototype for quality, accessibility and privacy issues. This tool has been used in addition the other accessibility checker tools just to verify that the prototype is in conformance to accessibility guidelines.
AccessColor	AccessColor[38] is a free online tool for analyzing the internal and external CSS of a web page and test the colour contrast and colour brightness between the text and background colours. It is using an algorithm recommended by the W3C[39]. The tool makes sure that the contrast on the web pages is high enough people with visual impairments. It will be used for ensuring that the colour difference and colour brightness on the web pages of the new prototype meet the recommended standard.
CSS Analyzer	CSS Analyser[40] is a free online service for checking the validity of a web page's CSS against the W3C's validation service, along with a colour contrast test using the W3C's colour contrast algorithm, for ensuring that relevant sizes are specified in relative units of measurement. This tool will be used for checking the CSS documents of the new prototype, to make sure that relevant sizes are specified in relative units of measurement.

Table 3: Technologies & tools and their descriptions

[35] The A-Checker tool can be found at this location: http://checker.atrc.utoronto.ca/index.html
[36] The A-Prompt tool can be downloaded from this location: http://www.aprompt.ca/
[37] WebXACT can be found at this location: http://webxact.watchfire.com/
[38] AccessColor can be found at this location: http://www.accesskeys.org/tools/color-contrast.html
[39] The recommendation can be seen here at this site: http://www.w3.org/TR/AERT#color-contrast
[40] CSS Analyser can be found at this location: http://juicystudio.com/services/csstest.php

5 Solution and Results

This chapter presents the solution and the results achieved by using the methods and tools described in Chapter 4 Research Approaches.

5.1 Overall results

The interviews performed during the project period will be presented with a summary and with the key findings from those interviews. These interviews are presented in section 5.2 and 5.3.

The results from the heuristic evaluations are located in chapter 8.3 on page 100.
The results from the user testing conducted with real users are presented in chapter 8.2 on page 93.
The results from the accessibility evaluations will be briefly described which are presented in chapter 8.6 on page 109.

During the design phase of the new prototype, paper-prototypes were developed. The paper-based prototypes can be seen from Figure 108 at page 142 to Figure 113 on page 147.
These paper-prototypes were discussed with our supervisors during the guidance meetings regarding the functionality and the design of the prototype. The meeting minutes are located in the Appendix from page 130 to page 140.
The final version of the new prototype is based on the paper-prototypes developed.

After analyzing the results from the initial evaluation interview, the interview with NAV, the heuristic evaluations, the user tests and the accessibility checking we were able to identify the new requirements for the new prototype.
The requirements that are presented in chapter 5.4 on page 48 are the list of new demands for our prototype and also the basis for the design of our new prototype.

The description of the project management for this project can be found in Appendix 11.1 on page 117. The results for this project are also aided by guidance meetings during the project period. The meeting minutes can be seen in Appendix 11.4 from page 130 to 140.

5.2 Summary of evaluation interview

5.2.1 Full summary

The initial evaluation was performed with our supervisor, hereby referred to as “test person”, in December 2006 at Vinkelbygget located at Grooseveien 36 in Grimstad.
Before the evaluation interview with the test person, we prepared a set of questions. During the interview the test person used the current parental benefit calculator on her computer. We asked the test person questions, received answers and took notes while she was using the calculator.
The aim of the initial evaluation was to identify the functionality of the parental benefit calculator located at NAV’s website earlier in 2006, and address the weaknesses of it regarding the design and usability.

According to the test person, the parental benefit calculator was easy to navigate to at NAV’s website. It was only 2 mouse clicks from the main page of the website.
She said that it was too much information about the parental benefit and leave presented to her on the web site. As a proposal, it could have been a list where the users could select what information they wished to get. The page could also have a little description of the calculator at the top of the page.

According to her, the texts that were meant for helping the user during the calculation were good, but some of the help texts could have a bit more information.
The design of the layout for the calculator was good, but the page had wrong colour contrast. It was hard to identify the tabs for navigation. The calculator was compatible with different web browsers, since there weren't any problems with the layout and functionality.
The presentation of the results after the calculation was unsatisfying. The calculator should also have the functionality of presenting which days a parent can take the maternity leave and not just how many days. The test person was also missing the opportunity to see what parents will get in parental benefit each week, month and year, and not just for each week. The calculator didn't support calculations of the parental benefit and leave if the users wanted to combine a part-time job with their leave.

The results from this first interview were achieved by using heuristic evaluation of the user interface. We used heuristic evaluation for identifying usability problems with NAV's current solution in the beginning of the project. This means that we looked at the current solution and systematically inspected its design for usability keeping the 10 recommended heuristics in mind. These 10 heuristics were presented in chapter 4.1.1 on page 29.
Then, we communicated with our supervisor during this initial evaluation. We discovered that new usability problems were discovered by our supervisor that we didn't discover ourselves.

Based on the initial evaluation of the current solution the functionality of the parental benefit calculator was identified. The weaknesses of the current solution were also addressed when it comes to the design of the calculator and the functionality.

The evaluation interview with our supervisor can be seen in Appendix 11.2 on page 120.

5.2.2 Key findings

These are the key findings from the initial evaluation of the current solution:

- The location of the calculator is good. Easy to find the calculator.
- The pages where the calculator is located have too much information presented. A solution for this is to have a list where the users can select the information they wish to see.
- Some help text may be improved with some more information about a topic.
- A short description of each page is missing. With a short description, the users will easily know what they can do at a specific page.
- Wrong colour contrast on the page.
- Tabs for switching between pages may be improved to be more visible for the users
- Calculator can't display which dates a parent can take the parental leave.
- Calculator can't display how much a parent can get in parental benefit each week, month or year.
- Calculator can't calculate the parental benefit if the parent has a part-time job
- The calculator doesn't discover some missing input fields
- The calculation ”hangs” if the user has missed to enter an input into the calculator

5.3 Summary of Interview with NAV Trygd Grimstad

5.3.1 Full summary

The second interview, which was arranged during the project period, is the interview with a case worker at NAV Trygd Grimstad. The interview took place at the social security office located at the address: Storgata 16B, 4876 Grimstad, on Tuesday 06.03.2007 at 10:00. Before this interview we also prepared a set of questions which were discussed over with my supervisors at a telephone meeting.
The aim of the interview was to get answers about the parental benefit and parental leave, get feedbacks on the prototype developed so far, and feedbacks on the planned functionality so that it would aid the design and development of the new prototype.

The most frequently asked questions that NAV received were questions about when parents could take the maternity leave and how long they could take it. People are asking because they wish to know what benefits they have demands on. NAV didn't have any statistics over how many that contacts the social security office for help regarding the parental leave and benefit, but assumed that there were about 40-50 enquiries per month. Most of the enquiries are over by phone and most of the parents that call NAV haven't used the internet for looking up information. NAV knew this because the answers on the questions that the parents had are already located at NAV's web site.
NAV thinks that the parental benefit calculator and the information on the internet are reducing the number of enquiries, but there aren't that many that knows about the calculator. They haven't received any feedbacks regarding the design, the functionality and usability from parents that have used the calculator. NAV is also offering parents to have a meeting with a caseworker if the parents wish to get some answers on questions regarding the parental benefit and leave.

The graded withdrawal option is new from 01.01.2007 so there haven't been many questions about it. The graded withdrawal option is a more flexible system than the previous arrangement which was called "time account arrangement" before 01.01.2007. Parents are now allowed to have a more flexible combination of part-time jobs with their parental leave.
The current parental benefit calculator didn't support this kind of calculation and the reason for this is that NAV wished to have a simple calculator as possible. If the calculator was too difficult to use, NAV was afraid of that no one would take the calculator in use.
They said that the calculator was just to act as an extra helper in addition to the information on the internet. It didn't support calculations with time account arrangements, because there weren't many parents that took this option in use. Most parents wish to spend their time at home with the newly born child/children.

NAV doesn't know if the new solution that they currently are working on shall support calculations with graded withdrawal, but they doubt it. The reason for this is that such a calculator can easily be too advanced and too difficult to use. Another reason is that the graded withdrawal arrangement is quite new, and just a few knows about this kind of arrangement.
NAV thinks that it will help them to reduce the enquiries from parents if a new calculator supported such calculations, but in order to achieve this, the parents are forced to get to know the rules for calculation of the parental benefit.
It is a great idea to let the calculator support such calculations, but one has to keep in mind that the calculator has to be easy to use.
The feedbacks from NAV on the prototype so far were good. It will be possible to let the new prototype support calculation of the parental benefit and leave in combination with graded withdrawal. The feedbacks on the planned functionality were also positive. It would give the parents a better overview and understanding of the parental benefit and leave.

NAV thinks that a new solution of the parental benefit calculator will help them to reduce the influx at the social security office over time if more parents use the internet and the calculator becomes more popular. But they doubt that the calculator will be able to take over their tasks completely. This is because of that some of the necessary information that the parents have to provide the calculator with has to be obtained through NAV, so that the calculation of the parental benefit can be correct.

We made use of a method called structured interview and from this interview we managed to understand the idea behind the parental benefit calculator and obtained information about the parental benefit and leave, together with the graded withdrawal arrangement. At the end of the interview we received some feedbacks on the prototype that we had developed so far.

The interview with NAV can be seen in Appendix 11.3 on page 120.

5.3.2 Key findings

These are the key findings from the interview with the NAV officer in Grimstad:

- The calculator should give answers on when a parent can take the parental leave and how long they can take it, because these are the most frequently asked questions.
- An online calculator is for sure reducing the number of enquiries.
- The graded-withdrawal option is rarely used by parents or parents-to-be, since it is a new arrangement.
- Most parents wish to stay at home with their children, so graded-withdrawal is rarely used.
- The current solution was designed and developed as simple as possible, because NAV thinks that no one will use it if it is too difficult to use. It was just meant as an extra support tool.
- The new prototype designed so far was good. NAV thinks that it will provide great support for the users.
- NAV thinks that a new solution of the parental benefit calculator will reduce the influx at the social security offices.
- NAV doubts that an online calculator will be able to take over their tasks completely.

5.4 Requirements

The following sections present the requirements for the parental benefit calculator prototype. These requirements are gathered and analyzed during the whole project period.

5.4.1 Functional Requirements

Requirement no.	**Description**	**Priority**
REQ-1	The prototype must be able to receive input from a user	High
REQ-1.1	The prototype must be able to store the user inputs to a database	High
REQ-1.2	The prototype must be able to update the user inputs to a database	High
REQ-2	The prototype must perform calculations based	High

	on the users' inputs	
REQ-2.1	The prototype must be able to calculate the parental benefit	High
REQ-2.2	The prototype must be able to calculate the dates for parental leave	High
REQ-2.3	The prototype must be able to calculate the parental benefit with graded-withdrawal	High
REQ-3	An administrator can modify the content of the database	High
REQ-3.1	An administrator can add, delete and change a calculation rule in the database	High
REQ-3.2	An administrator can add, delete and change a help text in the database	Medium
REQ-3.3	An administrator can add, delete and change a user in the database	Low
REQ-4	The prototype must be able to switch style on and off	High
REQ-4.1	A user can change the font size	High
REQ-4.2	A user can change the colour contrast	High
REQ-4.3	A user can choose to display the text in Norwegian	High
REQ-4.4	A user can choose to display the text in other languages	Medium
REQ-5	The prototype must be able to let users log in	Will not be implemented
REQ-5.1	The prototype must read out data belonging to a user from a database	Will not be implemented
REQ-5.2	The prototype must be able to update the data belonging to a user to a database	Will not be implemented
REQ-6	The prototype must be able to present the results from the calculations	High
REQ-6.1	Present the parental benefit as numbers	High
REQ-6.2	Present the parental leave in a graphical drawn calendar	High
REQ-7	The prototype must detect if users have missed to enter the required inputs into the calculation	High

Table 4: Requirements for the new prototype

5.4.2 External Interface Requirements

The external interface requirements are presented in Table 5.

User Interfaces	The user interfaces that are needed to operate the prototype are a mouse and a keyboard.
Hardware Interfaces	In order to use the prototype, it requires that the users have a computer with an internet connection.
Software Interfaces	The prototype needs to be deployed on a web server, Apache webs server for instance, and have a working connection with a MySQL database.
Communication Interfaces	The prototype is designed for using with any web browser that supports Javascripts. HTTP will be used as the communication standard. Internet Explorer, Mozilla Firefox, Opera etc.

Table 5: External Interface Requirements

5.4.3 Non-functional Requirements

The non-functional requirements are presented in Table 6.

Performance Requirements	The prototype must be able to manage multiple connections at the same time since it is going to be offered as a free public service. The prototype must handle up to 500 users at the same time. The prototype must be able to calculate and present the results to the users in less than 15 seconds. The prototype must be available 24 hours a day. Multiple administrators can edit the data in the database at the same time.

	Prototype must be able to store at least 3 million users.
Security Requirements	If and when Requirement-5 is be implemented in the future, the user authentication process must be secured. The users will have to enter their social security number and a PIN code before they can use the parental benefit calculator. The calculator will store the input data from the users and read them out the next time the users log in again.
Software Quality Attributes	The prototype must be developed as a web application and in such a way that it will be easy to maintain and further developed. It must also be easy to use, since the target group of users could be anyone, and therefore web usability and accessibility are important criteria for the prototype.

Table 6: Non-functional Requirements

5.4.4 Other Requirements

The prototype must be developed using Open Sources and Open Standards which makes use of a MySQL database or a PostreSQL database. The web pages shall be developed by using HTML. The transfer protocol shall be HTTP, and the web pages can be reached at an URL.
The preferred programming language for developing the web application is either Ruby or Python, but Java or PHP can also be used.
In other words, the prototype must make use of Internet standards, which are among others: web standards, email standards, communication standards, programming language etc.

6 Design Specification

In this chapter we will describe how the new prototype is designed.
The design of our solution is based on the initial evaluation of the current solution performed with our supervisor. The design is also based on the interview with NAV Trygd Grimstad.

Before we could design the new prototype we had to become familiar with the technologies and tools presented in chapter 4.4.2 on page 38. These technologies and tools were going to be use in order to develop the new prototype. The system architecture for our prototype is illustrated in Figure 17.

6.1 Architecture

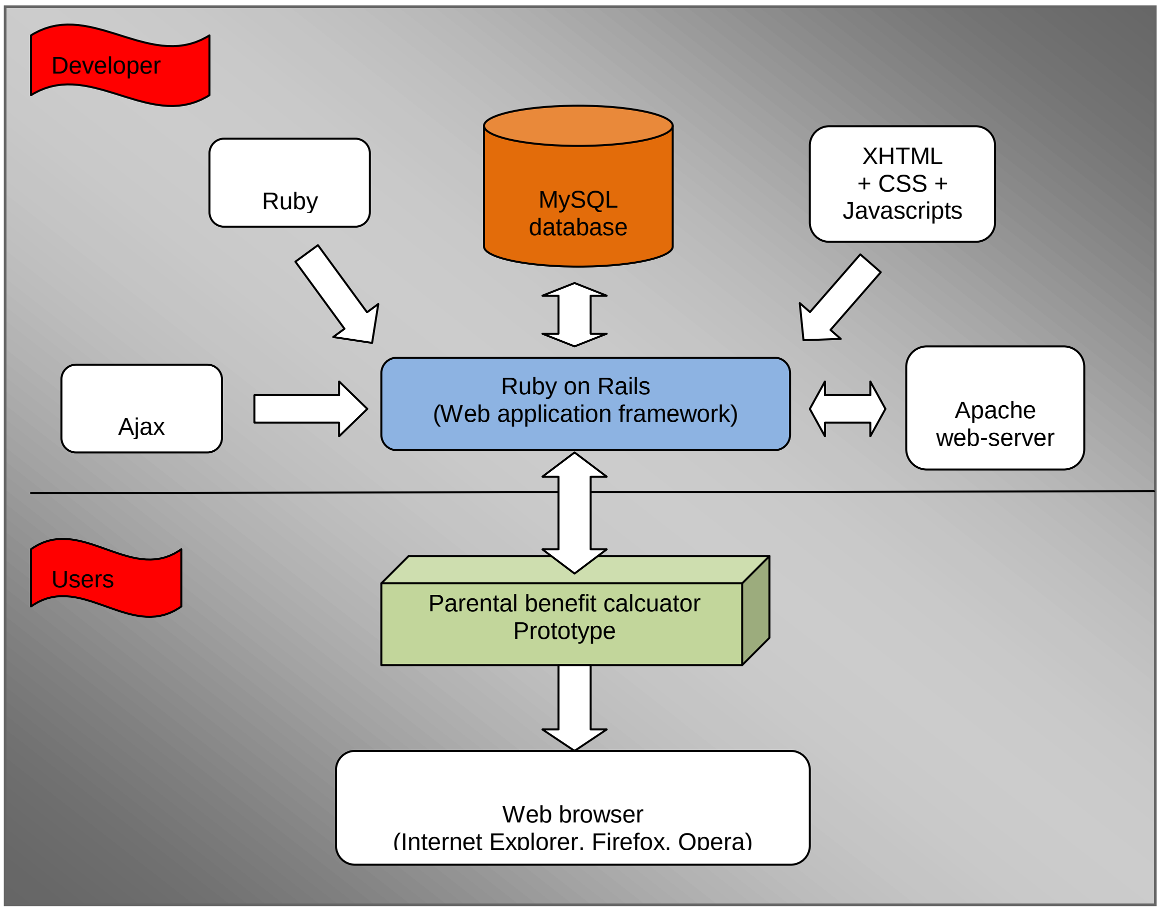

Developer
Ruby
MySQL database
XHTML + CSS + Javascripts
Ajax
Ruby on Rails (Web application framework)
Apache web-server
Users
Parental benefit calcuator Prototype
Web browser (Internet Explorer, Firefox, Opera)

Figure 17: An illustration of software tools and their relationship used during implementation

The prototype will be built as a web application by using Ruby on Rails, XHTML, CSS and Javascripts. The web application, which will be the parental benefit calculator, will make use of a MySQL database and deployed on a web server.

XHTML together with CSS will be used to create the view and layout of the prototype. Ruby shall be embedded into the view so that the view can have dynamic content, which means that the prototype will display different things to the users according to the users' choices. The view shall make use of Javascripts to offer the users some additional functionality.

The MySQL database will store the users' inputs when they use the prototype and Ruby on Rails will make use of the stored values to calculate the parental benefit and leave. In addition, the database will store the rules for calculation and the texts for helping out the users when they interact with the prototype. Ajax will be used to handle the displaying of the help texts.

The architectural pattern MVC, Model-View-Controller, will be used to separate the data that the prototype makes use of, the prototype's user interface and the user interactions with the prototype.

Model
The models are objects that are representing the data tables in the MySQL database. These models will be used to read data from the database and update data in the database.

View
The view is showing the users the graphical user interface of the parental benefit calculator prototype. This is where XHTML, CSS etc comes into play.

Controller
The controller ties the model to view. This means that it accepts inputs from the users and instructs the model and view to perform actions based on how the users use the prototype.

Figure 18 is illustrating the MVC pattern and how it works in Ruby on rails.

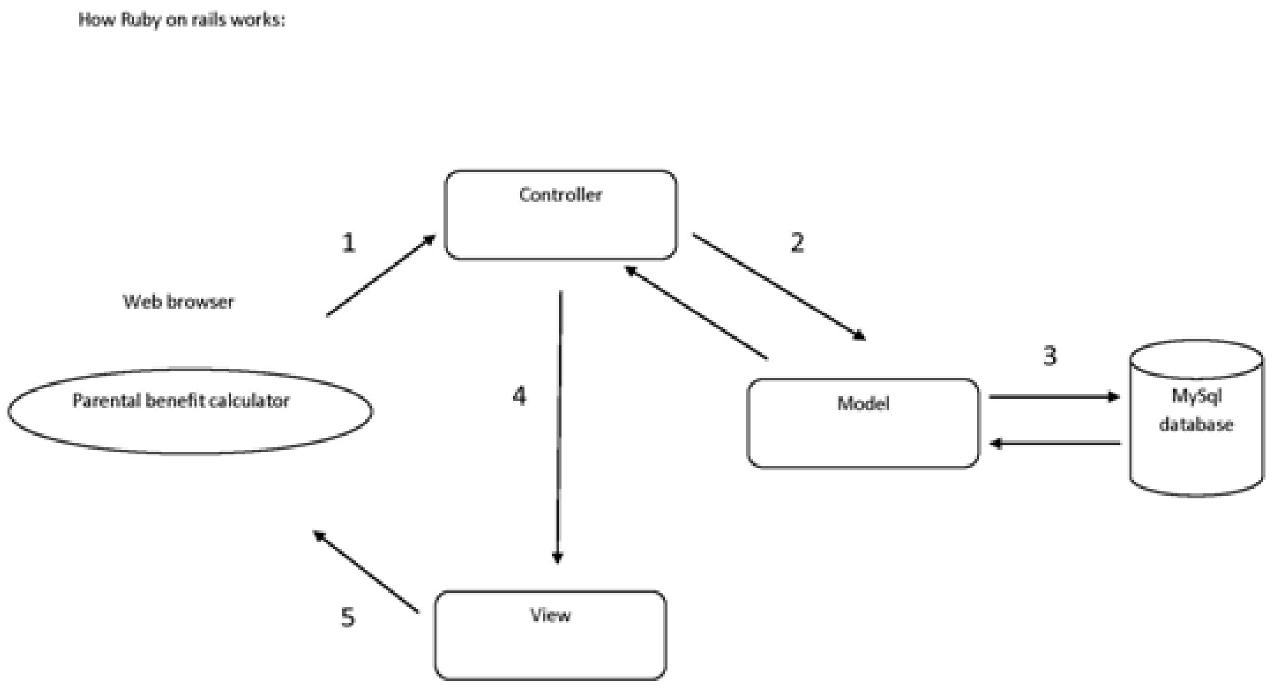


Figure 18: MVC in Ruby on rails

6.2 Graphical User Interface

The graphical user interface is designed according to our paper-prototypes which we developed in the beginning of the project. These paper-prototypes can be seen in Appendix 11.6 on page 142.

6.2.1 GUI for users

The choice to keep much of the same layout, that the current parental benefit calculator had, in the new prototype was based on the two interviews that were performed. The results from these interviews are presented in chapter 5.2 on page 45 and in chapter 5.3 on page 47.
The decision was also based the user testing with real users which is presented in chapter 8.2 on page 93.

It will be possible for the users to display the calculator with different colour contrasts, and with different font sizes. The calculator can also be shown in different languages and the users are able to switch the layout styles.
At the end of the calculation the users can also get an overview of the parental leave, which means that the mandatory days for taking the parental leave are highlighted in a calendar. This is used as an aid to the users for planning their parental leave.
The users will also be able to pick a date from a calendar instead of entering the date with numbers, when they are asked to provide the calculator with the period for calculation of the max date[41]. These requirements are also stated in chapter 5.4 on page 48.

6.2.2 GUI for administrator

The graphical user interface for the administrator will have a simple user interface design which provides the necessary functionalities for editing the database. The intention of this graphical user interface is to let the administrator easily manipulate the relevant data, such as users, calculation rules and help texts, in the database.

6.3 Functionality

The functionality for our new prototype is much the same as NAV's current solution, but our prototype has been designed to have enhanced functionality for improving the usability. Chapter 3.1.1 presented how the current solution offered by NAV works.

6.3.1 Functionality regarding users

Looking at the results from the interviews presented in chapter 5 on page 45, the needed functionality for the prototype was identified. The input data required for the calculations of the parental benefit and leave were identified by the papers that were provided by NAV Trygd Grimstad. These can be found in Figure 106 and Figure 107 on page 141.

The prototype has to support calculations of the parental benefit with graded withdrawal. This means that the new prototype will have much of the same functionality as the current parental benefit calculator, but it will have some extra added functionality and improved usability. It will be possible for the users to calculate the parental benefit when they combine it with graded

[41] Max date is the last day for taking out the parental benefit.

withdrawal. The users can choose to show how much benefit they will get per week, per month or per year.
The prototype will also have the functionality for finding out and showing the dates for when a mother must take the maternity leave.

6.3.2 Functionality regarding administrator

The needed functionality regarding an administrator is to manipulate the relevant data in the database. The functionality for adding, changing, removing a calculation rule or a help text is needed. . These are also stated as the requirements in chapter 5.4 on page 48 and are the functionalities behind the GUI for administrators described in chapter 6.2.2 on page 54.

6.4 Database

The database in the prototype will be used to store and retrieve information. The design of the database is based on the initial evaluation of the current parental benefit calculator, the interview with NAV, the requirements that are gathered during the project period and the discussions with the supervisors. As mentioned earlier, these results can be seen in chapter 5 on page 45.

The database is used to store the input from the users, read out the input to perform calculations and save the results. Its purpose is also to let the administrator easily add, change or delete calculation rules to be used in the calculator prototype. In addition, an administrator is able to define the help texts in several languages which are used by the users during a calculation. The administrator is also able to perform statistics with the data in the database, for example to see which help text is the most used. By getting to know such information the prototype can be further improved over time.

Following, the tables in the database and their relationships are presented and described.

6.4.1 Tables in the database

The database tables were designed to be able to store the information that we needed in order for the functionality of our prototype to work.

6.4.1.1 Users

The “users” table will consist of information about users that use the calculator prototype. The purpose of this table is to store each user’s information such as user id, user name, name and address etc. The idea of this table is to store the calculations that the users have performed if the prototype supports a log in system. Each user will be given a unique id and a unique user id; this could be the social security number for example. By reading out the data in this table, one can find what inputs the users have made before. This table will therefore be serving less functionality in this prototype, but may be extended in future works.

6.4.1.2 Calculations

The “calculations” table will store every single calculation. Each calculation will be given a unique id. The table will also store the language that the user has chosen when they use the parental benefit calculator prototype. By storing this information one can see which language is most used, and what language a specific user prefers. This table will also store the date and time for when the calculation started and ended. This information can be statistical analyzed to see how long the calculations took and at what time of the day the prototype is used etc. It will also be possible to see how many calculations each user has performed.

6.4.1.3 Userinputs

The “userinputs” table will store all the user inputs as the name suggests. The information that is stored will be used by the prototype to perform calculations and can also be used to perform statistics to maybe see what scenarios are the most common ones.

6.4.1.4 Results

The “results” table will contain the results for a calculation. These results are presented to the users at the end of the calculation.

6.4.1.5 Calculationrules

The “calculationrules” table will consists of all the calculation rules that are used for the calculation of the parental benefit and maternity leave. An administrator can add, change or delete the rules in this table. Each rule will be given a unique id and name.

6.4.1.6 Guihelptexts

The “guihelptexts” table will keep track of which help text a user has used during a calculation of the parental benefit and maternity leave. The table’s content will be the calculation id, which is the current number of calculation, and the id of the help text. An administrator has the possibility to add, change or delete a help text, in several different languages.
This table can be used to see which help texts that are most used, and this indicates what information that is hard to understand for the users.

6.4.1.7 Currentcalculationrules

The “currentusedrules” table will store the id of the users and the id of the rules. This will ensure that when the rules are changed, users’ calculations will be updated with the new rules.

6.4.1.8 Usedguihelptexts

The “usedguihelptexts” table will store the id of the users and the id of the helptexts. This table will keep track of which helptexts each user has used. This will indicate which help texts are the most used ones, and give an administrator the ability to improve the prototype later.

6.4.2 Database Relationship

The relationship between the database tables was decided in cooperation with our supervisors. Figure 19 shows the relationships between the tables in the database.
The aim by having such a relationship is to have a clean and efficient database. When a user is removed from the system, all the calculations that are performed by that user are no longer needed. Neither is the rules used in the calculations, the help texts that the user has used, the user's inputs and results are needed.

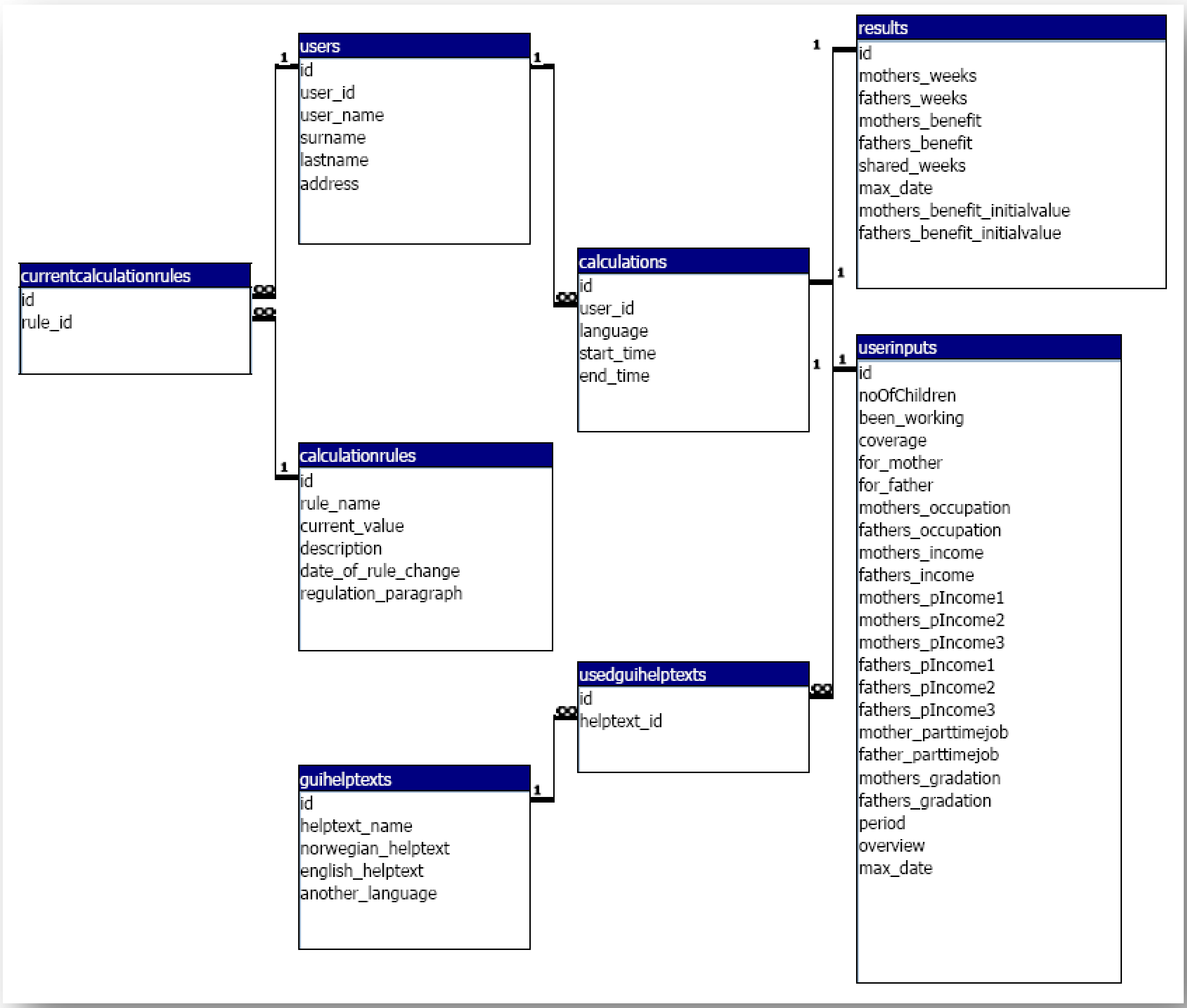


Figure 19: The database relationship

The following text will describe how the relationships are designed.

users – calculations

One-to-many: A user can have many calculations. When a user is deleted from the "users" table the calculations that belong to that user will also be deleted. If the user's id is changed in the "users" table the user's id in the "calculations" will be updated.

users – currentcalculationrules

One-to-many: A user can have many used rules. All the rules in the "rules" table will be connected to every single existing user id. If a user is deleted all the records belonging to that user in the "currentcalculationrules" table will be deleted. If the user id is changed the user id in the "currentcalculationrules" table will be updated.

calculationrules – currentcalculationrules
One-to-many: A rule can be found in many used rules. Each user id in the "currentcalculationrules" table can use all the rules, but each rule can only be used once. If a rule is deleted the record in the "currentcalculationrules" table will also be deleted. If an update of a rule occurs, the "currentcalculationrules" table will be updated.

calculations – userinputs
One-to-one: A calculation has one related record in the "userinputs" table. This kind of relationship is the least common type, since one can have all the related information in one table. However, for clarity reasons and because of the logical separation, the information is chosen to be broken out into two tables. The related calculation ids will always be the same in these two tables. If a calculation is deleted the related user inputs will be deleted.

calculations – results
One-to-one: A calculation has one related record in the "results" table. Because of logical separation and for clarity reasons the information is also chosen to be broken out into two tables here. The related calculation ids in these tables will always be the same. If a calculation is deleted or changed the related results will also be deleted or updated.

userinputs – usedguihelptexts
One-to-many: Each user id is related to many help texts. If the user is removed from the system, or the calculation is deleted all the record belonging to that user id will be deleted too. If the user id is changed the user id in the "usedguihelptexts" table will be updated.

guihelptexts – usedguihelptexts
One-to-many: A help text can be found in many used help texts. Each user id in the "usedguihelptexts" table can use any of the help texts. If a help text is deleted or changed, the related record in "usedguihelptexts" table will be deleted or updated.

6.5 Usage and behaviour

The parental benefit calculator prototype will be used by the users to calculate how much they will get in parental benefit and how long maternity leave they will get. The usage and the behaviour were specified after we had stated the requirements for the system. The requirements can be found in chapter 5.4 on page 48.

The following figures will show the different use cases for the prototype together with a description for each use case.

6.5.1 Usage for Users

When a parent uses the prototype to calculate the parental benefit and leave, the parent has to provide the calculator with the necessary data. When the prototype has received all those data it will make use of the calculation rules to perform the calculation. Figure 20 is illustrating this.

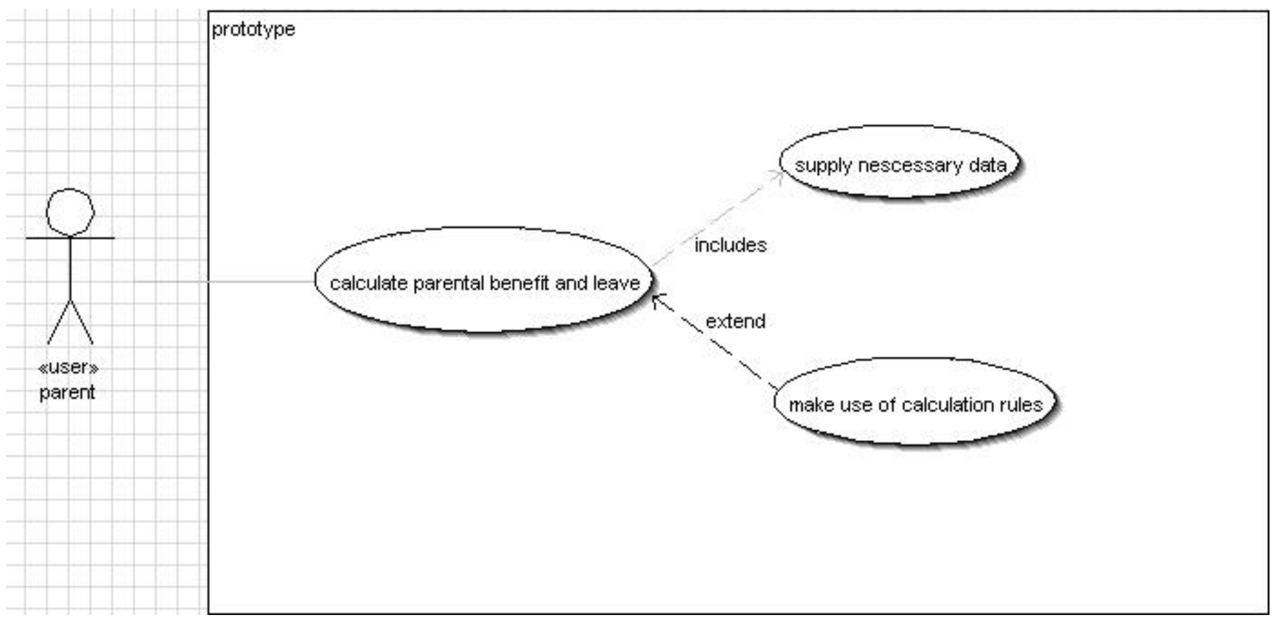

prototype
«user»
parent
calculate parental benefit and leave
supply nescessary data
includes
extend
make use of calculation rules

Figure 20: UML Use case diagram for calculation of parental benefit and maternity leave

When the prototype has finished the calculation, the results are presented to the users. A parent can choose to show and hide the results, and see the results as numbers or as a graphical presentation. This is illustrated in Figure 21.

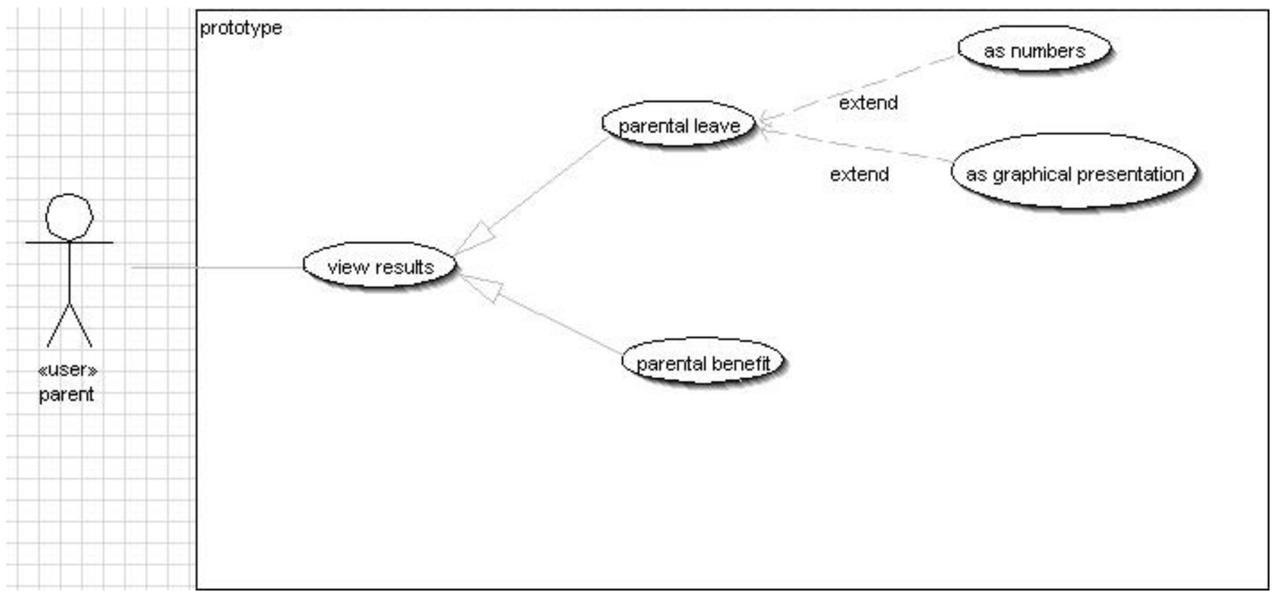


Figure 21: UML Use case diagram for viewing the results

If a parent has chosen to combine the maternity leave with a part-time job, the parent can provide the prototype with the necessary data and recalculate the parental benefit. The results will be presented to the parent immediately. Figure 22 illustrates the recalculation of the parental benefit.

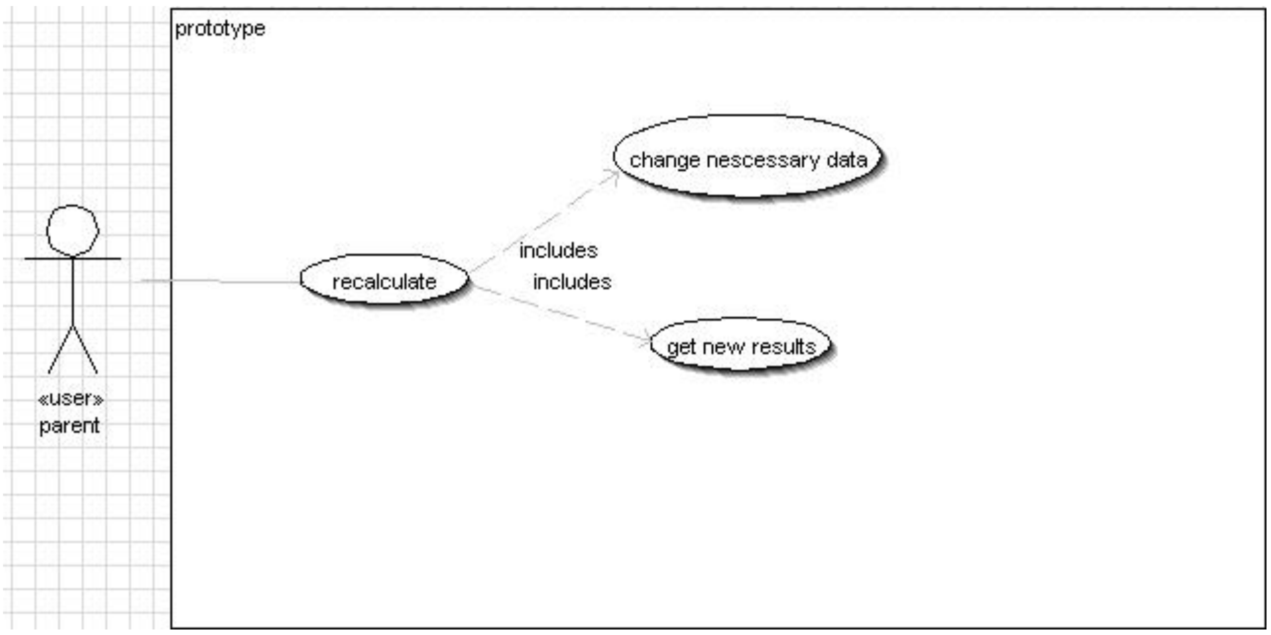


Figure 22: UML Use case diagram recalculation of the parental benefit and maternity leave

This is a proposal for future work and will not be implemented in this prototype. The prototype will ask a user to login to the system before calculation of the parental benefit and maternity leave. The prototype will then identify the user and find out if the user has performed any calculations earlier. If the user has performed calculations before, the prototype will show the last calculation to the user and let the user edit the inputs. If the user hasn't performed any earlier calculations, the prototype will let the user perform a new calculation. Figure 23 depicts the login system.

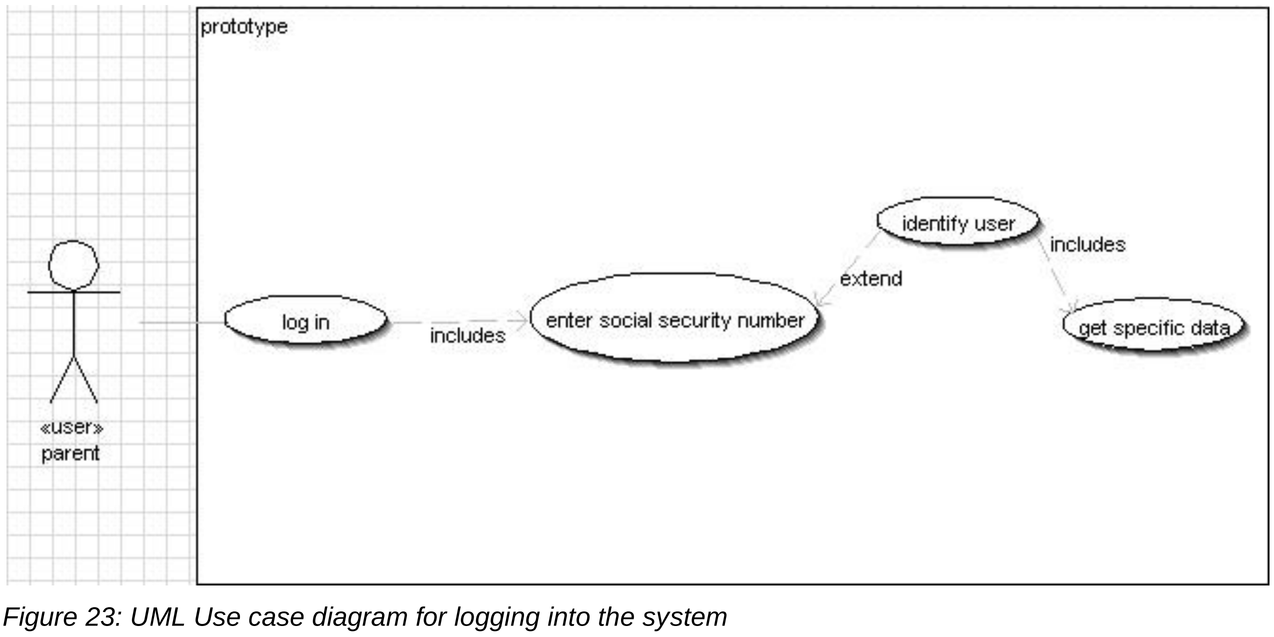


Figure 23: UML Use case diagram for logging into the system

6.5.2 Usage for Administrator

The administrator can get an overview of the content of the database. He or she will also be able to view the data in each table that exists in the database. Figure 24 illustrates this.

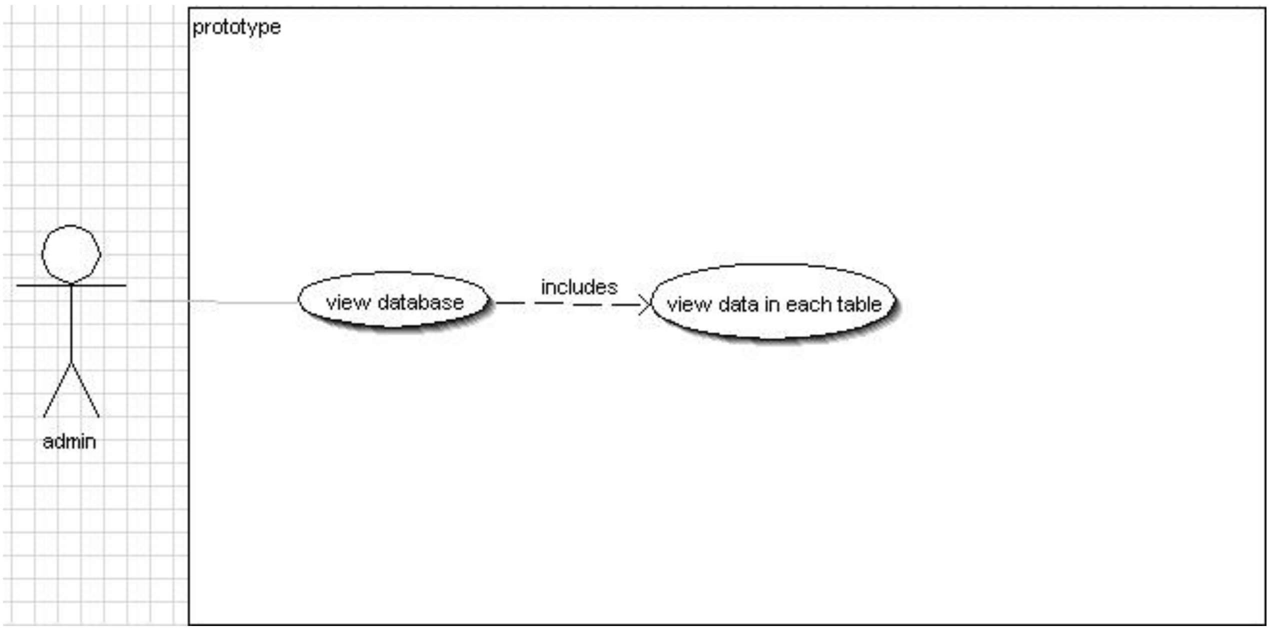


Figure 24: UML Use case diagram for viewing the database content

The administrator can maintain the database by modifying the database design and/or content. Else, the administrator can switch to a replica database so that the prototype temporarily uses that database and maintain the main database. This is illustrated in Figure 25.

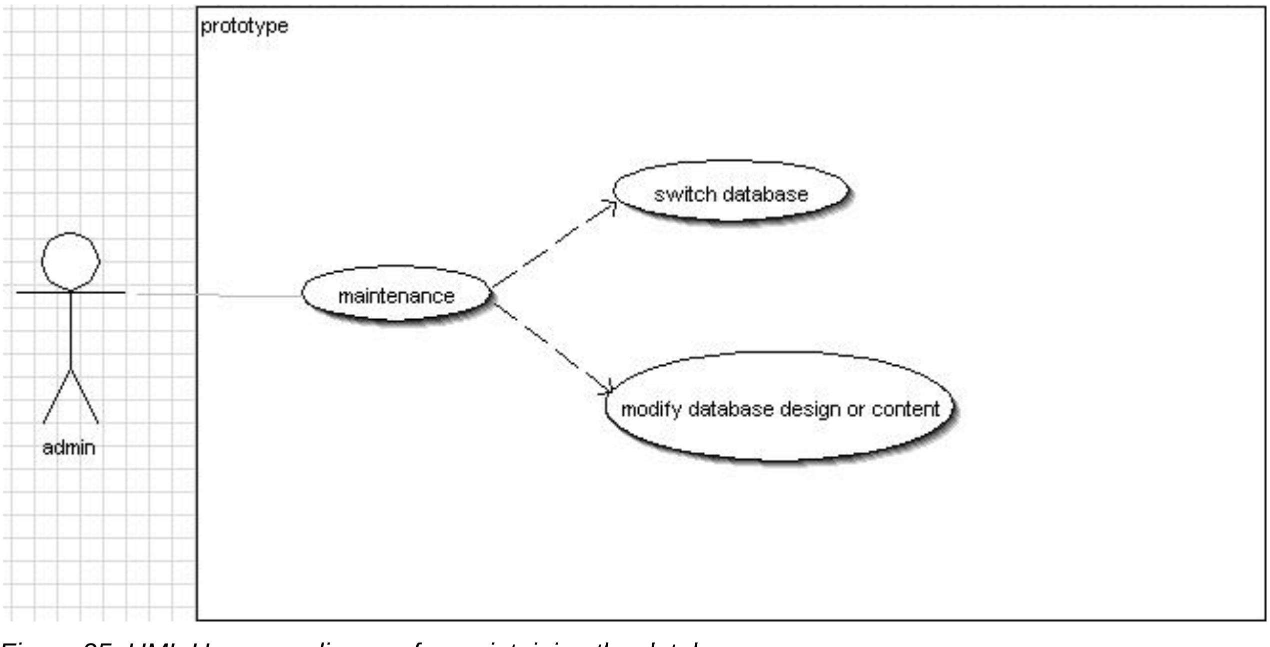


Figure 25: UML Use case diagram for maintaining the database

The administrator can modify the users that are registered. It will be possible to add a new user or modify an existing user, but it is also possible to a delete an existing one. This use case is illustrated in Figure 26.

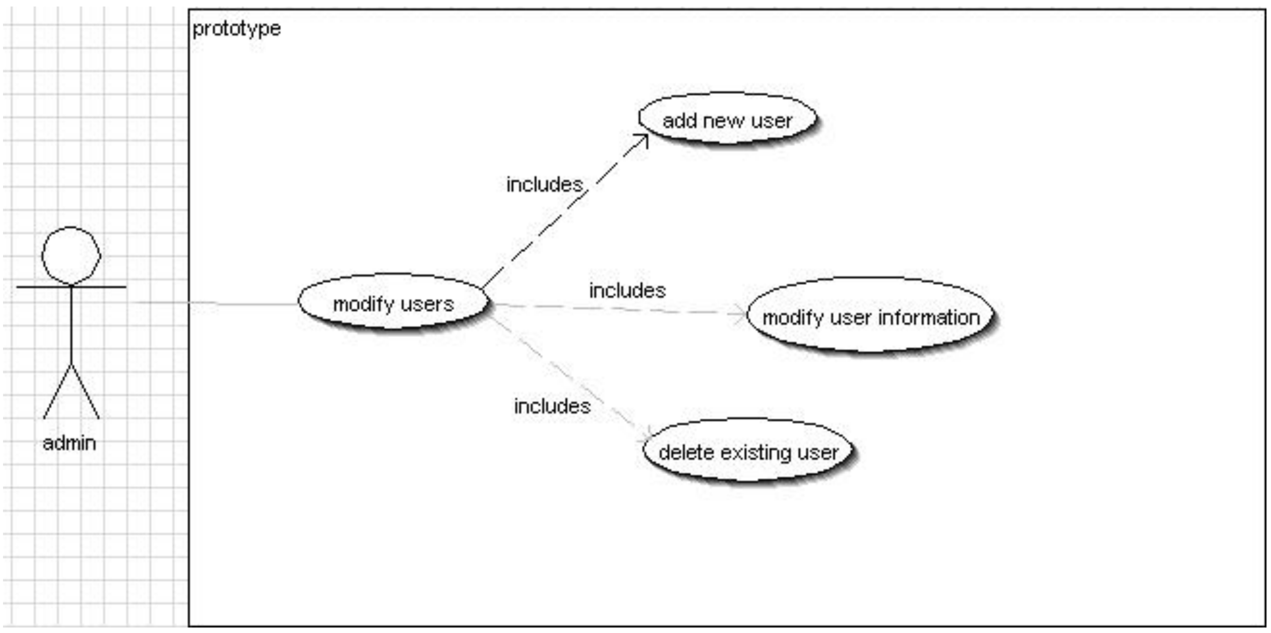


Figure 26: UML Use case diagram for modifying the users

The administrator can choose to modify the calculation rules in the database. It will be possible to add a new rule, edit an existing rule or delete one. Figure 27 is illustrating this.

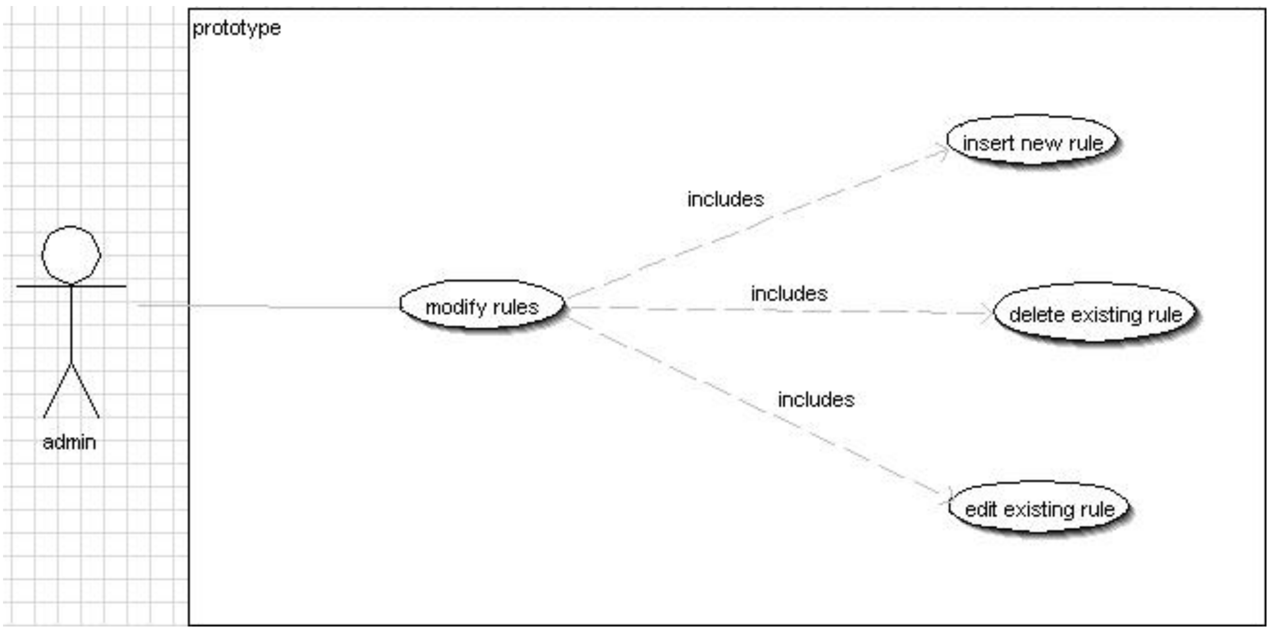


Figure 27: UML Use case diagram for modifying the calculation rules

The administrator also has the possibility to modify the help texts in different languages. The administrator can insert a new calculation rule, edit an existing rule or delete one. Figure 28 illustrates this.

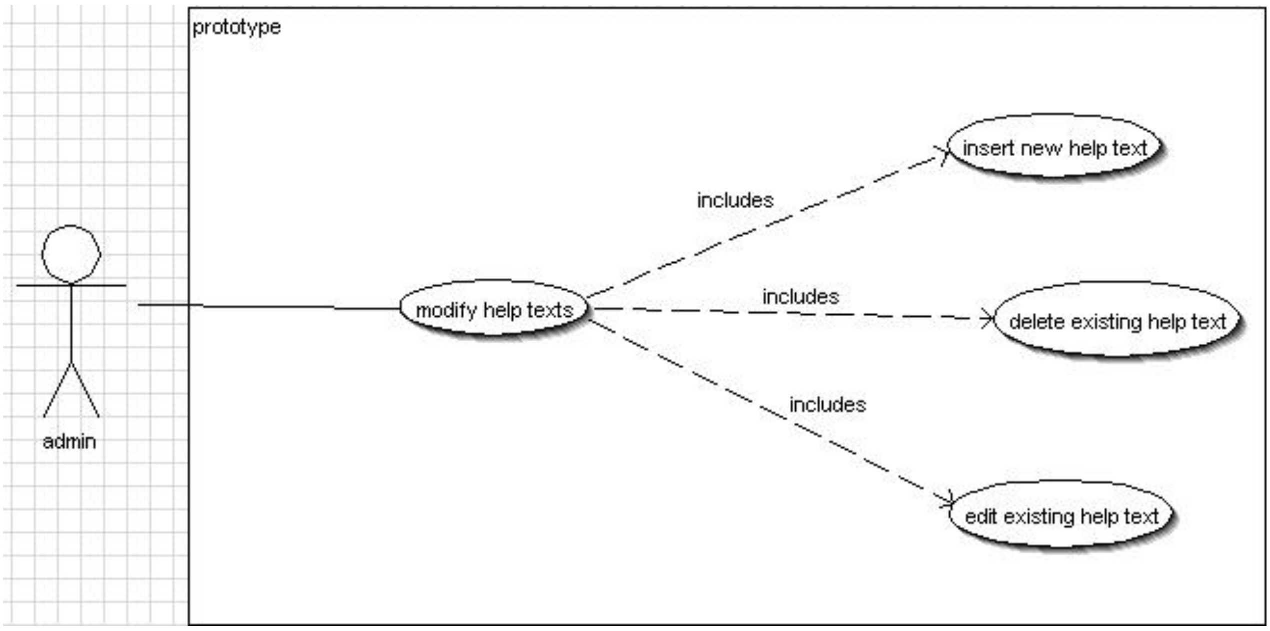


Figure 28: UML Use case diagram for modifying the help texts

This is a proposal for future work with the prototype and will not be implemented. The administrator can get the data out of the database, analyze them to compile statistics. This would be useful to get a clearer overview of the content of the database. Figure 29 is illustrating this use case.

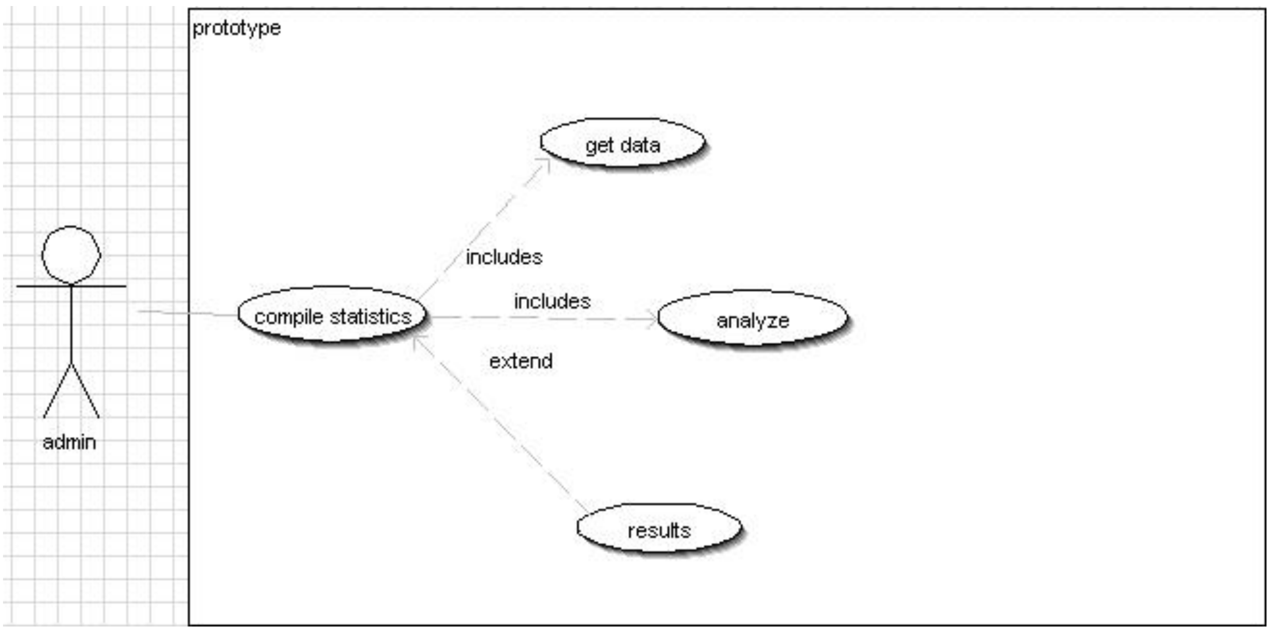


Figure 29: UML Use case diagram for compiling statistics of the database data

6.5.3 Behaviour

Any user can use the prototype by loading a web browser, such as Internet Explorer, Firefox, Opera etc. The user can then navigate to the location of the prototype which a specific address. When the prototype is being loaded, the controller will be notified and the actions defined in the controller will be executed. The controller will interact with the model to get the necessary data, and the model will be interacting with the database, which holds all the information, to become updated. Then, the controller will tie the model and the view together and display the prototype to the users in the web browser. Figure 30 presents the order of the events that are taking place when the users are interacting with the prototype.

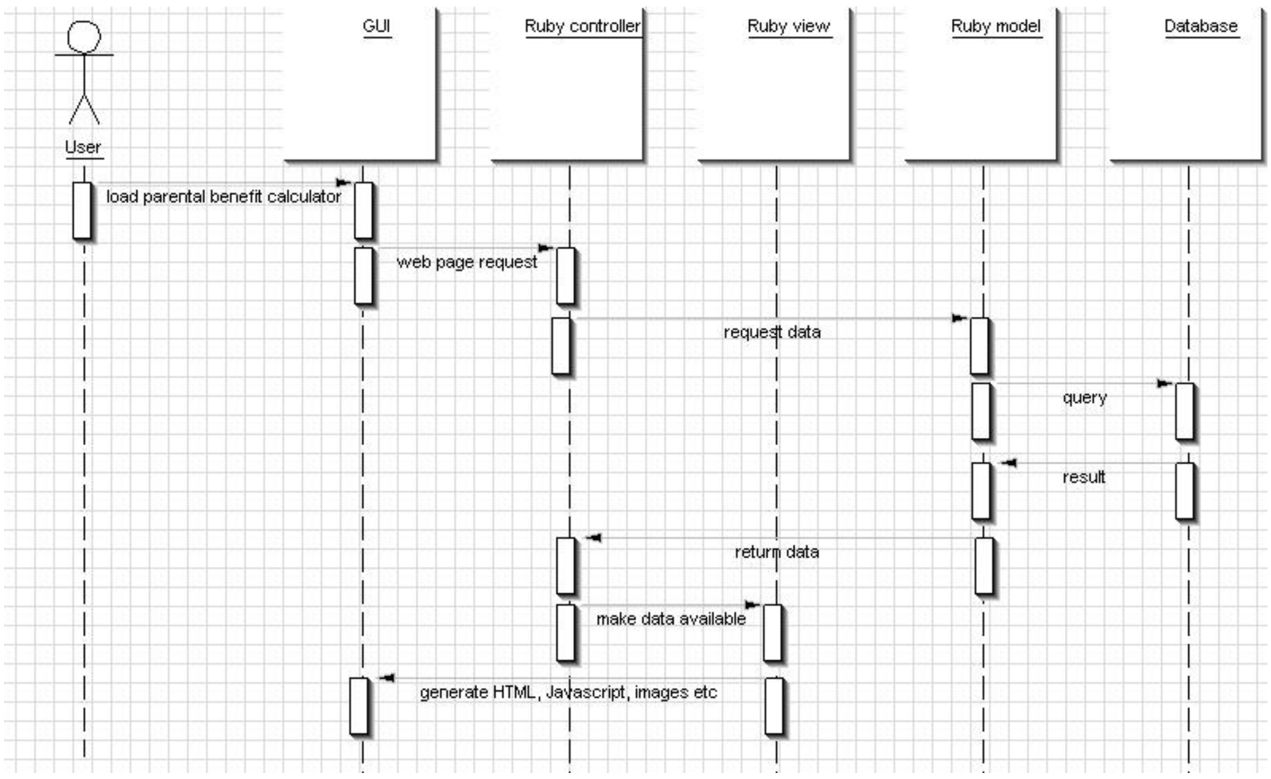


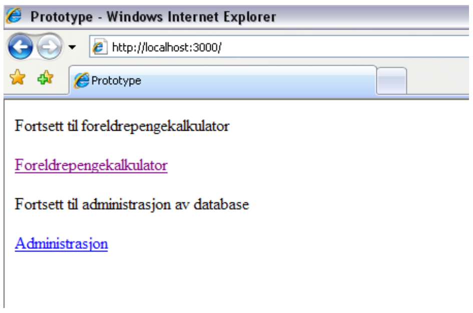
Figure 30: UML Sequence diagram for the system behaviour

7 Implementation

7.1 Graphical User Interface

The prototype is divided into two parts. The first part is the parental benefit calculator prototype and the second part is the administration of the calculator.

Our new parental benefit calculator prototype is deployed at this address: http://128.39.61.95:3000/.
By browsing to the web application located on the server, at http://localhost port 3030 on our development computer, one will see two links; one link for the parental benefit calculator and another link for the administration of the calculator as shown in Figure 31.



Fortsett til foreldrepengekalkulator

Foreldrepengekalkulator

Fortsett til administrasjon av database

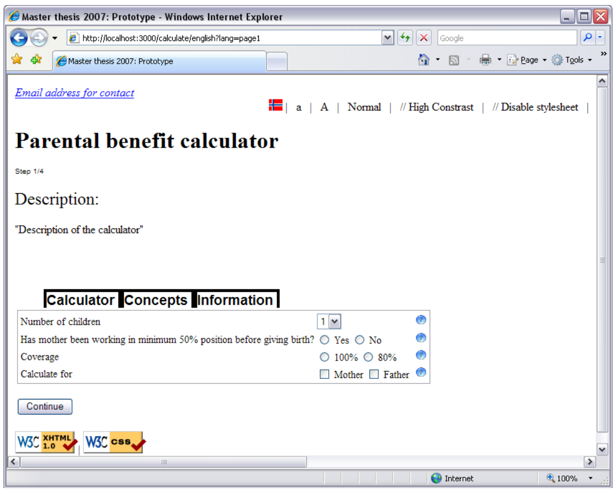
Administrasjon

Figure 31: Prototype screenshot - Main page

By following those links the user of the system will be directed to the specific pages. When the user clicks one of the links, the different controllers in Ruby on Rails will be initiated, and the web page will be rendered.

7.2 GUI for users

The first page of the parental benefit calculator consists of a menu on the top of the page, a heading, a brief description of the current page, an area where the users can input data, and at last a button where the users can proceed with the calculation of the parental benefit and leave. Figure 32 is showing the first page of the prototype.


Master thesis 2007: Prototype - Windows Internet Explorer
http://localhost:3000/calculate/english?lang=page1
Google
Master thesis 2007: Prototype
Page
Tools
Email address for contact
a | A | Normal | // High Constrast | // Disable stylesheet |
Calculator
Concepts
Information
Continue
Internet
100%

Parental benefit calculator

Step 1/4

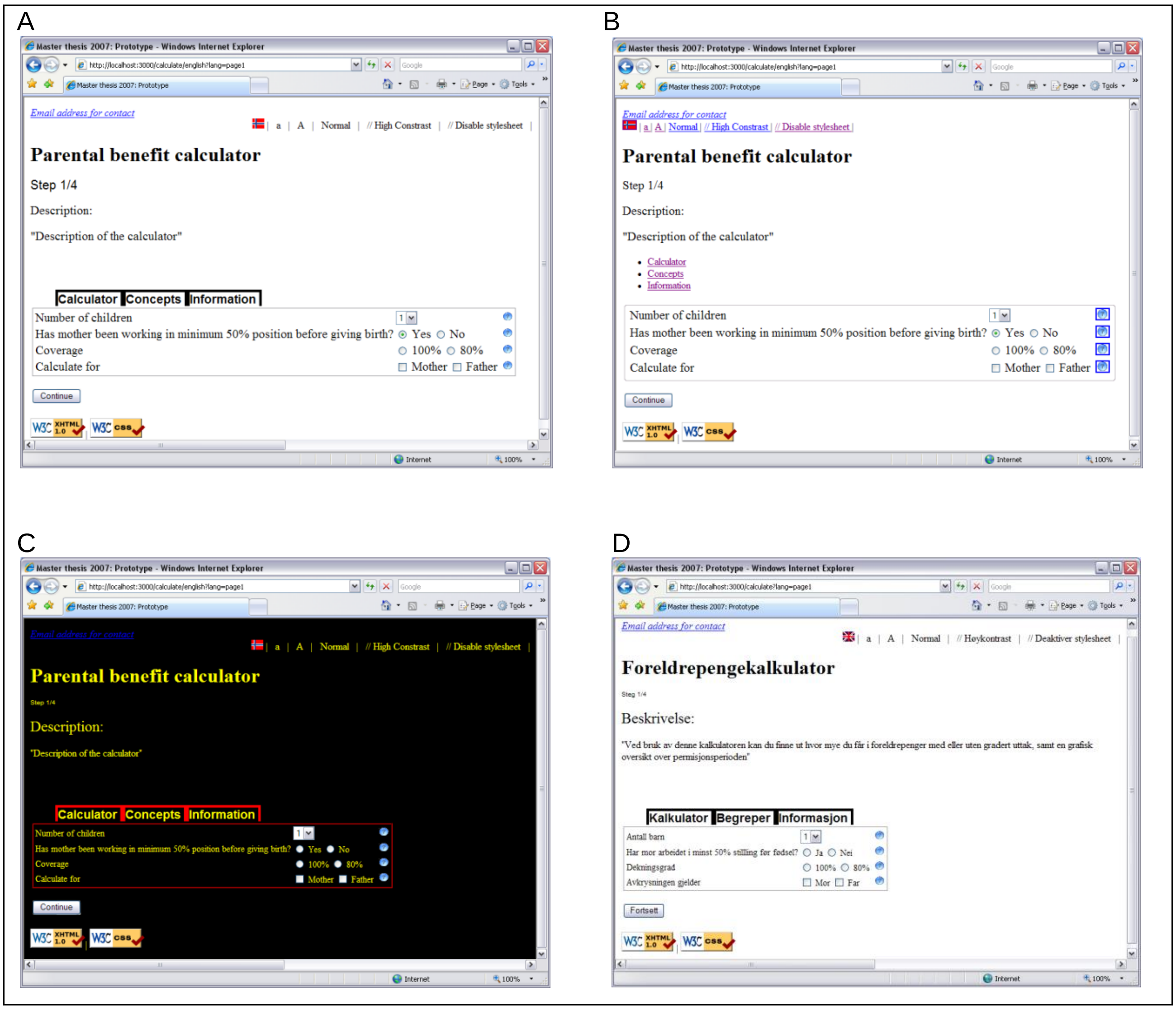
Description:

"Description of the calculator"

Number of children	1
Has mother been working in minimum 50% position before giving birth?	○ Yes ○ No
Coverage	○ 100% ○ 80%
Calculate for	☐ Mother ☐ Father

Figure 32: Prototype screenshot – Parental benefit calculator page 1

The content of the pages for the prototype is dynamically produced by the controller together with the view and model, in Ruby on Rails. The menu on the top of the page can be used to configure the look of the page such as changing the font size, deactivating the style sheet for the page, showing the page in high contrast or changing the displaying language as illustrated in Figure 33.


A
B
C
D
Master thesis 2007: Prototype - Windows Internet Explorer
http://localhost:3000/calculate/english?lang=page1
http://localhost:3000/calculate?lang=page1
Internet
100%

Email address for contact

a | A | Normal | // High Constrast | // Disable stylesheet |

Parental benefit calculator

Step 1/4

Description:

"Description of the calculator"

- Calculator
- Concepts
- Information

Calculator | Concepts | Information

Number of children	1
Has mother been working in minimum 50% position before giving birth?	[✓ Yes] No
Coverage	100% 80%
Calculate for	Mother Father

Continue

Email address for contact

a | A | Normal | // Høykontrast | // Deaktiver stylesheet |

Foreldrepengekalkulator

Steg 1/4

Beskrivelse:

"Ved bruk av denne kalkulatoren kan du finne ut hvor mye du får i foreldrepenger med eller uten gradert uttak, samt en grafisk oversikt over permisjonsperioden"

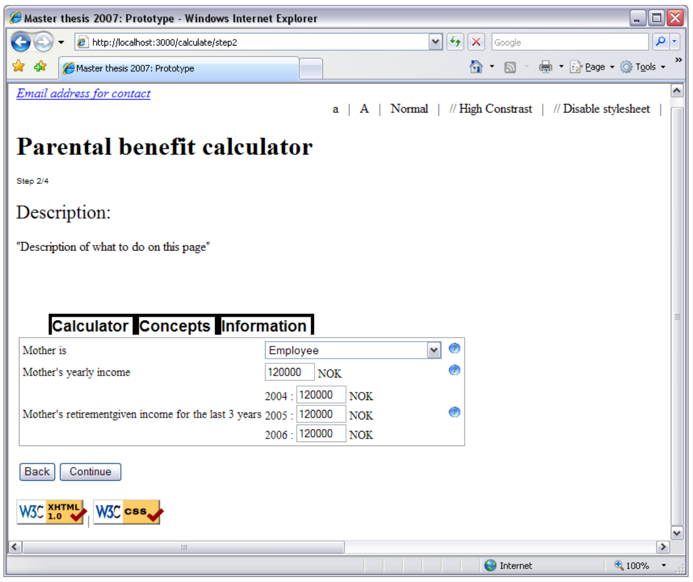
Kalkulator | Begreper | Informasjon

Antall barn	1
Har mor arbeidet i minst 50% stilling før fødsel?	Ja Nei
Dekningsgrad	100% 80%
Avkrysningen gjelder	Mor Far

Fortsett

Figure 33: Prototype screen shot Page 1 - with increased font size (A), without style sheet (B), with high contrast (C), in another language (D)

When the user has provided the calculator with the necessary inputs at page 1, he/she can continue the calculation by pressing the “Continue” button.


Master thesis 2007: Prototype - Windows Internet Explorer

http://localhost:3000/calculate/step2

Email address for contact

a | A | Normal | // High Constrast | // Disable stylesheet |

Parental benefit calculator

Step 2/4

Description:

"Description of what to do on this page"

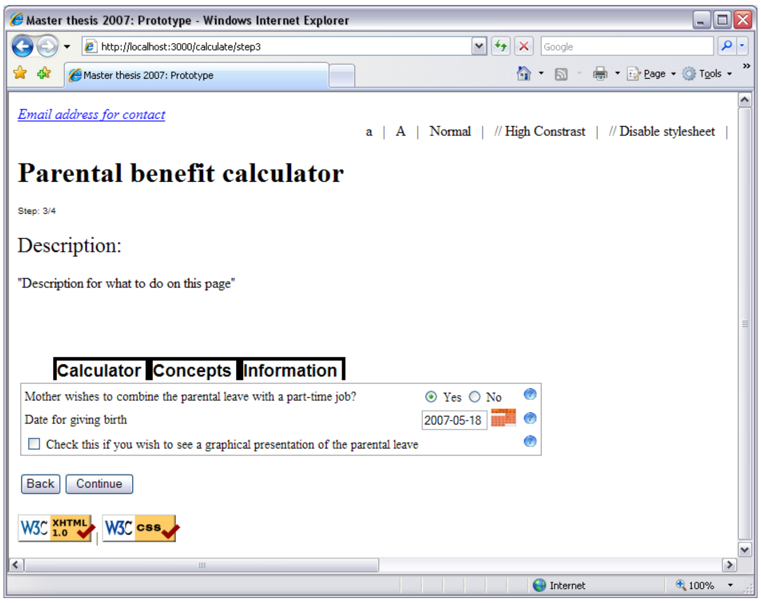
Calculator Concepts Information

Mother is	Employee
Mother's yearly income	120000 NOK
Mother's retirementgiven income for the last 3 years	2004 : 120000 NOK 2005 : 120000 NOK 2006 : 120000 NOK

Back Continue

Figure 34: Prototype screenshot – Parental benefit calculator page 2

The layout for page 2 looks exactly the same as page 1, because all the pages use the same layout and only the content is different. Based on the inputs that the user made on page 1, page 2 will be rendered according to that. From this moment the users are unable to change the language, but all the other "displaying functions" are still available. The calculation can be continued by entering the required data and by pressing the "Continue" button. This is illustrated in Figure 34.
The users are also able to go back to page 1 for altering the inputs that they made.


Master thesis 2007: Prototype - Windows Internet Explorer

http://localhost:3000/calculate/step3

Email address for contact

a | A | Normal | // High Constrast | // Disable stylesheet |

Parental benefit calculator

Step: 3/4

Description:

"Description for what to do on this page"

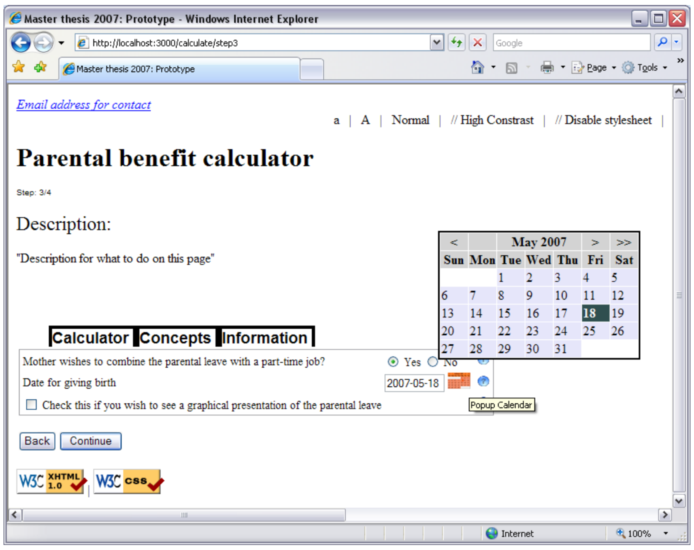
Calculator Concepts Information

Mother wishes to combine the parental leave with a part-time job?	[✓ Yes] No
Date for giving birth	2007-05-18
☐ Check this if you wish to see a graphical presentation of the parental leave	

Back Continue

Figure 35: Prototype screenshot – Parental benefit calculator page 3

Again, at page 3 the users have to provide the calculator with more information. Figure 35 illustrates this. The users can either proceed or go back at this point too.
On this page the users are also able to pick a date from a calendar by clicking on the calendar button on the page, instead of entering the date manually in the text-field as depicted in Figure 36.


Master thesis 2007: Prototype - Windows Internet Explorer
http://localhost:3000/calculate/step3
Master thesis 2007: Prototype
Page
Tools
Internet
100%

Email address for contact

a | A | Normal | // High Constrast | // Disable stylesheet |

Parental benefit calculator

Step: 3/4

Description:

"Description for what to do on this page"

<		May 2007		>		>>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Calculator Concepts Information

Mother wishes to combine the parental leave with a part-time job? [✓ Yes] No

Date for giving birth 2007-05-18

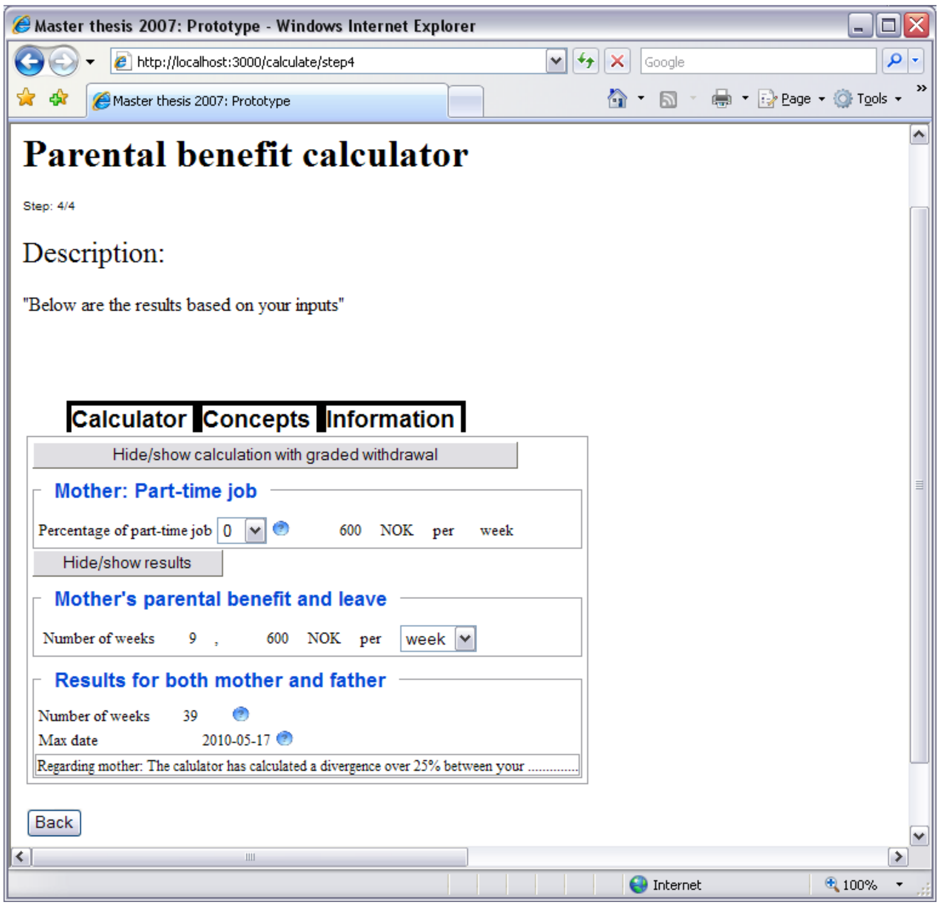
☐ Check this if you wish to see a graphical presentation of the parental leave

Popup Calendar

Back Continue

Figure 36: Prototype screenshot – Parental benefit calculator page 3

At page 4 as illustrated in Figure 37, the results of the calculation will be presented to the users. According to what the users have chosen in the previous steps, the presentation of the results can vary. The users can choose to display how much the parental benefit is in week, month or year. On this page the users also have the possibility to calculate the parental benefit in combination with a part-time job.


Master thesis 2007: Prototype - Windows Internet Explorer
http://localhost:3000/calculate/step4
Master thesis 2007: Prototype
Page
Tools
Internet
100%

Parental benefit calculator

Step: 4/4

Description:

"Below are the results based on your inputs"

Calculator Concepts Information

Hide/show calculation with graded withdrawal

Mother: Part-time job

Percentage of part-time job 0 600 NOK per week

Hide/show results

Mother's parental benefit and leave

Number of weeks 9 , 600 NOK per week

Results for both mother and father

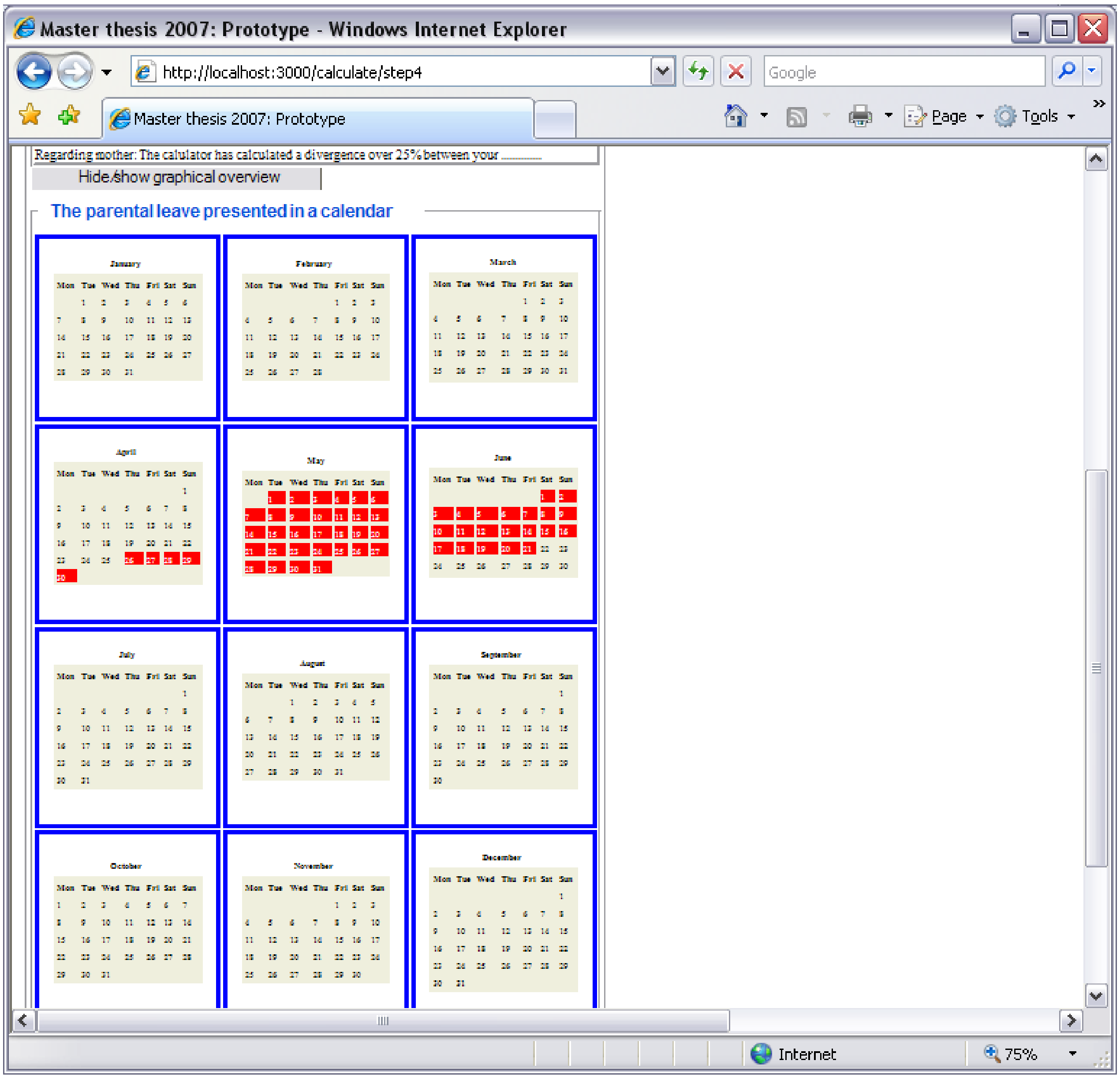
Number of weeks 39

Max date 2010-05-17

Regarding mother: The calulator has calculated a divergence over 25% between your ....

Back

Figure 37: Prototype screenshot – Parental benefit calculator page 4


Master thesis 2007: Prototype - Windows Internet Explorer

http://localhost:3000/calculate/step4

Regarding mother: The calulator has calculated a divergence over 25% between your

Hide/show graphical overview

The parental leave presented in a calendar

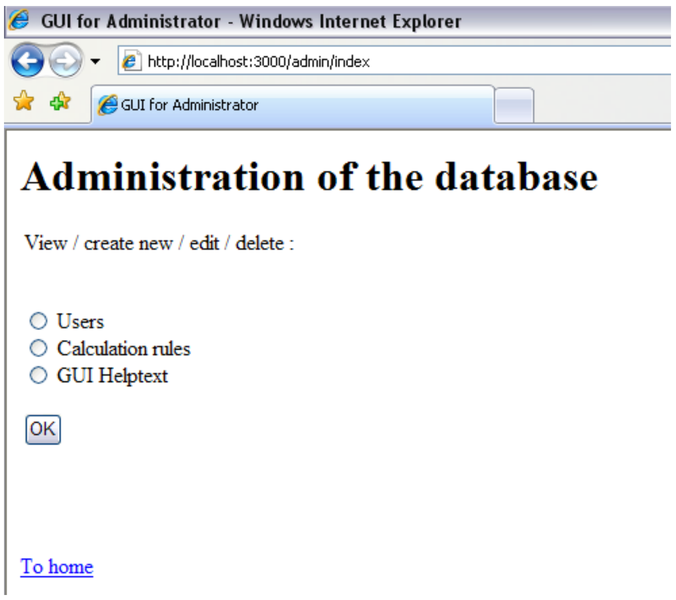
Figure 38: Prototype screenshot – Parental benefit page 4

The parental benefit calculator is also able to show the mandatory days for taking the parental leave highlighted in a calendar as illustrated in Figure 38. This gives the users the possibility to have a quick overview and may be used to plan their parental leave.

7.3 GUI for administrator

The graphical user interface for administration was simply designed and developed. The built-in method in Ruby on Rails, dynamic scaffolding[42], automatically generated all the appropriated data interfaces.
The administrator has the possibility to view, create, edit or delete the users, calculation rules or the GUI help texts when using the administration prototype.

[42] Scaffolding is a technique supported by Ruby on Rails which generates code that an application can use to create, read, update and delete database entries. This technique is often used for prototyping application and entering test data into a database. More information at http://wiki.rubyonrails.org/rails/pages/Scaffold


GUI for Administrator - Windows Internet Explorer
http://localhost:3000/admin/index
GUI for Administrator

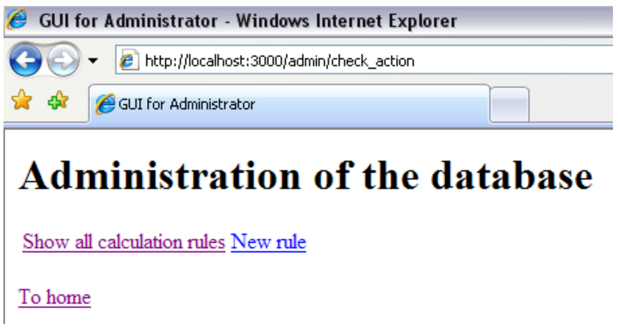
Administration of the database

View / create new / edit / delete :

- ○ Users
- ○ Calculation rules
- ○ GUI Helptext

Figure 39: Prototype screenshot – Admin main page

When the administrator chooses one of the three choices presented, the specific controller will be initiated and a new page will be rendered. The choices are illustrated in Figure 39. Below, in Figure 40, is an example of when the administrator has chosen one of these choices.

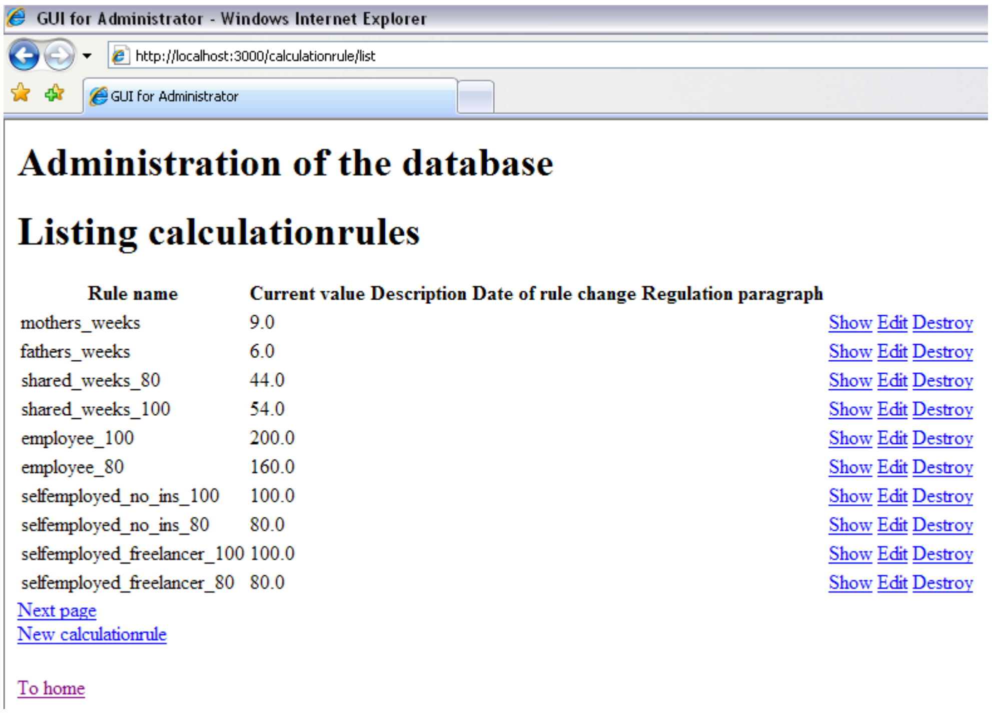

GUI for Administrator - Windows Internet Explorer
http://localhost:3000/admin/check_action
GUI for Administrator

Administration of the database

Show all calculation rules New rule
To home

Figure 40: Prototype screenshot – Admin page “Calculation rules” page 1

If the administrator then chooses “Show all calculation rules”, the controller will read out all the database entries of the calculation rules model. This is illustrated in Figure 41.


GUI for Administrator - Windows Internet Explorer

http://localhost:3000/calculationrule/list

GUI for Administrator

Administration of the database

Listing calculationrules

Rule name	Current value	Description	Date of rule change	Regulation paragraph	
mothers_weeks	9.0				Show Edit Destroy
fathers_weeks	6.0				Show Edit Destroy
shared_weeks_80	44.0				Show Edit Destroy
shared_weeks_100	54.0				Show Edit Destroy
employee_100	200.0				Show Edit Destroy
employee_80	160.0				Show Edit Destroy
selfemployed_no_ins_100	100.0				Show Edit Destroy
selfemployed_no_ins_80	80.0				Show Edit Destroy
selfemployed_freelancer_100	100.0				Show Edit Destroy
selfemployed_freelancer_80	80.0				Show Edit Destroy

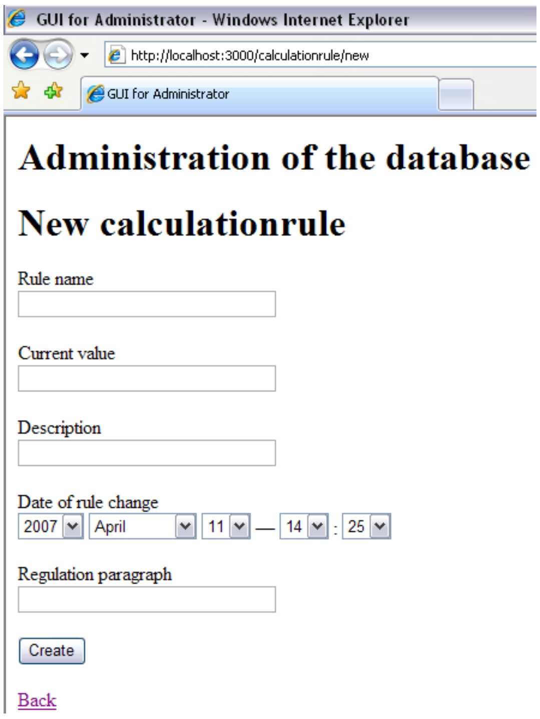
Next page
New calculationrule

To home

Figure 41: Prototype screenshot – Admin page “Calculation rules” page 2

From here the administrator can easily choose to view, create or edit or delete each calculation rule.

If the administrator chooses the “New rule” function, the controller automatically generates a page where the administrator can input the appropriated data as illustrated in Figure 42.


GUI for Administrator - Windows Internet Explorer

http://localhost:3000/calculationrule/new

GUI for Administrator

Administration of the database

New calculationrule

Rule name

Current value

Description

Date of rule change
2007 April 11 — 14 : 25

Regulation paragraph

Create

Back

Figure 42: Prototype screenshot – Admin page “Calculation rules” page 3

Manipulation of the users and the GUI help texts data in the database is happening in the same way.

7.4 Functionality

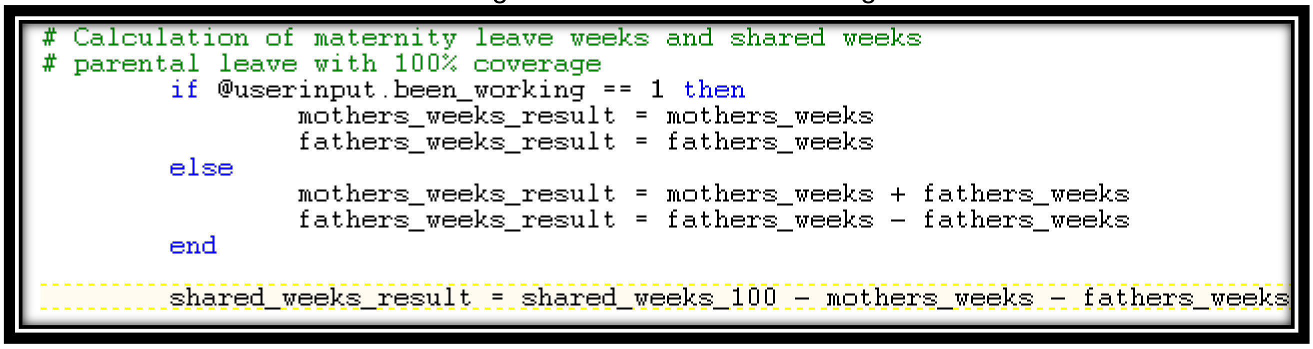
The following sections will describe the functionality of the parental benefit calculator prototype. The first section is describing the functionality for the calculation of the parental benefit and leave, and the second section will describe the administration part of the prototype.

7.4.1 Functionality regarding users

In this section we will describe how the functionality is when using the parental benefit calculator. We will illustrate how the functionality is implemented by using excerpts from the source code and explanations.

Calculation of the parental leave

The parental leave will be calculated if the users has answered “yes” on that mother has been working in minimum 50% position before giving birth. The parental leave is calculated by reading out the rules in the database and using the formula illustrated in Figure 43.



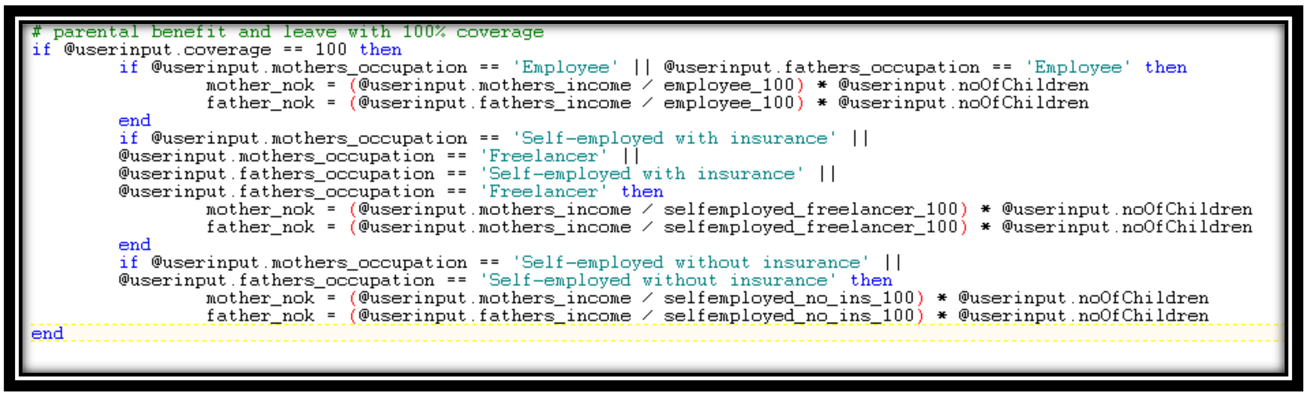
```
# Calculation of maternity leave weeks and shared weeks
# parental leave with 100% coverage
        if @userinput.been_working == 1 then
                mothers_weeks_result = mothers_weeks
                fathers_weeks_result = fathers_weeks
        else
                mothers_weeks_result = mothers_weeks + fathers_weeks
                fathers_weeks_result = fathers_weeks - fathers_weeks
        end

        shared_weeks_result = shared_weeks_100 - mothers_weeks - fathers_weeks
```

Figure 43: Excerpt of code: Function for calculating the parental leave

The weeks are stored as rules in the database. If the rules need to be changed so that the calculation of the parental leave shall be correct according to new rules, this can easily be done changing the rules in the database by using the administration interface.

Calculation of the parental benefit

The parental leave is calculated by checking which occupation mother or father has, and what coverage they have chosen. The necessary data are read out from the database.
Figure 44 illustrates how the function for calculating parental benefit with 100% coverage for a mother is implemented.



```
# parental benefit and leave with 100% coverage
if @userinput.coverage == 100 then
        if @userinput.mothers_occupation == 'Employee' || @userinput.fathers_occupation == 'Employee' then
                mother_nok = (@userinput.mothers_income / employee_100) * @userinput.noOfChildren
                father_nok = (@userinput.fathers_income / employee_100) * @userinput.noOfChildren
        end
        if @userinput.mothers_occupation == 'Self-employed with insurance' ||
        @userinput.mothers_occupation == 'Freelancer' ||
        @userinput.fathers_occupation == 'Self-employed with insurance' ||
        @userinput.fathers_occupation == 'Freelancer' then
                mother_nok = (@userinput.mothers_income / selfemployed_freelancer_100) * @userinput.noOfChildren
                father_nok = (@userinput.mothers_income / selfemployed_freelancer_100) * @userinput.noOfChildren
        end
        if @userinput.mothers_occupation == 'Self-employed without insurance' ||
        @userinput.fathers_occupation == 'Self-employed without insurance' then
                mother_nok = (@userinput.mothers_income / selfemployed_no_ins_100) * @userinput.noOfChildren
                father_nok = (@userinput.fathers_income / selfemployed_no_ins_100) * @userinput.noOfChildren
end
```

Figure 44: Excerpt of code: Function for calculating the parental benefit

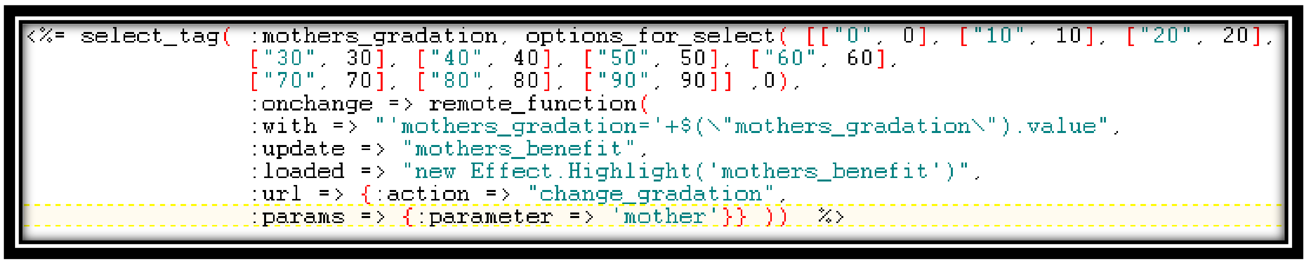
The values used in the calculation are the values received from the users and the rules stored in the database. These rules can be changed by using the administration interface.

Calculation of the parental benefit with graded withdrawal

The prototype has the functionality for calculating the parental benefit with graded withdrawal implemented. If the users wish to combine the parental leave with a part-time job, the calculator can calculate how much parental benefit they will get when they have provided the calculator the percentage of the part-time job.

On page 3 of the calculation the users are asked if mother and/or father wishes graded withdrawal. If the users answer “Yes”, they will have the functionality for calculating the benefit with graded withdrawal on page 4. On page 4 they will then have to choose the percentage from a list and after the choice the benefit will be updated with the correct amount of money.

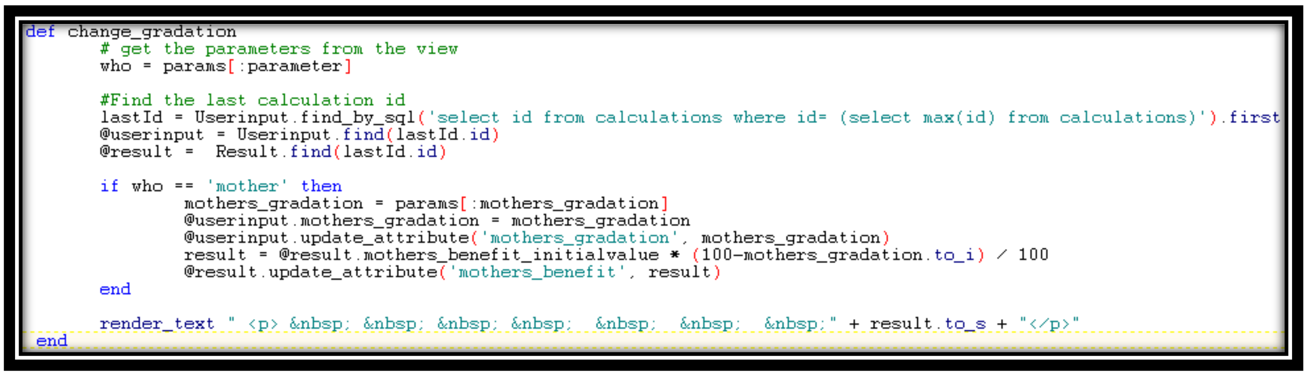
Figure 45 illustrates how the list for choosing the percentage for the part-time job for a mother is implemented in Ruby on Rails. The list can easily be changed to support other percentages.



```
<%= select_tag( :mothers_gradation, options_for_select( [["0", 0], ["10", 10], ["20", 20],
                ["30", 30], ["40", 40], ["50", 50], ["60", 60],
                ["70", 70], ["80", 80], ["90", 90]] ,0),
                :onchange => remote_function(
                :with => "'mothers_gradation='+$(\"mothers_gradation\").value",
                :update => "mothers_benefit",
                :loaded => "new Effect.Highlight('mothers_benefit')",
                :url => {:action => "change_gradation",
                :params => {:parameter => 'mother'}} ))  %>
```

Figure 45: Excerpt of code: List box for selecting the percentage of part-time job

When selecting a value from the list, the action “change_gradation” will calculate the parental benefit with a part-time job. This is illustrated in Figure 46.



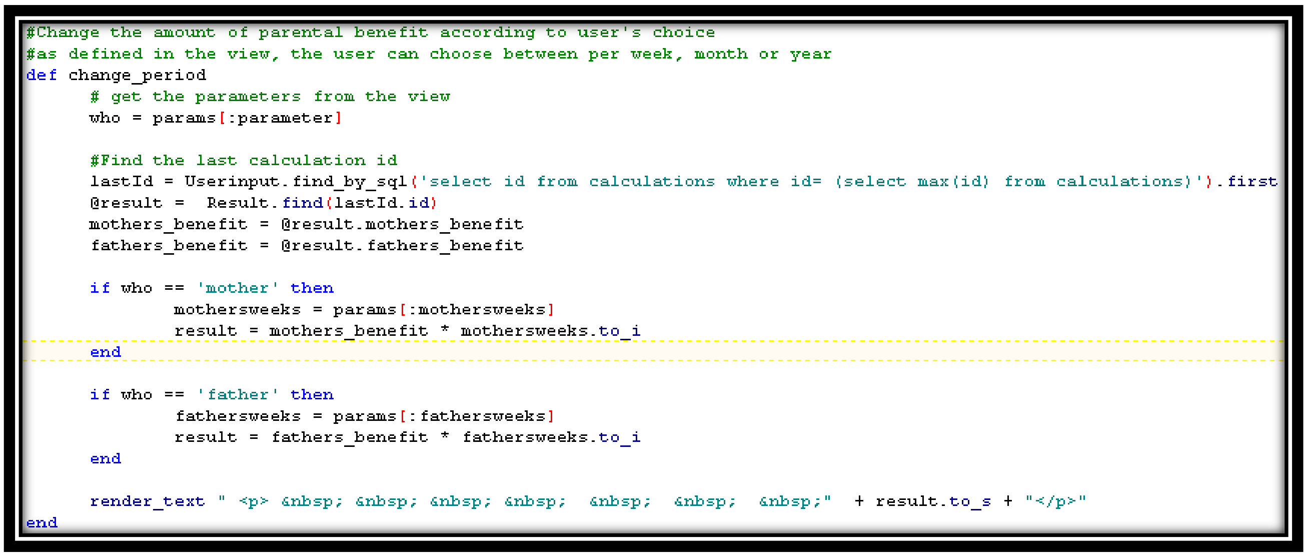
```
def change_gradation
       # get the parameters from the view
       who = params[:parameter]

       #Find the last calculation id
       lastId = Userinput.find_by_sql('select id from calculations where id= (select max(id) from calculations)').first
       @userinput = Userinput.find(lastId.id)
       @result =  Result.find(lastId.id)

       if who == 'mother' then
               mothers_gradation = params[:mothers_gradation]
               @userinput.mothers_gradation = mothers_gradation
               @userinput.update_attribute('mothers_gradation', mothers_gradation)
               result = @result.mothers_benefit_initialvalue * (100-mothers_gradation.to_i) / 100
               @result.update_attribute('mothers_benefit', result)
       end

       render_text " <p> &nbsp; &nbsp; &nbsp; &nbsp;  &nbsp;  &nbsp;  &nbsp;" + result.to_s + "</p>"
 end
```

Figure 46: Excerpt of code: Function for calculating the parental benefit with a part-time jobs

The value for the parental benefit will be read out from the database, recalculated and then updated in the database again after the calculation completes.

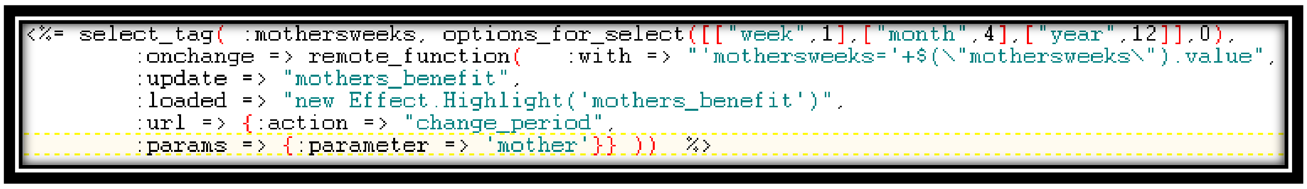
Show amount of benefit per week, month or year

The users are able to show the amount of parental benefit they get per week, per month or per year. Figure 47 and Figure 48 illustrate how this functionality is implemented.



```
#Change the amount of parental benefit according to user's choice
#as defined in the view, the user can choose between per week, month or year
def change_period
      # get the parameters from the view
      who = params[:parameter]

      #Find the last calculation id
      lastId = Userinput.find_by_sql('select id from calculations where id= (select max(id) from calculations)').first
      @result =  Result.find(lastId.id)
      mothers_benefit = @result.mothers_benefit
      fathers_benefit = @result.fathers_benefit

      if who == 'mother' then
              mothersweeks = params[:mothersweeks]
              result = mothers_benefit * mothersweeks.to_i
      end

      if who == 'father' then
              fathersweeks = params[:fathersweeks]
              result = fathers_benefit * fathersweeks.to_i
      end

      render_text " <p> &nbsp; &nbsp; &nbsp; &nbsp;  &nbsp;  &nbsp;  &nbsp;"  + result.to_s + "</p>"
end
```

Figure 47: Excerpt of code: Function for changing the parental benefit for each period of time



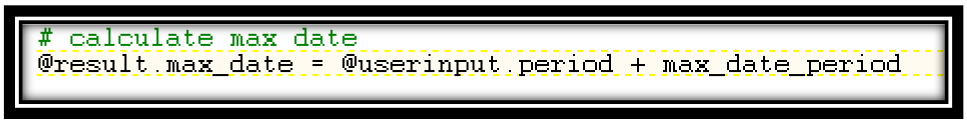
```
<%= select_tag( :mothersweeks, options_for_select([["week",1],["month",4],["year",12]],0),
        :onchange => remote_function(   :with => "'mothersweeks='+$(\"mothersweeks\").value",
        :update => "mothers_benefit",
        :loaded => "new Effect.Highlight('mothers_benefit')",
        :url  => {:action => "change_period",
        :params => {:parameter => 'mother'}} ))  %>
```

Figure 48: Excerpt of code: Changing the period of time

When the users has chosen the wished period to display the parental benefit in, the action “change_period” will be executed and the correct amount of parental benefit will be updated on page 4.

Calculation of the max date

The max date can be calculated by the calculator when the users have provided the calculator with the period. This date can either be enter manually in the text-field on page 3 or picked from a pop-up calendar on the same page.
Figure 49 illustrates how the max date is calculated. The data values are read out from the database and the max date is updated in the database. The max date is a rule that is read out, and this can easily be changed in the database by using the administration interface.

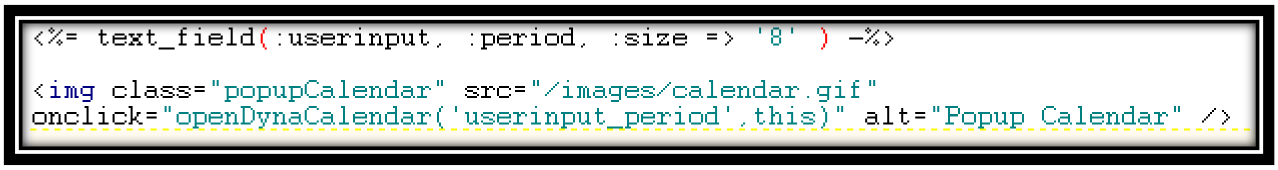


```
# calculate max date
@result.max_date = @userinput.period + max_date_period
```

Figure 49: Excerpt of code: Calculating the max date

Date picking

The date can picked from a Javascript calendar, and updated automatically in the text-field.
Figure 50 illustrates how the button for date picking is displayed on page 3.



```
<%= text_field( :userinput, :period, :size => '8' ) -%>

<img class="popupCalendar" src="/images/calendar.gif"
onclick="openDynaCalendar('userinput_period',this)" alt="Popup Calendar" />
```

Figure 50: Excerpt of code: Pop-up calendar for picking dates

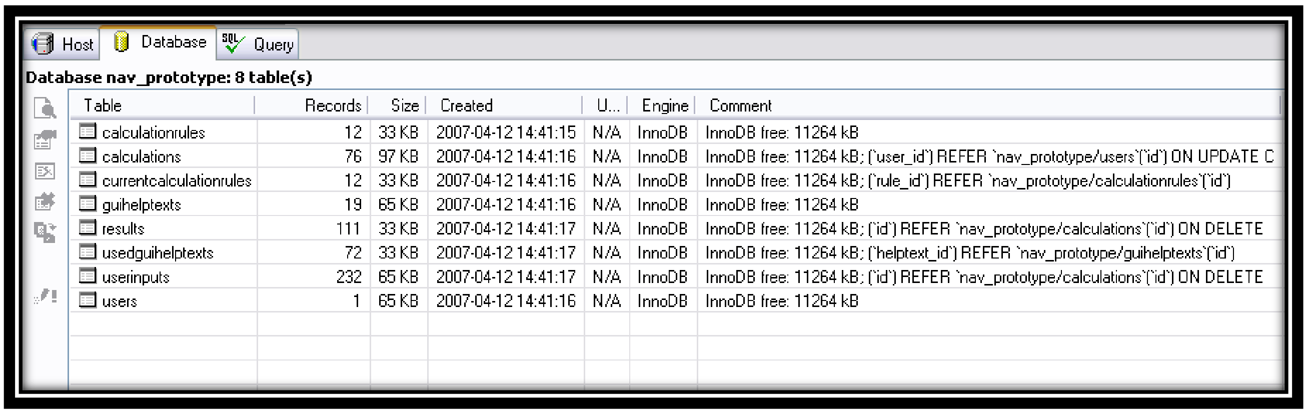
7.4.2 Functionality regarding administrator

This section describes the functionality for administrating the database for the parental benefit calculator.

The functionality for manipulating the data in the database is provided by Ruby on Rails. This method allowed us to quickly add, delete, show and change the data in the database. This method is called scaffolding and the actions are coming with both controller logic and default templates that already know which fields to display on the web site and which input types to use.

7.5 Database

The database in the prototype is implemented using MySQL and a tool called HeidiSQL.
The tables in the database from view in HeidiSQL are illustrated in Figure 51.


Host | Database | Query

Database nav_prototype: 8 table(s)

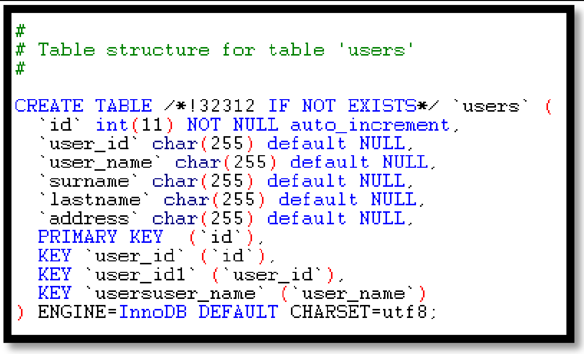
Table	Records	Size	Created	U...	Engine	Comment
calculationrules	12	33 KB	2007-04-12 14:41:15	N/A	InnoDB	InnoDB free: 11264 kB
calculations	76	97 KB	2007-04-12 14:41:16	N/A	InnoDB	InnoDB free: 11264 kB; (`user_id`) REFER `nav_prototype/users`(`id`) ON UPDATE C
currentcalculationrules	12	33 KB	2007-04-12 14:41:16	N/A	InnoDB	InnoDB free: 11264 kB; (`rule_id`) REFER `nav_prototype/calculationrules`(`id`)
guihelptexts	19	65 KB	2007-04-12 14:41:16	N/A	InnoDB	InnoDB free: 11264 kB
results	111	33 KB	2007-04-12 14:41:17	N/A	InnoDB	InnoDB free: 11264 kB; (`id`) REFER `nav_prototype/calculations`(`id`) ON DELETE
usedguihelptexts	72	33 KB	2007-04-12 14:41:17	N/A	InnoDB	InnoDB free: 11264 kB; (`helptext_id`) REFER `nav_prototype/guihelptexts`(`id`)
userinputs	232	65 KB	2007-04-12 14:41:17	N/A	InnoDB	InnoDB free: 11264 kB; (`id`) REFER `nav_prototype/calculations`(`id`) ON DELETE
users	1	65 KB	2007-04-12 14:41:16	N/A	InnoDB	InnoDB free: 11264 kB

Figure 51: Database tables in HeidiSQL

7.5.1 Tables in the database and their relationship

The following table (Table 7) with Figure 52 to Figure 75 is presenting how the tables are implemented by using MySQL and how the tables' properties and data look like in the database. For more information about each table, please refer back to chapter 6.4.1 and chapter 6.4.2.

Users



```
#
# Table structure for table 'users'
#

CREATE TABLE /*!32312 IF NOT EXISTS*/ `users` (
  `id` int(11) NOT NULL auto_increment,
  `user_id` char(255) default NULL,
  `user_name` char(255) default NULL,
  `surname` char(255) default NULL,
  `lastname` char(255) default NULL,
  `address` char(255) default NULL,
  PRIMARY KEY  (`id`),
  KEY `user_id` (`id`),
  KEY `user_id1` (`user_id`),
  KEY `usersuser_name` (`user_name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figure 52: MySQL query for creating the users table

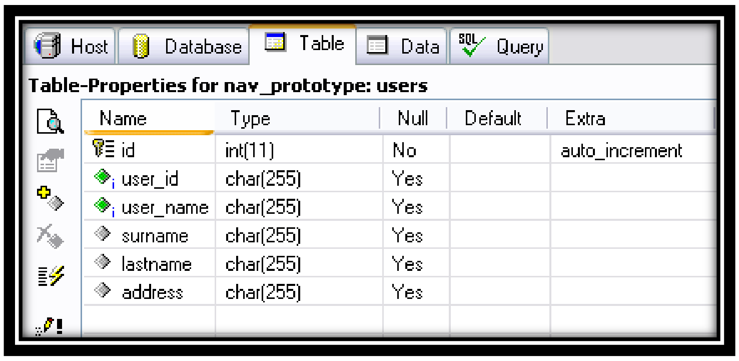
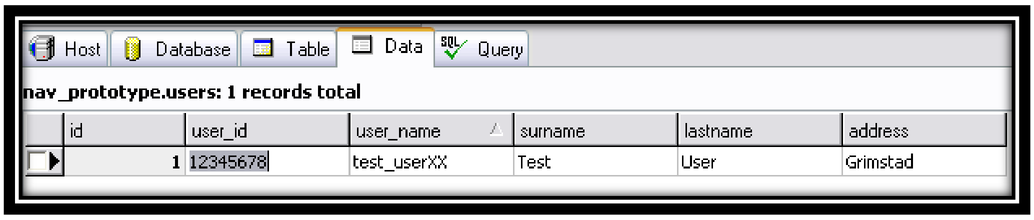

Host | Database | Table | Data | Query

Table-Properties for nav_prototype: users

Name	Type	Null	Default	Extra
id	int(11)	No		auto_increment
user_id	char(255)	Yes		
user_name	char(255)	Yes		
surname	char(255)	Yes		
lastname	char(255)	Yes		
address	char(255)	Yes		

Figure 53: Database table properties – users

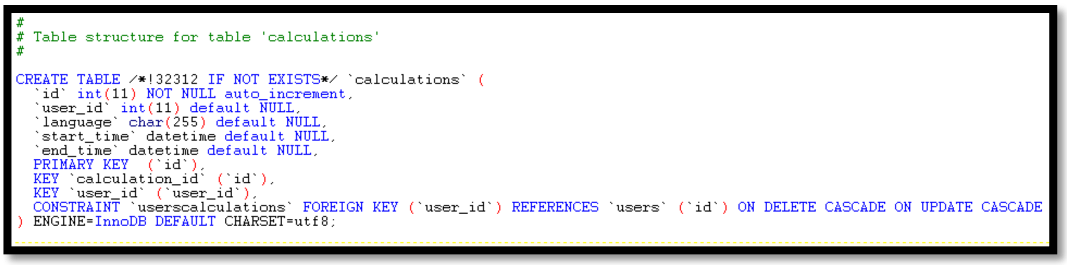

Host | Database | Table | Data | Query

nav_prototype.users: 1 records total

id	user_id	user_name	surname	lastname	address
1	12345678	test_userXX	Test	User	Grimstad

Figure 54: Database table data – users

Calculations



```
#
# Table structure for table 'calculations'
#

CREATE TABLE /*!32312 IF NOT EXISTS*/ `calculations` (
  `id` int(11) NOT NULL auto_increment,
  `user_id` int(11) default NULL,
  `language` char(255) default NULL,
  `start_time` datetime default NULL,
  `end_time` datetime default NULL,
  PRIMARY KEY  (`id`),
  KEY `calculation_id` (`id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `userscalculations` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figure 55: MySQL query for creating the calculations table

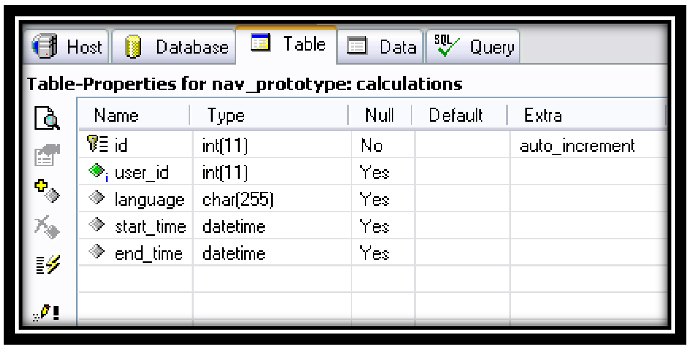
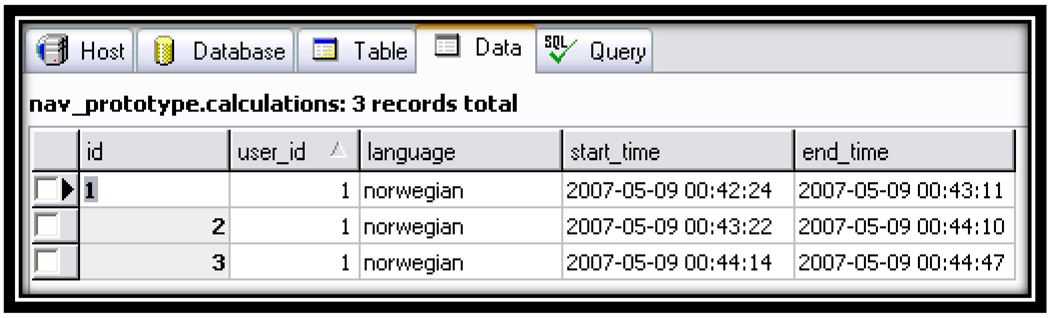


Table-Properties for nav_prototype: calculations

Name	Type	Null	Default	Extra
id	int(11)	No		auto_increment
user_id	int(11)	Yes		
language	char(255)	Yes		
start_time	datetime	Yes		
end_time	datetime	Yes		

Figure 56: Database table properties – calculations

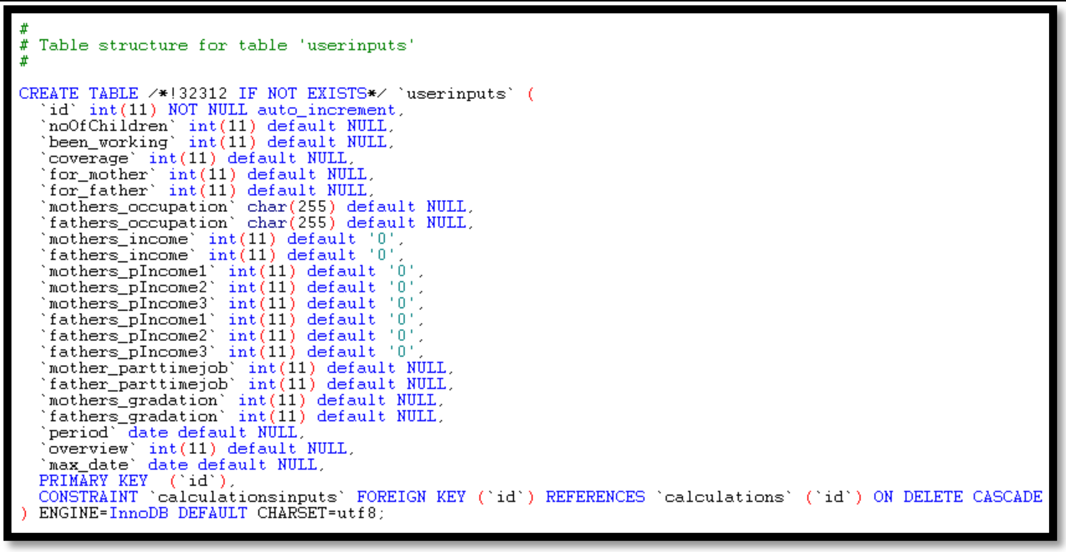


nav_prototype.calculations: 3 records total

id	user_id	language	start_time	end_time
1	1	norwegian	2007-05-09 00:42:24	2007-05-09 00:43:11
2	1	norwegian	2007-05-09 00:43:22	2007-05-09 00:44:10
3	1	norwegian	2007-05-09 00:44:14	2007-05-09 00:44:47

Figure 57: Database table data – calculations

Userinputs



```
#
# Table structure for table 'userinputs'
#

CREATE TABLE /*!32312 IF NOT EXISTS*/ `userinputs` (
  `id` int(11) NOT NULL auto_increment,
  `noOfChildren` int(11) default NULL,
  `been_working` int(11) default NULL,
  `coverage` int(11) default NULL,
  `for_mother` int(11) default NULL,
  `for_father` int(11) default NULL,
  `mothers_occupation` char(255) default NULL,
  `fathers_occupation` char(255) default NULL,
  `mothers_income` int(11) default '0',
  `fathers_income` int(11) default '0',
  `mothers_pIncome1` int(11) default '0',
  `mothers_pIncome2` int(11) default '0',
  `mothers_pIncome3` int(11) default '0',
  `fathers_pIncome1` int(11) default '0',
  `fathers_pIncome2` int(11) default '0',
  `fathers_pIncome3` int(11) default '0',
  `mother_parttimejob` int(11) default NULL,
  `father_parttimejob` int(11) default NULL,
  `mothers_gradation` int(11) default NULL,
  `fathers_gradation` int(11) default NULL,
  `period` date default NULL,
  `overview` int(11) default NULL,
  `max_date` date default NULL,
  PRIMARY KEY  (`id`),
  CONSTRAINT `calculationsinputs` FOREIGN KEY (`id`) REFERENCES `calculations` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figure 58: MySQL query for creating the userinputs table

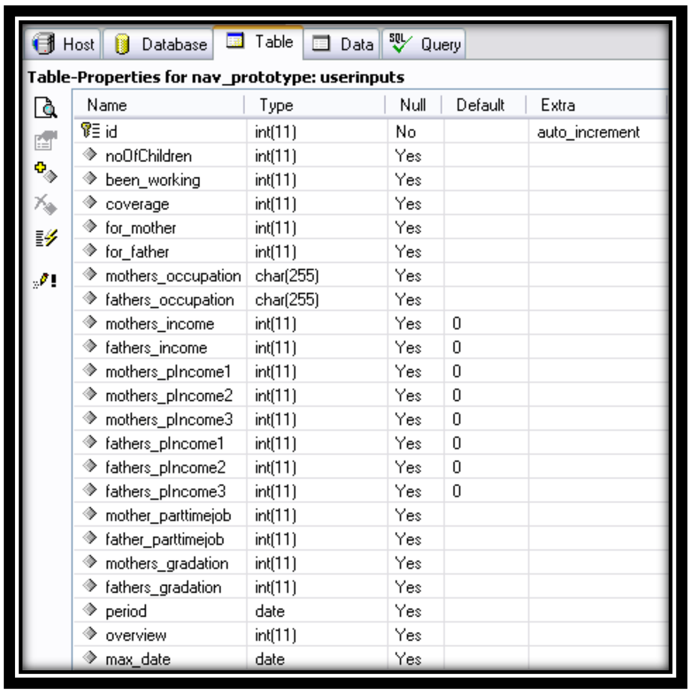
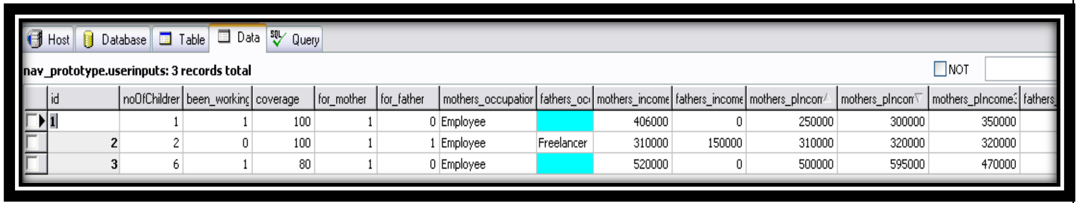


Table-Properties for nav_prototype: userinputs

Name	Type	Null	Default	Extra
id	int(11)	No		auto_increment
noOfChildren	int(11)	Yes		
been_working	int(11)	Yes		
coverage	int(11)	Yes		
for_mother	int(11)	Yes		
for_father	int(11)	Yes		
mothers_occupation	char(255)	Yes		
fathers_occupation	char(255)	Yes		
mothers_income	int(11)	Yes	0	
fathers_income	int(11)	Yes	0	
mothers_pIncome1	int(11)	Yes	0	
mothers_pIncome2	int(11)	Yes	0	
mothers_pIncome3	int(11)	Yes	0	
fathers_pIncome1	int(11)	Yes	0	
fathers_pIncome2	int(11)	Yes	0	
fathers_pIncome3	int(11)	Yes	0	
mother_parttimejob	int(11)	Yes		
father_parttimejob	int(11)	Yes		
mothers_gradation	int(11)	Yes		
fathers_gradation	int(11)	Yes		
period	date	Yes		
overview	int(11)	Yes		
max_date	date	Yes		

Figure 59: Database table properties – userinputs

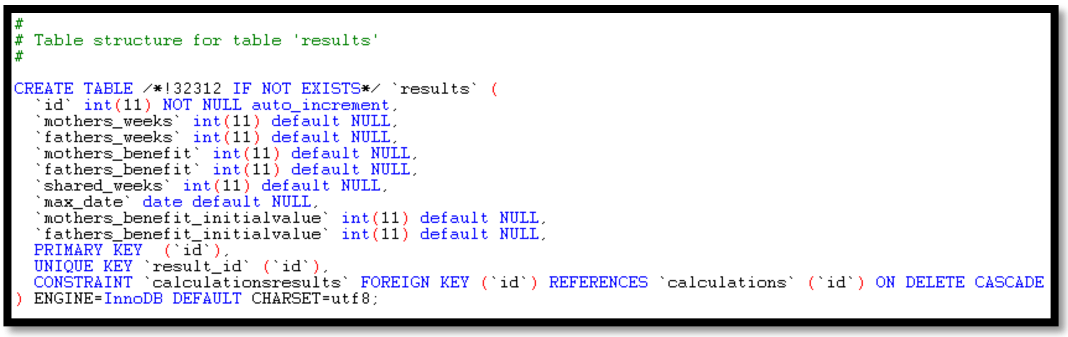


nav_prototype.userinputs: 3 records total

id	noOfChildrer	been_workinç	coverage	for_mother	for_father	mothers_occupatior	fathers_oc	mothers_income	fathers_income	mothers_pIncom	mothers_pIncom	mothers_pIncome3	fathers
1	1	1	100	1	0	Employee		406000	0	250000	300000	350000	
2	2	0	100	1	1	Employee	Freelancer	310000	150000	310000	320000	320000	
3	6	1	80	1	0	Employee		520000	0	500000	595000	470000	

Figure 60: Database table data – userinputs

Results



```
#
# Table structure for table 'results'
#

CREATE TABLE /*!32312 IF NOT EXISTS*/ `results` (
  `id` int(11) NOT NULL auto_increment,
  `mothers_weeks` int(11) default NULL,
  `fathers_weeks` int(11) default NULL,
  `mothers_benefit` int(11) default NULL,
  `fathers_benefit` int(11) default NULL,
  `shared_weeks` int(11) default NULL,
  `max_date` date default NULL,
  `mothers_benefit_initialvalue` int(11) default NULL,
  `fathers_benefit_initialvalue` int(11) default NULL,
  PRIMARY KEY  (`id`),
  UNIQUE KEY `result_id` (`id`),
  CONSTRAINT `calculationsresults` FOREIGN KEY (`id`) REFERENCES `calculations` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figure 61: MySQL query for creating the results table

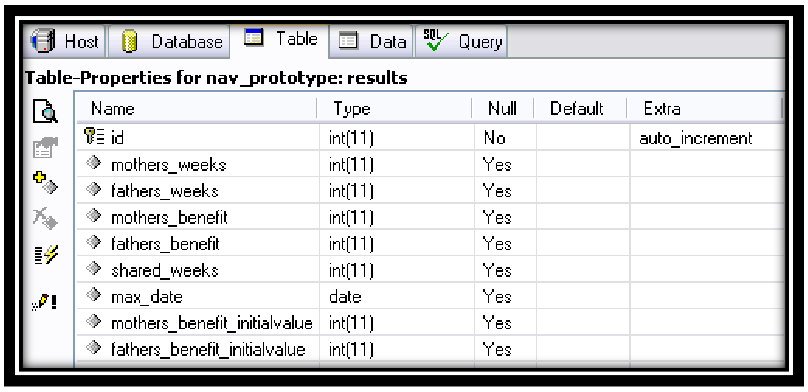
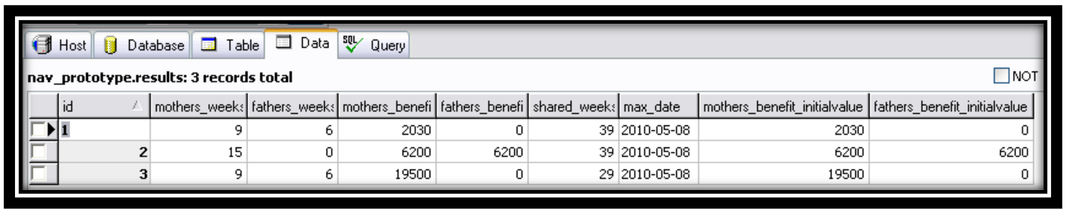


Table-Properties for nav_prototype: results

Name	Type	Null	Default	Extra
id	int(11)	No		auto_increment
mothers_weeks	int(11)	Yes		
fathers_weeks	int(11)	Yes		
mothers_benefit	int(11)	Yes		
fathers_benefit	int(11)	Yes		
shared_weeks	int(11)	Yes		
max_date	date	Yes		
mothers_benefit_initialvalue	int(11)	Yes		
fathers_benefit_initialvalue	int(11)	Yes		

Figure 62: Database table properties – results

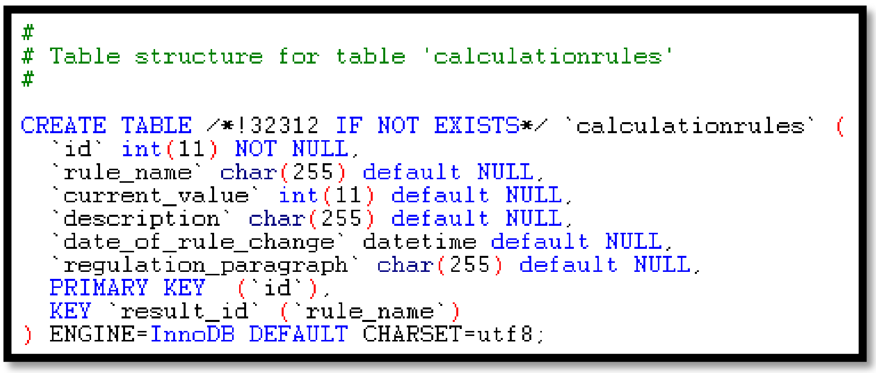


nav_prototype.results: 3 records total

id	mothers_week:	fathers_week:	mothers_benefi	fathers_benefi	shared_week:	max_date	mothers_benefit_initialvalue	fathers_benefit_initialvalue
1	9	6	2030	0	39	2010-05-08	2030	0
2	15	0	6200	6200	39	2010-05-08	6200	6200
3	9	6	19500	0	29	2010-05-08	19500	0

Figure 63: Database table data – results

Calculationr-
ules



```
#
# Table structure for table 'calculationrules'
#

CREATE TABLE /*!32312 IF NOT EXISTS*/ `calculationrules` (
  `id` int(11) NOT NULL,
  `rule_name` char(255) default NULL,
  `current_value` int(11) default NULL,
  `description` char(255) default NULL,
  `date_of_rule_change` datetime default NULL,
  `regulation_paragraph` char(255) default NULL,
  PRIMARY KEY  (`id`),
  KEY `result_id` (`rule_name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figure 64: MySQL query for creating calculationrules table

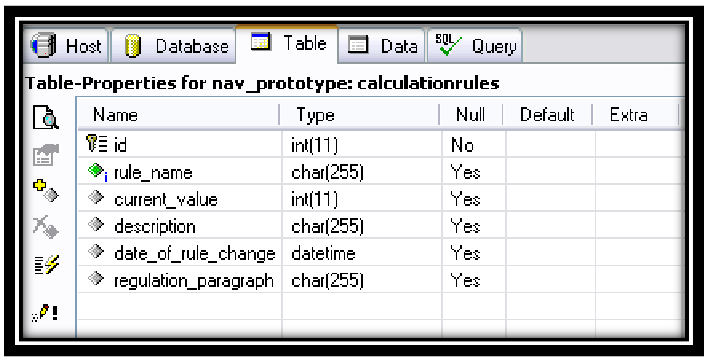
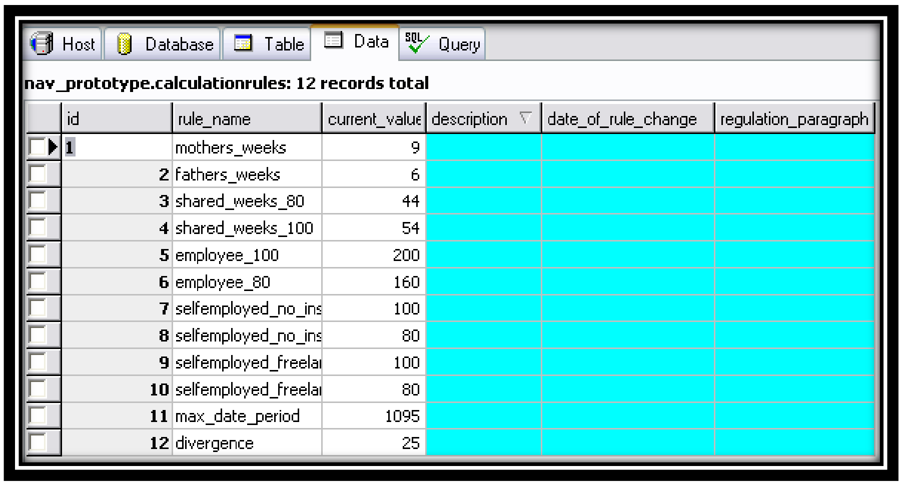


Table-Properties for nav_prototype: calculationrules

Name	Type	Null	Default	Extra
id	int(11)	No		
rule_name	char(255)	Yes		
current_value	int(11)	Yes		
description	char(255)	Yes		
date_of_rule_change	datetime	Yes		
regulation_paragraph	char(255)	Yes		

Figure 65: Database table properties – calculationrules

Figure 66 is illustrating how the data in this table could look like. Each of these rules can easily be changed through the administration interface prototype.

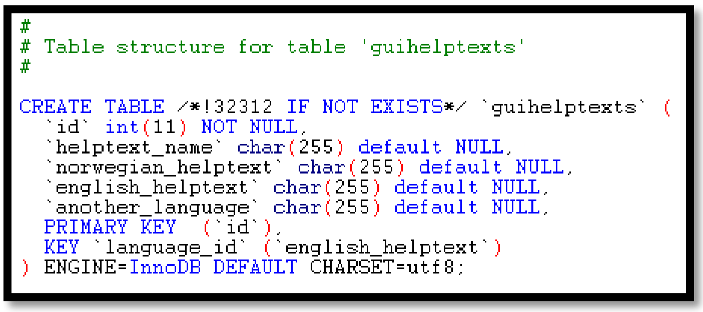


nav_prototype.calculationrules: 12 records total

id	rule_name	current_value	description	date_of_rule_change	regulation_paragraph
1	mothers_weeks	9			
2	fathers_weeks	6			
3	shared_weeks_80	44			
4	shared_weeks_100	54			
5	employee_100	200			
6	employee_80	160			
7	selfemployed_no_ins	100			
8	selfemployed_no_ins	80			
9	selfemployed_freelai	100			
10	selfemployed_freelai	80			
11	max_date_period	1095			
12	divergence	25			

Figure 66: Database table data – calculationrules

Guihelptexts



```
#
# Table structure for table 'guihelptexts'
#

CREATE TABLE /*!32312 IF NOT EXISTS*/ `guihelptexts` (
  `id` int(11) NOT NULL,
  `helptext_name` char(255) default NULL,
  `norwegian_helptext` char(255) default NULL,
  `english_helptext` char(255) default NULL,
  `another_language` char(255) default NULL,
  PRIMARY KEY  (`id`),
  KEY `language_id` (`english_helptext`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figure 67: MySQL query for creating the guihelptexts table

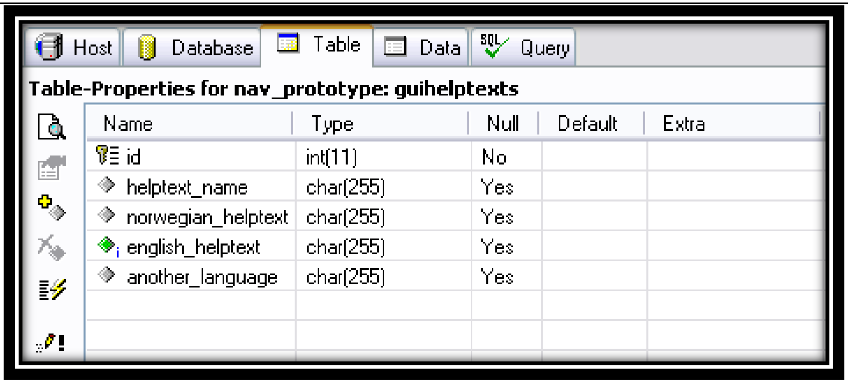
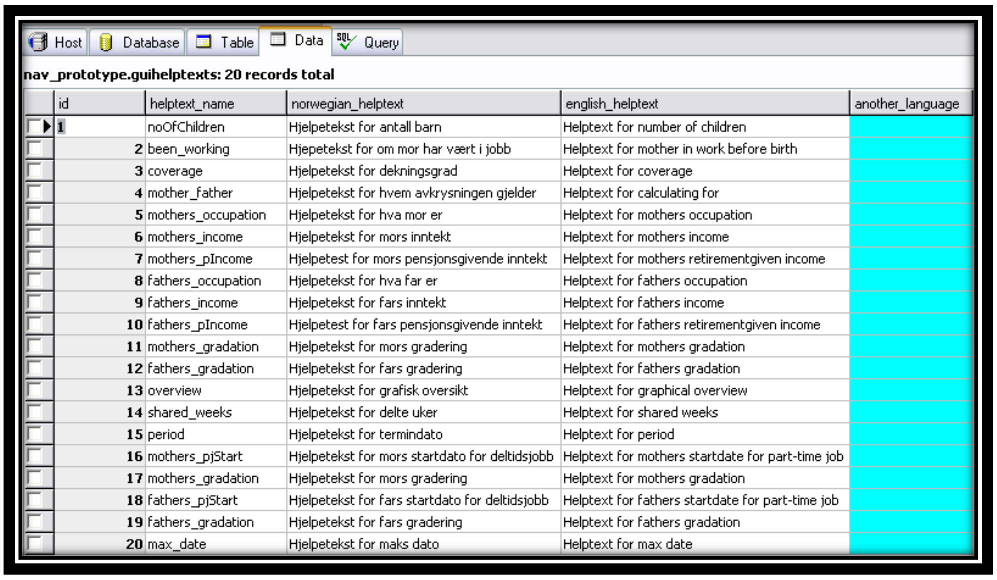


Table-Properties for nav_prototype: guihelptexts

Name	Type	Null	Default	Extra
id	int(11)	No		
helptext_name	char(255)	Yes		
norwegian_helptext	char(255)	Yes		
english_helptext	char(255)	Yes		
another_language	char(255)	Yes		

Figure 68: Database table properties – guihelptexts

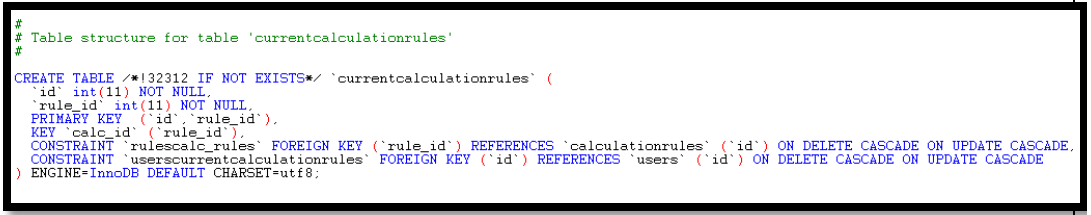


nav_prototype.guihelptexts: 20 records total

id	helptext_name	norwegian_helptext	english_helptext	another_language
1	noOfChildren	Hjelpetekst for antall barn	Helptext for number of children	
2	been_working	Hjepetekst for om mor har vært i jobb	Helptext for mother in work before birth	
3	coverage	Hjelpetekst for dekningsgrad	Helptext for coverage	
4	mother_father	Hjelpetekst for hvem avkrysningen gjelder	Helptext for calculating for	
5	mothers_occupation	Hjelpetekst for hva mor er	Helptext for mothers occupation	
6	mothers_income	Hjelpetekst for mors inntekt	Helptext for mothers income	
7	mothers_pIncome	Hjelpetest for mors pensjonsgivende inntekt	Helptext for mothers retirementgiven income	
8	fathers_occupation	Hjelpetekst for hva far er	Helptext for fathers occupation	
9	fathers_income	Hjelpetekst for fars inntekt	Helptext for fathers income	
10	fathers_pIncome	Hjelpetest for fars pensjonsgivende inntekt	Helptext for fathers retirementgiven income	
11	mothers_gradation	Hjelpetekst for mors gradering	Helptext for mothers gradation	
12	fathers_gradation	Hjelpetekst for fars gradering	Helptext for fathers gradation	
13	overview	Hjelpetekst for grafisk oversikt	Helptext for graphical overview	
14	shared_weeks	Hjelpetekst for delte uker	Helptext for shared weeks	
15	period	Hjelpetekst for termindato	Helptext for period	
16	mothers_pjStart	Hjelpetekst for mors startdato for deltidsjobb	Helptext for mothers startdate for part-time job	
17	mothers_gradation	Hjelpetekst for mors gradering	Helptext for mothers gradation	
18	fathers_pjStart	Hjelpetekst for fars startdato for deltidsjobb	Helptext for fathers startdate for part-time job	
19	fathers_gradation	Hjelpetekst for fars gradering	Helptext for fathers gradation	
20	max_date	Hjelpetekst for maks dato	Helptext for max date	

Figure 69: Database table data – guihelptexts

Currentcalcu-lationrues



```
#
# Table structure for table 'currentcalculationrules'
#

CREATE TABLE /*!32312 IF NOT EXISTS*/ `currentcalculationrules` (
  `id` int(11) NOT NULL,
  `rule_id` int(11) NOT NULL,
  PRIMARY KEY  (`id`,`rule_id`),
  KEY `calc_id` (`rule_id`),
  CONSTRAINT `rulescalc_rules` FOREIGN KEY (`rule_id`) REFERENCES `calculationrules` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `userscurrentcalculationrules` FOREIGN KEY (`id`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figure 70: MySQL query for creating the currentcalculationrules table

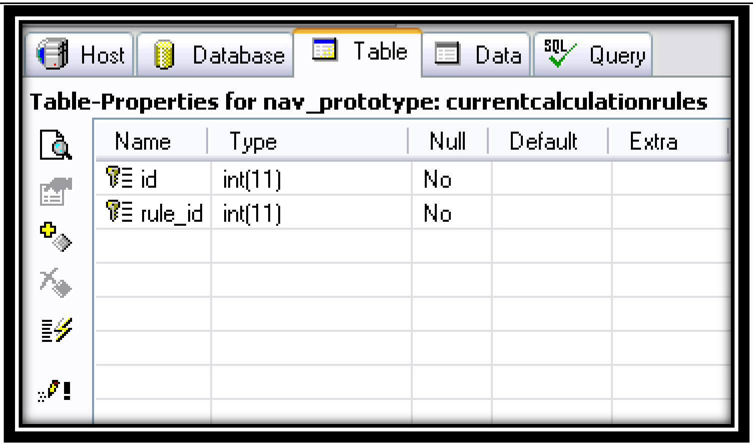
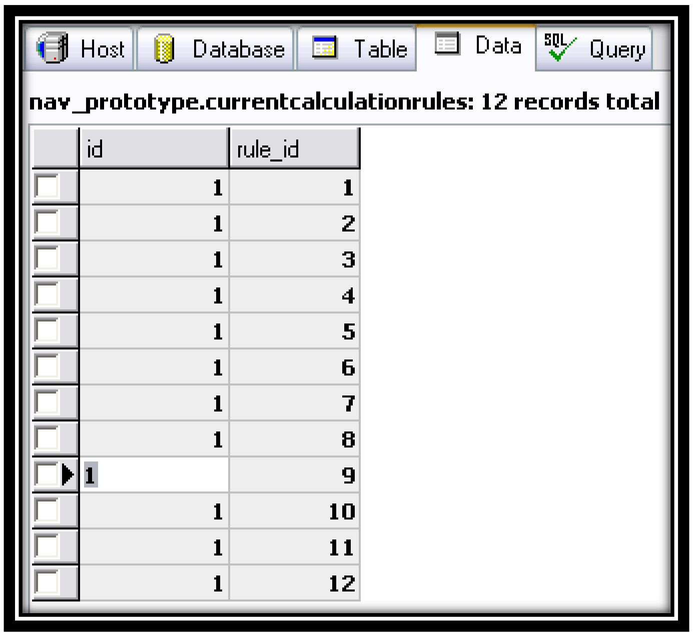


Table-Properties for nav_prototype: currentcalculationrules

Name	Type	Null	Default	Extra
id	int(11)	No		
rule_id	int(11)	No		

Figure 71: Database table properties – currentcalculationrules

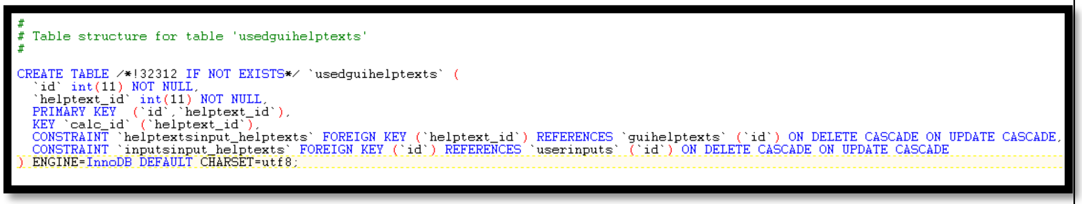


nav_prototype.currentcalculationrules: 12 records total

id	rule_id
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10
1	11
1	12

Figure 72: Database table data – currentcalculationrules

Usedguihelp-texts



```
#
# Table structure for table 'usedguihelptexts'
#

CREATE TABLE /*!32312 IF NOT EXISTS*/ `usedguihelptexts` (
  `id` int(11) NOT NULL,
  `helptext_id` int(11) NOT NULL,
  PRIMARY KEY  (`id`,`helptext_id`),
  KEY `calc_id` (`helptext_id`),
  CONSTRAINT `helptextsinput_helptexts` FOREIGN KEY (`helptext_id`) REFERENCES `guihelptexts` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `inputsinput_helptexts` FOREIGN KEY (`id`) REFERENCES `userinputs` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Figure 73: MySQL query for creating the usedguihelptexts

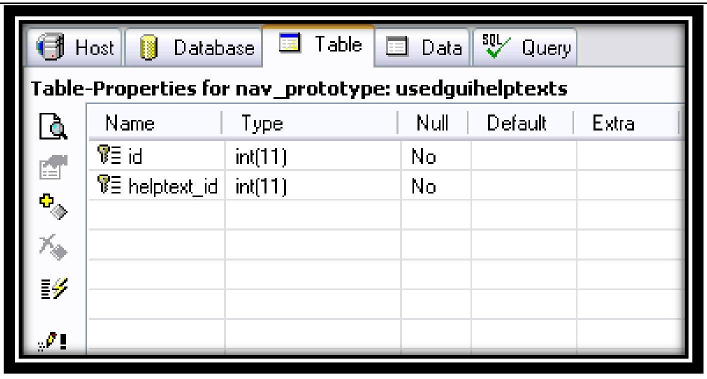
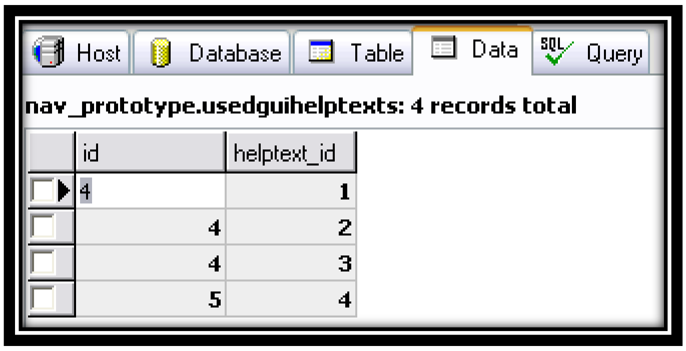


Table-Properties for nav_prototype: usedguihelptexts

Name	Type	Null	Default	Extra
id	int(11)	No		
helptext_id	int(11)	No		

Figure 74: Database table properties – usedguihelptexts



nav_prototype.usedguihelptexts: 4 records total

id	helptext_id
4	1
4	2
4	3
5	4

Figure 75: Database table data – usedguihelptexts

Table 7: Implementation of database tables

8 Validation and Testing

This chapter will explain how we validated and tested the parental benefit calculator for ensuring that it is working and satisfies the specific requirements presented in chapter 5.4. The user testing on the new prototype with real users and the heuristic evaluation performed by us will also be presented. In addition, this chapter also describes how the validation and testing regarding accessibility were performed, on both NAV's current solution and the new prototyping, using the accessibility evaluation tools and guidelines that were presented in chapter 4.1.3 on page 31.

8.1 Testing according to requirements

8.1.1 Test Plan

We will test the new prototype in regards to the requirements specified in chapter 5.4 to see if each requirement is successfully implemented.
We will make use of the graphical user interface of the new prototype and test the requirements. The next section will present what we have tested, how we tested and the result of each test by reporting whether a requirement is successfully implemented or not.

8.1.2 Test Results

The following tables, Table 8 to Table 13, will be presenting each test that was performed on the new prototype in regards to the requirements.

Test no. 1	
Test description	In this test we will test the requirements: REQ-1.1 and REQ-1.2.
Task description	REQ-1.1: The prototype must be able to store the user inputs to a database. REQ-1.2: The prototype must be able to update the user inputs to a database.
Performing the task	We choose to input “Yes”, “100%” and “Mother” as illustrated in Figure 76. Then we clicked on the “Continue” button.

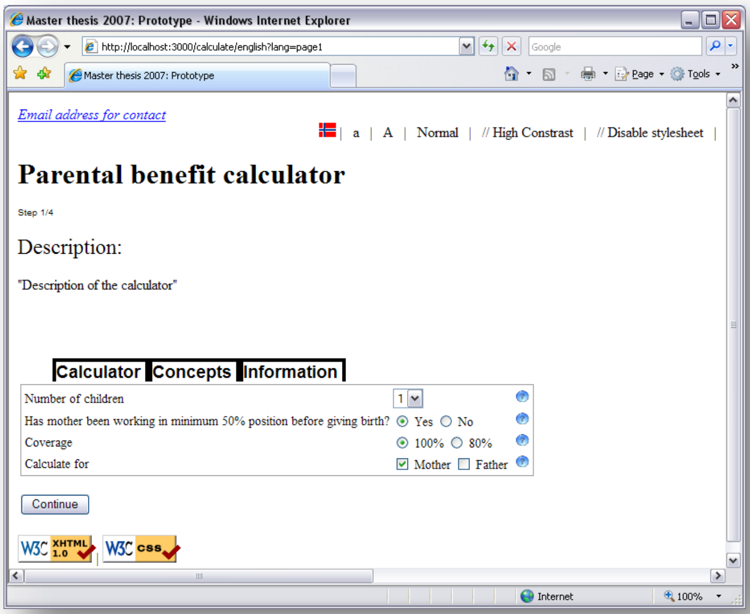
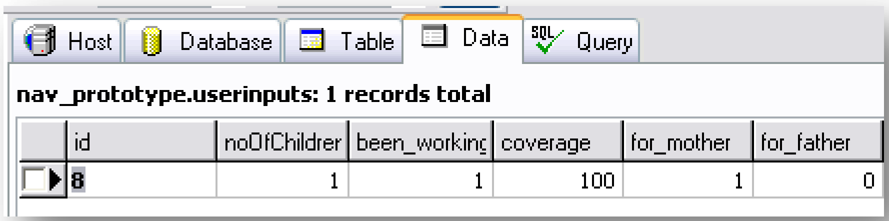
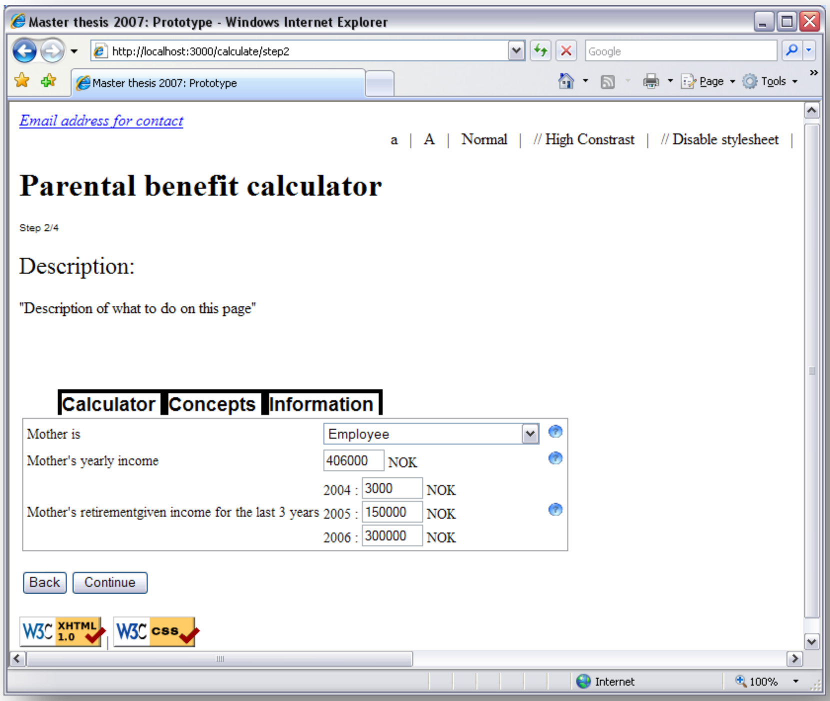
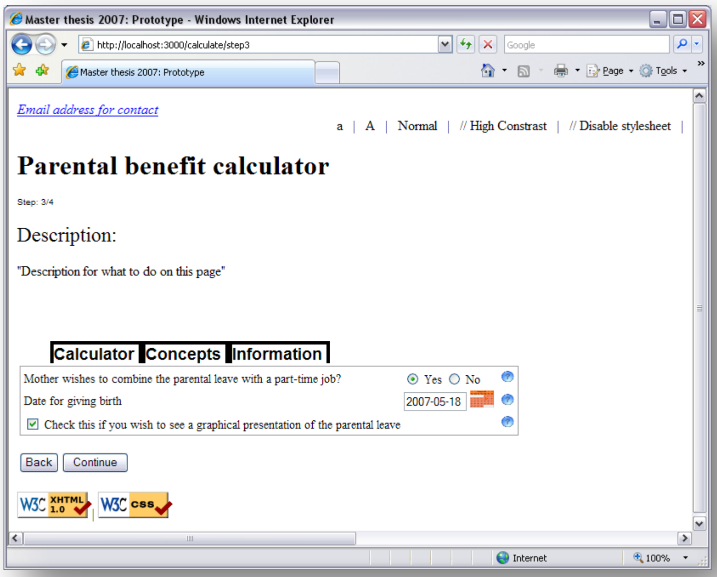
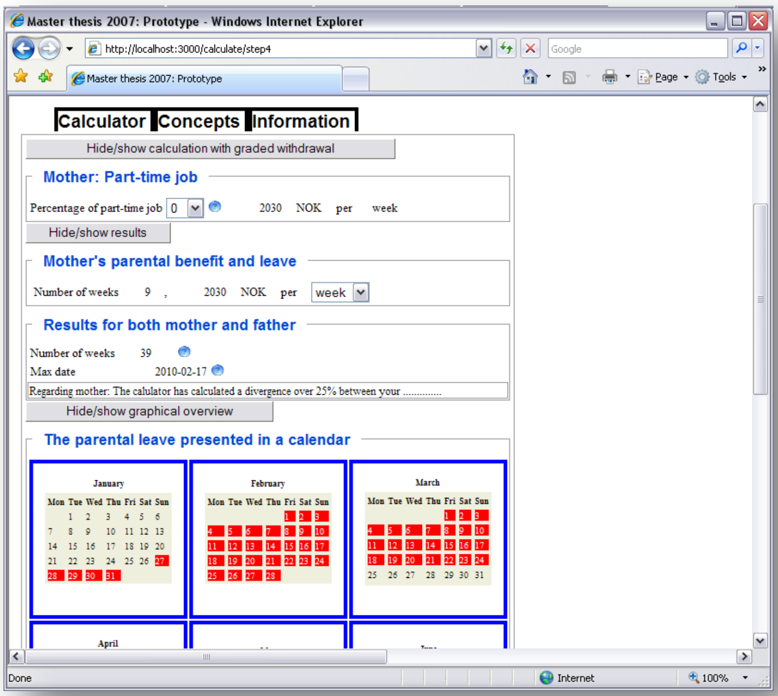
	 Master thesis 2007: Prototype - Windows Internet Explorer http://localhost:3000/calculate/english?lang=page1 Email address for contact a | A | Normal | // High Constrast | // Disable stylesheet | Parental benefit calculator Step 1/4 Description: "Description of the calculator" Calculator Concepts Information Number of children: 1 Has mother been working in minimum 50% position before giving birth? [✓ Yes] No Coverage: [✓ 100%] 80% Calculate for: [✓ Mother] Father Continue *Figure 76: Testing REQ-.11 and REQ-1.2 – GUI page 1* We opened HeidiSQL and check the database table, userinputs, to verify that the prototype had successfully received the user inputs. This is illustrated in Figure 77.  nav_prototype.userinputs: 1 records total id: 8; noOfChildrer: 1; been_workinç: 1; coverage: 100; for_mother: 1; for_father: 0 *Figure 77: Testing REQ-1.1 and REQ-1.2 – HeidiSQL table data*
Test result	REQ-1.1 and REQ-1.2 are implemented successfully.

Table 8: Test no. 1 – Testing REQ-1 and REQ-1.1

Test no. 2	
Test description	In this test we will test the requirements: REQ-1, REQ-2, REQ-2.1, REQ-2.2 and REQ-2.3
Task description	REQ-1: The prototype must be able to receive input from a user. REQ-2: The prototype must perform calculations based on the user's inputs. REQ-2.1: The prototype must be able to calculate the parental benefit.

	REQ-2.2: The prototype must be able to calculate the dates for parental leave. REQ-2.3: The prototype must be able to calculate the parental benefit with graded-withdrawal
Performing the task	At page 2, we enter mother's yearly income and her retirement-given income for the last 3 years as illustrated in Figure 78.  Master thesis 2007: Prototype - Windows Internet Explorer http://localhost:3000/calculate/step2 Email address for contact a | A | Normal | // High Constrast | // Disable stylesheet | Parental benefit calculator Step 2/4 Description: "Description of what to do on this page" Calculator Concepts Information Mother is: Employee Mother's yearly income: 406000 NOK Mother's retirementgiven income for the last 3 years: 2004 : 3000 NOK; 2005 : 150000 NOK; 2006 : 300000 NOK *Figure 78: Testing REQ-1, REQ-2, REQ-2.1, REQ-2.2 and REQ-2.3 – GUI page 2* When we clicked continue, we came to page 3 where we chose "Yes" to calculation with graded-withdrawal and checked the button for a graphical overview of the parental leave. This is illustrated in Figure 79. We then continued to the last page, page number 4, and we see that we the prototype has calculated the parental benefit, the dates for parental leave, and the parental benefit with grade-withdrawal as illustrated in Figure 80.

	 *Figure 79: Testing REQ-1, REQ-2, REQ-2.1, REQ-2.2 and REQ-2.3 – GUI page 3*  *Figure 80: Testing REQ-1, REQ-2, REQ-2.1, REQ-2.2 and REQ-2.3 – GUI page 4*
Test result	REQ-1, REQ-2, REQ-2.1, REQ-2.2 and REQ-2.3 are successfully implemented.

Master thesis 2007: Prototype - Windows Internet Explorer

http://localhost:3000/calculate/step3

Email address for contact

a | A | Normal | // High Constrast | // Disable stylesheet |

Parental benefit calculator

Step: 3/4

Description:

"Description for what to do on this page"

Calculator Concepts Information

Mother wishes to combine the parental leave with a part-time job? [✓ Yes] No

Date for giving birth 2007-05-18

[✓] Check this if you wish to see a graphical presentation of the parental leave

Back Continue

Master thesis 2007: Prototype - Windows Internet Explorer

http://localhost:3000/calculate/step4

Calculator Concepts Information

Hide/show calculation with graded withdrawal

Mother: Part-time job

Percentage of part-time job 0 2030 NOK per week

Hide/show results

Mother's parental benefit and leave

Number of weeks 9 , 2030 NOK per week

Results for both mother and father

Number of weeks 39

Max date 2010-02-17

Regarding mother: The calulator has calculated a divergence over 25% between your ....

Hide/show graphical overview

The parental leave presented in a calendar

January

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

February

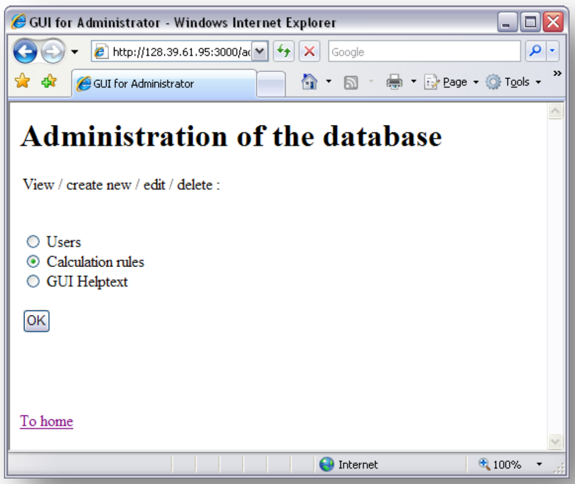
Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

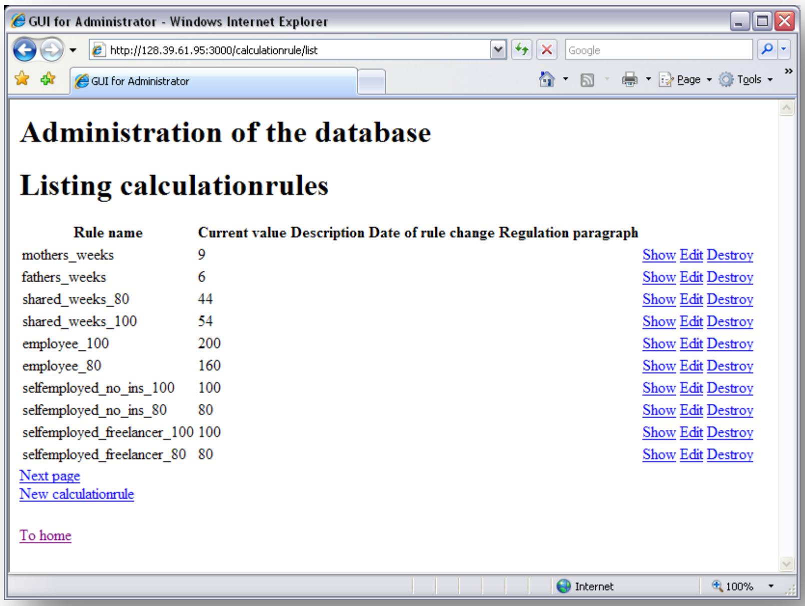
March

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

April

Table 9: Test no. 2 – Testing REQ-1, REQ-2, REQ-2.1, REQ-2.2 and REQ-2.3

Test no. 3	
Test description	In this test we will test the requirements: REQ-3, REQ-3.1, REQ-3.2 and REQ-3.3.
Task description	REQ-3: An administrator can modify the content of the database. REQ-3.1: An administrator can add, delete and change a calculation rule in the database. REQ-3.2: An administrator can add, delete and change a help text in the database. REQ-3.3: An administrator can add, delete and change a user in the database.
Performing the task	Since the functions for adding, deleting and changing a calculation rule, GUI help text and a user is an out-of-the-box solution in Ruby on Rails, we will just be testing adding, deleting and changing a calculation rule. If it works for the calculation rule, it will work for the GUI help text and users. We started the testing by starting the administration interface. We then chose to manipulate the calculation rules by selecting the radio button next to the text as illustrated in Figure 81.  GUI for Administrator - Windows Internet Explorer http://128.39.61.95:3000/a **Administration of the database** View / create new / edit / delete : ○ Users ● Calculation rules ○ GUI Helptext OK To home *Figure 81: Testing REQ-3, REQ-3.1, REQ-3.2 and REQ-3.3 – Admin interface page 1* Then, we proceed by pressing the OK button. We showed all the data in the database by clicking on “Show all calculation rules”. In Figure 82, all the rules are showed on the page.


GUI for Administrator - Windows Internet Explorer
http://128.39.61.95:3000/calculationrule/list

Administration of the database

Listing calculationrules

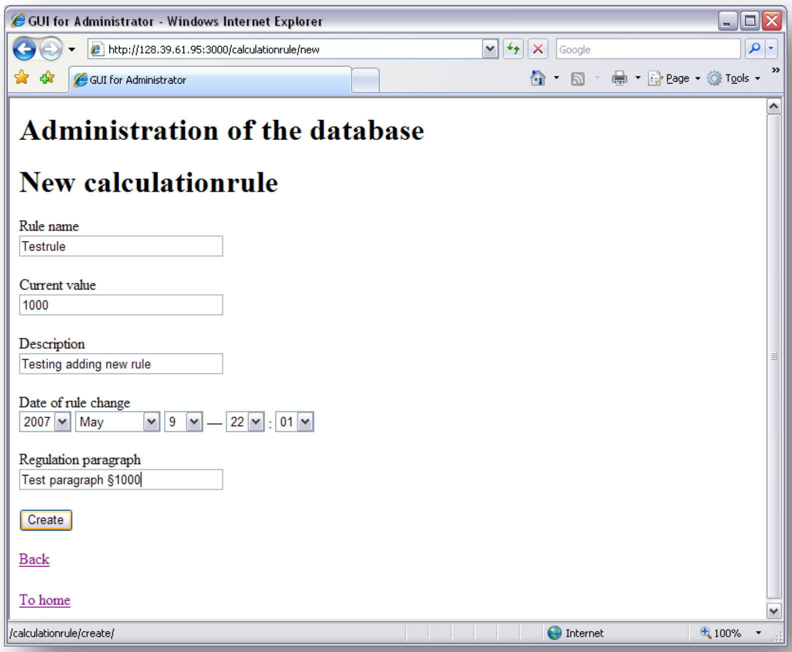
Rule name	Current value	Description	Date of rule change	Regulation paragraph	
mothers_weeks	9				Show Edit Destroy
fathers_weeks	6				Show Edit Destroy
shared_weeks_80	44				Show Edit Destroy
shared_weeks_100	54				Show Edit Destroy
employee_100	200				Show Edit Destroy
employee_80	160				Show Edit Destroy
selfemployed_no_ins_100	100				Show Edit Destroy
selfemployed_no_ins_80	80				Show Edit Destroy
selfemployed_freelancer_100	100				Show Edit Destroy
selfemployed_freelancer_80	80				Show Edit Destroy

Next page
New calculationrule

To home

Figure 82: Testing REQ-3, REQ-3.1, REQ-3.2 and REQ-3.3 – Admin interface show rules

We tested adding a new rule by clicking “New rule”, and provided the required data as illustrated in Figure 83.


GUI for Administrator - Windows Internet Explorer
http://128.39.61.95:3000/calculationrule/new

Administration of the database

New calculationrule

Rule name
Testrule

Current value
1000

Description
Testing adding new rule

Date of rule change
2007 May 9 — 22 : 01

Regulation paragraph
Test paragraph §1000

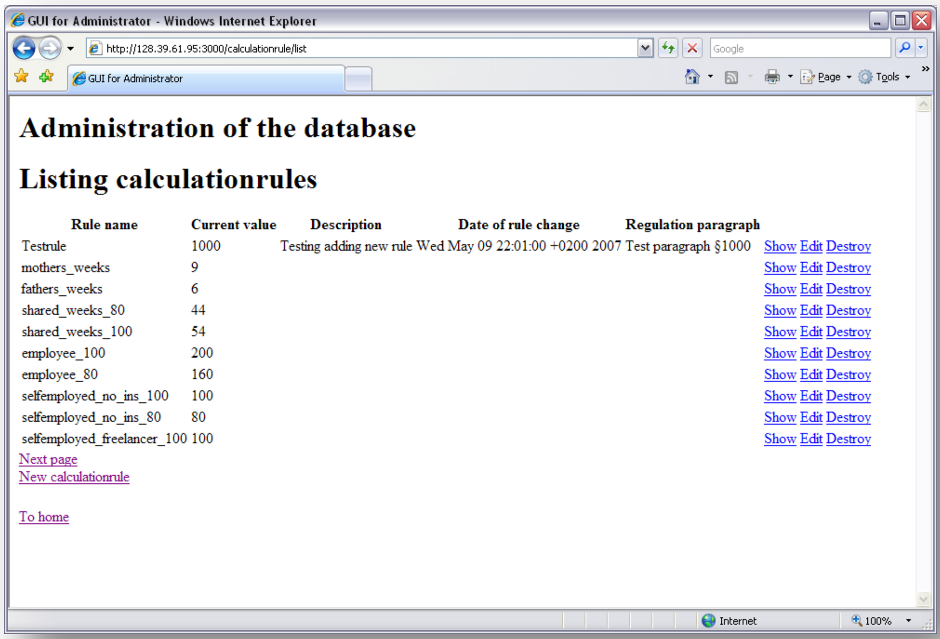
Create

Back

To home

Figure 83: Testing REQ-3, REQ-3.1, REQ-3.2 and REQ-3.3 – Admin interface new rule

By pressing the “Create” button, we could see that the new calculation rule was created as illustrated in Figure 84.


GUI for Administrator - Windows Internet Explorer
http://128.39.61.95:3000/calculationrule/list
GUI for Administrator
Internet
100%

Administration of the database

Listing calculationrules

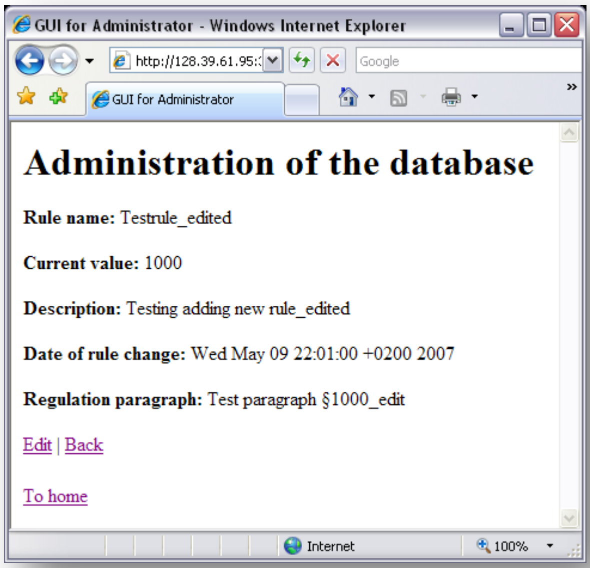
Rule name	Current value	Description	Date of rule change	Regulation paragraph	
Testrule	1000	Testing adding new rule	Wed May 09 22:01:00 +0200 2007	Test paragraph §1000	Show Edit Destroy
mothers_weeks	9				Show Edit Destroy
fathers_weeks	6				Show Edit Destroy
shared_weeks_80	44				Show Edit Destroy
shared_weeks_100	54				Show Edit Destroy
employee_100	200				Show Edit Destroy
employee_80	160				Show Edit Destroy
selfemployed_no_ins_100	100				Show Edit Destroy
selfemployed_no_ins_80	80				Show Edit Destroy
selfemployed_freelancer_100	100				Show Edit Destroy

Next page
New calculationrule

To home

Figure 84 Testing REQ-3, REQ-3.1, REQ-3.2 and REQ-3.3 – Admin interface list new rule

We then tested to edit to the new rule by clicking “Edit”. We were then directed to a new page and entered “_edited” at each field and clicked “Update”. Figure 85 illustrates the confirmation page for that we had edited the calculation rule.



Administration of the database

Rule name: Testrule_edited

Current value: 1000

Description: Testing adding new rule_edited

Date of rule change: Wed May 09 22:01:00 +0200 2007

Regulation paragraph: Test paragraph §1000_edit

Edit | Back

To home

Figure 85: Testing REQ-3, REQ-3.1, REQ-3.2 and REQ-3.3 – Admin interface editing rule

At last we tested to delete the new calculation rule by clicking “Destroy”. We had to confirm by clicking “OK” when a confirmation window popped up. The rule was deleted and the page looked the same at the window in Figure 82.

Test result	REQ-3, REQ-3.1, REQ-3.2 and REQ-3.3 are successfully implemented.

Table 10: Test no. 3 – Testing REQ-3, REQ-3.1, REQ-3.2 and REQ-3.3

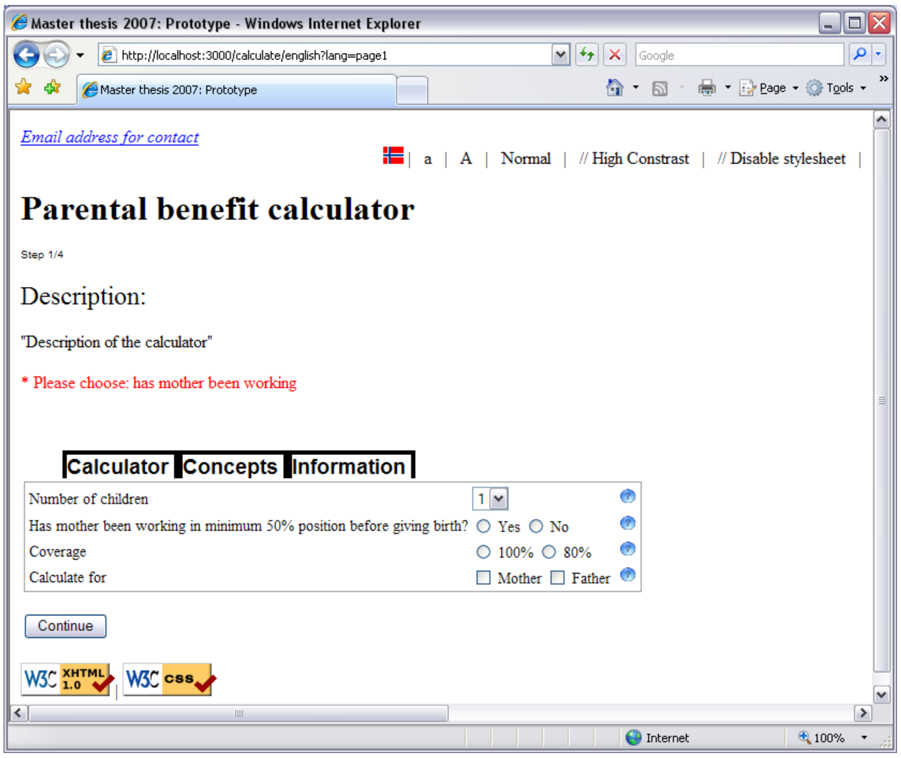
Test no. 4	
Test description	In this test we will test the requirements REQ-4, REQ-4.1, REQ-4.2, REQ-4.3 and REQ-4.4.
Task description	REQ-4: The prototype must be able to switch style on and off. REQ-4.1: A user can change the font-size. REQ-4.2: A user can change the colour contrast. REQ-4.3: A user can choose to display the text in Norwegian. REQ-4.4: A user can choose to display the text in other languages.
Performing the task	We tested these requirements by clicking on each of the button on the menu on the top of page. The results of these are the same as illustrated in Figure 33 at page 66.
Test result	REQ-4, REQ-4.1, REQ-4.2, REQ-4.3 and REQ-4.4 are successfully implemented.

Table 11: Test no. 4 – Testing REQ-4, REQ-4.1, REQ-4.2, REQ-4.3 and REQ-4.4

Test no. 5	
Test description	In this test we will test the requirements: REQ-6, REQ-6.1 and REQ-6.2.
Task description	REQ-6: The prototype must be able to present the results from the calculations REQ-6.1: Present the parental benefit as numbers. REQ-6.2: Present the parental benefit in a graphical drawn calendar.

Performing the task	The requirements were tested in Test no. 2. The validation can be seen in Figure 80 at page 87.
Test result	REQ-6, REQ-6.1 and REQ-6.2 are successfully implemented.

Table 12: Test no. 5 – Testing REQ-6, REQ-6.1 and REQ-6.2

Test no. 6	
Test description	In this test we will test the requirement: REQ-7.
Task description	REQ-7: The prototype must detect if users have missed to enter the required inputs into the calculation.
Performing the task	We tested to continue with the calculation by **not** entering a required input into the calculator. The calculator detected this and printed out an error message in red text on the screen as illustrated in Figure 86.  Master thesis 2007: Prototype - Windows Internet Explorer http://localhost:3000/calculate/english?lang=page1 Email address for contact a | A | Normal | // High Constrast | // Disable stylesheet | **Parental benefit calculator** Step 1/4 Description: "Description of the calculator" * Please choose: has mother been working Calculator Concepts Information Number of children 1 Has mother been working in minimum 50% position before giving birth? Yes No Coverage 100% 80% Calculate for Mother Father Continue *Figure 86: Testing REQ-7 – Missing input detection* We tested all the inputs and the result was the same for every inputs

Test result	REQ-7 is successfully implemented.

Table 13: Test no. 6 – Testing REQ-7

8.1.3 Summary

All the requirements stated in chapter 5.4 were successfully implemented.

8.2 User Testing

We made use of user testing with real users for our new prototype, in order to identify new usability problems and correct them into the final version of the new prototype.

The following sections will present the results from the usability testing of the new prototype with real test users and how these results impact the usability improvement of the prototype.

8.2.1 Test Plan

Before the user testing was conducted a test plan was composed. The test plan can be seen in Table 14.

The goal of the test	The goal for the test is to discover usability problems with the newly developed parental benefit calculator prototype. Learn what is good and bad with the interface. By identifying these usability problems the prototype can be further improved before the final version of it.
Where will the test take place?	The test will take place in a lab at Agder University College in Grimstad.
When will the test take place?	The test takes place on Friday 11.05.2007 at 20:00. All the test users will be available at that time.
How long is each session estimated to take?	Each test session will take about 20 minutes.

What computer support is needed for the test?	Any computer with internet access will do, as long as the server for hosting the prototype is running.
What software is needed for the test?	A web browser of any kind is all that is needed.
Who are the test users?	Test user 1: • Age: 19 • Gender: Female • Computer experience: Novice. This user has no experience with software usability. The user doesn't use a computer often. User has never used the parental benefit calculator before. Test user 2: • Age: 26 • Gender: Male • Computer experiene: Expert. This user is a system developer and has good experience with software usability. This user has used the calculator earlier. Test user 3: • Age: 25 • Gender: Male • Computer experience: Expert. This user is a computer engineer and has a bit experience with software usability. This user has not used the calculator before. Test user 4: • Age: 20 • Gender: Female • Computer experience: Novice. This user has good experience with using the computer for chatting, reading and writing emails, and reading newspapers. This user didn't even know about such a calculator. Test user 5: • Age: 44 • Gender: Female • Computer experience: None. This user never uses a computer. This user has not used the calculator before.
How to get hold of the users?	The users are all located in Grimstad.
What tasks will the users perform?	First of all, they will be asked to imagine how it

	would be to be a parent and to calculate the parental benefit with a part-time job. The users will be asked to perform calculations of the parental benefit and leave. The questions that the test users will have in front of them all the time are: • How much benefit can you get? • How much benefit can you get if you have a 50% part-time job? • How long can you stay at home? • When is the last day for taking out the parental benefit and leave? The users will also be asked questions about what they thought were difficult about using the prototype, at what was good and bad at the end of the test session.
What user aids will be made available to the test users?	The users will have all the information presented at http://www.nav.no at their disposal.
How much help can the test users get from the researcher?	Only explain to the test users when they click on the "question mark" icons, since no help text are implemented. Otherwise, the researcher will just observe the users.
What data is going to be collected?	The feedbacks on the use of the prototype must be collected. When observing the use during the test sessions, the difficulties that the users encounter must be registered.
How are the data going to be analysed?	The feedbacks from all the test sessions will be compared to each other. The observations of the use at each test session will also be compared. These data will be analysed by comparing those with each other, and draw conclusions about the design and usability of the prototype. The results for each task will be examined and presented.

Table 14: The test plan for usability testing

8.2.2 Test Results

According to the test plan presented in Table 14, the user testing was performed with 5 users. The following tasks that the users were given are numbered from 1 to 4:

1. How much benefit can you get?
2. How much benefit can you get if you have a 50% part-time job?
3. How long can you stay at home?
4. When is the last day for taking out the parental benefit and leave?

Each of the following tables, Table 15 to Table 19, will present the results from each test session.

Test user 1	
Difficulties experienced	The calculation was erased when going from the information tab back to the calculator tab. Some difficult information presented for the inputs.
Results for task no. 1	User managed to see this at once.
Results for task no. 2	User managed to calculate the benefit with a part-time job.
Results for task no. 3	User managed to find out how many weeks the leave consisted of.
Results for task no. 4	User managed how to find out the max date.
Feedback on design and fucntionality	Good design. The help texts were very useful for the calculation.

Table 15: Results from usability testing with Test User 1

Test user 2	
Difficulties experienced	User had problems seeing the tabs. User encountered problems on page 3. User decided to skip that page and the calculator didn't handle the missing user input. Income was 0 and the calculation failed, and an error message was printed on screen by Ruby on rails.

	User missed the functionality for seeing a calendar for the whole parental leave. User had problems with seeing the updated parental benefit when choosing the percentage of part-time job.
Results for task no. 1	User managed to see this after starting the calculation over after an error.
Results for task no. 2	User understood how to calculate the benefit with a part-time job, but did not see the results of the calculation at once.
Results for task no. 3	User managed to find out how many weeks the leave consisted of after on the second retry of the calculation.
Results for task no. 4	User managed how to find out the max date.
Feedback on design and functionality	Easy to understand. Good calculator. The needed information was found. Help texts were used all for every single input, and provided good information.

Table 16: Results from usability testing with Test User 2

Test user 3	
Difficulties experienced	User had problems with understanding some information presented for the inputs. User didn’t understand how many children to choose, number of children the user had in total or number of children that that are newly born or will be born. The page was colourless. The user had visited NAV’s web sites earlier and missed the red coloured frame in the prototype.
Results for task no. 1	User managed to see this at once.
Results for task no. 2	User managed this.
Results for task no. 3	User managed to find out how many weeks the

	leave consisted of.
Results for task no. 4	User managed to find out the max date.
Feedback on design and functionality	Good design and it was easy to use calculator. The description for each page was very useful.

Table 17: Results from usability testing with Test User 3

Test user 4	
Difficulties experienced	User didn’t notice the description for the page. User had problems when entering the date manually. User didn’t understand that it was a pop up calendar functionality. User didn’t see that there were tabs for finding additional information. The calculator didn’t stop the user when the user continued the calculation without inputting the income.
Results for task no. 1	User managed to see this at once.
Results for task no. 2	User managed to find out how many weeks the leave consisted of.
Results for task no. 3	User managed to find out how many weeks the leave consisted of after retry.
Results for task no. 4	User managed how to find out the max date.
Feedback on design and functionality	Good design.

Table 18: Results from usability testing with Test User 4

Test user 5	
Difficulties experienced	User had problems with seeing the updated parental benefit when choosing the percentage of part-time job.

	User had problems with seeing the original sat fonts on the page. Used didn’t understand all the information presented on the page.
Results for task no. 1	User managed to see this at once.
Results for task no. 2	User managed this.
Results for task no. 3	User managed to find out how many weeks the leave consisted of, but knew this in beforehand.
Results for task no. 4	User managed to find out the max date.
Feedback on design and functionality	Easy to perform calculation. Every help texts were used by the user. User made use of the font size changing functionality, and thought it was good.

Table 19: Results from usability testing with Test User 5

8.2.3 Summary

After the usability testing with the test users the results were analysed in order to further improve the parental benefit calculator prototype. The key findings from the user testing are:

- The description for each page need to be more visible
- Unclear information on the page were clarified with help texts
- Tabs must be more visible
- Wrong link in tab. When going back to calculator from other tabs, the first page is displayed
- Calculator must prevent the user to continue if the income has not been inputted
- Graphical overview with calendar could be interactive and support pagination between several years
- The result for the parental benefit with graded-withdrawal could be easier to find.
- The help text for the coverage was used by all users
- The help text for number of children was used by all users
- 4 of the users didn’t understand the word graded withdrawal
- 2 of the users didn’t understand the pension-given income fields
- The result of the benefit calculated with graded withdrawal was hard to find. Must move it to the top of the page to be more visible
- Must add an extra field for displaying the parental benefit to make it more clearly
- Overall good design
- Easy-to-follow calculation

8.3 Heuristic evaluation of NAV's current solution

8.3.1 Test Plan

We performed this test by looking at the parental benefit calculator offered by NAV which is located at http://tjenester.nav.no/fodselspenger/fpenger/fpenger1.do .
We inspected the calculator systematically for usability problems in regards to the 10 heuristics presented in chapter 4.1.1 on page 29. When inspecting the calculator we noted every usability problem that we identified.

8.3.2 Test Results

For each of the 10 heuristics, we will present our findings in Table 20.

Heuristic	Findings
1. Visibility status	The calculator is keeping the user informed about what is going on. An example, is when the user does not input the required inputs, the calculator is alerting the user about this.
2. Match between system and the real world	There were some words that were unfamiliar to the user, but this was explained in form of a help text by the calculator.
3. User control and freedom	A user is able to go back and forth between the calculation steps.
4. Consistency and standards	User does not have to wonder whether different situations, actions mean the same thing.
5. Error prevention	The calculator has error prevention implemented, but not all of them are working correctly. An example is to leave an input field empty and continue the calculation. The user was alerted with a message, and a solution was presented.
6. Recognition rather than recall	User had to remember some information from page to another, but some of them were easy to retrieve. All in all, most of the information, options and actions were visible to the user all the time. The different steps of the calculation could have a description for how to use the

	calculator.
7. Flexibility and efficiency of use	The solution is flexible and efficient to use.
8. Aesthetic and minimalist design	The solution doesn’t contain irrelevant information. It has also a minimalist design.
9. Help users recognize, diagnose, and recover from errors	The error messages are easy to understand and self explaining.
10. Help and documentation	On the “Informasjon” tab, too much information was presented.

Table 20: Results from heuristic evaluation of NAV’s calculator

8.3.3 Summary

The usability problems were identified and became a part of the iterative design process in this project. We designed, developed and tested our new prototyping and kept these usability problems in mind in order to avoid them for occurring in our new prototype.

8.4 Testing according to W3C recommendations

8.4.1 Test Plan

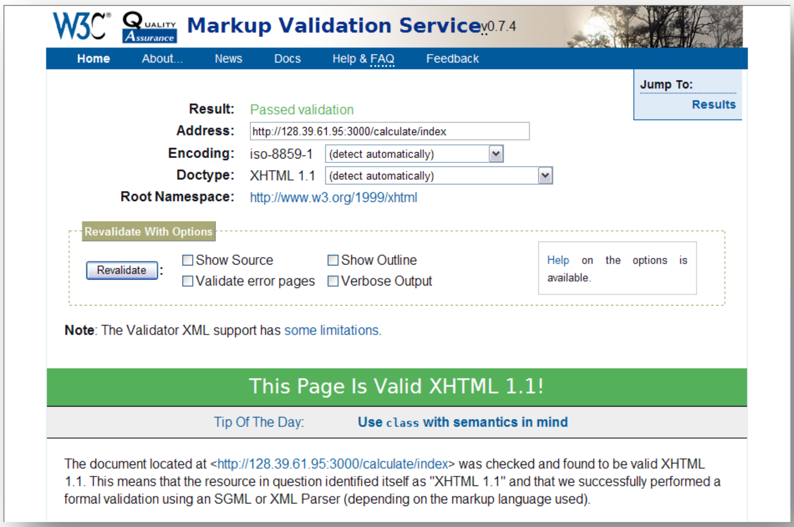
On every page in the new parental benefit calculator, there are two buttons on the bottom of the page for validation of the current page according to W3C recommendation. These buttons can for example be seen in Figure 86 on page 92.
To test web pages in regards to the W3C recommendations we will click on these buttons, and wait for the result from the validator.

By clicking them we will be directed to the W3C Markup Validation Service or the W3C CSS Validation Service. The first mention service is a free service that checks documents in formats like HTML and XHTML for conformance to W3C Recommendations and other standards.

The second service is also a free service that checks CSS files, either in (X)HTML documents or standalone for conformance to W3C recommendations.

8.4.2 Test Results

These services have been used during the whole development phase of the new prototype to ensure that the final version of the prototype will be in conformance to the W3C recommendations. The services are illustrated in Figure 87 and Figure 88.


W3C Quality Assurance Markup Validation Service v0.7.4
Home About... News Docs Help & FAQ Feedback
Jump To: Results
Revalidate With Options
Revalidate
Show Source
Show Outline
Validate error pages
Verbose Output
Help on the options is available.

Result: Passed validation
Address: http://128.39.61.95:3000/calculate/index
Encoding: iso-8859-1 (detect automatically)
Doctype: XHTML 1.1 (detect automatically)
Root Namespace: http://www.w3.org/1999/xhtml

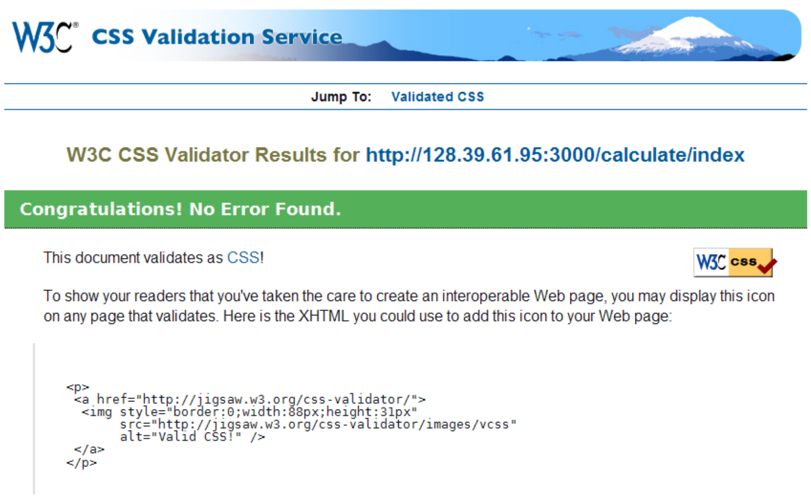
Note: The Validator XML support has some limitations.

This Page Is Valid XHTML 1.1!

Tip Of The Day: Use `class` with semantics in mind

The document located at <http://128.39.61.95:3000/calculate/index> was checked and found to be valid XHTML 1.1. This means that the resource in question identified itself as "XHTML 1.1" and that we successfully performed a formal validation using an SGML or XML Parser (depending on the markup language used).

Figure 87: Quality Assurance - Markup Validation Service



W3C CSS Validator Results for http://128.39.61.95:3000/calculate/index

Congratulations! No Error Found.

This document validates as CSS!

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

```
<p>
 <a href="http://jigsaw.w3.org/css-validator/">
  <img style="border:0;width:88px;height:31px"
       src="http://jigsaw.w3.org/css-validator/images/vcss"
       alt="Valid CSS!" />
 </a>
</p>
```

Figure 88: CSS Validation Service

8.4.3 Summary

When we tested the web pages of our new prototype, the validator checked the source code of those pages. If the validator found an error it would report the error, and present where in the source code the error occurred so that we could correct the problem. All the web pages were checked using these validators, and all of them passed the validation in the final version of the new prototype.

8.5 Testing according to accessibility guidelines

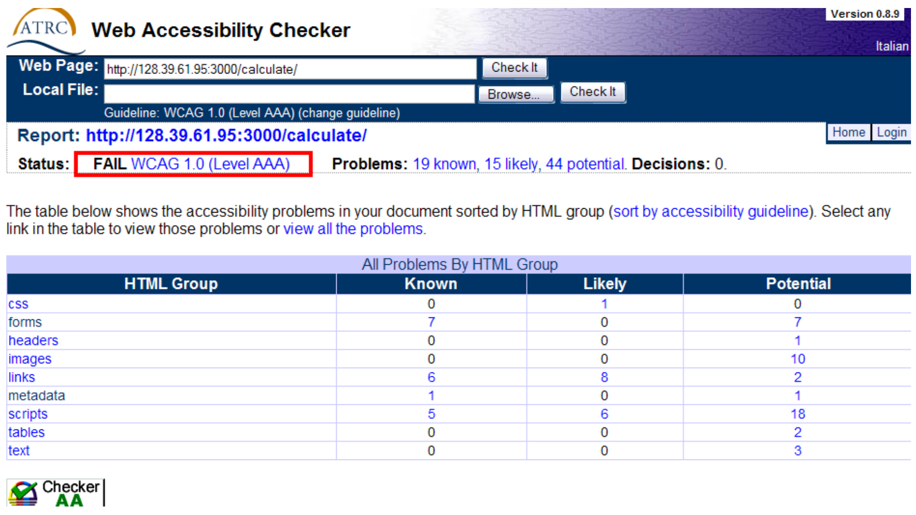
8.5.1 Test Plan

We tested the new prototype for accessibility problems by using tools which checked the web pages to see if they conformed to the accessibility guidelines.

The accessibility of the new prototype was evaluated early and at the end of the development phase. Different tools were used to evaluate the accessibility of the new prototype. The same tool was used for checking both NAV's current solution and our new prototype. The tools are discussed in chapter 4.4.2 on page 38.
The problems that were detected were improved, and evaluations were then performed again. This continued until all the problems were solved and the web pages were in conformance to the accessibility guidelines.

8.5.2 Test Results

We tested the main page and received 19 known problems, 15 likely and 44 potential as illustrated in Figure 89.



Guideline: WCAG 1.0 (Level AAA) (change guideline)

Report: http://128.39.61.95:3000/calculate/

Status: FAIL WCAG 1.0 (Level AAA) Problems: 19 known, 15 likely, 44 potential. Decisions: 0.

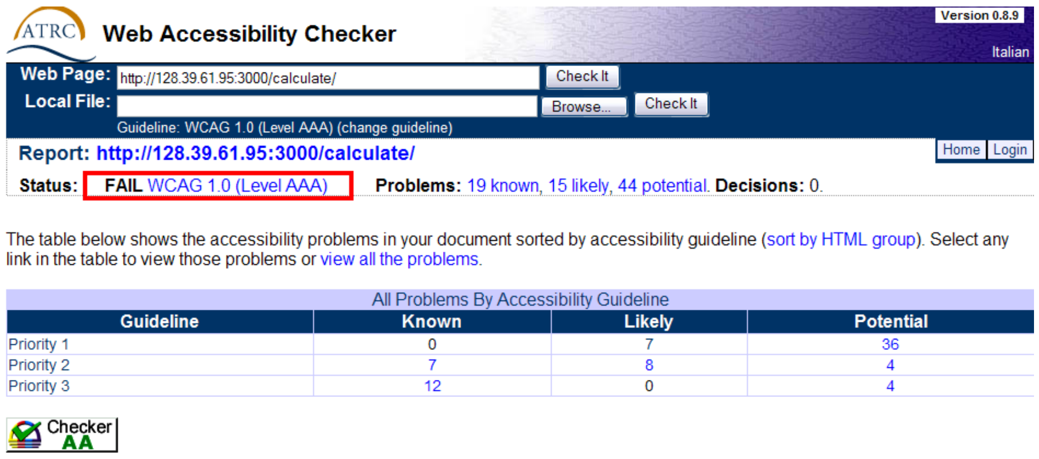
The table below shows the accessibility problems in your document sorted by HTML group (sort by accessibility guideline). Select any link in the table to view those problems or view all the problems.

All Problems By HTML Group

HTML Group	Known	Likely	Potential
css	0	1	0
forms	7	0	7
headers	0	0	1
images	0	0	10
links	6	8	2
metadata	1	0	1
scripts	5	6	18
tables	0	0	2
text	0	0	3

Figure 89: A-Checker : Checking main page for accessibility problems

Then, we sorted the problems by accessibility guidelines as illustrated in Figure 90.



Guideline: WCAG 1.0 (Level AAA) (change guideline)

Report: http://128.39.61.95:3000/calculate/

Status: FAIL WCAG 1.0 (Level AAA) Problems: 19 known, 15 likely, 44 potential. Decisions: 0.

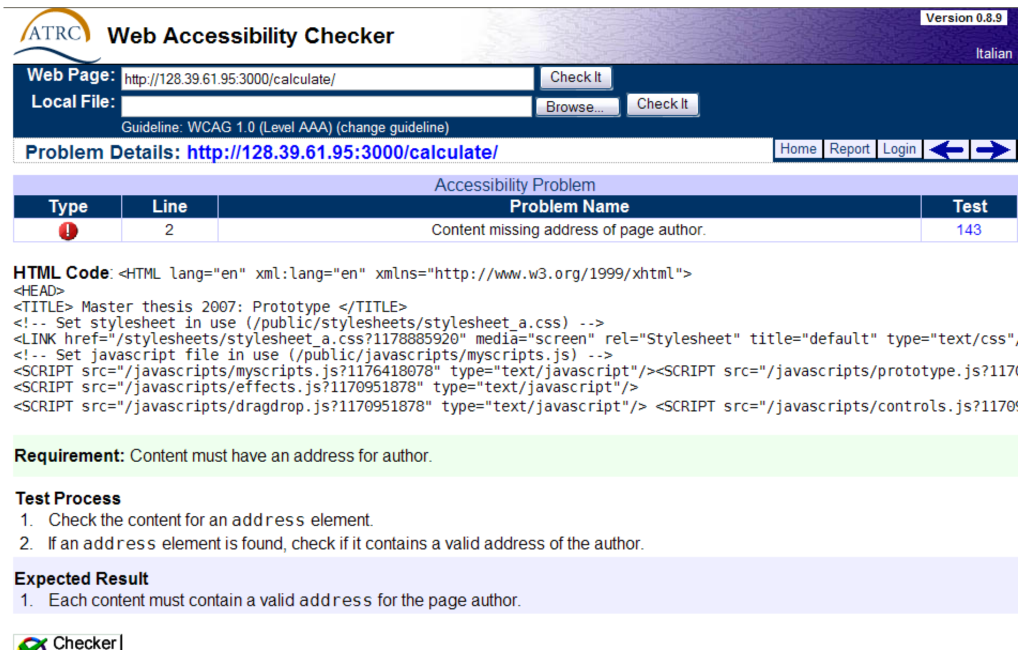
The table below shows the accessibility problems in your document sorted by accessibility guideline (sort by HTML group). Select any link in the table to view those problems or view all the problems.

All Problems By Accessibility Guideline

Guideline	Known	Likely	Potential
Priority 1	0	7	36
Priority 2	7	8	4
Priority 3	12	0	4

Figure 90: A-Checker: Sort problems by accessibility guidelines

By clicking on the priority, we could see all the problems in that priority group. We clicked on one problem and were directed to a page with a description for what to test and what the expected result of the accessibility guideline was as illustrated in Figure 91.


Web Accessibility Checker

Version 0.8.9

Italian

Web Page: http://128.39.61.95:3000/calculate/

Local File:

Guideline: WCAG 1.0 (Level AAA) (change guideline)

Problem Details: http://128.39.61.95:3000/calculate/

Accessibility Problem

Type	Line	Problem Name	Test
●	2	Content missing address of page author.	143

HTML Code:
```
<HTML lang="en" xml:lang="en" xmlns="http://www.w3.org/1999/xhtml">
<HEAD>
<TITLE> Master thesis 2007: Prototype </TITLE>
<!-- Set stylesheet in use (/public/stylesheets/stylesheet_a.css) -->
<LINK href="/stylesheets/stylesheet_a.css?1178885920" media="screen" rel="Stylesheet" title="default" type="text/css",
<!-- Set javascript file in use (/public/javascripts/myscripts.js) -->
<SCRIPT src="/javascripts/myscripts.js?1176418078" type="text/javascript"/><SCRIPT src="/javascripts/prototype.js?117
<SCRIPT src="/javascripts/effects.js?1170951878" type="text/javascript"/>
<SCRIPT src="/javascripts/dragdrop.js?1170951878" type="text/javascript"/> <SCRIPT src="/javascripts/controls.js?1170
```

Requirement: Content must have an address for author.

Test Process
1. Check the content for an `address` element.
2. If an `address` element is found, check if it contains a valid address of the author.

Expected Result
1. Each content must contain a valid `address` for the page author.

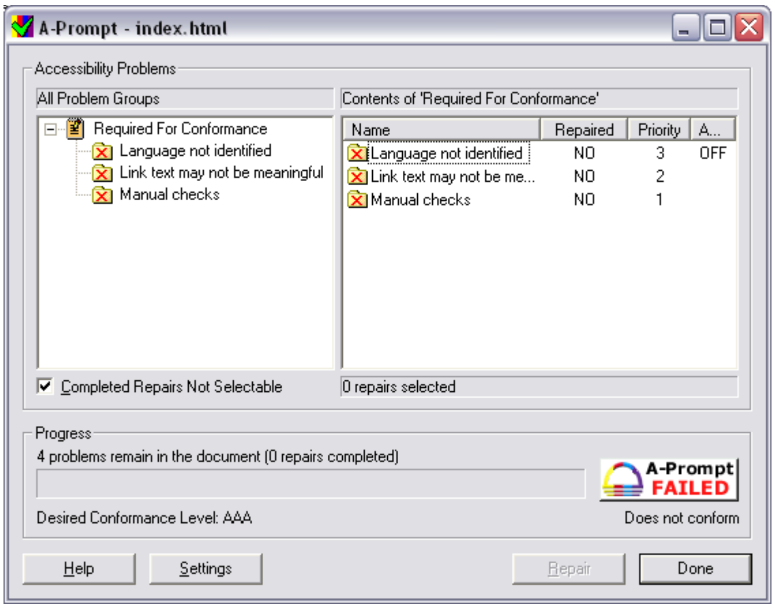
Checker

Figure 91: A-Checker: Viewing the accessibility problem

This tool was used to check every single pages of the new prototype, and the known accessibility problems were corrected and we achieved the expected results.

We also tested the prototype by using a tool called A-Prompt[43]. This was a free tool that was downloaded and installed on the computer. We tested the prototype with this tool which also supported automatic repair of the HTML files. After providing the tool with the HTML file, the errors were displayed to us as illustrated in Figure 92.

[43] The home page of A-Prompt, a Web accessibility verifier, can be found at http://www.aprompt.ca/ . Online version is located at this link: http://checker.atrc.utoronto.ca/index.html



A-Prompt - index.html

Accessibility Problems

All Problem Groups

- Required For Conformance
  - Language not identified
  - Link text may not be meaningful
  - Manual checks

Contents of 'Required For Conformance'

Name	Repaired	Priority	A...
Language not identified	NO	3	OFF
Link text may not be me...	NO	2	
Manual checks	NO	1	

☑ Completed Repairs Not Selectable

0 repairs selected

Progress

4 problems remain in the document (0 repairs completed)

A-Prompt FAILED

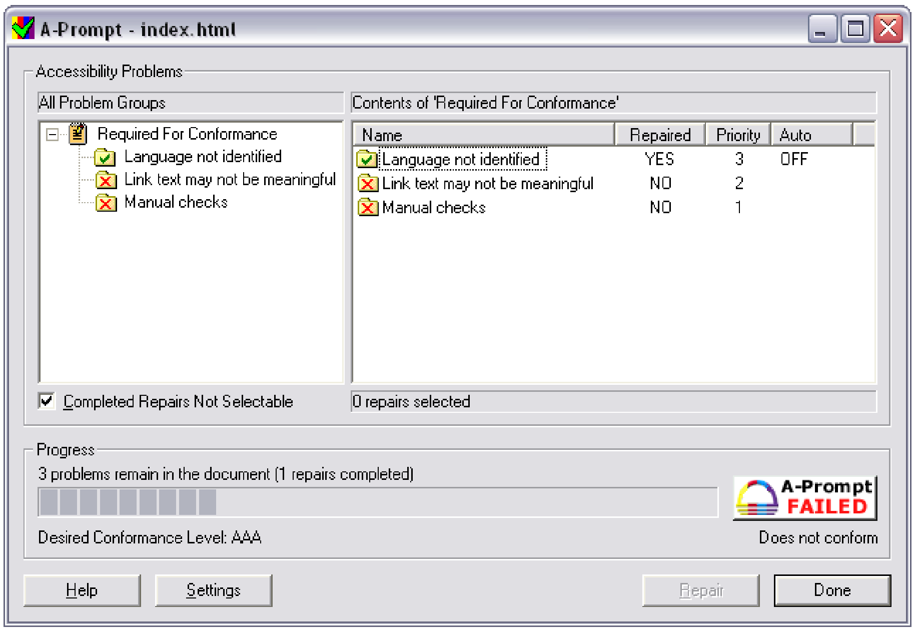
Desired Conformance Level: AAA

Does not conform

Help Settings Repair Done

Figure 92: A-Prompt – detecting accessibility problems

By double clicking on each of the errors we had the opportunity to correct the accessibility problems.



A-Prompt - index.html

Accessibility Problems

All Problem Groups

- Required For Conformance
  - Language not identified
  - Link text may not be meaningful
  - Manual checks

Contents of 'Required For Conformance'

Name	Repaired	Priority	Auto
Language not identified	YES	3	OFF
Link text may not be meaningful	NO	2	
Manual checks	NO	1	

☑ Completed Repairs Not Selectable

0 repairs selected

Progress

3 problems remain in the document (1 repairs completed)

A-Prompt FAILED

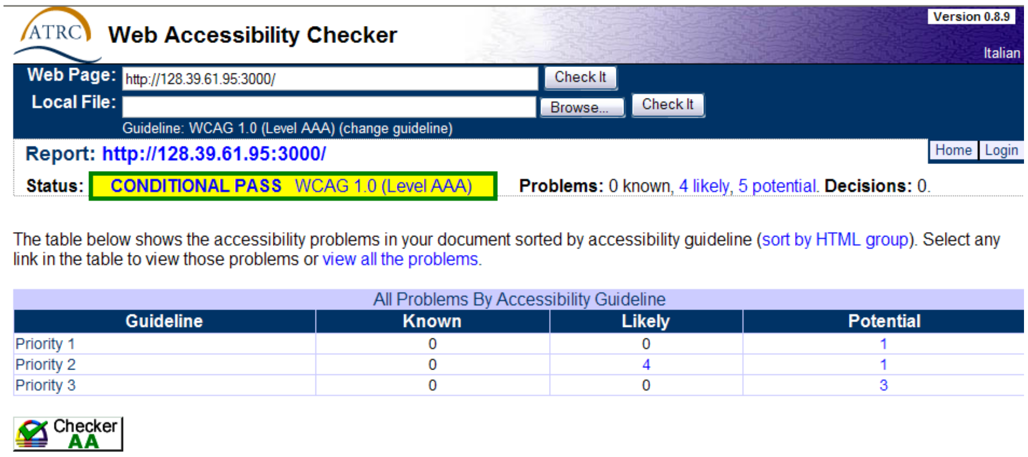
Desired Conformance Level: AAA

Does not conform

Help Settings Repair Done

Figure 93: A-Prompt – accessibility problem fixed

We double checked this as illustrated in Figure 94, by using the A-Checker tool, and no accessibility problems were found with the HTML file we just fixed with the A-Prompt tool. The page was in conformance with the accessibility guidelines.


Web Accessibility Checker

Version 0.8.9

Web Page: http://128.39.61.95:3000/

Local File:

Guideline: WCAG 1.0 (Level AAA) (change guideline)

Report: http://128.39.61.95:3000/

Status: CONDITIONAL PASS WCAG 1.0 (Level AAA)

Problems: 0 known, 4 likely, 5 potential. Decisions: 0.

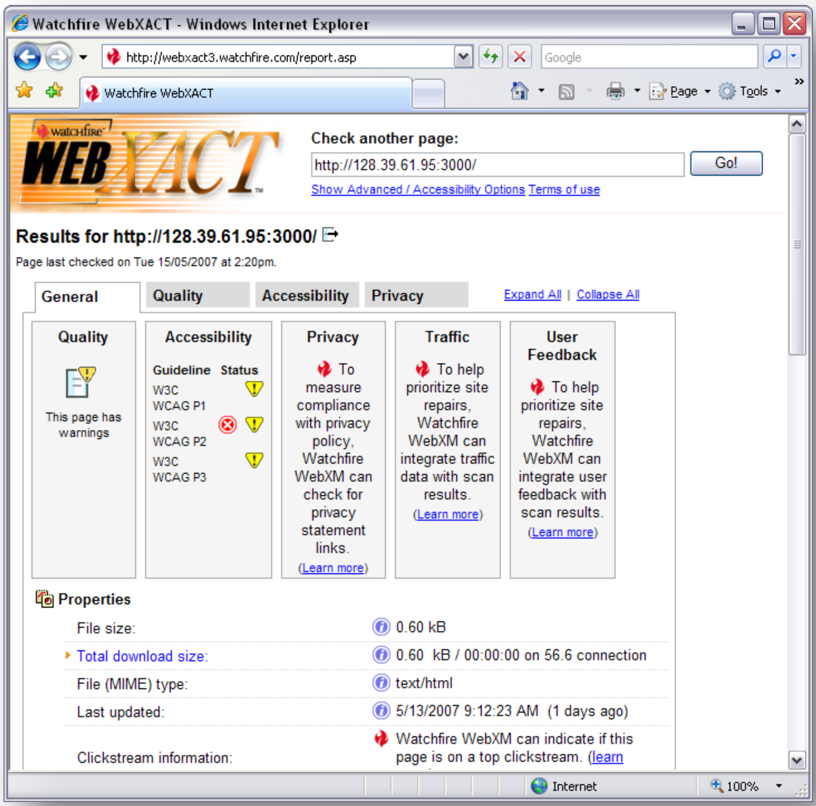
The table below shows the accessibility problems in your document sorted by accessibility guideline (sort by HTML group). Select any link in the table to view those problems or view all the problems.

All Problems By Accessibility Guideline

Guideline	Known	Likely	Potential
Priority 1	0	0	1
Priority 2	0	4	1
Priority 3	0	0	3

Figure 94: A-Checker – Check for accessibility problems after fix with A-Prompt

Another tool was also used for testing the accessibility of the prototype. When using this tool we received the feedback on that the page had warnings as illustrated in Figure 95: WebXACT - Checking main page for accessibility problems


Watchfire WebXACT - Windows Internet Explorer

http://webxact3.watchfire.com/report.asp

Check another page: http://128.39.61.95:3000/

Results for http://128.39.61.95:3000/

Page last checked on Tue 15/05/2007 at 2:20pm.

Quality

This page has warnings

Accessibility

Guideline	Status
W3C WCAG P1	⚠
W3C WCAG P2	⊗ ⚠
W3C WCAG P3	⚠

Privacy

To measure compliance with privacy policy, Watchfire WebXM can check for privacy statement links. (Learn more)

Traffic

To help prioritize site repairs, Watchfire WebXM can integrate traffic data with scan results. (Learn more)

User Feedback

To help prioritize site repairs, Watchfire WebXM can integrate user feedback with scan results. (Learn more)

Properties

File size:	0.60 kB
Total download size:	0.60 kB / 00:00:00 on 56.6 connection
File (MIME) type:	text/html
Last updated:	5/13/2007 9:12:23 AM (1 days ago)
Clickstream information:	Watchfire WebXM can indicate if this page is on a top clickstream. (learn

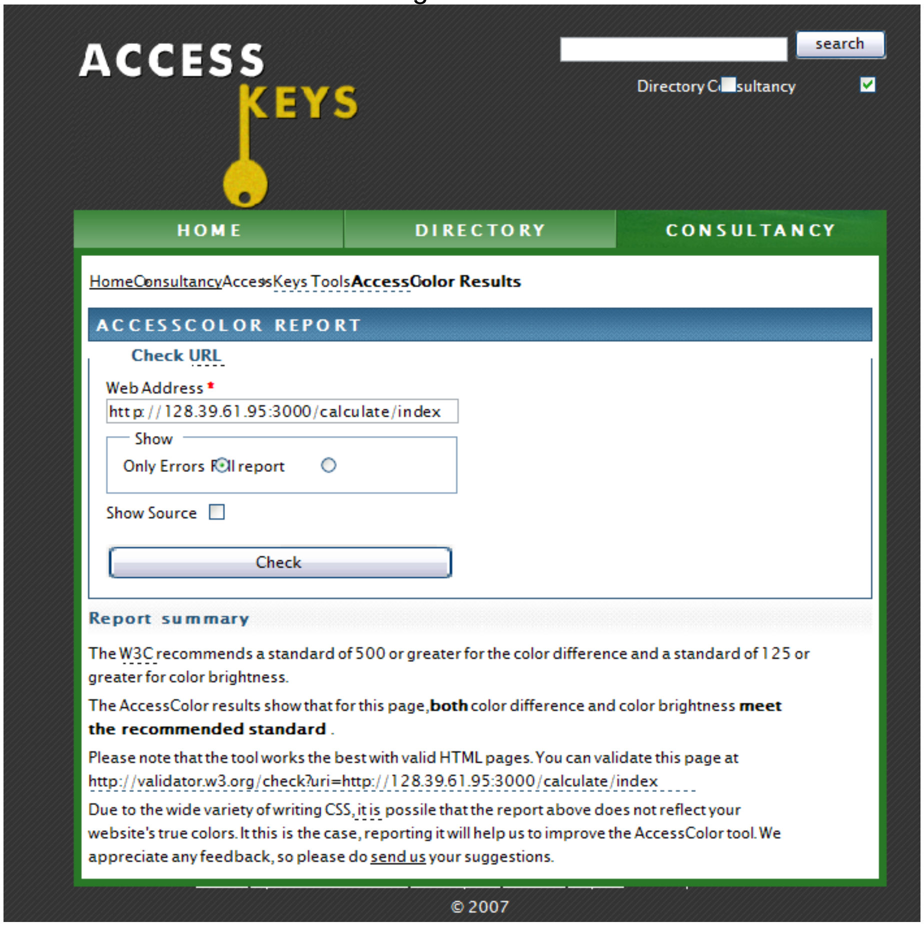
Figure 95: WebXACT - Checking main page for accessibility problems

The same approach was used for every page in the new prototype.

We also tested the web pages for colour contrast and colour brightness for conformance to the accessibility guidelines and W3C recommendations.

The first tool used, AccessColor, was used for checking the colour contrast and colour brightness between the foreground and background of all the elements on the web page ensuring that the contrast was high enough with visual impairments. The second tool, CSS Analyser, tested the same thing but in addition it also made sure that relevant sizes are specified in relative units of measurement.

Since all the pages of the new prototype used the same layout we only needed to test one of the pages with AccessColor. We decided to test the first page of the calculation and received the feedback that the colour contrast and colour brightness for the page met the recommended standard. This is illustrated in Figure 96.



HomeConsultancyAccessKeys Tools**AccessColor Results**

ACCESSCOLOR REPORT

Check URL

Web Address *

http://128.39.61.95:3000/calculate/index

Show

Only Errors [✓] Full report ○

Show Source ☐

Check

Report summary

The W3C recommends a standard of 500 or greater for the color difference and a standard of 125 or greater for color brightness.

The AccessColor results show that for this page, **both** color difference and color brightness **meet the recommended standard** .

Please note that the tool works the best with valid HTML pages. You can validate this page at http://validator.w3.org/check?uri=http://128.39.61.95:3000/calculate/index

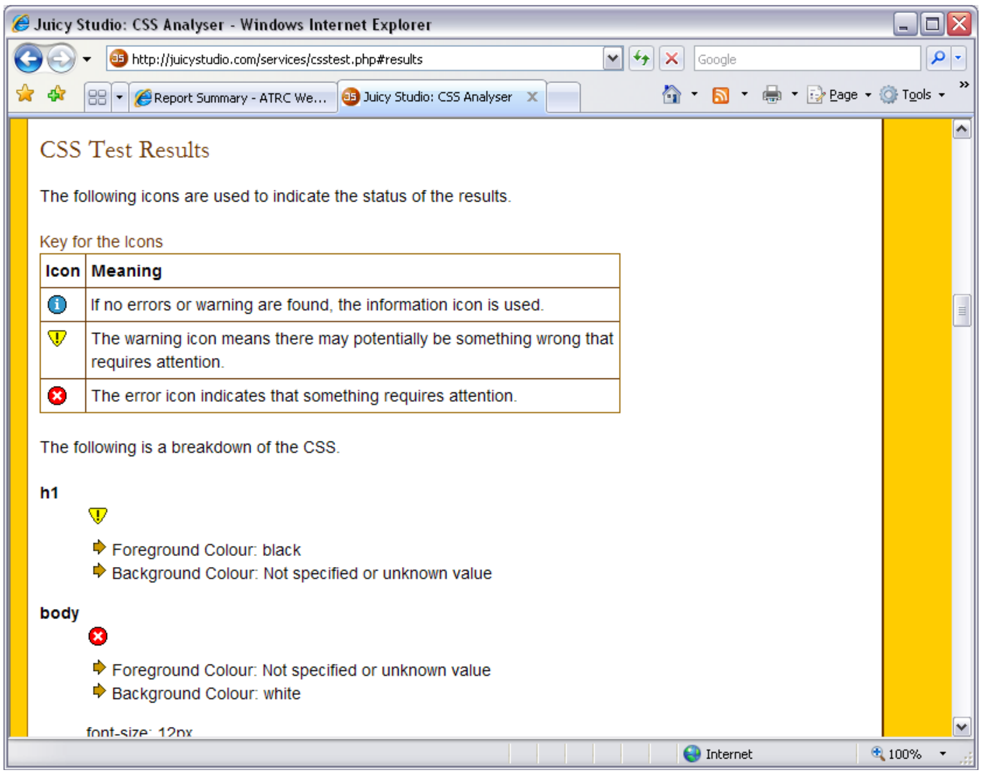
Due to the wide variety of writing CSS, it is possile that the report above does not reflect your website's true colors. It this is the case, reporting it will help us to improve the AccessColor tool. We appreciate any feedback, so please do send us your suggestions.

© 2007

Figure 96: AccessColor – Testing colour contrast and colour brightness

We also used the CSS Analyser tool to test the web pages of the new prototype to make sure that relevant sizes were specified in relative units of measurement.
This was performed in the end of the development phase and we received some errors when testing the CSS document. The test results were printed out to the screen, and we had to correct each problem one by one. Figure 97 is illustrating an example of the test results we got from the CSS Analyser.

By following the error description from the test result, the CSS document was corrected and tested several times until everything was correct. This is illustrated in Figure 98.


Juicy Studio: CSS Analyser - Windows Internet Explorer
http://juicystudio.com/services/csstest.php#results
Google
Report Summary - ATRC We...
Juicy Studio: CSS Analyser
Page
Tools
Internet
100%

CSS Test Results

The following icons are used to indicate the status of the results.

Key for the Icons

Icon	Meaning
ⓘ	If no errors or warning are found, the information icon is used.
⚠	The warning icon means there may potentially be something wrong that requires attention.
✖	The error icon indicates that something requires attention.

The following is a breakdown of the CSS.

h1

⚠

- Foreground Colour: black
- Background Colour: Not specified or unknown value

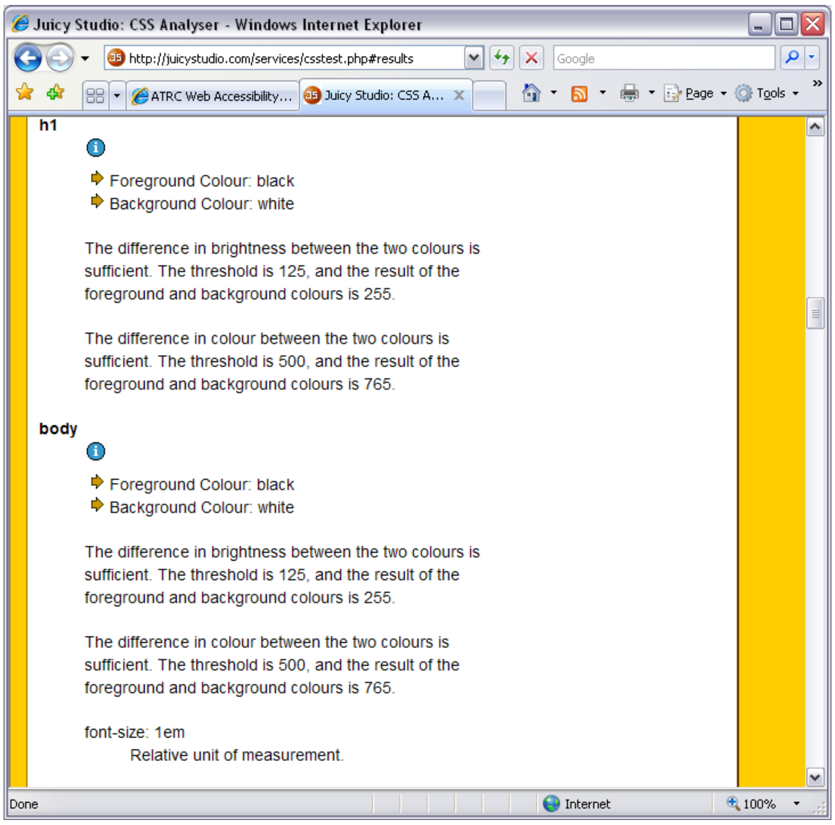
body

✖

- Foreground Colour: Not specified or unknown value
- Background Colour: white

font-size: 12px

Figure 97: CSS Analyser – Testing the CSS document: errors


Juicy Studio: CSS Analyser - Windows Internet Explorer
http://juicystudio.com/services/csstest.php#results
Google
ATRC Web Accessibility...
Juicy Studio: CSS A...
Page
Tools
Done
Internet
100%

h1

ⓘ

- Foreground Colour: black
- Background Colour: white

The difference in brightness between the two colours is sufficient. The threshold is 125, and the result of the foreground and background colours is 255.

The difference in colour between the two colours is sufficient. The threshold is 500, and the result of the foreground and background colours is 765.

body

ⓘ

- Foreground Colour: black
- Background Colour: white

The difference in brightness between the two colours is sufficient. The threshold is 125, and the result of the foreground and background colours is 255.

The difference in colour between the two colours is sufficient. The threshold is 500, and the result of the foreground and background colours is 765.

font-size: 1em
Relative unit of measurement.

Figure 98: CSS Analyser – Testing and validating the CSS document: successfulness

8.5.3 Summary

The web pages of the new prototype were checked for accessibility problems several times. When the tools detected a problem we needed to correct them and check the web pages again. We continued this process until all the known accessibility problems had been corrected.
The accessibility problems have also been reviewed by us to ensure that the prototype is accessible, since the use of the accessibility checker tools are not enough for determining the accessibility of the web pages.

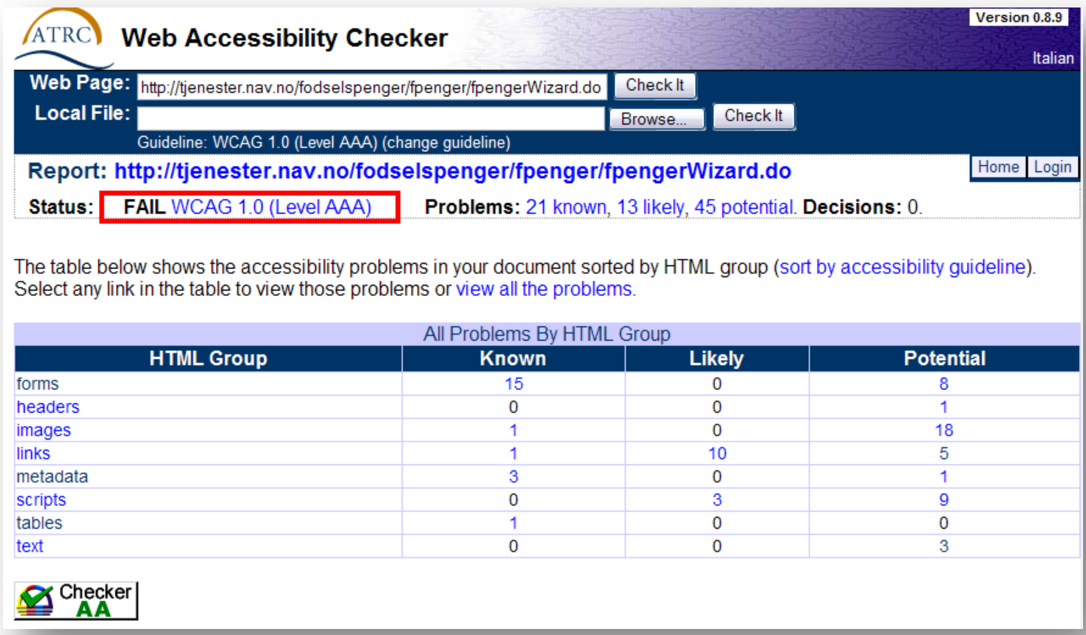
8.6 Accessibility evaluations of NAV's parental benefit calculator

8.6.1 Test plan

The parental benefit calculator offered by NAV today was checked for accessibility problems.
We used the A-Checker tool for evaluating it, and used the web content accessibility guidelines version 1.0 and tested it for level AAA conformance.

8.6.2 Test results

The content of the web page where the calculator was located had 21 known accessibility problems, 13 likely and 45 potential accessibility problems. For details about what the page had problems with, this is illustrated in Figure 99.



http://tjenester.nav.no/fodselspenger/fpenger/fpengerWizard.do

Guideline: WCAG 1.0 (Level AAA) (change guideline)

Report: http://tjenester.nav.no/fodselspenger/fpenger/fpengerWizard.do

Status: FAIL WCAG 1.0 (Level AAA) **Problems:** 21 known, 13 likely, 45 potential. **Decisions:** 0.

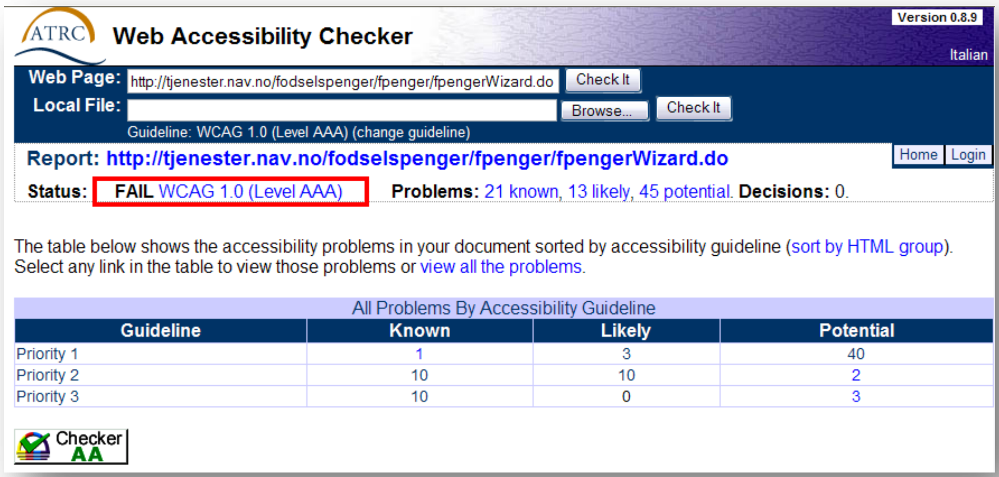
The table below shows the accessibility problems in your document sorted by HTML group (sort by accessibility guideline). Select any link in the table to view those problems or view all the problems.

All Problems By HTML Group

HTML Group	Known	Likely	Potential
forms	15	0	8
headers	0	0	1
images	1	0	18
links	1	10	5
metadata	3	0	1
scripts	0	3	9
tables	1	0	0
text	0	0	3

Figure 99: A-Checker - Testing NAV's calculator for accessibility problems (1)

Then, we sorted the accessibility problems by guidelines and found out that there were 1 known accessibility problem with Priority 1, and 10 problems in Priority 2 and also 10 problems in Priority 3. This can be seen in Figure 100.



Web Page: http://tjenester.nav.no/fodselspenger/fpenger/fpengerWizard.do

Guideline: WCAG 1.0 (Level AAA) (change guideline)

Report: http://tjenester.nav.no/fodselspenger/fpenger/fpengerWizard.do

Status: FAIL WCAG 1.0 (Level AAA) Problems: 21 known, 13 likely, 45 potential. Decisions: 0.

The table below shows the accessibility problems in your document sorted by accessibility guideline (sort by HTML group). Select any link in the table to view those problems or view all the problems.

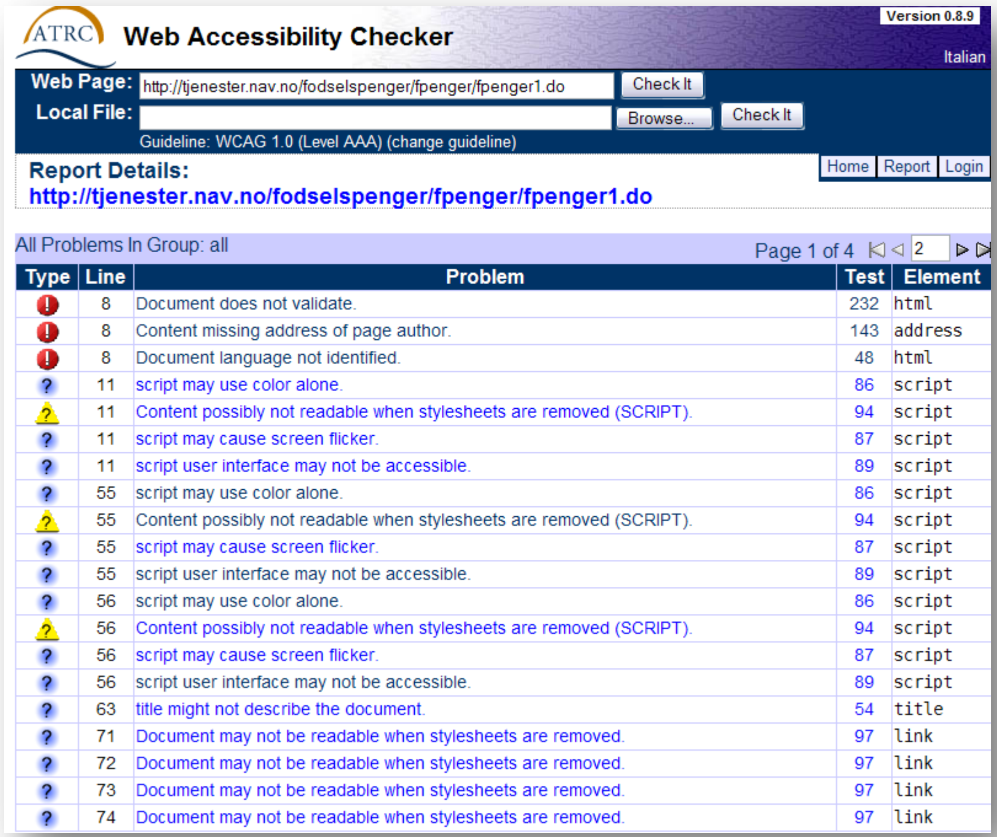
All Problems By Accessibility Guideline

Guideline	Known	Likely	Potential
Priority 1	1	3	40
Priority 2	10	10	2
Priority 3	10	0	3

Checker AA

Figure 100: A-Checker - Testing NAV's calculator for accessibility problems (2)

The tool also allowed us to view all the accessibility problems with the page. By clicking on each of the problem, we received detailed information about the problem and information about how we could fix the accessibility problem. Figure 101 illustrates the first page of how all the accessibility problems are presented.



Web Page: http://tjenester.nav.no/fodselspenger/fpenger/fpenger1.do

Guideline: WCAG 1.0 (Level AAA) (change guideline)

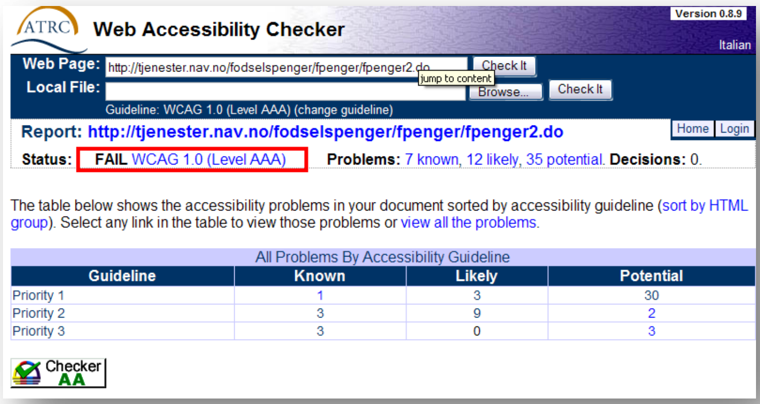
Report Details:
http://tjenester.nav.no/fodselspenger/fpenger/fpenger1.do

All Problems In Group: all Page 1 of 4 2

Type	Line	Problem	Test	Element
❗	8	Document does not validate.	232	html
❗	8	Content missing address of page author.	143	address
❗	8	Document language not identified.	48	html
?	11	script may use color alone.	86	script
⚠	11	Content possibly not readable when stylesheets are removed (SCRIPT).	94	script
?	11	script may cause screen flicker.	87	script
?	11	script user interface may not be accessible.	89	script
?	55	script may use color alone.	86	script
⚠	55	Content possibly not readable when stylesheets are removed (SCRIPT).	94	script
?	55	script may cause screen flicker.	87	script
?	55	script user interface may not be accessible.	89	script
?	56	script may use color alone.	86	script
⚠	56	Content possibly not readable when stylesheets are removed (SCRIPT).	94	script
?	56	script may cause screen flicker.	87	script
?	56	script user interface may not be accessible.	89	script
?	63	title might not describe the document.	54	title
?	71	Document may not be readable when stylesheets are removed.	97	link
?	72	Document may not be readable when stylesheets are removed.	97	link
?	73	Document may not be readable when stylesheets are removed.	97	link
?	74	Document may not be readable when stylesheets are removed.	97	link

Figure 101: A-Checker - Testing NAV's calculator for accessibility problems (3)

We also tested the second page of the parental benefit calculator for accessibility problems. This page had fewer problems than the first page. This is illustrated in Figure 102.



http://tjenester.nav.no/fodselspenger/fpenger/fpenger2.do

Guideline: WCAG 1.0 (Level AAA) (change guideline)

Report: http://tjenester.nav.no/fodselspenger/fpenger/fpenger2.do

Status: FAIL WCAG 1.0 (Level AAA) **Problems:** 7 known, 12 likely, 35 potential. **Decisions:** 0.

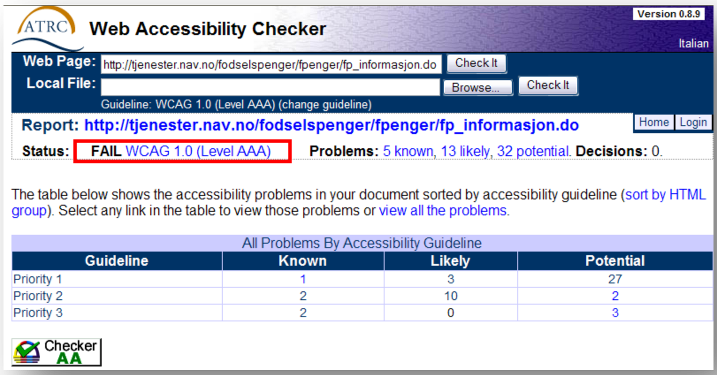
The table below shows the accessibility problems in your document sorted by accessibility guideline (sort by HTML group). Select any link in the table to view those problems or view all the problems.

All Problems By Accessibility Guideline

Guideline	Known	Likely	Potential
Priority 1	1	3	30
Priority 2	3	9	2
Priority 3	3	0	3

Figure 102

In addition, we tested the page with information about the parental benefit and leave, and the page with the explanations for the different concepts. This is illustrated in Figure 103 and Figure 104.



http://tjenester.nav.no/fodselspenger/fpenger/fp_informasjon.do

Guideline: WCAG 1.0 (Level AAA) (change guideline)

Report: http://tjenester.nav.no/fodselspenger/fpenger/fp_informasjon.do

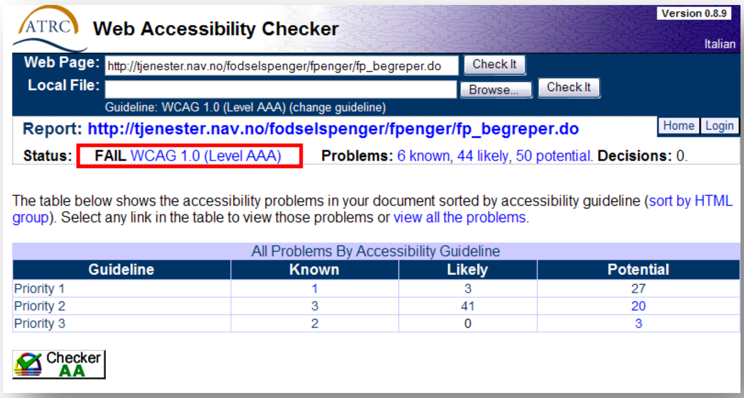
Status: FAIL WCAG 1.0 (Level AAA) **Problems:** 5 known, 13 likely, 32 potential. **Decisions:** 0.

The table below shows the accessibility problems in your document sorted by accessibility guideline (sort by HTML group). Select any link in the table to view those problems or view all the problems.

All Problems By Accessibility Guideline

Guideline	Known	Likely	Potential
Priority 1	1	3	27
Priority 2	2	10	2
Priority 3	2	0	3

Figure 103



http://tjenester.nav.no/fodselspenger/fpenger/fp_begreper.do

Guideline: WCAG 1.0 (Level AAA) (change guideline)

Report: http://tjenester.nav.no/fodselspenger/fpenger/fp_begreper.do

Status: FAIL WCAG 1.0 (Level AAA) **Problems:** 6 known, 44 likely, 50 potential. **Decisions:** 0.

The table below shows the accessibility problems in your document sorted by accessibility guideline (sort by HTML group). Select any link in the table to view those problems or view all the problems.

All Problems By Accessibility Guideline

Guideline	Known	Likely	Potential
Priority 1	1	3	27
Priority 2	3	41	20
Priority 3	2	0	3

Figure 104

8.6.3 Summary

We tested all of the pages for the parental benefit calculator offered by NAV, and none of them in were in conformance to the accessibility guidelines. These test results were involved later in the design and the development phases of our new prototype. The aim of this was to achieve a new prototype without these accessibility problems.

8.7 Overall Summary

All the requirements were tested and validated by using the prototype developed. Every single requirement was successfully implemented. The requirements are presented in chapter 5.4 on page 48.
We have validated the web pages of the prototype in regards to W3C recommendations and all the web pages passed the validations.

With the user testing with real users we were able to identify new usability problems, and corrected these into the final version of the new prototype.

We have also performed heuristic evaluations of NAV’s current solution according to the 10 heuristics presented in chapter 4.1.1 on page 29.

The accessibility problems with the new prototype were detected by using the online tools, and corrected into the completed prototype. The tests were performed on all the web pages of the prototype, but not all of them are shown in this chapter because it would just cause duplicates of figures.

We used the accessibility checker tools to check NAV’s parental benefit calculator for accessibility problems as presented in chapter 8.6 on page 109. We also checked our new prototype with the same tools as presented in chapter 8.5 on page 102 and compared the results with each other. From these results we could determine that NAV’s parental benefit calculator had not been developed in conformance to accessibility guidelines. Since we used the accessibility checker tools to check our prototype and correct the problems, our new prototype was developed in regards to the accessibility guidelines.

9 Discussion

The result of this project is an accessible parental benefit calculator prototype with improved usability. This was achieved by the use of the methodologies presented in chapter 4 on page 28.
Our prototype can calculate the parental benefit for parents that have a part-time job during the parental leave. It is also able to calculate the last day for when parents must take out the parental benefit. It can also present the parental leave in a graphical calendar. These results are presented in chapter 7 on page 65. The advantages with these functionalities are that the prototype will provide more usability since parents can get more information from a calculator with such functionalities. The prototype is also supporting different languages and with this functionality we will also include people that don't read Norwegian so that they are able to use the prototype too. In addition, the prototype is also able to calculate the parental benefit for each week, month and year. Then, the parents won't have to calculate this by themselves.
If we were to do this again, we would implement an interactive calendar into the prototype so that the parents will be able to plan their parental leave by choosing different dates in the calendar.

It is important to notice that the research approaches we utilized and the results from this project can be utilized in other development projects. Our work was to show how a public service offered to the citizens could be more usable and accessible to each and every one. The research approaches presented in chapter 4 on page 28 made it possible for us to achieve the results. By following the research approaches utilized in this project and by using them correct, any developed service may become fully accessible by everybody and provide great usability.

The results from the analytical work, which are presented in chapter 5 on page 45, are the key part to the new design of the new prototype. The results from this work produced a new prototype with improved usability. The advantages with these results are that it became easier to design and develop the new prototype.

Paper-based prototyping was a fast and good technique for developing prototypes. Although it seemed to be unprofessional, the results from using these prototypes were quick feedback on the interface design and the system functionality. The advantage with this kind of prototyping is that people without computer skills can design prototypes. We did not have to implement anything, and this saved us a lot of time. (Nielsen J. , 1993) stated that this technique is very efficient since we can test early design ideas and fix usability problems before the implementation phase. We received feedback on the design from our supervisors and we managed to address the needed functionality by the use of this technique.
The paper prototypes were utilized when we designed and developed our new prototype, and this made it easier for us to develop it since we had a sketch of how the new prototype looked like. These can be seen in Appendix 11.6 on page 142.

We made use of structured interview when we interviewed the caseworkers at NAV. This technique was an efficient way to acquire information about the needed functionality for the new prototype. During this interview we also received feedback on our prototype, and this resulted in that we were able to verify that our prototype was correctly designed.

Heuristic evaluation of the parental benefit calculator was a good method in order to identify the usability problems with the interface design. As stated by (useit.com) it was a quick, cheap and easy evaluation method of a user interface design.
If we would have done this step over again, we would have used more evaluators to perform heuristic evaluation. This would have resulted in that more usability problems are detected and a new prototype with even less usability problems.

The user testing that was performed on the new developed prototype provided us very useful information. The test was performed on 5 real users. The users were able to identify usability problems that we didn't discover when we evaluated the new prototype. This method provided us with direct information about how the test users used the prototype and what their exact problems were with the user interface. This also agrees to what (Nielsen J. , 1993) has stated.
The results from these tests had a great positive impact on the final version of our new prototype. We were able to correct more usability problems, and if we were to do this again we would also have involved users with disabilities. The advantage with this is that we will get a better understanding of how these people use the web, and hopefully this will result in that we are able to develop a more usable prototype.

Beside usability problems, accessibility problems had to be taken into account. We made use of the web content accessibility guidelines. These accessibility guidelines were quite technical. Since we had managed to familiarize us with the required technologies in this project we were able to understand how to correct the accessibility problems. The parental benefit calculator offered by NAV was checked using accessibility checker tools. The results from these checks were kept in mind when we developed our new prototype. The advantages with these tools were that they saved us time, since they automatically checked the web pages of the prototype for accessibility problems. The result from the use of these checker tools, was that we were able to correct all the known problems detected by the accessibility tools and this resulted in an accessible prototype.

We developed the web pages with XHTML with embedded Ruby code. This was very efficient since we could get values directly from the Ruby controller from the HTML file. CSS was used to style the web pages and we chose CSS so that we could separate the HTML code and the code for styling the pages. The advantage with this is that it will be easier to maintain and further develop the web pages in the future which are important for an eGovernment service.
By doing this we were also able to check the web pages and the style sheets through quality assurance with W3C validation services.
MySQL was used to build up the database for the new prototype and HeidiSQL was used to gain control over the data in the database. Ruby on Rails supported MySQL, and therefore we chose to use this kind of database. The programming language Ruby was easy to learn. If we would have done this again, we would definitely have chosen the same technologies and tools again. These tools were quite efficient to use for developing. We learned these technologies and used them to achieve the solution in this project.

In this project we learned about how we could utilize existing methodologies and technologies to develop an accessible public service with high usability by making use of different approaches. By following these principles we were able to solve the research problem. If we were to start again we would have spent more time on investigating the accessibility guidelines, since we have learned that accessibility of a service is much more than just caring for people with disabilities to be able to access them. Accessibility is about making it easier for everybody to use and caring for what people need and want. We have also learned that usability is the next step after accessibility, and that step is to ensure that everybody can make use of the services in an easy and intuitive way.

10 Conclusion

The presented prototype in this project illustrates how an online eGovernment service may be improved to be more usable and accessible. Our prototype provides significantly greater functionality and interactivity. It is not limited to the parental leave duration calculation, which is the most trivial aspect of the parental leave planning, but supports also estimation of the financial support as well as simulation of various graded-withdrawal scenarios. A calculator developed based on the proposed prototype would therefore be likely to provide much more useful tool to the public than the previously available solution. As such, it may lead to less influx of people with inquiries about the parental benefit and leave to all local NAV offices.

Our prototype is also offering NAV a support for monitoring the use of the calculator. The developed database will store all data by collecting data from different users of the calculator. This allows NAV to analyze in detail how the online calculator is used.
For example, what is the average duration of the session, how frequently an individual is using the calculator, how many queries are investigated, etc. These data are valuable knowledge about how the calculator is performing and about how it can be further improved. All tools and technologies that are utilized in the development of the prototype are open and adhere to Open Standards, and they ensure that further development and maintenance of the calculator can be easily performed.

The prototype has been developed adhering to usability and accessibility guidelines. Usability testing has been performed with several different test users. The results from these have been used in the designing and development of the prototype. The prototype has been improved in several stages before it became the final version.
The known accessibility problems in the prototype, discovered by accessibility checkers have also been corrected in conformance to guidelines. The result of this is that the prototype will become fully accessible by everybody, even the people with disabilities.
The web content accessibility guidelines 1.0 has been followed during the accessibility checking of the prototype since this version was supported by the tools we used and is the stable and reference able version that was approved in May 1999 according to (W3 - WAI).
The tools we used to check our prototype for accessibility problems were: A-Checker, A-Prompt and WebXACT. The parental benefit calculator offered by NAV was also checked for accessibility problems. These results are presented in chapter 8.6 on page 109. By comparing the results between their solution and our prototype, we conclude that our prototype has been improved both in usability and accessibility.

The goals of eGovernment are to provide better, more effective services. If the new services are not fully accessible, part of our societies will be excluded and become increasingly isolated. If they do not address the actual needs of the public, they will not be used. By following the principles presented in this project, a developer may be able to develop both accessible and usable services in the future. We believe that only eGovernment services that are fully accessible and highly usable will ensure that the goals of truly inclusive Information Society for all are met[44].

[44] http://ec.europa.eu/information_society/policy/accessibility/eincl/index_en.htm

10.1 Research Limitations and Future Work

Although our prototype indeed is providing enhanced functionality and improved usability, it can be further improved in the future.
If the prototype is approved by NAV and taken in use by them, the administration part of the prototype can be improved to support statistics compilation.
This functionality will aid the administrator of the prototype to get a quick overview of the prototype. For example, by compiling statistics of the use of the prototype and present it with graphs and percentages.

The prototype can also be improved to have a login system for the users. There are many advantages with such a login system. For example, the users can log into the system before performing calculations of the parental benefit and leave. Once they have performed it, and they use the calculator later again, they won't have to input all the data over again. The calculator can store the user inputs and connect it to specific users by giving each calculation a specific reference number.
Another advantage with this could be that the users can contact their local NAV offices by only pressing a button on the web site when they have performed a calculation and need further assistance with the parental benefit and leave. The calculator will make use of the stored data about the users, and automatically find the local NAV office for the users, for example by using the postal code. All the user data, the user inputs, and a short description of the needed assistance will be sent directly to their local NAV office. The reference number can then be used by NAV to sort out and process inquiries, but it can also be used by the users for when they are contacting NAV for assistance. In this way NAV can become more efficient with responding to inquiries.

The prototype could also be improved so that it is offered as 2 different versions:

- An advanced version which required special input data from the users and gave detailed information and answers on the parental benefit and leave.

- A basic version which is easy to use and gives simple answers on the parental benefit and leave

In the advanced version of the prototype, the prototype could be improved so that it also supported an interactive scheduler for the whole parental leave. The scheduler could be a calendar displayed on the screen to the users. The users will then be able to interact with the calendar to plan their parental leave and take a printout for example.

The accessibility guidelines are true helpful when developing an accessible service. As a future work, all the likely and all the potential accessibility problems can be corrected so that the prototype doesn't contain any single accessibility problem. The newer versions can also be used for checking accessibility problems with the prototype when those versions become approved by the W3C in the future.
More usability testing can be performed on the prototype to ensure that all usability problems can be detected and removed. Usability testing questionnaires that are developed and reviewed can be used for the usability testing.

11 Appendices

11.1 Project Management

The project title of this project is: “Accessible eGovernment services – Developing a parental benefit calculator with improved usability”. This project was carried out by one person, Peter Le, and was given by NAV Trygd Grimstad in cooperation with Agder University College.

The project was sat to be started on Monday 8. January 2007 at 08:00 hours. The work was taking place in Grimstad and the delivery date for this project was sat to be Tuesday 29. May 2007 within 15:30 hours.

During the project period there were several dates that were important. The table below presents these dates and the tasks that had to be done.

Date - hours:	Tasks:
08.01.2007 – 08:00	Start of master project
05.02.2007 – 15:00	Master’s thesis freeze of title and definition
18.05.2007 – 15:00	Deliver first “full” version of master thesis to supervisors
29.05.2007 – 15:30	Delivery of master thesis report
31.05.2007 – 15:00	Reserve room for producing a multimedia presentation
02.06.2007 - 15:00	Delivery of poster to be presented at poster session in the foyer
08.06.2007 – 12:00	Place poster on the web
09.06.2007 – 12:00	Place master thesis report on the web
06.06.2007 – 15:00	Produce multimedia presentation of 12-15 minutes
07.06.2007 – 12:00	Delivery of PowerPoint presentation
13.06.2007 – 09:00	Plenary presentation of master project

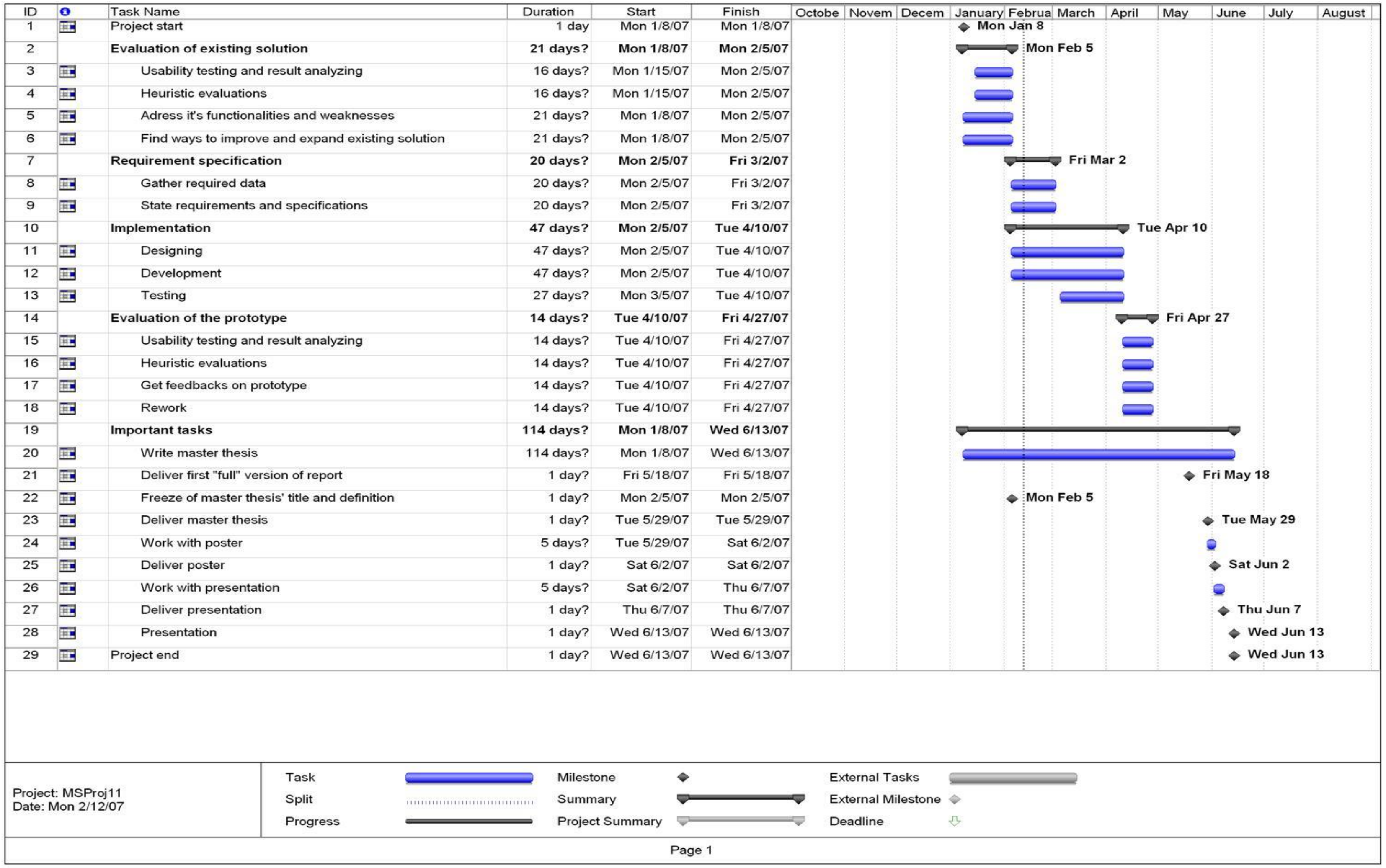
Table 21: Important dates and tasks

The working hours for this project was every weekdays, besides the weekends and the public holidays. These holidays would become working hours in case of emergency. This means that we had to work on these days if unforeseen problems had arisen and the completion of my work was in danger. The dates for the public holidays are presented in the table below.

Date	Public holiday
05.04.2007	Maundy Thursday
06.04.2007	Good Friday
09.04.2007	Easter Monday
01.05.2007	Public festival
17.05.2007	Ascension Day & Constitution Day

Table 22: Public holidays

Since this project was carried out by only one person, we had work with the thesis in parallel with the other tasks in this project. These tasks are presented in Figure 105 and this was how we organised my work in this project.
We managed this project by using the principles and techniques described at (Home of Principle Based Project Management).



ID		Task Name	Duration	Start	Finish
1	▦	Project start	1 day	Mon 1/8/07	Mon 1/8/07
2		**Evaluation of existing solution**	**21 days?**	**Mon 1/8/07**	**Mon 2/5/07**
3	▦	Usability testing and result analyzing	16 days?	Mon 1/15/07	Mon 2/5/07
4	▦	Heuristic evaluations	16 days?	Mon 1/15/07	Mon 2/5/07
5	▦	Adress it's functionalities and weaknesses	21 days?	Mon 1/8/07	Mon 2/5/07
6	▦	Find ways to improve and expand existing solution	21 days?	Mon 1/8/07	Mon 2/5/07
7		**Requirement specification**	**20 days?**	**Mon 2/5/07**	**Fri 3/2/07**
8	▦	Gather required data	20 days?	Mon 2/5/07	Fri 3/2/07
9	▦	State requirements and specifications	20 days?	Mon 2/5/07	Fri 3/2/07
10		**Implementation**	**47 days?**	**Mon 2/5/07**	**Tue 4/10/07**
11	▦	Designing	47 days?	Mon 2/5/07	Tue 4/10/07
12	▦	Development	47 days?	Mon 2/5/07	Tue 4/10/07
13	▦	Testing	27 days?	Mon 3/5/07	Tue 4/10/07
14		**Evaluation of the prototype**	**14 days?**	**Tue 4/10/07**	**Fri 4/27/07**
15	▦	Usability testing and result analyzing	14 days?	Tue 4/10/07	Fri 4/27/07
16	▦	Heuristic evaluations	14 days?	Tue 4/10/07	Fri 4/27/07
17	▦	Get feedbacks on prototype	14 days?	Tue 4/10/07	Fri 4/27/07
18	▦	Rework	14 days?	Tue 4/10/07	Fri 4/27/07
19		**Important tasks**	**114 days?**	**Mon 1/8/07**	**Wed 6/13/07**
20	▦	Write master thesis	114 days?	Mon 1/8/07	Wed 6/13/07
21	▦	Deliver first "full" version of report	1 day?	Fri 5/18/07	Fri 5/18/07
22	▦	Freeze of master thesis' title and definition	1 day?	Mon 2/5/07	Mon 2/5/07
23	▦	Deliver master thesis	1 day?	Tue 5/29/07	Tue 5/29/07
24	▦	Work with poster	5 days?	Tue 5/29/07	Sat 6/2/07
25	▦	Deliver poster	1 day?	Sat 6/2/07	Sat 6/2/07
26	▦	Work with presentation	5 days?	Sat 6/2/07	Thu 6/7/07
27	▦	Deliver presentation	1 day?	Thu 6/7/07	Thu 6/7/07
28	▦	Presentation	1 day?	Wed 6/13/07	Wed 6/13/07
29	▦	Project end	1 day?	Wed 6/13/07	Wed 6/13/07

Project: MSProj11
Date: Mon 2/12/07

Page 1

Figure 105: The project plan

11.2 Interview: Initial evaluation of the current parental benefit calculator

1. Er det lett å finne fram til kalkulatoren fra forsiden til NAV ?

Veldig greit. Pga tidligere besøk på siden...den nye siden er organisert på samme måten som den gamle siden. 2 museklikk fra forsiden til kalkulatoren.

2. Den informasjonen som er presentert, er den lettleselig og forståelig?

Ja. Litt mye opplysningner på en side. Kanskje en liste med ting man vil se på . (listbox)
Begreper er greit.

3. Hvordan er designen av selve siden?

Fint med tabs øverst. Kanskje ikke så lett å se at det er tabs. Feil fargekontrast. Kalkulator på første tabs.
2 setninger med innledning før kalkulasjonen.

4. Var hjelpetekstene under beregningen forklarende ?

Hjelpespørsmål veldig greie.
Forvirrende hjelpespørsmål når det gjelder det andre hjelpespørsmålet. "har mor abeidet i minst 50% stilling før fødsel?"
Første hjelpespørsmål: 5 uker med 80% dekning, 7 uker med 100% dekning. Bør stå oppgitt.

5. Går det raskt å navigere seg gjennom beregningen?

Ja.

6. Var det noe som var uforståelig ved bruk av kalkulatoren?

Fast / korte arbeidsforhold...
Rotete presentasjon av oppgitt informasjon etter endt beregning.
Dårlig framstilling av resutatet. Får man like mye ?
Hva menes med grunnbeløp ? Årsbeløp eller månedsbeløp?
Meldingen som kommer opp er uforståelig.
Mor inntekt : 10000 , får beskjed
Mor inntekt: 20000, får ikke beskjed
Presentere hva man får og mister.
Presentasjon i mnd eller uke.

7. Har du prøvd å bruke kalkulatoren i andre nettlesere? Hvis ja, var det noe forskjell ?

Ingen forskjell.

8. Hva likte du med siden ? og hva likte du ikke ?

Presentere hvilke dager man kan ta fødselsperioden og ikke bare tall.
Forventer å få grafisk framstilling av ting.
Når man krysser av noe, forventer man noe i tillegg. Utvidelse ....
Design stort sett ok.
Presentasjon av resultat er uforståelig. Hvordan stønadspengene er beregnet
Framstilling av stønadsperiode

Table 23: Evaluation interview with our supervisor

11.3 Interview: NAV Trygd Grimstad

Tidligere løsning

Fødselspermisjon:

1) Hvilke er de 3-4 oftest stilte spørsmålene angående fødselspermisjonen?

De mest stilte spørsmålene er når foreldre kan ta permisjon og hvor lenge kan de ta den, og om fedrekvoten som når far kan begynne fødselspermisjonen.

   a) Hvorfor tror dere at de er så ofte stilt?

Folk vil benytte seg av sine rettigheter og vil vite hva de har krav på. Det er økonomiske årsaker som ligger i grunnen.

      i) Hva kan være grunnen til at fødselspermisjonen er vanskelig å forstå/finne/beregne?

Beregningen av fødselspenger er komplisert. Det er mye nødvendig informasjon som trenges, og man må sette seg inn i beregningsgrunnlaget.

2) Hva må dere gjøre for å kunne besvare disse henvendelsene?
   a) Hva må dere gjøre for å håntere disse henvendelsene?

NAV avtaler et møte og setter av en time til de som tar kontakt.

   b) Hva kan være til hjelp for foreldre som tar kontakt?

De kan bli henvist til informasjonen som ligger på nettet.

   c) Hvor lang tid tar det som regel å behandle disse henvendelsene?
      i) Tiden en saksbehandler bruker?
      ii) Tiden som foreløper fra at en henvendelse har kommet inn til den er besvart?

Vanlige spørsmål om fødselspermisjonen tar cirka 15 til 30 minutter.
   d) Kunne kalkulatoren besvart noen av de oftest spurte spørsmålene?

Ja, det gjør den. Men ikke alle bruker nettet så de ringer inn til NAV istedet. Det er oftest nyutdannende eller personer i helsevesenet, som sykepleiere, som tar kontakt når det gjelder avvik i beregningen. Dette er fordi de typisk får et stort avvik mellom de 3 års siste pensjons givende inntekt og nåværende inntekt.

   e) Hvorfor tror dere at kalkulatoren ikke ble brukt til å besvare disse henvendelsene?

Det er ikke mange som lurer på det, og dessuten var kalkulatoren bare ment som en liten hjelp i tillegg til all informasjonen om fødselspenger som ligger på nettet.

3) Har dere statistikk over hvor mange henvendelser dere har fått i form av besøk, over telefon

eller mail angående fødselspermisjonen?

Fører ingen statistikk, men det er mest henvdelser over telefon. Det er cirka 40-50 henvendelser per måned.

4) For de tilfeller når foreldre har tatt kontakt flere ganger :
   a) Hva kunne man ha gjort slik at ting kunne blitt klarlagt ved første besøk?

At folk har satt seg mer inn i regelverket og lest den informasjonen som er tilgjengelig på nettet.

   b) hva var problemet?

Beregningsgrunnlag og datoer.

Gradert uttak:
5) Hvilke er de 3-4 oftest stilte spørsmålene angående gradert uttak?

De oftest stilte spørsmålene ligger på NAV sine hjemmesider. Man kan finne dem ved å navigere seg til NAV -> Familie og Omsorg -> Nye regler – ofte stilte spørsmål

   a) Hvorfor tror dere at de er så ofte stilt?

Folk har ikke hørt så mye om det, og derfor har det også vært lite snakk om det imellom blivende mødre.

      i) Hva kan være grunnen til at gradert uttak er vanskelig å forstå/finne/beregne?

Man må kjenne til beregningsgrunnlagene. For å kunne finne ut det eksakte beløpet må man kunne reglene.

6) Hva må dere gjøre for å kunne besvare disse henvendelsene?
   a) Hva må dere gjøre for å håntere disse henvendelsene?

NAV avtaler et møte og setter av en time til de som tar kontakt.

   b) Hva kan være til hjelp for foreldre som tar kontakt?

De kan bli henvist til informasjonen som ligger på nettet.

   c) Hvor lang tid tar det som regel å behandle disse henvendelsene?
      i) Tiden en saksbehandler bruker?
      ii) Tiden som foreløper fra at en henvendelse har kommet inn til den er besvart?

Cirka. 15 til 30 minutter pr. Henvendelse.

d) Kunne kalkulatoren besvart noen av de oftest spurte spørsmålene?

Ja, kalkulatoren kunne ha besvart de oftest spurte spørsmålene, hvis flere hadde tatt i bruk nettet for å skaffe seg informasjon først. Men som regel så ringer folk inn og spør om det de lurer på istedenfor å lese all informasjonen på nettet.

e) Hvorfor tror dere at kalkulatoren ikke ble brukt til å besvare disse henvendelsene?

Tidligere ordning het tidskonto. Men veldig få som benyttet seg av det. De fleste ønsker å være hjemme med barnet sitt etter fødsel. Gradert uttak er nytt fom 1.1.2007.

7) Har dere statistikk over hvor mange henvendelser dere har fått i form av besøk, over telefon eller mail angående gradert uttak?

Fører ingen statistikk, men det er mest henvdendelser over telefon. Veldig få som har tatt kontakt angående gradert uttak, fordi det er en ny ordning.

8) For de tilfeller når foreldre har tatt kontakt flere ganger :
   a) Hva kunne man ha gjort slik at ting kunne blitt klarlagt ved første besøk?
   b) hva var problemet?

Ingen ennå.
Kalkulator:
9) Har du brukt kalkulatoren selv?

Nei

10) Hva er dine egne erfaringer med den?

Ingen

11) Fører dere statistikk over hvor mange som har benyttet seg av kalkulatoren?
   a) Hvor effektiv er kalkulatoren når det gjelder å lette arbeidet deres?

Det hjelper nok litt, i og med at det er folk som kan data som bruker kalkulatoren så har de også lest informasjonen som ligger tilgjengelig på nettet.

b) Hva trur dere kan hjelpe til å gjøre den enda mer effektiv?

At den blir hørt om av flere.

12) Har dere fått tilbakemeldinger på kalkulatoren fra brukere? Hvis ja, hva slags?
   a) Tilbakemelding på design?
   b) Tilbakemelding på funksjonalitet?
   c) Hva var dårlig, hva var bra?
   d) Hva er de tilbakemeldingene som dere mottar oftest?

Ingen!

Ny løsning

13) Kan flere deler av arbeidsoppgavene for å svare på henvendelsene angående fødselspermisjon og gradert uttak bli automatisert ved hjelp av en ny type online kalkulator?
    a) Hvordan?

Det kan det, men da må brukere sette seg inn i reglene og forstå beregningsgrunnlaget. Kalkulatoren kan be om flere opplysninger fra brukere, men flere av de opplysningene de må oppgi krever at de tar kontakt med NAV for å få vite.

14) Tror dere at det vil effektivisere arbeidet deres hvis den nye kalkulatoren klarer å besvare henvendelser angående gradert uttak?

Ja, det vil det. Hivs flere bruker internett og tar i bruk kalkluatoren, så vil de mest sannsynlig snakke sammen, og da vil de finne ut av ting selv sammen og mange spørsmål vil de få svar på av hverandre.

15) Ser dere noen måter å forbedre kalkulatoren på, slik at de ofte stilte spørsmålene om fødselspermisjon og gradert uttak kan besvares av kalkuatoren?
    a. Hvilke måter?

Den kan for eksempel vise bruker hvor lenge en kan ta fødselspermisjonen når man ønsker å forskyve. Få fram forskjellene mellom tidskonto og gradert uttak. Mor eller far trenger ikke å velge minst 50% deltidsjobbing lenger. 14 dager er minst nå med gradert uttak.

16) Har dere noen tanker på hva som bør støttes av en ny type kalkulator for å gjøre det lettere for:
    a) NAV?

At brukere vil få svar på alle de kompliserte variantene, vil nok ikke la seg løse. Men hvis kalkulatoren tilbyr generell informasjon, og hvis den gir svar på grunnlag, prosenter og delvis uttak så vil det hjelpe NAV.

    b. Foreldre?

Kalkulatoren må være veldig ryddig og lett betjenlig. Det de fleste foreldre har misforstått er tidspunkter. Hvor lenge kan man få gradert uttak. Få fram at begge kan ta gradert uttak samtidig, men at den sammenlagte prosenten av jobbingen er 100% for begge. Vise at maksdatoen for uttak av foreldrepenger forskyves når mor eller far velger å ta gradert uttak.

17) Hva er de nødvendige opplysningene som trengs å innhente fra foreldre for å kunne beregne fødselspermisjonen med gradert uttak?

Se utdelt dokument . REG GRADERING etter 1/1 2007.

18) Har du eller noen her på kontoret vært med på å spesifisere den nye løsningen eller kjenner til den?

Nei

a) Hva skal være nytt?
b) Hva skal være forbedret?

Det som er annerledes er de nye reglene. Tidskonto opphører fra 1/1-2007.

Skjønnsmessige grunner:
Gjelder mor: Kalkulatoren har beregnet et avvik på over 25% mellom din nåværende inntekt og din gjennomsnittlige pensjonsgivende inntekt siste tre år. Dette kan ha betydning for hva du faktisk kan komme til å få i fødselspenger. Kalkulatorens beregninger kan være feil fordi viktige skjønnsmessige vurderinger ikke ivaretas av kalkulatoren.

19) Hva er disse "viktige skjønssmessige vurderingene" i teksten ovenfor ?

NAV må finne ut hvorfor det er så stort avvik i de tallene brukeren har oppgitt.

20) Hvor stort utslag kan de skjønnsmessige vurderingene gi for resultatet av beregningen?

Ganske store. NAV vil helst ikke at kalkulatoren oppgir for mye fødselspenger før de har manuelt beregnet ut hvor mye en foreldre får.

21) Har du noen eksempler for skjønnsmessige vurderinger?

Det er mest folk i helsevesenet og nyutdannende som får dette avviket. Sykepleiere har en fastlønn og i tillegg jobber de mye ekstravakter. Det er ikke alle nyutdannende som kommer i arbeid med en gang og da kan de for eksempel tjene:
** 2004: 40000kr*
** 2005: 250000kr*
** 2006: 270000kr*

Og har en månedslønn på 30000kr i måneden nå. Da oppstår 25% avviket.

22) Hvorfor ivaretas ikke dette av tidligere løsning av kalkulatoren?

Fordi NAV må finne ut av hvorfor avviket har oppstått for hver enkelt person som kommer bort i dette tilfellet.

23) Skal disse ivaretas av den nye løsningen som dere holder på å jobbe med?

Nei. Det trur de nok ikke. Det kan fort bli for komplisert at folk ikke bruker kalkulatoren i det hele tatt.

24) Vi vil gjerne ivareta disse vurderingene i den nye løsningen av kalkulatoren. Lar dette seg gjøre?

Da må man vite reglene veldig eksakt. Det kan kanskje lages 2 alternativer av kalkulatoren. 1 for de vanlige opplysningene og den andre for spesielle opplysninger. Man må også ha tilgang til opplysninger som kun NAV har og kan få tak i.

25) Kjenner du til andre tjenester som NAV driver og som kunne egne seg for tilsvarende type implementasjon?

Sykepenger for eksempel. Alt kan brukes for beregning av en kalkulator.

26) Kan du nevne noen fordeler for NAV ved å tilby foreldre en ny type kalkulator som hjelper dem med å få svar på fødselspermisjon med gradert uttak?

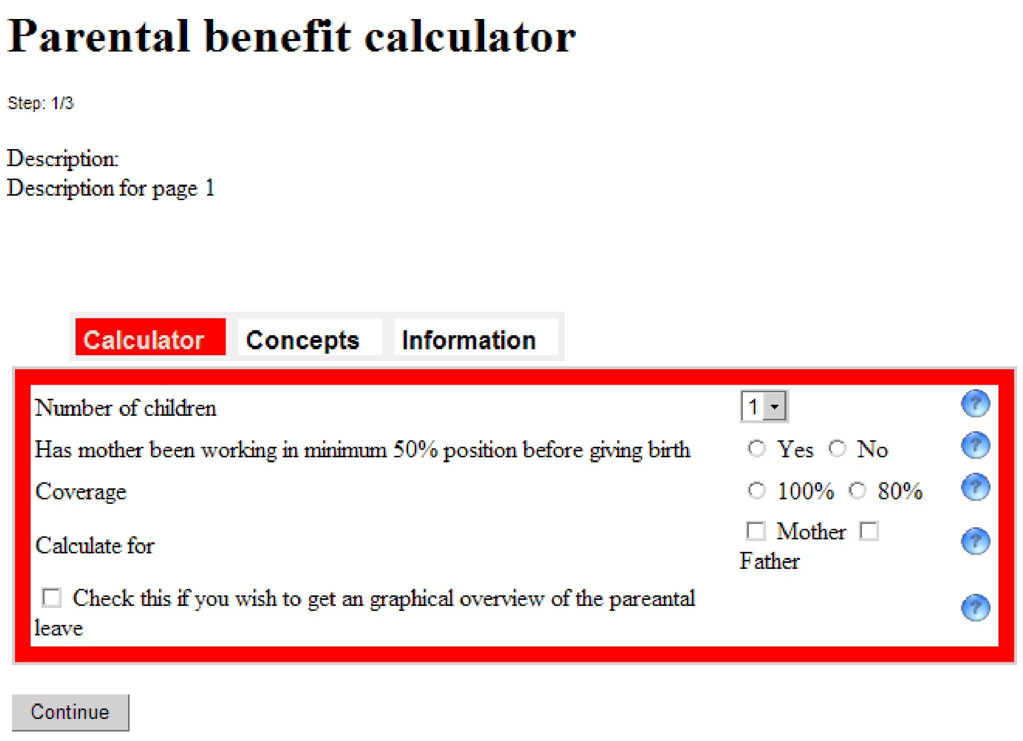
Det er en inngangsport for folk til å få vite hva man kan få, Men når det blir for komplisert så gir de opp. Og da tar de kontakt med NAV istedet for å prøve å bruke det. Noen opplysninger har folk tilgjengelig, mens andre opplysninger som kreves for å fullføre en hel beregning må uansett innhentes fra NAV.

27) Finnes det andre ting som kan knyttes til fødselspengekalkulatoren som en tilleggstjeneste som kan tilbys foreldre?

Det er det, men man må passe på at ting ikke blir for komplisert. De som bruker en kalkulator på nett, har som regel satt seg litt inn i informasjonen som er tilgjengelig på nettet og vil se fort over hvor mye de kan forvente å få utbetalt og krysssjekke at forståelsen er riktig.

Få tilbakmelding på av NAV: (hvis tid til overs)

Merk: Tar ingen hensyn til de forskjellige scenarioene. Demonstrerer kun 1 scenario hvor alle de mulige dynamiske elementene er vist.
Steg 1 av 3:



Parental benefit calculator

Step: 1/3

Description:
Description for page 1

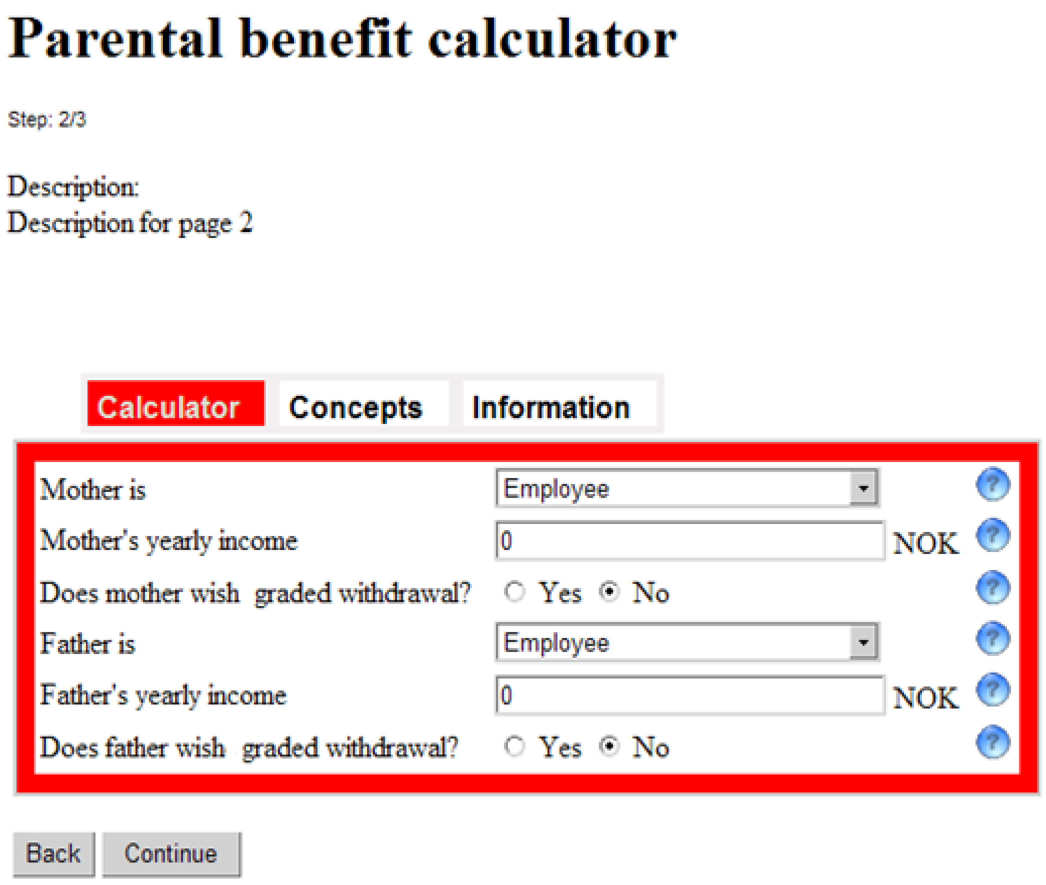
Calculator | Concepts | Information

Number of children	1
Has mother been working in minimum 50% position before giving birth	○ Yes ○ No
Coverage	○ 100% ○ 80%
Calculate for	☐ Mother ☐ Father
☐ Check this if you wish to get an graphical overview of the pareantal leave	

Continue

Jeg har valgt å beholde de samme spørsmålene som tidligere løsning. Det siste valget gir brukeren muligheten til å få en grafisk oversikt over permisjonsperioden i Steg 3.

Steg 2 av 3:



Parental benefit calculator

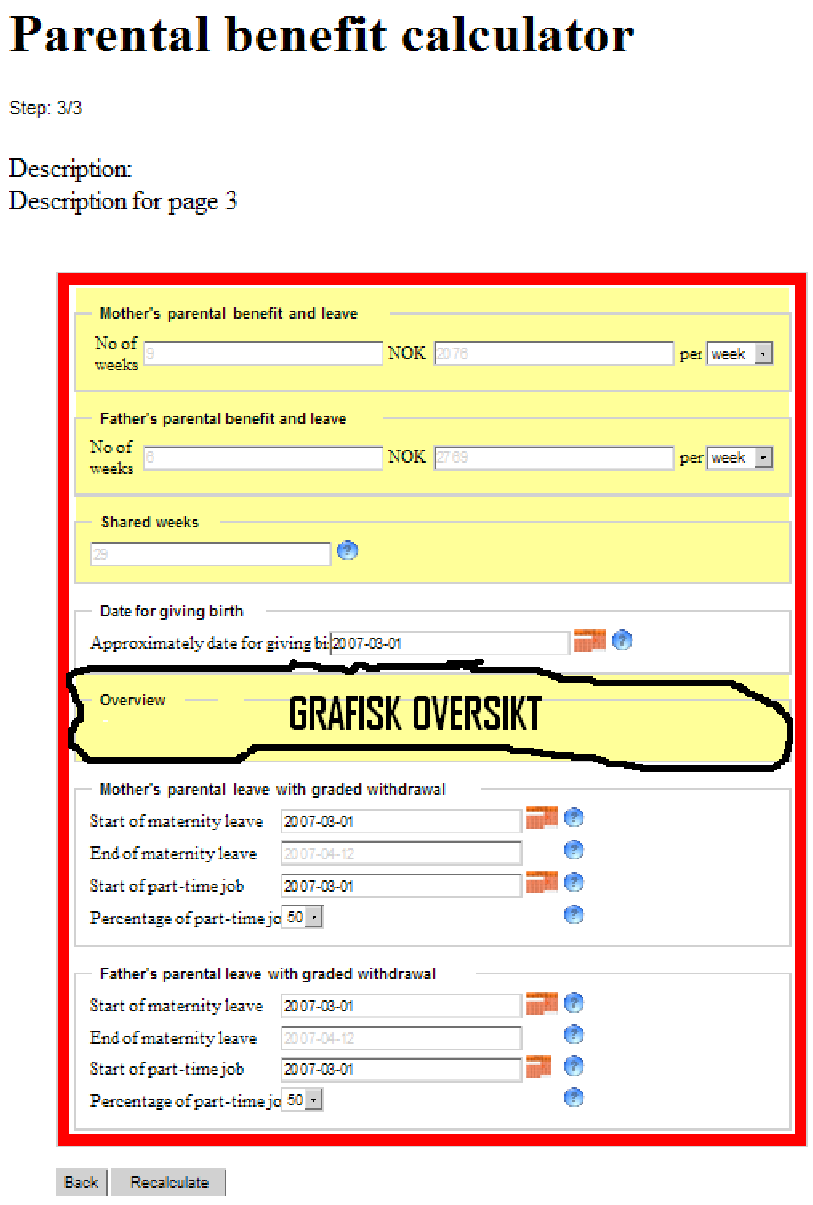
Step: 2/3

Description:
Description for page 2

Mother is	Employee	
Mother's yearly income	0	NOK
Does mother wish graded withdrawal?	Yes [✓ No]	
Father is	Employee	
Father's yearly income	0	NOK
Does father wish graded withdrawal?	Yes [✓ No]	

Her kan bruker bestemme om far eller mor ønsker gradert uttak av fødselspengene(tidskonto ordning). Er svaret ja, blir bruker bedt om å fylle inn flere opplysninger i Steg 3.

Steg 3 av 3:



Parental benefit calculator

Step: 3/3

Description:
Description for page 3

Mother's parental benefit and leave

No of weeks	9	NOK	2076	per	week

Father's parental benefit and leave

No of weeks	8	NOK	2769	per	week

Shared weeks

29

Date for giving birth

Approximately date for giving bi 2007-03-01

Overview

GRAFISK OVERSIKT

Mother's parental leave with graded withdrawal

Start of maternity leave	2007-03-01
End of maternity leave	2007-04-12
Start of part-time job	2007-03-01
Percentage of part-time jo	50

Father's parental leave with graded withdrawal

Start of maternity leave	2007-03-01
End of maternity leave	2007-04-12
Start of part-time job	2007-03-01
Percentage of part-time jo	50

Her får bruker presentert resultatene.
Hvis bruker valgte gradert uttak i Steg 2 må brukeren velge ønsket prosent.
Hvis bruker valgte grafisk oversikt i Steg 1 må bruker taste inn dato for fødselen av barnet, startdato for permisjonen. Og startdato for deltidsjobbingen hvis og bare hvis bruker valgte gradert uttak i tillegg.

Grafisk oversikt: (som popup vindu ? => bedre plass og oversikt)
Presentere en tabell med flere rader og kolonner.
Rad : Datoer med 1 ukers intervall for hver kolonne
Rad: Hjemmeværende
Rad: I arbeid
Merk datoen for fødsel av barnet(som bruker har angitt) med en farge (rød?). Og sett merk av i 2. Rad at mor skal være hjemme 3 uker før fødselen og 6 uker etter fødselen. (Ifølge reglene).
Har bruker valgt gradert uttak vil 3. Rad være synlig og her kan bruker velge de forskjellige ukene å være hjemmeværende på og i arbeid på ved å klikke på de ulike tabellrutene. Disse ukene vil bli markert med prosentdelen(prosentandel av deltidsjobbingen som bruker har angitt).

_Eksempel 1:

Week no	9	10	11	12	13	14	15	
	01.03.07	08.03.07	15.03.07	22.03.07	29.03.07	05.04.07	12.04.07	....
Hjemmeværende	X	X	X	X	X	X	X	
I arbeid	0	0	0		0	0	0	

Week no	16	17	18	19	20	21	22	
	19.04.07	26.04.07	03.05.07	10.05.07	17.05.07	24.05.07	31.05.07	....
Hjemmeværende	X	X	50%	50%	🟥	50%	50%	
I arbeid	0	0	50%	50%		50%	50%	

_Eksempel 2:

MARS							
	Ma	Ti	On	To	Fr	Lø	Sø
9	26	27	28	1	2	3	4
10	5	6	7	8	9	10	11
11	12	13	14	15	16	17	18
12	19	20	21	22	23	24	25
13	26	27	28	29	30	31	1

APRIL							
	Ma	Ti	On	To	Fr	Lø	Sø
13	26	27	28	29	30	31	1
14	2	3	4	5	6	7	8
15	9	10	11	12	13	14	15
16	16	17	18	19	20	21	22
17	23	24	25	26	27	28	29
18	30	1	2	3	4	5	6

MAI							
	Ma	Ti	On	To	Fr	Lø	Sø
18	30	1	2	3	4	5	6
19	7	8	9	10	11	12	13
20	14	15	16	17	18	19	20
21	21	22	23	24	25	26	27
22	28	29	30	31	1	2	3

Her vil de obligatoriske permisjonsukene bli markert med en farge(rød?). De ukene brukeren velger å jobbe på vil bli markert med en farge(grønn?).

Ønsker å få tilbakemelding på prototypen etter visning og forklaring av skjermbildene, slik at den kan bli forbedret.
Den uka mor føder på er også obligatorisk permisjonsuke. En slik grafisk oversikt kan være nyttig for å øke forståelsen hos foreldre.

Table 24: Interview with Nav Trygd Grimstad

11.4 Meeting minutes

First meeting			
30.11.2006		09:00	Vinkelbygget
Meeting called by	Agata Sawicka		
Attendees	Mikael Snaprud, Agata Sawicka, Peter Le		
Issues	This was the first meeting, therefore no issues this time.		
Actions	i. Read about Parental benefit and time account. ii. Study the present solution of the calculator at NAV's website and find out how the calculations have been done. iii. Prepare questions about the calculator for next meeting. iv. Prepare an evaluation interview.		
Decisions	• Thursday at 9 o'clock is the tentative meeting time. • Meetings should only be held if there are any issues, but in the beginning it would be nice to have meetings frequently. • Prepare agenda before every meeting. • Write meeting minutes after every meeting.		
Discussion	We discussed why I should prepare some questions about the calculator to the next meeting, like an interview, instead of getting all the information right away. We also discussed why the supervision should be commented if necessary, and talked a bit about why working with the project on full-time can't be started yet, and a bit about the literature review in the problem proposal.		

Table 25: First meeting

Second meeting		
07.12.2006	09:00 – 10:50	Vinkelbygget
Meeting called by	Scheduled	
Agenda		
09:00 – 10:30		
Attendees	Agata, Peter	
Perform an evaluation interview with Agata		
Discussion	We talked about the parental benefit calculator. We discussed the information presented on the web site, and about how the calculator can be improved. We discussed how the design could be different, how the information presented could be improved to be more clearly, how the results could be presented differently and we also discussed a bit about how the prototype should be. We also looked at some spreadsheets, which Agata had developed, for calculating some different time account scenarios.	
10:30 - 10:50		
Attendees	Agata, Peter	
Discuss the next steps to take Present what I have planned to do the 2 next weeks		
Discussion	Agata showed examples of how to use cross references and Endnote. We discussed the styles, cross reference function and automatic table of contents in Microsoft Word and how important it is to make use of these functions. A little discussion about which text editor I am familiar with. This is Microsoft Word We discussed what to do in the next meeting. This was about deciding which text editor tool to use to write documents. By doing this the supervisors can more easily review and give comments on the documents handed in. We also discussed that I should write down a date for when I need a reply on my requests.	
Issues	• I wonder if I can get some literature to work with my full proposal (within 15.12.2006) • I just have some general questions about the parental benefit	
Actions	• Redo the document structure of the minutes, and add discussion area in the agenda • Create a list with questions about the calculations of parental benefit, for sending to NAV Trygd Grimstad • Contact PEP regarding the license for Endnote	
Decisions	• Write down a date for when I need reply on my request, to mark that the subject is urgent • Take a look at Open Office • Work with the questions I have about the calculations of the parental benefit, for sending in to NAV Trygd Grimstad	

Table 26: Second meeting

Third meeting		
18.12.2006	09:00	Vinkelbygget
Meeting called by	Scheduled	
Agenda		
09:00 – 10:00		
Attendees	Agata, Peter	
Discuss about which text editor to use when writings documents Other things you want to discuss		
Issues	No particular issues right now. Currently, I am working with the full proposal, which I plan to hand in to PEP for a review on Monday 18.12.2006.	
Actions	Rework of the proposal: • Create a GANTT chart. • Rearrange and redo some of the sub problems. • Insert background literature.	
Decisions	The work will continue on Monday 8. January 2007. Take a closer look at the literature in the beginning of January, especially eGovernment. Learn a bit about Python and MySQL in the beginning of January if time allows.	
Discussion	We didn’t discuss which text editor that must be used, since Mikael was away, but we did talked about a text editor, and came to conclusion that I felt most comfortable with using Microsoft Word. We also discussed the proposal I am working with. I received help on the chapters in the proposal which I had problems with, and received a book about eGovernment, a book about usability and a book about prototyping. We discussed these things at the meeting.	

Table 27: Third meeting

Fourth meeting		
11.01.2007	14:00	Vinkelbygget
Meeting called by	Peter	
Agenda		
14:00 – 14:45		
Attendees	Mikael, Agata, Peter	
I would like some more information about the work that has to be done from now on and until the date for freezing the title and definition which is the 5. February 2007. What I intend to is to find out how the different chapters in my thesis report shall look like, and find out what to write in it as soon as possible. I received this information from Stein Bergsmark on Monday 08.01.07 15:15. I may need some help and advices here, because I am not so sure what to include here. I will be happily to receive all additional information from you.		
Issues	I have taken a look at www.nav.no, and it seems that they have updated the web site. I am not able to locate the parental benefit calculator anymore. Shall I follow the plan that I created and presented in my full proposal in IKT505? Some comments on the plan at the meeting would be nice. • If I follow the description I wrote in the proposal, then I can't perform evaluations of the existing solution with several test users because I can't locate the calculator at NAV's web site • I have experimented a bit with Python and MySQL. What technology should I use to design the web page, and is it possible to obtain the database from NAV? • Can I use Microsoft Word to write and edit my master thesis report?	
Actions	Redo the project plan so that I use an iterative model instead. Look into the background literature and begin to make an outline for the literature review. Take a look at Ruby On Rails. Supervisors shall contact NAV and ask for the source code for the old solution and get the specifications of it, and clarify the project with NAV. Supervisors shall send me a template for the master thesis.	
Decisions	Using Ruby On Rails for developing the prototype is a good choice. The thesis can be written using Microsoft Word. If NAV doesn't respond in time about the old solution, I shall design the prototype based on the initial evaluation of the old solution performed with Agata. Wait with the change of the project title until the deadline for freezing of title and definition is approaching.	
Discussion	We discussed a bit about the significance of the project and the title of the project. We also discussed the project plan, the structure of the thesis and the thesis' contents. All issues that needs an answer must be included in the agenda was brought up again. This is extremely important, because the supervisors will then have enough time to prepare themselves to give me advices.	

Table 28: Fourth meeting

Fifth meeting		
19.01.2007	09:00	Vinkelbygget
Meeting called by	Agata	
Agenda		
09:00 – 11:00		
Attendees	Agata, Peter	
• Status about making contact with NAV regarding the calculator, the source code for it and the database contents? • Discuss the draft of the thesis' structure and the outline for each bullet in the TOC. • Take a closer look and discuss the writing process in this project. • Work together with the thesis title and definition on a day close to the 5. February. My suggestion is at a meeting on Thursday 01.02.2007.		
Issues	• I have difficulties with writing the literature review. I thought I understood what to do at our previous meeting, but I was wrong. I am not so sure what to write in the literature review and neither how to create the outline for it. • Will it be possible for us to obtain the source code and the database contents within January? • I have a clear view of the development part for this project, but the research part is still very unclear.	
Actions	• Find guides and examples for writing thesis and literature review • Work with the outline for chapters: Research problem and Approach • Do a literature review for chapter: Background/Literature review • Start looking at methods and tools.	
Decisions	• Use the template handed out by Agata for the document structure of the thesis	
Discussion	• Mostly about the outline for the thesis, its contents and the writing process • Discussed the research part of this project and also the development part.	

Table 29: Fifth meeting

Sixth meeting		
02.02.2007	09:00	Vinkelbygget
Meeting called by	Scheduled	
Agenda		
09:00 – 10:00		
Attendees	Agata, Peter	
• Final thesis title and definition for sending to Sissel on Thursday 05.02.2007 within 15:00 hours. • Comments on the review of literature review • Comments on thesis structure		
Issues	• No issues. Working with paper prototyping. Literature review on hold until receiving feedback.	
Actions	• Correct date error, date for freezing of thesis title and definition, in agenda and minutes. • Add a “evaluations” sub chapter to Chapter 5 • Continue working with paper prototype until Thursday 08.02.2007 • Start looking at how to code the necessary parts of the user interface	
Decisions	• Wait for feedback from Mikael on final thesis title until Monday 05.02.2007 11:00 hours. • Get feedback on literature review at meeting on Thursday 08.02.2007 • Supervisors will try to get the old parental benefit calculator from NAV • Must send in final thesis title and definition within 15:00 hours on Monday 05.02.2007	
Discussion	• Discussed the thesis structure • Read over the mail Mikael sent to NAV • Discussed my first paper prototype • Talked a bit about accessibility. Try to get the old calculator from NAV to perform accessibility evaluations.	

Table 30: Sixth meeting

Seventh meeting		
09.02.2007	09:00	Vinkelbygget
Meeting called by	Peter	
Agenda		
09:00 – 10:00		
Attendees	Agata, Peter	
• Feedback on things in thesis that are written so far		
Issues	• No issues. Currently, working with HTML. Will continue with the literature review next week	
Actions	• Redo GANTT chart. Start earlier with implementation. • Correct things in thesis and continue to write.	
Decisions	• Send a copy to Mikael and Agata when sending in final thesis title and definition to Sissel. • Ask for assistance if problems with Ruby haven't been solved within next week.	
Discussion	• Received feedback on the review of the thesis.	

Table 31: Seventh meeting

Eighth meeting		
01.03.2007	09:00	Vinkelbygget
Meeting called by	Peter	
Agenda		
09:00 – 10:00		
Attendees	Mikael, Agata, Peter	
	• Take a look at the prototype and UML diagrams and discuss functionality, page design and database design • NAV contact status update	
Issues	• I have some problems with HTML, but I am sure that I will solve them soon.	
Actions	• Redesign the database. Create a new class diagram. • Create new use case diagrams for the different user roles and send in to supervisors for feedback; mother, father, both, administrator • Rename "Web browser" to "GUI" in UML sequence diagram.	
Decisions	• Appointment with NAV Trygd Grimstad on Tuesday 06.03.07 at 10:00. • Mikael will search for some documentation for the services that is being deployed on minside.no	
Discussion	Talked about what should be written in the discussion section in the thesis. Discussed the future work of this project when it's finished. Should write about that the prototype may be used to fill out a complete application form for the parental leave, why some data are stored in the database and how much load this is for the server etc. Discussed the use case diagrams for the prototype and the design of the database. Which tables are needed and their relationships.	

Table 32: Eighth meeting

Telephone meeting		
02.03.2007	15:00	Vinkelbygget
Meeting called by	Mikael	
Agenda		
15:00 –		
Attendees	Mikael, Agata, Peter	
	• Discuss the questions prepared for the meeting with NAV on Tuesday 06.03.2007 10:00.	
Issues		
Actions	• Reorder the questions that I have • Add new questions • Leave out the questions about the calculation of the parental benefit	
Decisions	• Send in draft for the interview to Mikael on Monday 05 March before lunch. • Ask for permission and record the interview at the meeting with NAV on Tuesday for later use.	
Discussion	• What adjustments the questions that I had prepared need, and ideas to new questions that I can ask the case handler at NAV.	

Table 33: Ninth meeting

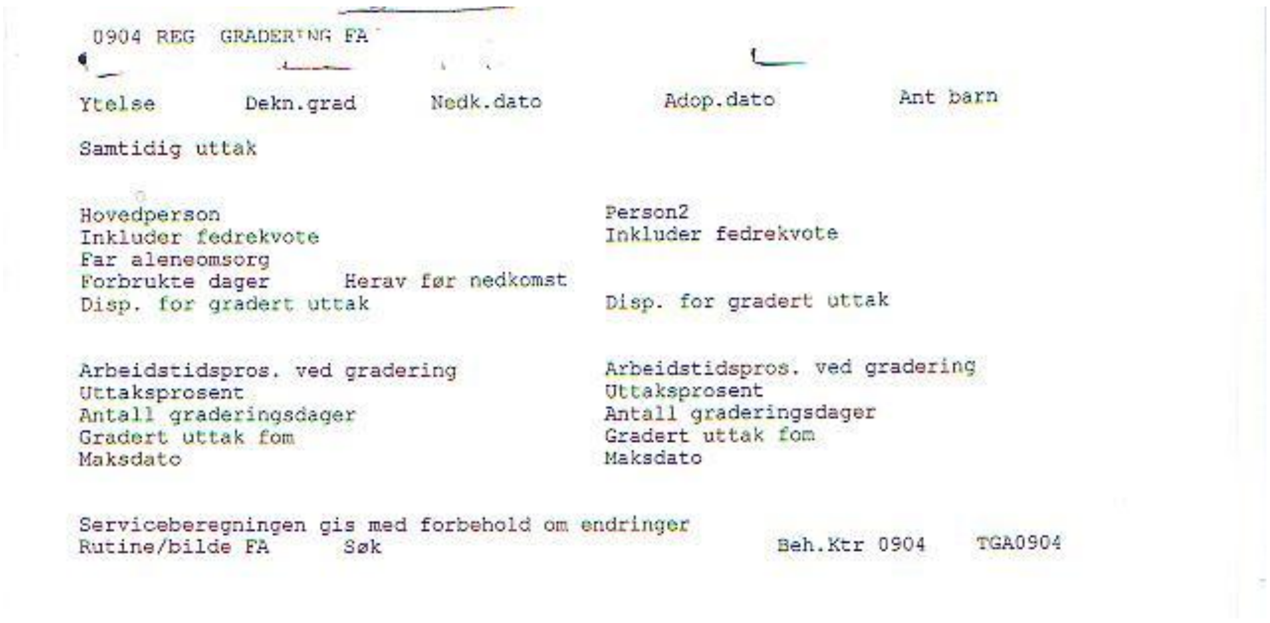
Tenth meeting		
15.03.2007	09:00	Vinkelbygget
Meeting called by	Peter	
Agenda		
09:00 – 10:00		
Attendees	Mikael, Agata, Peter	
1. Take a look at the use case diagrams and class diagram for database design 2. Status update of prototype		
Issues	I don't understand the database and would like some more detailed guidance if possible. I don't quite understand the relationship between the tables and which attributes each table needs. I am also not sure if I need any foreign keys in the tables. As I have understood it, I just need to store the inputs from the users in a table in the database, make use of the values to calculate and then present it on the web page to the users. I also need a table to store the rules for calculation, so that it will be easy for the administrator to change the rules. Therefore I wonder why we have to have the Scenarios table. These scenarios are created automatically when the users are giving the calculator different inputs, so why store them in the database? Should the database store every calculation, which means, should a record be created each time a calculation is performed? Would it be a good idea to create a table in the database which contains the help texts so that it would be easy for the administrator to change them? These help texts are the texts that will	
Actions	• Redesign the database. • Make a new diagram for each use cases. • Write a summary for the interview with NAV. • Write a summary for the initial evaluation with Agata in December 2006.	
Decisions	• Start thinking about what to write in the design chapter. Important to mention how this research approach can be applied to other similar projects, the advantages and weaknesses of the previous solution, and why the prototype is designed as it is etc. • Mikael will continue to try to get some documentation of the services located at www.minside.no • Notify Mikael within Friday 16.03.2007 if the computer problems still haven't been resolved.	
Discussion	Discussed the database content and the interview with NAV. Also talked about some important things to write in the thesis.	

Table 34: Tenth meeting

Eleventh meeting		
29.03.2007	09:00	Vinkelbygget
Meeting called by	Peter	
Agenda		
09:00 – 10:00		
Attendees	Mikael, Agata, Peter	
• I wish to get your opinion about the final database design		
Issues	• I am unable to show æ ø å on the page when loading the prototype. I have added : <?xml version="1.0" encoding="iso-8859-1"?> but with no results. Do you have a solution for this?	
Actions	• Change database design: ○ Rules table : rename to calculationrules, rename rate attribute to current_value, add attribute: date of change of rule, add attribute: description ○ Connect rules to calculations ○ Usedrules table : rename to currentcalculationrules ○ Add support for connecting the rules to the regulations at www.lovdata.no ○ Helptexts and usedhelptexts: rename to guihelptexts and usedguihelptexts • Work with the Design chapter in thesis and send in to Mikael and Agata when finished. Estimated delivery date is: Friday 06.04.2007. • Design chapter: Short summary -> key findings/conclusion. (bullet points)	
Decisions	• Mention multi language support in Discussion chapter • Mikael will look into how to display æ ø å in a web page • Send in the abstract of my work for review before sending it to the T4P Conference by Saturday 31.03.07	
Discussion	• Discussed the database design, the current status of the work and about giving out a publication on The T4P Conference (www.t4p.no)	

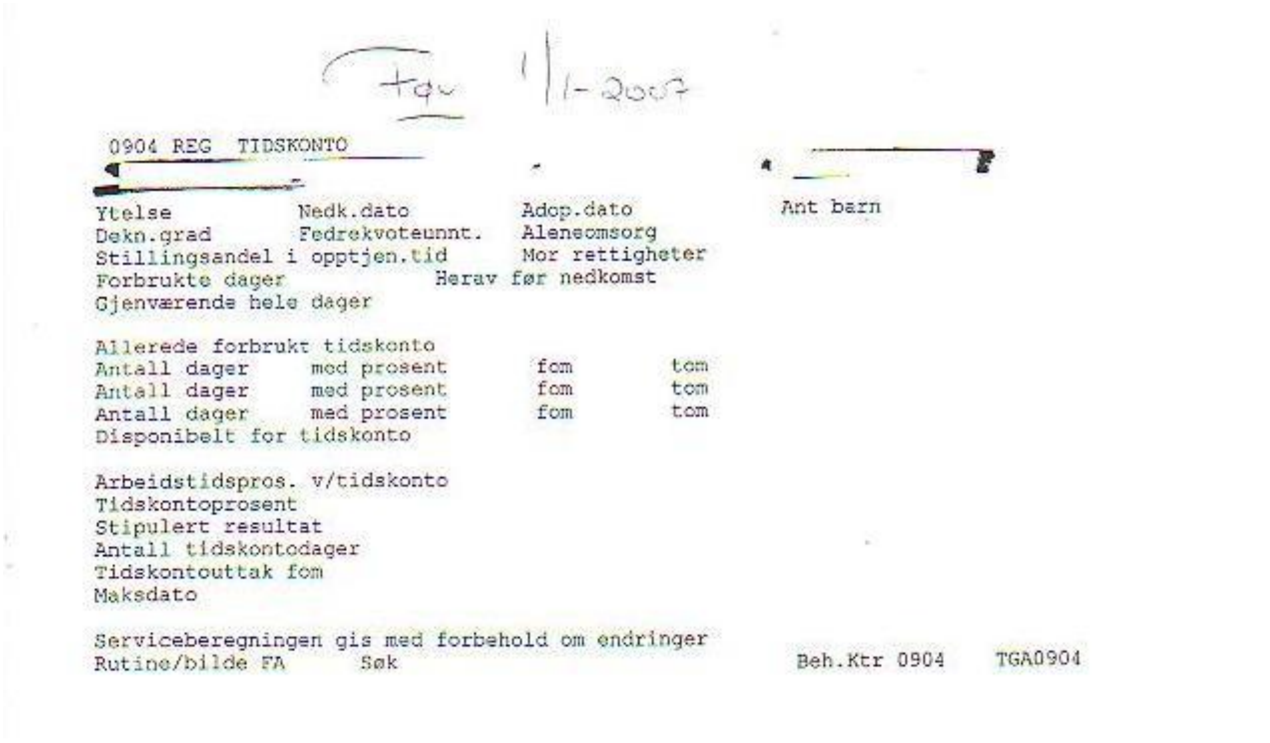
Table 35: Eleventh meeting

11.5 Miscellaneous papers supporting the design



```
0904 REG  GRADERING FA

Ytelse        Dekn.grad      Nedk.dato        Adop.dato        Ant barn
Samtidig uttak

Hovedperson                                  Person2
Inkluder fedrekvote                          Inkluder fedrekvote
Far aleneomsorg
Forbrukte dager      Herav før nedkomst
Disp. for gradert uttak                      Disp. for gradert uttak

Arbeidstidspros. ved gradering               Arbeidstidspros. ved gradering
Uttaksprosent                                Uttaksprosent
Antall graderingsdager                       Antall graderingsdager
Gradert uttak fom                            Gradert uttak fom
Maksdato                                     Maksdato

Serviceberegningen gis med forbehold om endringer
Rutine/bilde FA      Søk                          Beh.Ktr 0904    TGA0904
```

Figure 106: Screenshot of needed user inputs from NAV's system from 2006 and earlier, and from April 2007 towards

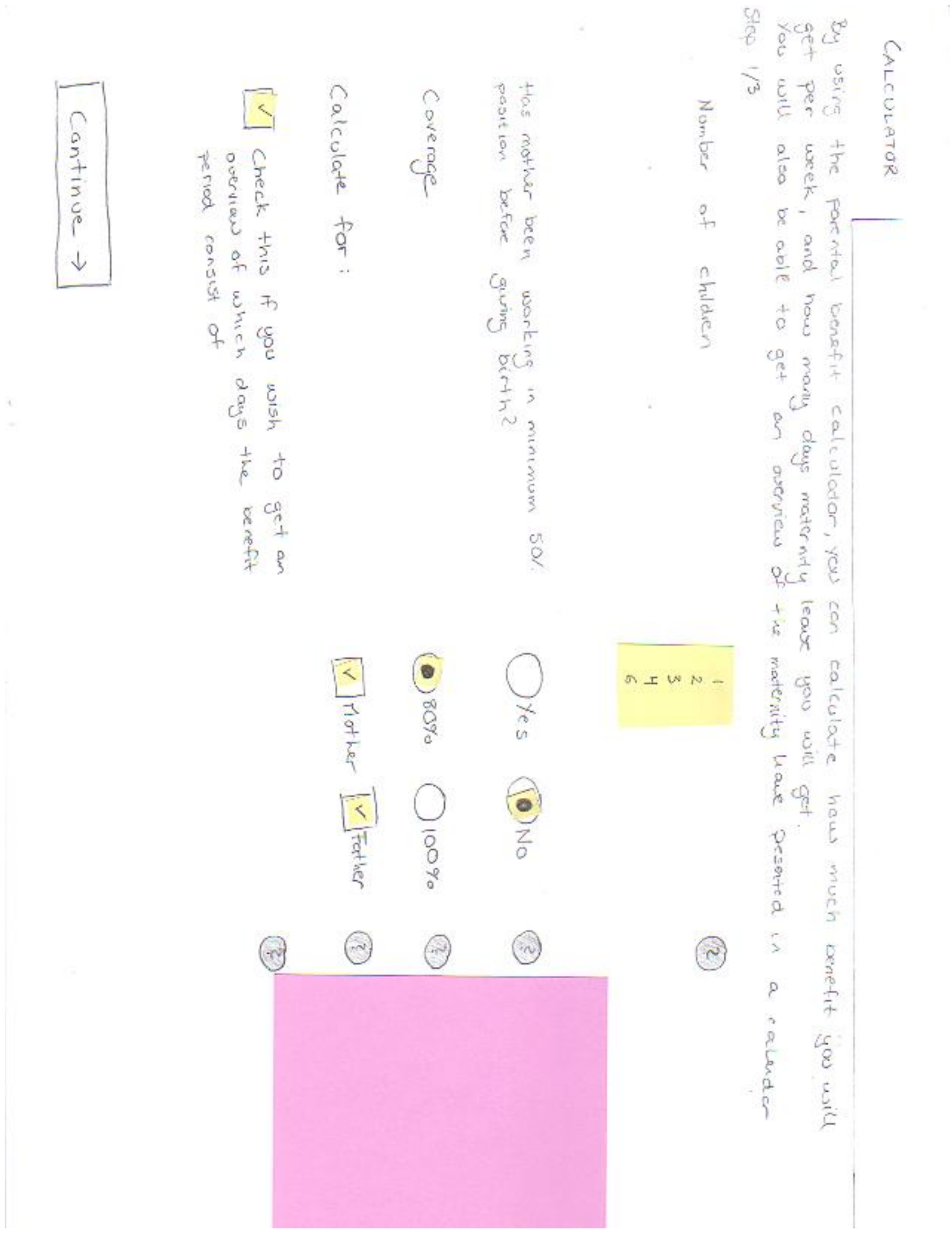


Fqv 1/1-2007

```
0904 REG  TIDSKONTO

Ytelse          Nedk.dato          Adop.dato            Ant barn
Dekn.grad       Fedrekvoteunnt.    Aleneomsorg
Stillingsandel i opptjen.tid       Mor rettigheter
Forbrukte dager           Herav før nedkomst
Gjenværende hele dager

Allerede forbrukt tidskonto
Antall dager     med prosent       fom       tom
Antall dager     med prosent       fom       tom
Antall dager     med prosent       fom       tom
Disponibelt for tidskonto

Arbeidstidspros. v/tidskonto
Tidskontoprosent
Stipulert resultat
Antall tidskontodager
Tidskontouttak fom
Maksdato

Serviceberegningen gis med forbehold om endringer
Rutine/bilde FA      Søk                          Beh.Ktr 0904    TGA0904
```

Figure 107: Screenshot of needed user inputs from NAV's system from 2007 and on

11.6 Paper prototype



CALCULATOR

By using the parental benefit calculator, you can calculate how much benefit you will get per week, and how many days maternity leave you will get.
You will also be able to get an overview of the maternity leave presented in a calender

Step 1/3

Number of children 1 2 3 4 6

Has mother been working in minimum 50% position before giving birth? Yes [✓ No]

Coverage [✓ 80%] 100%

Calculate for: [✓ Mother] [✓ Father]

[✓] Check this if you wish to get an overview of which days the benefit period consist of

Figure 108: Paper prototype page 1

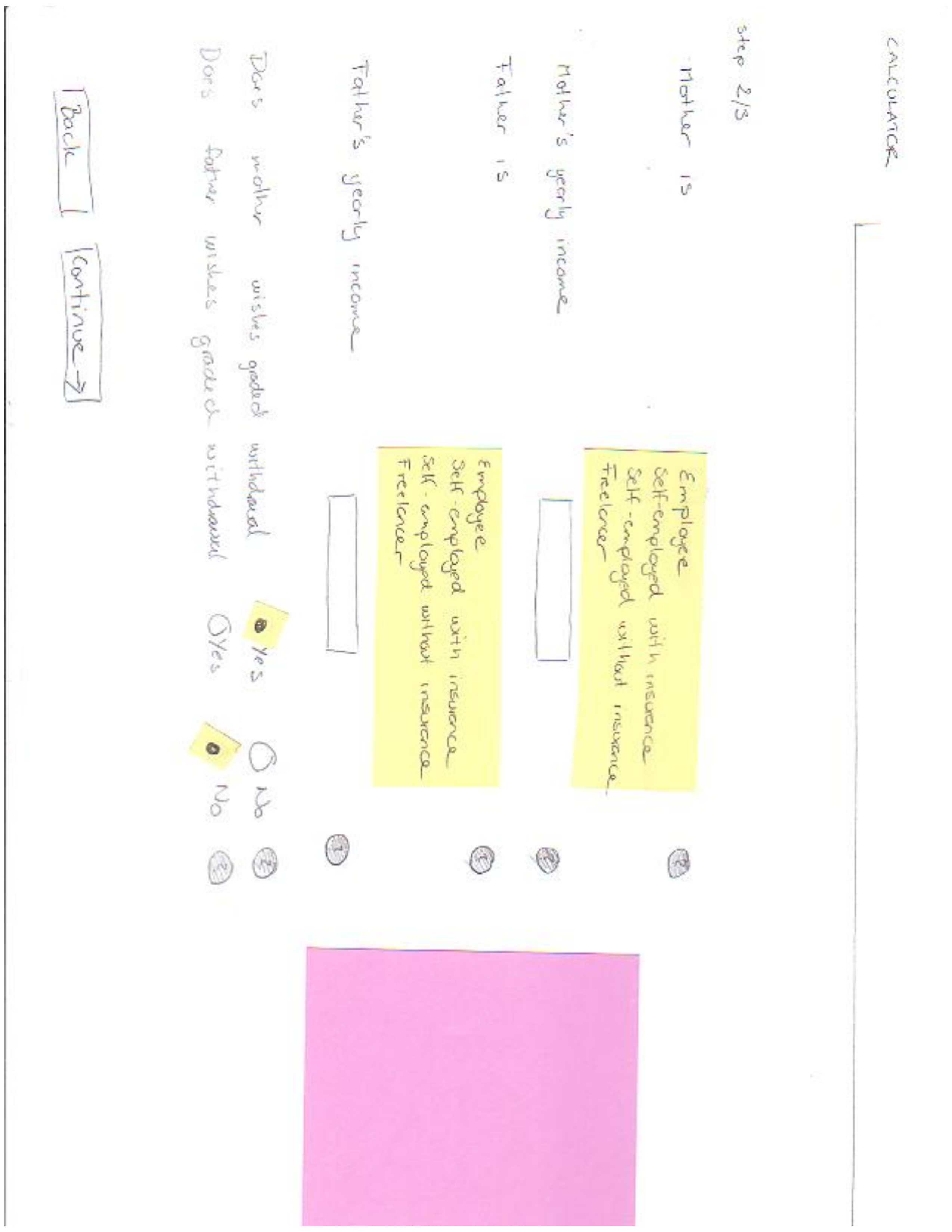

CALCULATOR
step 2/3
Mother is
Employee
Self-employed with insurance
Self-employed without insurance
Freelancer
Mother's yearly income
Father is
Employee
Self-employed with insurance
Self-employed without insurance
Freelancer
Father's yearly income
Does mother wishes graded withdrawal
Yes
No
Does father wishes graded withdrawal
Yes
No
Back
Continue →

Figure 109: Paper prototype page 2

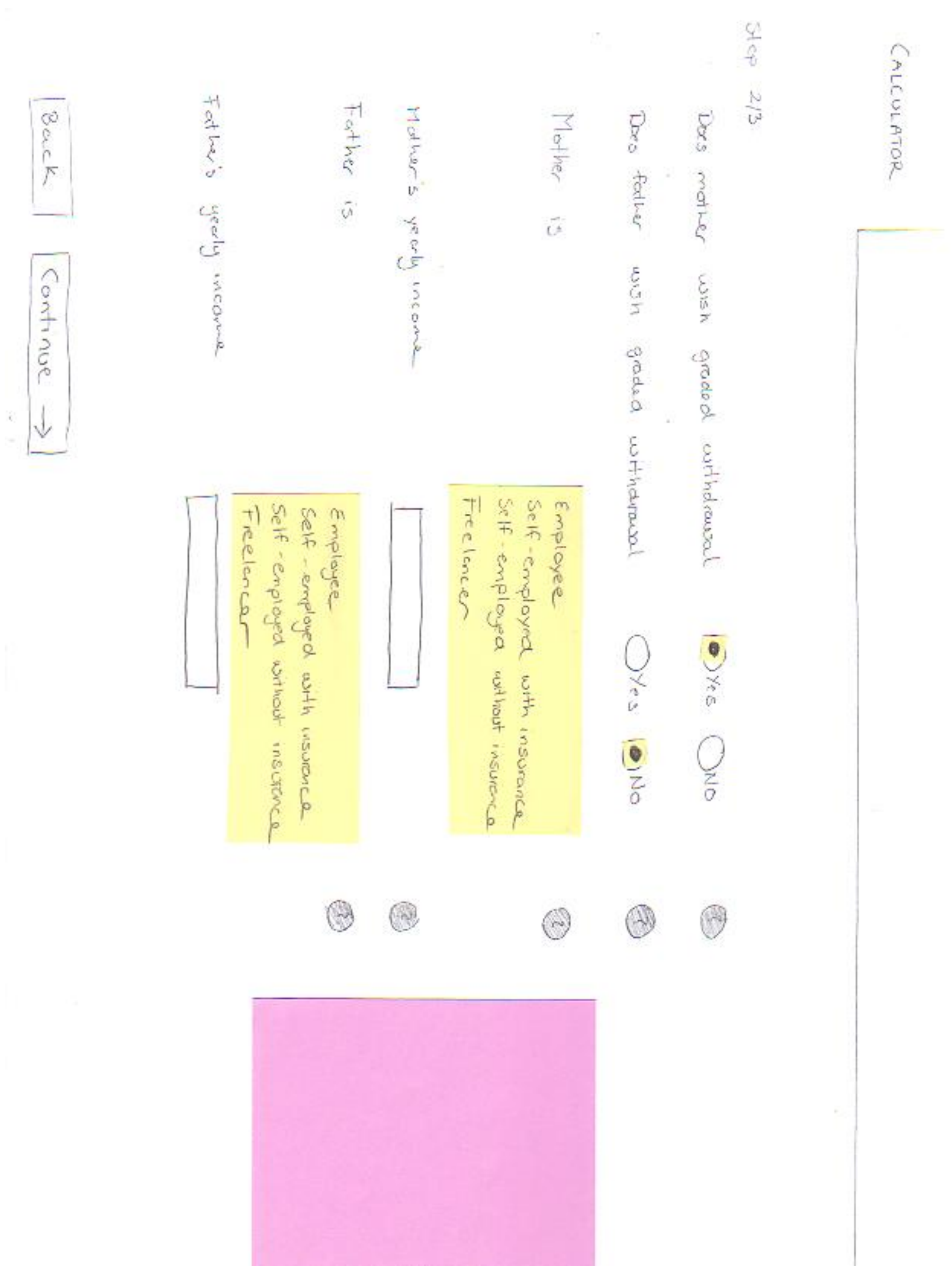

CALCULATOR
Step 2/3
Does mother wish graded withdrawal [✓ Yes] No
Does father wish graded withdrawal Yes [✓ No]
Mother is
Employee
Self-employed with insurance
Self-employed without insurance
Freelancer
Mother's yearly income
Father is
Employee
Self-employed with insurance
Self-employed without insurance
Freelancer
Father's yearly income
Back
Continue →

Figure 110: Paper prototype page 2 alternate version

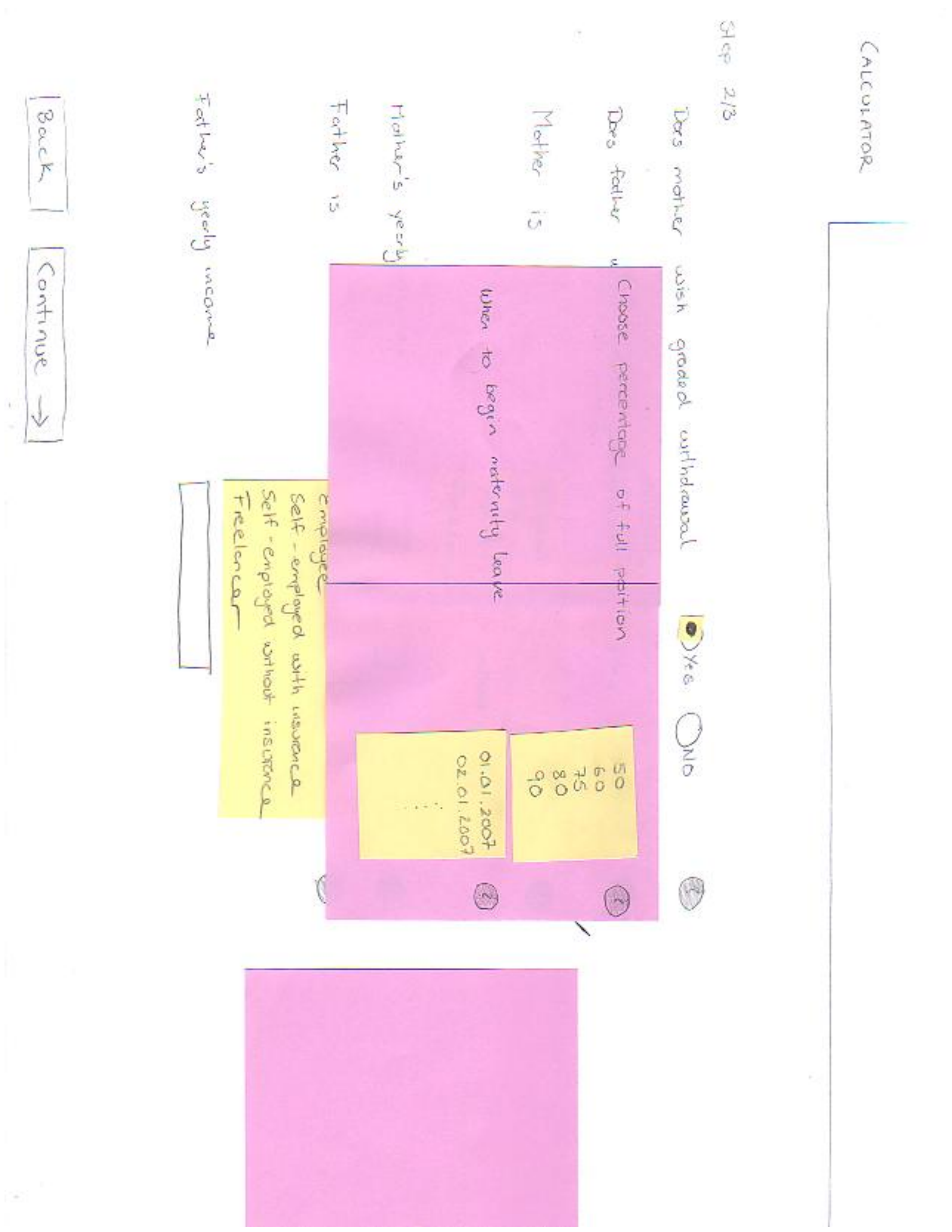

CALCULATOR
Step 2/3
Does mother wish graded withdrawal
Yes
No
Does father
Choose percentage of full position
50
60
75
80
90
Mother is
When to begin maternity leave
01.01.2007
02.01.2007
Father is
Mother's yearly
Employee
Self-employed with insurance
Self-employed without insurance
Freelancer
Father's yearly income
Back
Continue →

Figure 111: Paper prototype page 2 alternate version with visible dynamic content

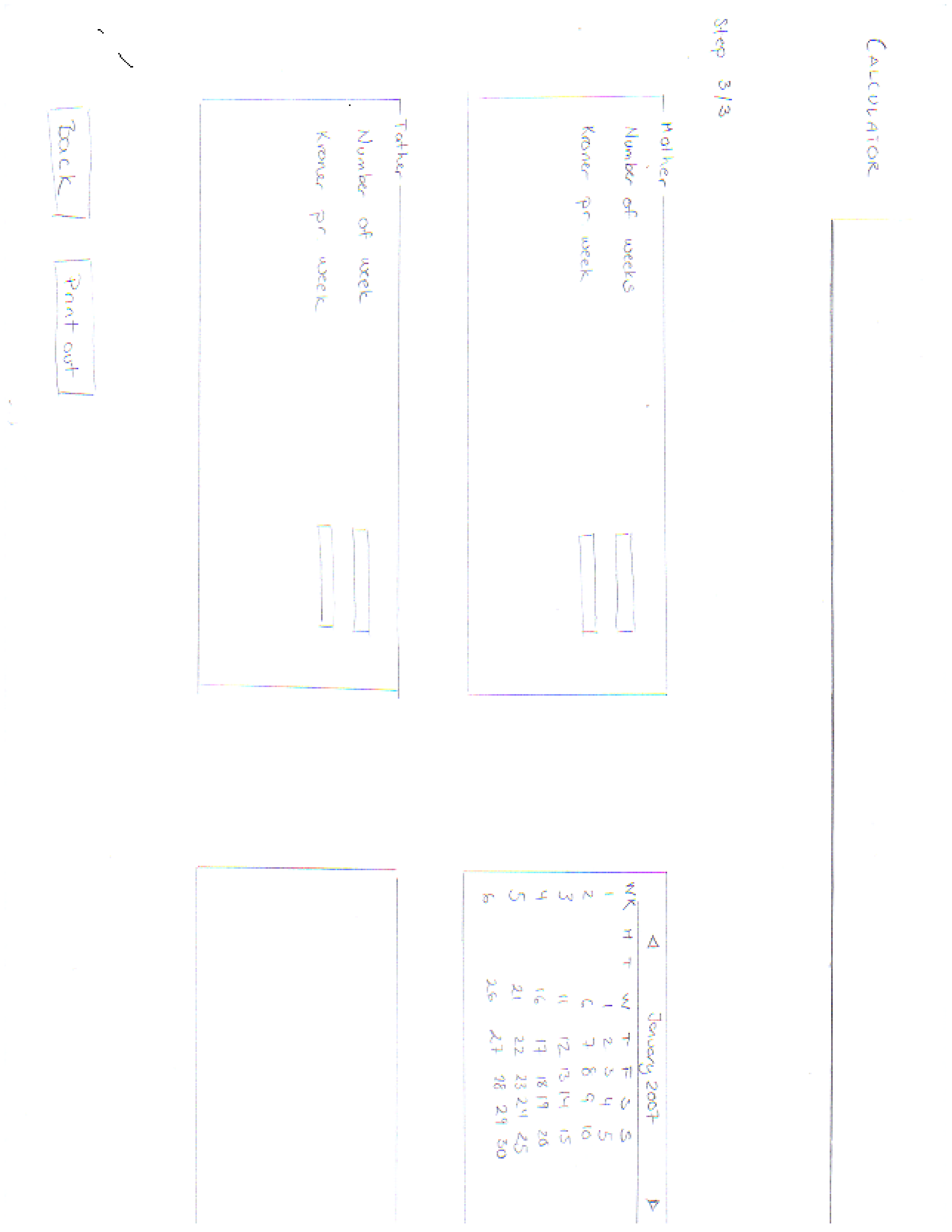


Figure 112: Paper prototype page 1

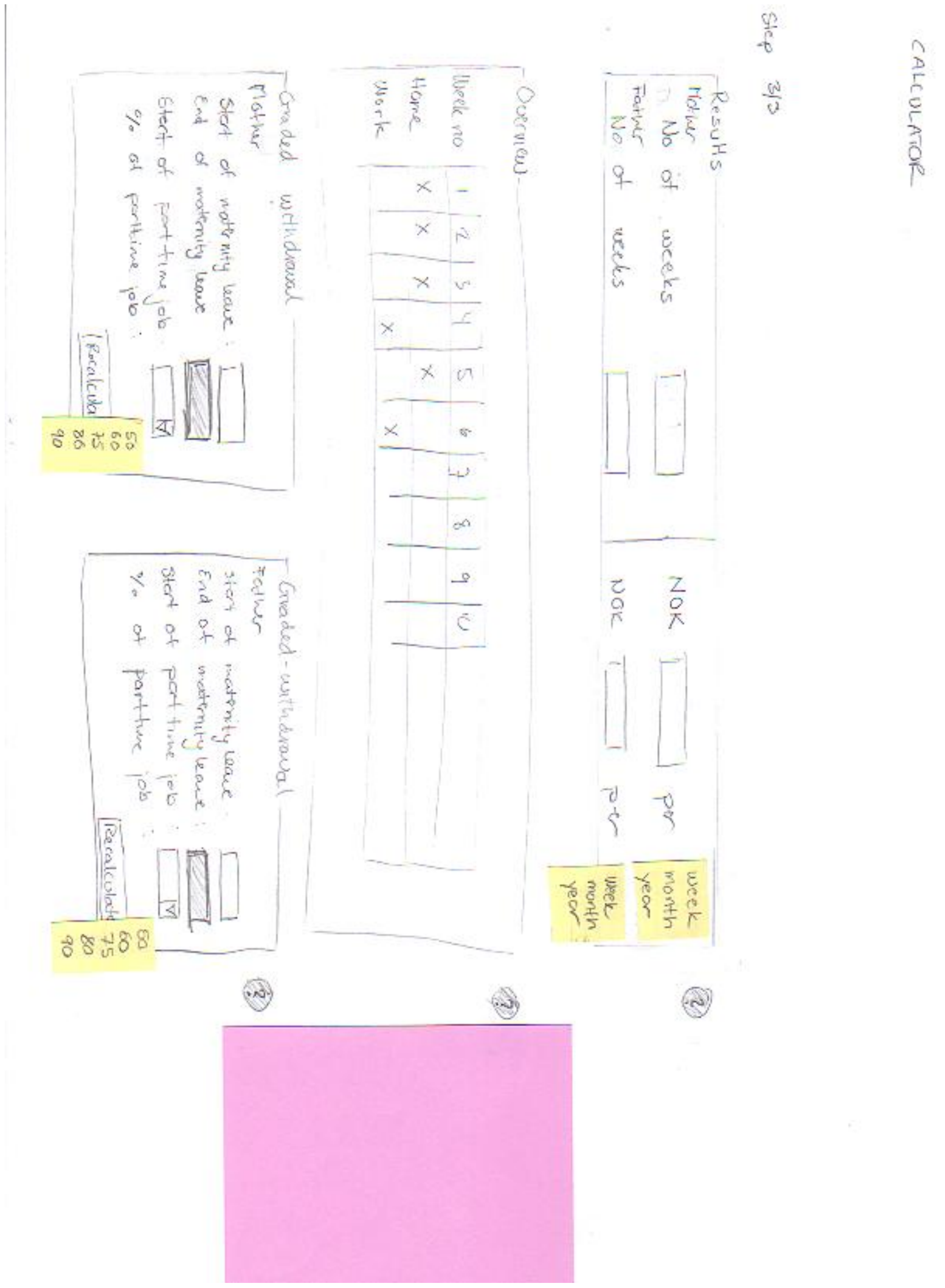


Figure 113: Paper prototype page 3 alternate version

11.7 Bibliography

Agile modeling. (2007, 03). Retrieved 05 25, 2007, from http://www.agilemodeling.com/essays/umlDiagrams.htm

Apache.org. (n.d.). Retrieved 05 25, 2007, from http://httpd.apache.org/

Blue Ink . (2005, 10 14). Retrieved 05 07, 2007, from Blue Ink - web applications instantly: http://www.blueink.biz/RapidApplicationDevelopment.aspx

Cascia, A. (n.d.). *Encyclopedia of Educational Technology*. Retrieved 05 25, 2007, from http://coe.sdsu.edu/eet/articles/usabilitytestin/start.htm

eclipse.org. (2007). Retrieved 05 07, 2007, from eclipse - an open development platform: http://www.eclipse.org

eNode. (2002). Retrieved 01 24, 2007, from eNode : Powerful XML Infrastructure: http://www.enode.com/x/markup/tutorial/mvc.html

Garshol, L. M. (1999, 10 12). *En introduksjon til CSS*. Retrieved 02 06, 2007, from css-intro.html: http://www.garshol.priv.no/download/text/css-intro.html

Grimmer, L. (2006, 02). *Interview with David Heinemeier Hansson from Ruby on Rails*. Retrieved 01 24, 2007, from MySQL: http://dev.mysql.com/tech-resources/interviews/david-heinemeier-hansson-rails.html

Heeks, R. (2006). *Implementing and Managing eGovernment.*

HeidiSQL. (n.d.). Retrieved 04 06, 2007, from www.heidisql.com

Hibbs, C. (2005, 10 13). *ONLamp*. Retrieved 05 25, 2007, from http://www.onlamp.com/pub/a/onlamp/2005/10/13/what_is_rails.html

Home of Principle Based Project Management. (n.d.). Retrieved 01 22, 2007, from Information and Training Site: http://hyperthot.com/project.htm

Kaufmann, M. (2003, 4 16). *Frontpage, Paper prototyping*. Retrieved 01 31, 2007, from Paper prototyping : The fast and easy way to design and refine user interfaces: http://www.paperprototyping.com/what.html

London South Bank University. (1997, 11 21). Retrieved 05 08, 2007, from London South Bank University - Business, Computing & Information Management: http://www.scism.sbu.ac.uk/inmandw/tutorials/kaqu/qu8.htm

Meyer, M. A. (2005, 06). *eNorge 2009 - det digitale spranget*. Retrieved 02 13, 2007, from Fornyings- og Administrasjonsdepartementet: http://www.regjeringen.no/nb/dep/fad/Tema/IT-politikk__eNorge/eNorge-2009.html?id=439499

mozilla developer center. (2007, 02 19). Retrieved 05 25, 2007, from http://developer.mozilla.org/en/docs/About_JavaScript

Nielsen, J. (1993). Usability Engineering. In U. Engineering.

Nielsen, J. (1993). Usability Engineering. In *Usability Engineering* (p. 155).

Nielsen, J. (2003). *Use-it*. Retrieved 05 25, 2007, from http://www.useit.com/alertbox/20030414.html

Object Management Group. (2007, 01 02). Retrieved 05 07, 2007, from Object Management Group - Unifield Modelling Language: http://www.uml.org/

OpenSource.org. (2007, 03 13). Retrieved 05 13, 2007, from http://www.opensource.org/

Perens, B. (n.d.). *perens.com*. Retrieved 05 13, 2007, from perens.com: http://perens.com/OpenStandards/Definition.html

Preece, J. (1994). Human-Computer Interaction. In *Human-Computer Interaction* (p. 14).

Preece, J. (1994). Human-Computer Interaction. In *Human-Computer-Interaction* (p. 3).

Pressemelding Nr. 41/2006. (2006, 10 06). Retrieved 02 13, 2007, from Fornyings- og Administrasjonsdepartementet - Storsatsing på opne IT-standardar og open kjeldekode i 2007: http://www.regjeringen.no/nb/dep/fad/pressesenter/pressemeldinger/2006/Storsatsing-pa-opne-IT-standardar-og-open-kjeldekode-i-2007.html?id=271707

Pressemelding Nr.: 24/2007. (n.d.). Retrieved 05 13, 2007, from Pressemelding Nr.: 24/2007: http://www.regjeringen.no/nb/dep/fad/pressesenter/pressemeldinger/2007/--Forste-skritt-mot-en-offentlig-sektor-.html?id=466602

Pressemelding Nr.:17/2007. (2007, 04 16). Retrieved 05 13, 2007, from Pressemelding Nr.:17/2007: http://www.regjeringen.no/nb/dep/fad/pressesenter/pressemeldinger/2007/MinSide-blir-fri-programvare.html?id=463325

Pressemelding Nr: 01/2007. (2007, 01 08). Retrieved 02 13, 2007, from Fornyings- og Administrasjonsdepartementet - Over 100.000 brukarar på Miside: http://www.regjeringen.no/nb/dep/fad/pressesenter/pressemeldinger/2007/Over-100000-brukarar-pa-Miside.html?id=440838

Rails : Web development that doesn't hurt. (n.d.). Retrieved 01 23, 2007, from www.rubyonrails.org: http://www.rubyonrails.org/

Ruby - A Programmer's Best Friend. (n.d.). Retrieved 01 23, 2007, from www.ruby-lang.org/en: http://www.ruby-lang.org/en/about/

Ruby on rails: Understanding MVC. (n.d.). Retrieved 03 21, 2007, from Wiki.rubyonrails.org: http://wiki.rubyonrails.org/rails/pages/UnderstandingMVC

SDN. (n.d.). Retrieved 05 25, 2007, from Sun Developer Network: http://java.sun.com/blueprints/patterns/MVC.html

sincerechoice.org. (n.d.). Retrieved 05 13, 2007, from sincerechoice.org: http://sincerechoice.org/Principles/Open_Standards.html

Snyder, C. (2004, 06 29). *Paper protoyping*. Retrieved 01 31, 2007, from Snyder Consulting: http://www.snyderconsulting.net/us-paper.pdf

St.meld. nr. 17. (2006-2007, 12 15). Retrieved 01 22, 2007, from Fornyings- og Administrasjonsdepartementet - Eit informasjonssamfunn for alle: http://www.regjeringen.no/nb/dep/fad/dok/regpubl/stmeld/20062007/Stmeld-nr-17-2006-2007-.html?id=441497&epslanguage=NO

Stewart, B. (2001, 11 29). *An Interview with the Creator of Ruby*. Retrieved 01 23, 2007, from www.linuxdevcenter.com: http://www.linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html

University of Rhode Island. (n.d.). Retrieved 05 25, 2007, from http://www.uri.edu/home/help/www/mysql.html

Usabiliy first. (n.d.). Retrieved 05 25, 2007, from Usability in Website and Software Design: http://www.usabilityfirst.com/glossary/term_296.txl

useit.com. (n.d.). Retrieved from useit.com: http://www.useit.com/papers/heuristic/

W3 - WAI. (n.d.). Retrieved 05 08, 2007, from W3: http://www.w3.org/WAI/intro/accessibility.php

W3 CSS. (n.d.). Retrieved 04 30, 2007, from www.w3.org: http://www.w3.org/Style/CSS/

W3C - XHTML. (n.d.). Retrieved 05 25, 2007, from http://www.w3.org/TR/xhtml1/

W3C: XHTML. (n.d.). Retrieved 02 06, 2007, from W3C: http://www.w3.org/TR/xhtml1/#xhtml

W3schools. (n.d.). Retrieved 05 25, 2007, from http://www.w3schools.com/ajax/ajax_intro.asp

W3schools - HTML. (n.d.). Retrieved 05 25, 2007, from http://www.w3schools.com/html/html_intro.asp

W3Schools: Ajax Tutorial. (n.d.). Retrieved 03 21, 2007, from www.w3schools.com: http://www.w3schools.com/ajax/default.asp

W3Schools: Javascript Intro. (n.d.). Retrieved 03 21, 2007, from www.w3schools.com: http://www.w3schools.com/js/js_intro.asp

WEBrick. (2004, 08 31). Retrieved 01 24, 2007, from WEBrick: http://www.webrick.org/

Wikipedia, UML. (n.d.). Retrieved 05 07, 2007, from Wikipedia, UML: http://en.wikipedia.org/wiki/Unified_Modeling_Language#Modeling