USABILITY AND PRESENTATION LAYER OF A WEB BASED MANAGEMENT APPLICATION

vrije Universiteit
Faculty of Sciences

August 2008

Laura Garrido Puras

Acknowledgements

Preface

Chapter 1. *Introduction*

1.1 Introduction

In this paper, I present my Project *Usability and Presentation layer of a web based management application*. In this project I will measure and improve the quality of user's experience when interacting with the Staff and Salary application, PersoneelOnline.

1.2 Who am I?

I am a student on a Erasmus scholarship at the Vrije Universiteit of Amsterdam, studying Informatics Engineering at the Polytechnic University of Madrid.
For my Final Project I studied the usability and developed the presentation layer of a Staff and Salary web based application.

1.3 The goal of the project

The goal of this project is to develop a final user interface that enables the user to think he is working in a desktop application meanwhile he is in fact running a web application.

In the remainder of this paragraph you further explain the goal. For whom, but especially why, have you created the system in the way it is presented in this paper. Just keep in mind that this is the introductory chapter, so don't go to deep yet.

1.4 Project set-up

In the project set-up you explain how the project has been executed. To come to a design a couple of steps in the design process have to be followed. Here you define these steps and explain them shortly.

1.4.1 Task Analysis

The task analysis describes the current situation, if relevant, and the future situation as envisioned by the design team from the point of view of the client and users. The current situation in case of a web site is often none existent. The most important job of the task analysis is to get a good idea of the future situation. The tasks that have to be fulfilled at the site have to answer the demands of both the client and the users.

This project has started with a task analysis to come to a good design for the client. An inventarisation has been made of the current situation and the demands for the future situation.

The task analysis is based on a visit of the company and several interviews taken with the client and future users of the system.

1.4.2 Prototyping and Evaluation

The prototype group will implement the specifications delivered by the technical design group. Their task is to build a working, but not necessarily complete, prototype of the site.

The evaluation group will evaluate all the previous steps during the design process. Their task is to make sure that they give feedback to the appropriate group whenever they find a possible inconsistency or design flaw.

1.4.3 Technical Design

The Technical Design Group will make a representation of the requirements of the site. Their task is to work out the requirements in the task model into a complete and consistent detailed model. They will specify the functionality as well as the structure of the site. The task of the technical design group can be decomposed in a couple of subtasks:

-
-

Together these subtasks are the technical design. This is also called the User Virtual Machine.

1.5 Organization

Since the design has to be made in a relatively short period of time, the organization of the project is crucial. Every step in the design process has to be carefully planned and the team members have to make certain deadlines and milestones. To realize this the communication within the team has to be well organized. Describe in this paragraph how this was done in your team.

Besides that you can describe here who is part of your project world. That will not only be the team members, but also your clients, the future users, your teacher and his assistants. Name the roles of all the team members and their tasks.

Be very aware of the spelling of all the names. This is a document that is meant for your client. He won't like it if his, or a name from a customer, is misspelled.

1.6 Documentation organization

Describe here shortly the way this document is built up.

This document exists of six chapters, including this chapter, and several appendices.

Chapter 3 describes the task analysis of the project. It gives the methods and techniques used and what choices are made. With Chapter 3 comes **Appendix A**, in which the results of the task analysis are given, including the full transcriptions of the interviews that were taken.

Chapter 4 contains information about the prototype. The corresponding **Appendix B** gives a detailed description of the prototype.

Finally Chapter 5 gives an overview of all the evaluations during the project. **Appendix C** describes the detailed evaluation techniques used.

In Chapter 6 the technical design of the web site is described. Some general information about the field of work, global tasks and goals is given. The chapter includes information about the methods and techniques used in this project, as well as information about the dialog interface of the site. **Appendix D** holds the specific formal specification of the technical design.

Chapter 2. *Background*

2.1 Introduction

2.2 Usability Engineering

2.2.1 What is Usability?

Contrary to what some might think, usability is not just the appearance of the user interface (UI). Usability measures the quality of a user's experience when interacting with a product or system. In general, usability refers to how well users can learn and use a product to achieve their goals and how satisfied they are with that process. Usability, as defined by Joseph Dumas and Janice (Ginny) Redish, means that people who use the product can do so quickly and easily to accomplish their tasks. Usability may also consider such factors as cost, effectiveness and usefulness.

2.2.2 What does usability measure?

It is important to realize that usability is not a single, one-dimensional property of a user interface. Usability is a combination of factors including:

- **Ease of learning** How fast can a user who has never seen the user interface before learn it sufficiently well to accomplish basic tasks?. How easy it is to learn the main system functionality and gain proficiency to complete the job?. We usually assess this by measuring the time a user spends working with the system before that user can complete certain tasks in the time it would take an expert to complete the same tasks. This attribute is very important for novice users.

- **Efficiency of use** Once an experienced user has learned to use the system, how fast can he or she accomplish tasks? In other words, the number of tasks per unit of time that the user can perform using the system. We look for the maximum speed of user task performance. The higher system usability is, the faster the user can perform the task and complete the job.

- **Memorability** If a user has used the system before, can he or she remember enough to use it effectively the next time or does the user have to start over again learning everything? It is critical for intermittent users to be able to use the system without having to climb the learning curve again. This attribute reflects how well the user remembers how the system works after a period of nonuse.

- **Error frequency and severity** How often do users make errors while using the system, how serious are these errors, and how do users recover from these errors? This attribute contributes negatively to usability. It does not refer to system errors. On the contrary, it addresses the number of errors the user makes while performing a task. Good usability implies a low error rate. Errors reduce efficiency and user satisfaction, and they can be seen as a failure to communicate to the user the right way of doing things.

- **Subjective satisfaction** How much does the user like using the system? This shows a users subjective impression of the system.

2.2.3 Scenario-Based Usability Engineering

Computers do more than just provide information and services for people to use. The design of computing systems is part of an ongoing cycle in which new technologies raise new opportunities for human activity; as people's tasks change in response to these opportunities, new needs for technology arise. The basic argument behind scenario-based methods is that descriptions of people using technology are essential in discussing and analyzing how the technology is (or could be) reshaping their activities. A secondary advantage is that scenario descriptions can be created before a system is built and its impacts felt.

A user interaction scenario is a story about people and their activities (Caroll & Rosson 1990). Scenarios describe the setting, actor(s), and events of a user-computer interaction, but also include information about user's mental activities (goals, plans, and reactions).Scenarios integrate the many tasks of system development by first organizing the analysis of user needs, and then serving as central representations of user needs that are developed in a systematic manner through design, evaluation, and documentation activities. It does not explicitly describe the use of software or other technological support to achieve a task. Using the vocabulary and phrasing of users means that the scenarios can be understood by the stakeholders, and they are able to participate fully in the development process. In fact, the construction of scenarios by stakeholders[1] is often the first step in establishing requirements. Scenarios can be used to guide usability engineering - the scenario-based development (SBD) framework (Figure ??). The framework should not be understood as a waterfall, even though the diagram shows a "downward" flow from problem analysis to design and then to evaluation. At each step of the process, scenarios are analyzed and transformed in support of different development goals. We assume that all activities in SBD happen in an iterative and interleaved fashion, but for explanatory purposes we organize them into an idealized progression.

2.2.4 Requirements Analysis

In requirements analysis, the problem situation is studied through interviews with clients and other users (the stakeholders), field studies of the current situation, and brainstorming among users and developers. This input is used to formulate problem scenarios that convey important characteristics of the users, the typical and critical tasks they engage in, the tool they use, and their organizational context.

A key contribution of scenarios during requirements analysis is that they evoke reflection and discussion. Writing down a narrative of one situation almost immediately raises questions about other situations, about why this situation works the way it does, and how others situations might work differently. The concrete and narrative character of scenarios also facilitates mutual understanding and communication among the different groups who participate in requirements analysis.

There are two aims. One aim is to understand as much as possible about the users, their work, and the context of that work, so that the system under development can support them in achieving their goals. Building on this, our second aim is to produce, from the needs identified, a set of stable requirements that form a sound basis to move forward into thinking about design. This will be the requirements analysis' result. The **requirements specification**; a document that lists all functions and features that the proposed system must satisfy.

Conceptually, requirements analysis includes three types of activity:

- **Eliciting requirements**: communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering. Several techniques can be employed to elicit the requirements from the customer. Historically, this has included such things as holding interviews, holding surveys, making recordings or just workplace observations.

- **Analyzing requirements**: the task of determining whether the stated requirements are incomplete, unclear, ambiguous, or contradictory, and then resolving these issues.

- **Recording requirements**: Requirements may be documented in various forms, such as natural language documents, HTA[2], use cases, user stories, or process specifications.

Different kinds of requirements

[1] The stakeholders name the groups or individuals who should be consulted or observed in the fieldwork

[2] Hierarchical Task Analysis

In software engineering, two different kinds of requirements have traditionally been identified: functional requirements , which say what the system should do, and non-functional requirements , which say what constraints there are on the system and its development.
A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that show how a use case is to be fulfilled. It is a requirement that, when satisfied, will allow the user to perform some kind of function. They are supported by non-functional requirements, which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

However, instead of referring to all requirements that are not functional as simply "non-functional" requirements, we could refine this into further categories:

- **Data requirements** capture the type, volatility, size/amount, persistence, accuracy, and value of the amounts of the required data.

- **Environmental requirements** refer to the circumstances in which the interactive product will be expected to operate. For example, what technologies will the product run or need to be compatible with and what technological limitations might be relevant.

- **User requirements** capture the characteristics of the intended user group.

- **Usability requirements** capture the usability goals and associated measures for a particular product.

Main techniques and steps in analyzing requirements

Stakeholders identification

Stakeholder analyses are now arguably more important than ever because of the increasingly interconnected nature of the world. Choose any public problem and it is clear that 'the problem' encompasses or affects numerous people, groups and organizations. Figuring out what the problem is and what solutions might work are actually part of the problem, and taking stakeholders into account is a crucial aspect of problem solving. At a minimum, stakeholder analyses should help us figure out who the key stakeholders are and what would satisfy them. The technique used in this particular step could be brainstorm the list of potential stakeholders.

Preparing for the Field Study

Questions about the current situation the system is hoping to address, will come up as you develop the root concept. Now we have identified the different stakeholders, we are ready to develop different guides for conducting our interviews with each stakeholder . Each guide should support the questioning process. At the top there is a reminder about what the interviewer is trying to accomplish. The goal is to learn what the participants think about their own activities, so the guide should avoid specific and pointed questions early in the interview. Instead, begin with open-ended prompts that explore general background and how the interviewees think about their work. More specific questions are listed at the end, reminding the interviewer to address these issues if they have not yet been raised in the earlier discussion.

In addition to preparing an interviewing guide, we must decide how to document the field observations . If the work setting involves considerable physical manipulation of objects, a videotape may be helpful. Otherwise, a small tape recorder can be used to record conversations or maybe we can use some video screen capture software.

Summarizing the field data

Here we discuss several summary representation that can be used to organize the findings about a project's stakeholders and their activities.

Stakeholders

The observations and interviews for each stakeholder group are organized into stakeholder profiles. These profiles summarize the general characteristics of each group, and are based on the observations and interviews from the field study. We organize the summaries into background and expectations for the proposed system.

Task Analysis

Another set of summaries is developed to document the tasks of each stakeholder group. A simple list of tasks that were observed could be one way of summarizing the task analysis.
For tasks that have many steps, or that are particularly important in an activity, a hierarchical task analysis may be developed. This analysis decomposes complex tasks into subtasks. Figure 2.4 present a hierarchical task analysis. Each box in the task analysis diagrams represents a task step. Vertical lines indicate decomposition of a step into two or more subtasks; the subtasks are gathered together under the horizontal lines. Numbering indicates how a task is decomposed, and the plans show the logical ordering or dependencies among subtasks.

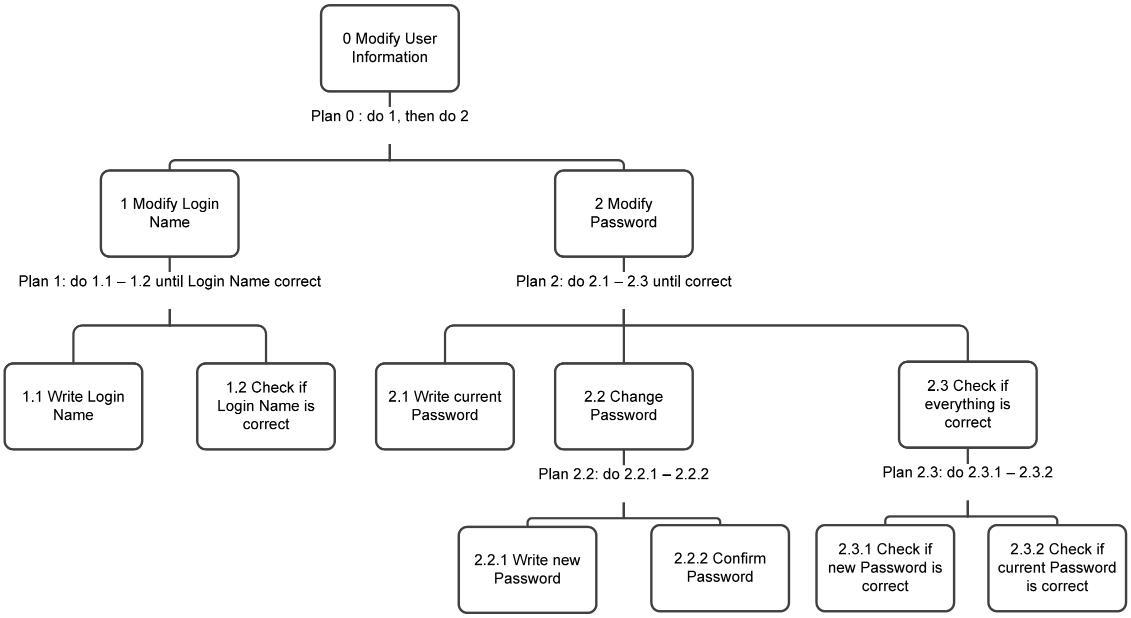

0 Modify User Information
Plan 0 : do 1, then do 2
1 Modify Login Name
2 Modify Password
Plan 1: do 1.1 – 1.2 until Login Name correct
Plan 2: do 2.1 – 2.3 until correct
1.1 Write Login Name
1.2 Check if Login Name is correct
2.1 Write current Password
2.2 Change Password
2.3 Check if everything is correct
Plan 2.2: do 2.2.1 – 2.2.2
Plan 2.3: do 2.3.1 – 2.3.2
2.2.1 Write new Password
2.2.2 Confirm Password
2.3.1 Check if new Password is correct
2.3.2 Check if current Password is correct

Figure 2.4: Hierarchical task analysis of Modify User Information

2.2.5 Prototyping

Prototypes are constructed and evaluated to guide redesign and refinement. A prototype is a concrete but partial implementation of a system design. A user interface prototype is a prototype built to explore usability issues (Wasserman & Shewmake 1982).

Usability testing is the core of usability engineering practice: Representative users are asked to interact with system prototypes, and their behavior and subjective reactions are studied. Prototypes can be used to test all aspects of usability for a system – what users will expect when they encounter parts of the system, how they will go about pursuing their goals, how they will respond to system feedback, and what subjective reactions they will have.

The most convenient prototype for usability testing is an early working version of the system. A working system brings a sense of realism to test tasks. Task instructions can be minimal, because evaluators can rely on the system to guide users through the test tasks. Issues associated with overall complexity or internal consistency can be examined, and the measures of satisfaction or irritation that are collected will be much more meaningful.

Unfortunately, waiting for a running version of a system may mean that usability testing must be postponed well into the development process. A popular alternative is to build a realistic simulation with rapid prototyping tools (e.g., Visual Basic or Macromedia Director), with the understanding that the prototype is temporary and will be replaced eventually by the real system. A discardable prototype can be an excellent option if the usability professionals already have expertise with an appropriate tool, but if not, building the prototype may become a major implementation effort itself. There is also the risk that a prototype will exhibit enough functionality that the team comes to believe that what began as a prototype is the final system.

Some Advantages of Prototyping:

- Reduces development cost.
- Requires user involvement.
- Developers receive quantifiable user feedback.
- Facilitates system implementation since users know what to expect.
- Results in higher user satisfaction.
- Exposes developers to potential future system enhancements.

Some Disadvantages of Prototyping:

- Can lead to insufficient analysis.
- Users expect the performance of the ultimate system to be the same as the prototype.
- Can cause systems to be left unfinished and/or implemented before they are ready.
- Sometimes leads to incomplete documentation.
- If sophisticated software prototypes (4th GL or CASE Tools[3]) are employed, the time saving benefit of prototyping can be lost.

2.2.6 Evaluation

Usability evaluation is a central activity in the usability process. It can determine the current version's usability level and whether the design works.

There are generally three types of usability evaluation methods: *Testing*, *Inspection*, and *Inquiry*.

Usability Testing

In Usability Testing approach, representative users work on typical tasks using the prototype and the evaluators use the results to see how the user interface supports the users to do their tasks. Testing methods include the following:

Coaching Method: This technique can be used when the participants are allowed to ask any system-related questions of an expert tester who will answer to the best of his ability.
The purpose of this technique is to discover the information needs of users in order to provide better *training* and *documentation*, as well as possibly redesign the interface to avoid the need for the questions.

Co-discovery Learning: During a usability test, two test users attempt to perform tasks together while being observed. They are to help each other in the same manner as they would if they were working together to accomplish a common goal using the product. They are encouraged to explain what they are thinking about while working on the tasks. Compared to thinking-aloud protocol, this technique makes it more natural for the test users to verbalize their thoughts during the test.

Performance Measurement: This technique is used to obtain quantitative data about test participants' performance when they perform the tasks during usability test. This will generally prohibit an interaction between the participant and the tester during the test that will affect the quantitative performance data. It should be conducted in a formal usability laboratory so that the data can be collected accurately and possible unexpected interference is minimized. Quantitative data is most useful in doing comparative testing, or testing

[3] Computer-aided software engineering (CASE) is the use of software tools to assist in the development and maintenance of software

against predefined benchmarks. To obtain dependable results, at least 5 user participants are needed, while 8 or more participants would be more desirable.

Question-asking Protocol: During a usability test, besides letting the test users to verbalize their thoughts as in the thinking aloud protocol, the testers prompt them by asking direct questions about the product, in order to understand their mental model of the system and the tasks, and where they have trouble in understanding and using the system.

Remote Testing: Remote usability testing is used when testers are separated in space and/or time from the participants. This means that the testers cannot observe the testing process directly and that the participants are usually not in a formal usability laboratory. There are different types of remote testing. One is same-time but different-place, where the tester can observe the test user's screen through computer network, and may be able to hear what the test user says during the test through speaker telephone. Another is different-time different-place testing such as journal sessions, where the user's test session is guided and logged through a special piece of software as well as additional code added to the system being tested.

Retrospective Testing: If a videotape has been made of a usability test session, the tester(s) can collect more information by reviewing the videotape together with the user participants and asking them questions regarding their behavior during the test. So this technique should be used along with other techniques, especially those where the interaction between the testers and the participants is restricted. But using this technique means that each test takes at least twice as long. Another obvious requirement for using this technique is that the user's interaction with the computer needs to be recorded and replayed.

Shadowing Method: During a usability test, the tester has an expert user sit next to him and explain the test user's behavior to the tester. This technique is used when it's not appropriate for the test user to think aloud or talk to the tester while working on the tasks.

Teaching Method: During a usability test, let the test users interact with the system first, so that they get familiar with it and acquire some expertise in accomplishing tasks using the system. Then introduce a naive user to each test user. The Novice users are briefed by the tester to limit their active participation and not to become an active problem-solver. Each test user is asked to explain to the novice how the system works and demonstrate to him a set of pre-determined tasks.

Thinking Aloud Protocol: During the course of a usability test, the test users are asked to verbalize their thoughts, feelings, and opinions while interacting with the system. It is very useful in capturing a wide range of cognitive activities. Two variations of thinking-aloud protocol technique are:

Critical response This requires the user to be vocal only during the execution of certain predetermined subtasks.

Periodic report This is used when the task is complex and makes it difficult for users to think aloud while performing the task at the same time. The user, therefore, verbalizes at predetermined intervals of time and describes what he is currently trying to achieve. The length of the interval depends upon the complexity of the task.

Usability Inspection

In Usability Inspection approach, usability specialists – and sometimes software developers, users and other professionals – examine usability-related aspects of a user interface. Commonly used Inspection methods are:

Cognitive Walkthroughs: Cognitive walkthrough involves one or a group of evaluators inspecting a user interface by going through a set of tasks and evaluate its understandability and ease of learning. The user interface is often presented in the form of a paper mock-up or a working prototype, but it can also be a fully developed interface. The input to the walkthrough also include the user profile, especially the users' knowledge of the task domain and of the interface, and the task cases.

Feature Inspection: This inspection technique focuses on the feature set of a product. The inspectors are usually given use cases with the end result to be obtained from the use of the product. Each feature is analyzed for its availability, understandability, and other aspects of usability. For example, a common user scenario for the use of a word processor is to produce a letter. The features that would be used include entering text, formatting text, spell-checking, saving the text to a file, and printing the letter. Each set of features used to produce the required output (a letter) is analyzed for its availability, understandability, and general usefulness.

Heuristic Evaluation: A heuristic is a guideline or general principle or rule of thumb that can guide a design decision or be used to critique a decision that has already been made. Heuristic evaluation, developed by Jakob Nielsen and Rolf Molich, is a method for structuring the critique of a system using a set of relatively simple and general heuristics.
The general idea behind heuristic evaluation is that several evaluators independently evaluate a system to come up with potential usability problems. It is important that there be several of these evaluators and that the evaluations be done independently. Nielsen's experience indicates that around 5 evaluators usually results in about 75% of the overall usability problems being discovered.

Pluralistic Walkthrough: A group of users, developers, and human factors engineers meet together to step through a set of tasks, discussing and evaluating the usability of a system. Group walkthroughs have the advantage of providing a diverse range of skills and perspectives to bear on usability problems. As with any inspection, the more people looking for problems, the higher the probability of finding problems. Also, the interaction between the team during the walkthrough helps to resolve usability issues faster.

Brainstorming: A method for generating ideas, intended to inspire the free-flowing sharing of thoughts of an individual or a group of people, typically while withholding criticism in order to promote uninhibited thinking.

Inquiry

Here, usability evaluators obtain information about users' likes, dislikes, needs, and understanding of the system prototype by talking to them, observing them using the system prototype in real work (not for the purpose of usability testing), or letting them answer questions verbally or in written form. Inquiry methods include:

Field Observation: Human factors engineers go to representative users’ workplace and observe them work, to understand how the users are using the system to accomplish their tasks and what kind of mental model the users have about the system.

Focus Groups: This is a data collecting technique where about 6 to 9 users are brought together to discuss issues relating to the system. A human factors engineer plays the role of a moderator, who needs to prepare the list of issues to be discussed beforehand and seek to gather the needed information from the discussion. This can capture spontaneous user reactions and ideas that evolve in the dynamic group process.

Interviews: In this technique, human factors engineers formulate questions about the product based on the kind of issues of interest. Then they interview representative users to ask them these questions in order to gather information desired. It is good at obtaining detailed information as well as information that can only be obtained from the interactive process between the interviewer and the user.
In an evaluation interview, an interviewer reads the questions to the user, the user replies verbally, and the interviewer records those responses. The methods of interviewing include unstructured interviewing and structured interviewing.
Unstructured interviewing methods are used during the earlier stages of usability evaluation. The objective of the investigator at this stage is to gather as much information as possible concerning the user's experience. The interviewer does not have a well-defined agenda and is not concerned with any specific aspects of the system. The primary objective is to obtain information on procedures adopted by users and on their expectations of the system.
Structured interviewing has a specific, predetermined agenda with specific questions to guide and direct the interview. Structured interviewing is more of an interrogation than unstructured interviewing, which is closer to a conversation.

Logging Actual Use: Logging involves having the computer automatically collect statistics about the detailed use of the system. It is useful because it shows how users perform their actual work and because it is easy to automatically collect data from a large number of users working under different circumstances. Typically, an interface log will contain statistics about the frequency with which each user has used each feature in the program and the frequency with which various events of interest (such as error messages) have occurred. Statistics showing the frequency of use of commands and other system features can be used to optimize frequently used features and to identify the features that are rarely used or not used. Statistics showing the frequency of various error situations and the use of online help can be used to improve the usability of future releases of the system by redesigning the features causing the most errors and most access for online help.

Proactive Field Study: Before designing a system, in order to understand the users, their tasks, and their working environment, human factors engineers go to representative users' workplace and talk to them, observe them work, and ask them questions, to understand the user characteristics, the work flow, the system features they need, etc. This technique should be used during the requirement or early design stage of software development. This should be the first step of usability work for a project.

Questionnaires: They are widely used instruments for usability evaluation, but their correct construction is often a complex task since several previous administrations are required to obtain a fine-tuned version of the questionnaire.

2.3 Salary and Personnel Management System

Salary and Personnel Management System Systems encompass:

1. Payroll
2. Work Time
3. Benefits Administration
4. HR management Information system

The **Payroll module** automates the pay process by gathering data on employee time and attendance, calculating various deductions and taxes, and generating periodic pay checks and employee tax reports. Data is generally fed from the human resources and time keeping modules to calculate automatic deposit and manual check writing capabilities. This module can encompass all employee-related transactions as well as integrate with existing financial management systems.

The process to carry out the payment of the payroll is complex because we have to take in account different factors as the salary base, the social security, the loans …. Each one of these aspects are contemplated in the system, which will help significantly to make the process easier.

Other advantages of this system is that permits the integration with technological tools as Internet, helps to resolve particularities of the client; the modules of finances and costs are integrated real-time offering more control, flexibility and reduction of costs by optimization of resources.

It provides a flexible solution that allows to automate the process of payroll in any company of any size, controlling the information of the process of payroll with confidence, maximizing the profit value of the business and bearing legal changes.

The **Work Time** gathers standardized time and work related efforts. The most advanced modules provide broad flexibility in data collection methods, labor distribution capabilities and data analysis features. Cost analysis and efficiency metrics are the primary functions.

The **Benefits Administration module** provides a system for organizations to administer and track employee participation in benefits programs. These typically encompass, insurance, compensation, profit sharing and retirement.

The **HR management module** is a component covering many other HR aspects from application to retirement. The system records basic demographic and address data, selection, training and development, capabilities and skills management, compensation planning records and other related activities. Leading edge systems provide the ability to "read" applications and enter relevant data to applicable database fields, notify employers and provide position management and position control. Human resource management function involves the recruitment, placement, evaluation, compensation and development of the employees of an organization. Initially, businesses used computer based information system to:

- Produce pay checks and payroll report.
- Maintain personnel records.
- Pursue Talent Management.

2.4 Web-Based Application

In software engineering (Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software) , a Web application is an application that is accessed via Web over a network such as the Internet or an intranet.

It is used when the site is for “doing” things rather than finding information. Web applications are popular due to the ubiquity of a client, sometimes called a thin client. The ability to update and maintain Web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity. A web-based application is an application that could just as well be a normal application. It now just runs in a web browser.

In earlier types of client-server computing, each application had its own client program which served as its user interface and had to be separately installed on each user's personal computer. An upgrade to the server part of the application would typically require an upgrade to the clients installed on each user workstation, adding to the support cost and decreasing productivity.

In contrast, Web applications dynamically generate a series of Web documents in a standard format supported by common browsers such as HTML/XHTML. Client-side scripting in a standard language such as JavaScript is commonly included to add dynamic elements to the user interface. Generally, each individual Web page is delivered to the client as a static document, but the sequence of pages can provide an interactive experience, as user input is returned through Web form elements embedded in the page markup. During the session, the Web browser interprets and displays the pages, and acts as the universal client for any Web application.

Chapter 3. *The state of the art*

3.1 Introduction

3.2 Case Description Personeel en Zo BV:

Personeel en Zo BV is a business specialised in the field of salary administration. They provide salary administration support and payslip calculations for third parties with the main focus on medium and small size enterprises. In the last couple of years Personeel en Zo BV has undergone a growth of on average 40% in turnover and expects to continue this growth the coming year as well.

The service of Personeel en Zo BV consists of providing a web based product called PersoneelOnline!, through which their customers are able to provide (online, at every moment desired) Personeel en Zo BV with the necessary input data for the salary administration like (new) employees, hours worked, salary, date of start and end dates of employment et cetera, after which Personeel en Zo BV takes care of the wage and tax calculations, creates company overviews like payment lists and provides a legal check-up of the administration. The output is (online) made available for the client and after client approval, the necessary data is sent to the tax authority and the client can book the data in his company administration. The total provided services vary per client from just making the calculations, providing administrative advice/support to taking care of the wage payments and tax reports.

For the actual calculations for the salary administration Personeel en Zo BV currently uses an (offline) software tool of an other company, several fulltime employed salary administrators and an online application which through the use of a server enables the clients to write the administration data into an Access database. The salary administrator uses the offline software tool to take the necessary data from the database to make the wage, tax calculations and company overviews while providing the control on the administration and if necessary warn about for example input mistakes made by the client. The (new) data is written back into the database and the output is made available for the client.

In the current system the clients and the salary administrators are not able to simultaneously dispose over all the administration data because after a client has provided the (monthly) input data it will have to be generated by one of the salary administrators of Personeel en Zo BV using the (offline) software tool before it can be made available online for the client to approve. This means that after the calculation and control by the salary administrator there might still be some errors in the output as a result of mistakes made during input by the client. One can think of faulty payslips because of a wrong number of days , hours worked by the employee or a false start date of a new employee filled in through PersoneelOnline!, which the salary administrator can not identify. This results in double work and time lost for both parties, because some calculations have to be redone and send to the client for approval more than once before the payments and tax reports can finally be done.

Because in the current system intervention of the salary administrator is needed in order to generate the payslips, tax reports and company overviews the control on the salary administration is done during the calculation phase. But it is also possible to check and if necessary make corrections after the tax reports have been send to the tax authority. As a result of the control over the salary administration being done during the calculation phase, and as most clients do their salary administration and payments around the 20th of the month, there is a huge spike in the amount of workload for Personeel en Zo BV every month around the 20th. If calculation, overview and tax report generation could be separated from control, Personeel en Zo BV could spread the workload more evenly over the month and create an efficiency gain in their work process.

Also in the current system Personeel en Zo BV is heavily dependent on the (offline) software tool which they do not own and for which they have to pay to use. This makes Personeel en Zo BV as a company dependent on another company.

The strategic importance of owning a software tool of their own, together with the possible efficiency gain lead Personeel en Zo BV to the decision to develop their own (completely) web based salary administration tool. And I'll be part of the team, taking care of the usability and presentation layer for this new salary administration tool.

3.3 About the technical Design

3.3.1 Web parts

A web part is an ASP.NET server control which is added to a Web Part Zone on Web Part Pages by users at run time. Web Parts are an integrated set of controls for creating Web sites that enable end users to modify the content, appearance, and behavior of Web pages directly from a browser. The Web Parts control set is a group of components that work together to enable developers to create Web pages where users can modify the user interface (UI) directly from a browser. The UI controls all derive from the Part class, and they comprise the primary UI on a Web Parts page.

The Web Parts framework contains a set of controls that lets you organize a portal page in a way that allows users of the portal to customize the appearance, content, and behavior of its contents directly from a web browser. The changes are then saved for the user and recalled for subsequent visits

Applications can be integrated within a third party website by the placement of a small snippet of code. The code brings in 'live' content – advertisements, links, images – from a third party site without the web site owner having to update.

3.3.2 Pop-Ups

A pop-up is a graphical user interface (GUI) display area, usually a small window, that suddenly appears ("pops up") in the foreground of the visual interface. Pop-ups can be initiated by a single or double mouse click or rollover, and also possibly by voice command or can simply be timed to occur. A pop-up window must be smaller than the background window or interface; otherwise, it's a replacement interface.

We can define two different kinds of pop ups. The *normal* pop ups and the modal pop ups.

Normal Pop-Up

A normal pop-up is just a child window for a parent application. You can always interact with the rest of the page.
They are very easy to implement as you only have to use the *Open Method* which opens a new window and loads the document specified by a given URL:

```
NewWindow = window.open( [strURL] [, strName] [, strFeatures] [, boolReplace])
```

Where:
strURL: String that specifies the URL of the document to display.

strName: String that specifies the name of the window.

strFeatures: String that contains a list of items separated by commas. The following features are supported: *channelmode, directories, fullscreen, height, leftlocation, menubar, resizable, scrollbars, status, titlebar, toolbar, top, width and zoominherit*.

boolReplace: Boolean that specifies whether the sURL creates a new entry or replaces the current entry in the window's history list. This parameter only takes effect if the sURL is loaded into the same window.

Modal Pop-Up

The Ajax Toolkit Modal Popup extender allows a page to display content to the user in a "modal" manner which prevents the user from interacting with the rest of the page. The modal content can be any hierarchy of controls and is displayed above a background that can have a custom style applied to it. When displayed, only the modal content can be interacted with; clicking on the rest of the page does nothing. When the user is done interacting with the modal content, a click of an OK/Cancel control dismisses the modal content and optionally runs custom script. The custom script will typically be used to apply whatever changes were made while the modal mode was active. If a postback is required, simply allow the OK/Cancel control to postback and the page to re-render.

You can also absolutely position a modal popup by setting the X and Y properties. By default it is centered on the page, however if just X or Y is specified then it is centered vertically or horizontally.

Chapter 4. *Task Analysis*

4.1 Introduction

In the introduction you give the goal of this chapter and the way it is built up.

This chapter will try to explain something about task analysis in general and the results of the task analysis in this project. First of all we start with a small introduction to the task analysis method. Next, some explanation is made about the techniques and methods that we used. Then we tell something about the analysis that we made and some results will be shown.
The goal of the task analysis group was to obtain a model of the current situation in the project-domain and to construct a model of the desired situation (3.3). To retrieve the relevant information used to construct the models, different techniques are available and used, these will be explained in 3.2.

4.2 Techniques used

In this paragraph you tell something about the available techniques for a task analysis. Keep it short, this is not a textbook on task analysis! Explain why you have chosen for a particular technique and why you have considered another technique but decided not to use it.

4.3 The task model: analysis of current and future situation

Root Concept

Before going into the field, I am going to develop an understanding of project's high-level goals. In SBD we document this understanding as a root concept. See Table Root Concept.

Statement of project vision and rationale
There already exists a working system used by many companies. Personeel online is the online solution for a company staff and salary administration. It is a completely online staff system but it needs some improvements to increase its functionalities and usability to make better user's experience. A new application that will replace the first one and which includes most of the required functionality is being developed. However its interface is not ready to handle the final users yet.
The technical part is almost developed but without taking in account the usability of its interface. This project intends to develop the interface for a web-based management application following a user-centered design approach and usability evaluations.

Component	*Contributions to the Root Concept*
High-level vision	*PersoneelOnline* is the online solution for a company staff and salary administration. It is a completely online staff system.
Basic rationale	A new application that will replace the first one and which includes most of the required functionality is being developed. However its interface is not ready to handle the final users yet. The technical part is almost developed but without taking in account the usability of its interface. This project develops the interface for a web-based management application following a user-centered design approach and usability evaluations.
Stakeholder group	
Payroll Specialist at Personeel en Zo	• *Background*: The Payroll Specialists at *Personeel en Zo* are mainly working with the Cobra system. • *Expectations:* They will use the system to make all the pay slip calculations, set everything about a new company, establish the

	social security settings and other things as make corrections and proformas.
Payroll Specialist of clients	• *Background*: The Payroll Specialists of clients are accountants with Know-how of payroll. They are only working with the PZ Web system. And they have to handle with multiple clients. • *Expectations:* They will use the system to make all the pay slip calculations.
Normal User	• *Background*: The normal users are payroll administrators but with no knowledge of payroll. They are working with PZ Web, using entire system. And they use to handle 1 to 3 companies with a maximum of 50 employees per company. • *Expectations:* They want to use the system for enter new employees, enter looncomponents, receive signals and they like to work with wizards.
Payroll Specialist of one client	• *Background*: This group is for the ones who are payroll specialist of one client but with more than 50 employees. They just introduce information to build the pay slips. They never get closer to a company and don't know what is going on them. We can resume the pay slip generation into three steps: Introduce the information. Make a request to get the pay slip processed and print the pay slip. They are also the ones where people go to complain about mistakes on their pay slips.
Human Resource Specialist of clients	• *Background*: They are Human Resource specialist. They have to handle with only one company with more than 50 employees.
Employee	• *Background*: This group is formed by the employees of a company. They don't necessary have to have know-how about payroll or HR. • *Expectations:* They will use the system for seeing the pay slips, add signals, change personal information and enter some other information as worked hours.
Debtor	• *Background*: A debtor is everyone who owes money. • *Expectations:* They will use the system to see the invoices, see where the invoices are going; see an invoice history and get invoices by post between others.
Starting assumptions	The new system is based in the current Cobra, PZ Web and PersoneelOnline desktop applications. The implementation of the system engine is very advanced. I will be working with and supervised by the current developer.

Table: Root Concept

Analysis of current Practice

In order to get a feedback about the system from the real users I have developed different guides for each stakeholder. With these guides I want to understand the kind of users they are, the tasks they work with, and overall, which features of the system they are using, they like the most and which ones they will like to improve, change or add in the new system.

Most of the questions are quite the same as they will be working in the future with the same system. But this is also very useful so I can see the different answers that different kinds of user give to the same question.

The interviews can be thought of as a "conversation with a purpose" (Kahn and Cannell, 1957). I have thought that the best will be write the interviews with open-ended questions as my goal is to gain first impressions about how users work with the current system and what users expect for the new one. These interviews are semi-structured interviews. I will have a basic script for guidance, so that the same topics are covered with each interviewee. I will start with preplanned questions and then I will probe the interviewee to say more until no new information was forthcoming.

See Appendix A.1

Summarizing the Field Data

Interviews outcomes

The observations and interviews for each stakeholder group are organized into stakeholders profiles . These profiles summarize the general characteristics of each group, and are based on the observations and interviews shown above.

	Payroll Specialist at Personeel en Zo	Payroll Specialist of clients	Normal User	Payroll Specialist of one client	Human Resource Specialist of clients	Employee
Open-ended questions	- Payroll administrator. - Currently working with Cobra.	- Payroll specialist. - Currently working with PZ web.	- Grew up with computers and use it every day. - Use the computer to get all kinds of information and to communicate.	- Book keepers. - Office applications.	- They work as users and administrators.	
Talking about Cobra system	- They use it: * To make payslip's calculations, * set new companies, * establish the social security settings, * make corrections, proformas ... - They would like it to be more friendly and trustable.					
Talking about PZ Web system	- They know how PZ Web works. - They have to process and answer the client requests.	- They use this application to provide the payroll administration to their customers.	- They use it to provide their employees the pay slips.	- They use it for processing pay slips.	- They find it very easy to work with and friendly.	
Conduct task analysis and prioritize tasks	- They use the system to: * Implement new clients. * Processing the salary. * Make changes in the settings. * Make analysis. * Download reports in Excel and make overviews. * Make offers for a company. * Put back the payroll period. - Critical tasks are: * The deadline of the payroll administration. * Speed of the system. - The most important for them is to process payslips.	- The feature on the system they use the most is managing looncomponents. - Critical tasks are: * Immediately payroll process. * Make less mistakes. - The most important for them is the payroll processing system and to be able to make proformas.	- They use the system: * To have some insight to all the different employees information. * As a way of communication with their employees. - Critical tasks are: * To pay the employees on time. * To give to the employees a correct payslip. * Everything correct to the tax company. * They need to work in a quick and handy way. - Features they use the most are: * Quick possibility to insert	- They use the system to: * Enter payroll processing mutations. * Fill in loon components. * Add and dismiss employees. - Critical task is the Standard Register. - Very important are the tasks of entering looncomponents and making fast calculations.	- They use the system to: * Make contracts. * Make the EDM. * Salary changes. * Creating an employee. * The Verlof and Verzuim tabs.	

			the monthly mutations. * To ask for a salary calculation. * To see the monthly results. - Tasks to accomplish easily with few errors: * Hanging a result of a meeting or a contract. * Vacation questionnaires.			
Expectations regarding the new Personeel Online	- Save button always visible. - Don't have to type the same information several times. - Group tasks. - To solve the problem in the updates. - To have a list in which you can see when a company is already implemented. - To have a Data Management System .	- Be able to create/modify companies/employees. - Be able to make proformas. - Process the pay slips themselves. - Give their clients access to the system and the feeling that it is their system. - To see what a customer has done on the system. - Know the result of what they are doing immediately. - Manage signals. - Access from every location at every time. - Easy to use. - Speed in processing.	- Chance of asking the system for support at any time. - Information good organized. - They would hide all the irrelevant information. - To give to their employees his own digital pay slip. - Access from every location at every time. - Speed in processing.	- Save button always visible. - To have the import function. - To be able to make Proformas. - To have an overview about all the changes made by someone else.	- Save button always visible. - Overviews. - To have "Suggestions" and check lists. - Previous run so they can check if everything is going ok. - Employee photo. - Save several things at the same time. - Possibility of checking what you have changed or done. - Improve the management of an out of order employee.	- See/Ask for holidays (Verlof) - See news. - Handbook. - See and fill in some kind of document. - See/change his own personal information. - Hour registration. - See their own pay slip online.

As I said before, the interviews were semi-structured interviews. Because of that maybe some questions were not covered or even answered in any other question as the result of the conversations that came up when talking about each topic.

See Appendix A.2

Task Analysis
Tasks carried out by stakeholders, which were observed and discussed before with the users and the client.

See Appendix A.3

4.4 Evaluation

Explain shortly why it is important to evaluate the task model.

4.4.1 Technique used

Describe why you have chosen a particular evaluation technique (for example scenarios) and how you have used it. The details are given in Appendix D.

4.4.2 Conclusions

Give the conclusion from the evaluation of the task model.

Chapter 5. *Prototyping*

5.1 Introduction

5.2 First Prototype

When I started with this project some studies about how the interface should be were already made. They were working on what we can call a *Prototype Cero*, made on Excel files in which each file was a different application page or view. It was a really good starting point for me as I could get a brief idea about what the aim of this application was and it was very helpful when building the First Prototype.
However it was very difficult to make a quick and easy revision and layout of pages. And I had to find an easy way to demonstrate the prototype to clients and team members. At this point is when the First Prototype came up.

5.2.1 Characteristics

Tool used

This First Prototype is completely based on the Excel files they used to have to handle everything about the interface, navigation and presentation layer of the new web based salary application they were developing. Looking for prototyping tools I found visual or text based HTML tools and diagramming tools.
In June 2002 GUUUI [4]conducted a survey on web prototyping tools usage. The purpose of the survey was to find out what tools are used for prototyping, what requirements interaction designers have for their tools, and how happy they are with the tools they are using. Some key findings on this survey are:

- There is currently no distinct consensus on what is the best tool to use for web prototyping.
- HTML and diagramming tools are the most commonly type of tools used for web prototyping.
- Macromedia Dreamweaver and Microsoft Visio are the most widely used tools.
- Respondents feel that the tools they use fall short on some high priority criteria, such as laying out and making changes to pages.
- Few seem to be perfectly happy with the tools they are using though users of HTML tools are strikingly more content than users of diagramming tools.

I found very nice starting with HTML because:

- It's quick and easy to layout pages. It will be very easy and quick to navigate through the application.
- It's quick and easy to make changes to pages.
- It's quick and easy to do site wide revisions to the prototype.
- It's able to simulate basic functionality.
- It's easy to demonstrate the prototype to clients and team members.
- It's suitable for carrying out usability testing.

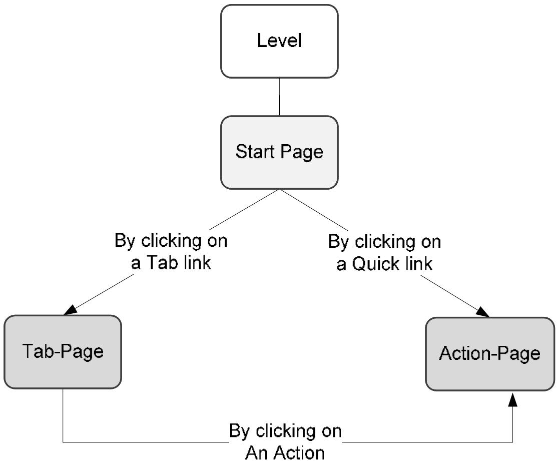
[4] GUUUI is a site for people engaged in the various fields of making the web a better experience for the users. You can find the survey outcomes I am talking about following this link: http://www.guuui.com/issues/01_03_02.php

- It produces good looking prototype.
- It fits in with existing development tools. This is very important as in the end I will have to develop the application with ASP.NET.
- It's easy to learn and easy to use, so I wouldn't spend too much time on building the prototype having more time to evaluate it.

I started this prototype already in Microsoft Visual Studio 2008, working only with the HTML part, so it meant to me a first approach to this development system.

Navigation

In this new web based salary management application we will be able to distinguish between two different levels: Company and Employee level.
Based on the existing PersoneelOnline! used by Personeel en Zo BV, we are going to have a left menu which will let us to switch between both Company and Employee level by clicking on an employee or company name. Because it is a too complex functionality to develop on a First Prototype I added two button for this purpose.
In both Company and Employee level the following navigation structure is applied:



Start Page
We have a Start Page per Level. This starting page consist of:

- Employee/Company Photo/Logo
- Employee/Company Information
- Several links to Tab-Pages and Action-Pages

Tab-Page
We can group the different kinds of information in different Tab-Pages. We have the following Tab-Pages:

Employee Level

- *Personalia (Personal Information)*
- *Dienstverbad (Employee information)*
- *Loonelementen (Remunerations)*
- *Loonheffing (Levy)*
- *Personeelsdossier (Personal Documents)*
- *Loonstrook (Pay slips)*

Company Level

- *Salaris verwerking (Salary processing)*
- *Personeels zaken (Staff Management)*
- *Bedrijfsinstellingen (Company settings)*

Some of these Tab-Pages have buttons or links to Action-Pages

Action-Page
Action-Page is the page in which we are able to carry out a certain action as editing or including new information. We can reach these pages since the start page through a quick link or since a Tab-Page by clicking on a button or a link.

Patterns

As we will see in the next section, *Evaluation,* the patterns are not very well established in this prototype.
Most of the pages consist of a editable form or table, already filled in with the default or current values. If you want to change the current information or even add new information you have to delete the information on the form, write the new one and save.
The *Save Button* usually is at the bottom of the page.
There is also a table with a historic in most of the pages, with all the past information on it.

See Appendix B.1

5.2.2 Evaluation

Technique used

For this First Prototype evaluation I used the *Brainstorming.* The client, the developer, an employee working for this particular project and I (presentation layer developer) met together to discuss and evaluate this prototype.
I found it as the best way of evaluating this prototype as there were a lot of features to talk about and new ideas to discuss. As it is a prototype completely built in HTML it doesn't contain functionality enough to use any other type of evaluation method in which a group of tasks is needed to go through them.

Evaluation

Current situation

- In the first prototype we are missing patterns for the different panels: It make the site a bit messy and not uniform.

- The way to add new information is not the best. Actually we can see two ways:

  - The current information is shown in a table. If you want to add new information you have to press the "new button" and we are led to another panel. In this new panel we can see a form already filled in with the current information. And we have to change this information in the form in order to write the new one.

  - The current information is in a form and we have to change this information to add the new one.

With this way of working we have the next **problems**:

- As we mentioned before there is a problem with the patterns: we have two different ways of showing the current information.

- The current information is shown in a form:
  - This could be a bit confusing for the user. He don't know if the values on the form are default values; what happen if he changes these values? Is a new information created or do they remove the information forever?

  - The forms take a lot of space in the screen and sometimes you have to scroll down to see all the information. Due to this, the "save button" is not always visible and the user can forget about saving changes.

    - Sometimes is not necessary to show so much information and several text boxes in the form are empty.

- The historic form is always visible when adding new information:
    - Why to show always information that the user hasn't asked for?.
    - It make the system slower as we have to load the historic every time.

Discussing all these problems in a brainstorming way we obtained the following **ideas** to improve the First prototype and build the Second one:

- We have first, to find the different panel patterns to make the site more visual uniform and intuitive.
- We think it is better to remove all the forms showing current information in a way that:
    - The current information is always fixed. Not in forms anymore. This new model supposes a big change in the First prototype. As the current information is not in forms anymore, it doesn't take too much space in the pages. The idea of building a start page in which we can see an overview of all the current information is born at this point.
    - We will have a page with all the possible overviews in a dashboard. We will have two buttons to either edit/create or show details. And this page will be configurable as the user will be able to close or add dash-lets in order to make it more personal and not see information that he doesn't want to see.
    - If we edit/create information a pop-up appears that contains a form:
        - The form can have more information than the one showed in the overview of the current information.
        - The form will appear already filled in with default values or/and the current information. Most of the times we don't need to change all the information but just some fields, so why should we fill all the fields in again?
        - We will only have two ways of closing the pop-up. We will not be able to click anywhere else outside the pop-up; so we will have to click either Save or Cancel to close it. There will be two buttons:
            - Save button, to save the changes and close the pop-up.
            - Cancel button, to cancel the operation and close the pop-up without saving.
    - The details button led us to another page that shows the details of the current information. Inside this page we will have a button to go to the historic panel which will appear as a pop-up. In this way we only see the historic when we want to. In this page we will also have the chance of editing/creating information the same way as mentioned above.

Conclusions

- We have found patterns.
- We save space so we don't need too much tabs.
- With the four modules mentioned above we organize the information in a better way.

Before Evaluation	After Evaluation
– **Missing patterns.** – **No good way to add new information.** – **Historic always visible.** – **Difficult navigation.**	– Patterns found. – *Current* information fixed. – Customizable Dashboard as starting page with overviews. – Pop-ups to edit or create information. – Much more easy to navigate.

5.3 Second Prototype

5.3.1 Characteristics

Tool used

In this Second Prototype I keep on working with HTML for almost everything but some JavaScript functions. I also use some very simple ASP.NET functionalities as *Master* and *Content Pages.*
Most of features will be represented on this prototype although the functionality wouldn’t be there. There will be a lot of possible clickable buttons and links but they will be just HTML images with no functionality at all. This way I try to represent what the user will be able to do on the application, but as still it is not the final prototype, I cannot go too much into technical details as it would be a waste of time if we decide to change something.

Navigation

With the Second Prototype we have a completely new way of thinking. The diagram *Second Prototype Navigation Tree* shows the navigation tree we came up after the First Prototype evaluation.

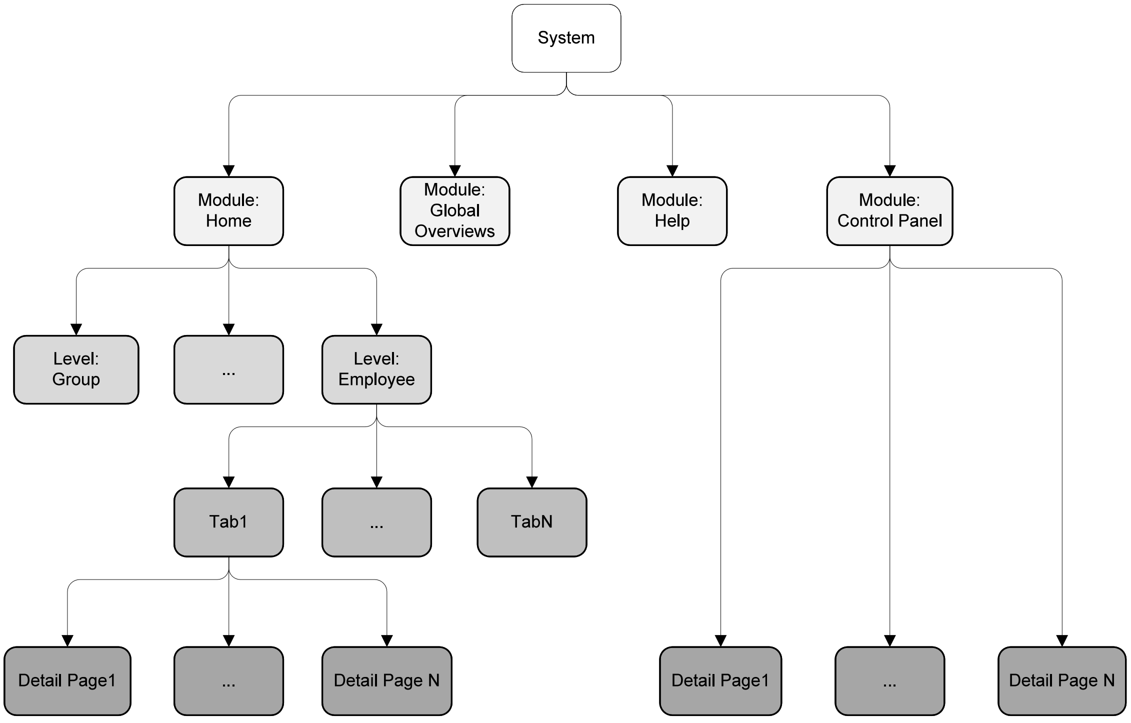


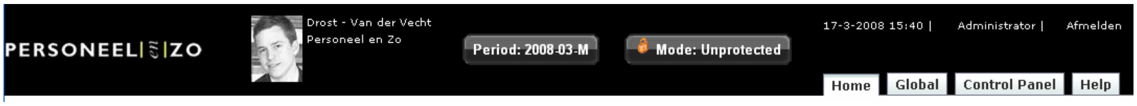
Diagram: Second Prototype Navigation Tree

The system is now divided into four different modules: *Home*, *Global*, *Control Panel* and *Help*[5].

See *Navigation Diagram* at Appendix B.2.

Header

The header is going to place both static and dynamic information. Static information because we need to show some elements as the application logo, date and time, user name and the logout link. And dynamic because we need some elements that could change depending on the user level or the current selection: employee photo, company logo, employee and company name, period selector and system mode.
After thinking what will be the best layout the following header came up.

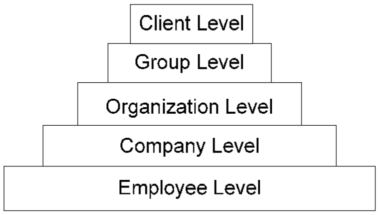


Application logo:
Employee photo:
Company logo:
Employee name:
Company name:
Period selector:
System mode:
Date and time:
User name:
Logout link:

Home

This module represents the application itself, where all the calculations will be done.

It is divided into four levels: *Group*, *Organization*, *Company* and *Employee Level.* Each user will belong to a level depending on his user type. And will be able to manage all the levels below his own level including his own level. For this we have to take in account the following hierarchy:



For example, if an organization user gets into this application, he will be able to manage his own organization, all the companies that belong to his organization, and all the employees belonging to the companies of that organization. But he could never manage or even see anything at Group Level.

Left Menu: In this new prototype the left menu I talk about in the first prototype is included. And it appears when the user is on the *Home Module.* This left menu will contain tree panels:

[5] These are the names given to the different modules just to call them in some way. But maybe they won't be the final ones. It is not so important at this point to worry about these small details.

- *Organization* panel. It will show all the organizations a group user is allowed to see. In case it was a organization user, it will show the organization the user belongs to. On Company and Employee Level this panel won't even appear on the system.

- *Company* panel. It will show all the companies an organization user is allowed to see. In case it was a company user, only the company the user belongs to will be shown. On Employee Level this panel won't even appear on the system.

- *Employee* panel. It will show all the employees a company user is allowed to see. In case it was a employee user, his own information will be the only one shown in this panel.

The user will be able to collapse the panels if he wants to. Each panel will have a text box to look for the names in an easy way. And the user will be able to sort each panel list by name or number. They will have tree buttons: *New*, *Management* and *Filter* button:

- *New* button. This button is a quick link to create a new Organization/Company/Employee.

- *Management* button. Here we have another quick link this time to open the management page for the Organizations/Companies/Employees.

- *Filter* button. By clicking on this button a small pop-up with a check list appears and the user will be able to select which filters does he want to apply to filter the panel list. An examples of filter could be, filter by department.

This left menu will be mostly used to switch between levels. By clicking on an organization, company or employee name the user will see information regarding that selection.

As we can observe, we have almost one panel per level, although client and group panels are not going to be include.
Client panel won't be there because we don't need to switch between clients; there is always only one, and we are not going to have any information to show about a client.
Group panel. At first, it was thought to show all the groups a client user is allowed to see. In case it was a group user, it will show the group the user belongs to. And on Organization, Company and Employee Level this panel won't even appear on the system. But after thinking more deeply about if it was really necessary, we get to the conclusion that it was not. We are not going to have overviews per group, so we don't need this panel on the left menu.

We still have to think more deeply about how the client and group levels are going to be.

Tabs per level: Each level is going to have at the most three different tabs. This is a very important improvement on the application user interface, as the one on the systems this one is based on, used to have several tabs what made the site very difficult to navigate.

For Group and Organization Level, the tabs are not defined yet.

On Company Level there will be the three following tabs:

- *Company Information* Tab Page: This tab page will consist of a dashboard with several dash-lets that the user will be able to move around or even close in order to customize his own application interface. In this tab page all the dash-lets will have to do with information related to the company selected. Information as company contact person, signals, salary details…

- *Settings* Tab Page: In the same way as the *Company Information Tab Page* it will consist of another dashboard but this time its dash-lets will contain information about the company settings: default schedule, salary settings …

- *Overviews* Tab Page: Here all the company overviews will be shown using different kinds of chart representations as pie, column or area charts. In this prototype we are not going into details on this tab.

There will be three tabs on Employee Level as well:

- *Personal Information* Tab Page: It will be like the *Company Information Tab Page* but all the information will now have to do with the employee selected: Employee schedule, absences,.. if he has a lease car..

- *Pay slips* Tab Page: The idea of this tab is that the user could see or even print his pay slips.

- *Overviews* Tab Page: Employee overviews will be shown.

Dash-lets and Detail Pages: We could say that the dash-lets are the main feature on this Second Prototype. As we discuss during the First Prototype evaluation, on this prototype the current information is not going to be shown in editable forms or tables anymore, but is going to be fixed, so we need much less space to show the same information. Every dash-let contains an overview with the most important information. And this information is shown in detail on that dash-let's detail page.
A problem was found on the detail pages: we have quite a lot of information per detail page. When we want to change some information we don't want to save it all because it is too much. For this reason we made different areas within one detail page. And to keep the consistence with the dash-lets, they are also divided in different areas where each dash-let area contains a detail page area overview.
Usually every dash-let is going to have a default action when clicking over the dash-let's area. And this action will be the same as the default action on its detail page area (usually edit).
As in the dash-lets we are only going to have that default action, is in the detail pages where all the functionality is going to take place.

Global Overviews

This module will give us the possibility of having different types of overviews for the organizations, companies or employees without having to select one in particular.
For this prototype we haven't gone into details.

Control Panel

Here we will have all the different functions and information regarding settings and system configuration.
The information is organized in three columns.
The first column has to do with salary settings: *SVW Settings*, *Sector*, *Risk Group*, *CAO Salary* table, *Loonmodel*, *Sattelite Data*, *Personnel Options* and *Engene Log.* The second one has to do with uses roles, profiles and accounts and the third column is for clients, groups, organizations, companies and employees management.

Help

Here is where the help functionality should be place, but we didn't think about this module for the moment. We just thought it should be here.

Patterns

A very important task to be developed in this prototype was to find patterns for the pages, so the application was more uniform and easy to learn and navigate.

Four different patterns for the details pages and three for the pop-ups were found. See Appendix B.2

Talking about the pop-ups: for this application, we want to use modal pop-ups in which when a user opened a pop-up, he couldn't click anywhere else but inside the pop-up. If we remember the user interview outcomes, a very common problem was that they forget about saving changes quite a lot of times. With these pop-ups, the user will never forget, as to close the pop-up he has to click either Close or Save.

5.3.2 Evaluation

Technique used

Conclusions

We change the way the historic is working: before, the delete button was on the detail page, but it is not working at all because, the delete button will remove the last information that was added into it, and as we are able to change information in the past or in the future, the one shown on the detail page might not be the one that we are willing to remove. This will make the user get confuse. That's way we decided to add the remove functionality on the first row of the historic table. This way we always know which information we are deleting.

5.4 Third Prototype

5.4.1 Characteristics

5.4.2 Evaluation

Technique used

Conclusions

5.4.3 What is missing from the prototype?

You can never make a prototype that is fully functional, since it then wouldn't be a prototype anymore. Describe here which functionality's are not implemented in the prototype.

Chapter 6. *Technical Design*

6.1 Introduction

In the introduction you give the goal of this chapter and the way it is built up.

6.2 Techniques used

State the techniques that are available or that you have looked at to come to the technical design. Explain why you have chosen for a particular technique and specify this technique, if necessary in subparagraphs.

6.2.1 WebParts

6.2.2 Pop-ups

6.2.3 Resizable Left Panel Menu

6.3 Presentation interface

Describe in words the features of your design. How are the interfaces built up, what are their functionalities, how does navigation and dialogue take place, etc. Use the task model to explain your choices. The details of the design will be given in Appendix B.

6.4 Evaluation of the Technical Design

Give a short description of the importance of evaluation of the technical design.

6.4.1 Technique used

Describe why you have chosen a particular evaluation technique (for example scenarios) and how you have used it. Discuss the different functionalities of the prototype separately. The details are given in Appendix D.

6.4.2 Conclusions

Give the conclusion from the evaluation of the technical design.

6.5 Changes made after evaluation

Explain the evaluation techniques that you have used during the technical design process and what changes you have made.

Appendix A. Task Analysis

Appendix A.1 Interviews

Internal Users Interviewing Guide

Here you have some questions I need to know about you as future user of the Personeel Online system. Please, answer as much as you can and take the time you need. I prefer good rather than fast answers.

1. Open-ended questions.

   a. How would you describe yourself as a user? (User characteristics, i.e., age, experience, education, etc.)

   b. What background do you have in using computers?

2. Talking About Cobra system

   a. How long have you been working with this application?

   b. Why do you use this application? (Needs, interests, and goals)

   c. When and where do you access the system?

   d. How do you access the system? (Connection speed, resolution, etc.)

3. Talking About PZ Web system

   a. How long have you been working with this application?

   b. Why do you use this application? (Needs, interests, and goals)

   c. When and where do you access the site?

   d. How do you access the site? (Connection speed, resolution, etc.)

4. Conduct task analysis and prioritize tasks

   a. What do you do on the system? (Tasks, content, features and functionality)

   b. Which tasks are critical to the organizations success on the system?(Criticality)

   c. Which tasks are critical to your success on the system?

   d. Which tasks are most important to you? (Importance)

   e. Which features of the system do you use the most? (Frequency)

   f. Which features are prone to usability issues? (Vulnerability)

5. Measurable usability objectives

    a. Which tasks should you be able to accomplish easily with few errors?(Efficiency)

    b. Which tasks should you be able to finish quickly and efficiently?(Effectiveness)

    c. What level of satisfaction should you have after using the system?(Enjoyability)

    d. What level of satisfaction do you actually have?

6. Expectations, requirements and preferences regarding the new Personeel Online

    a. What is your vision of what the system should do?

    b. Would you add, change or remove something in the new system? (Tasks, content, features or functionality)

    c. Are there any interface style that you prefer or feel very comfortable with? Give some real examples.

7. Any other comments

Thank you for your time !

Manager Interviewing Guide

Here you have some questions I need to know about you as future user of the Personeel Online system. Please, answer as much as you can and take the time you need. I prefer good rather than fast answers.

1. Open-ended questions

    a. How would you describe yourself as a user? (User characteristics, i.e., age, experience, education, etc.)

    b. What background do you have in using computers?

2. Talking About Cobra system

    a. How long have you been working with this application?

    b. Why do you use this application? (Needs, interests, and goals)

    c. When and where do you access the system?

    d. How do you access the system? (Connection speed, resolution, etc.)

3. Talking About PZ Web system

    a. How long have you been working with this application?

    b. Why do you use this application? (Needs, interests, and goals)

    c. When and where do you access the system?How do you access the system? (Connection speed, resolution, etc.)

4. Develop goals for the system

    a. How would you define a successful system for your organization?

    b. What does success look like?

    c. How will you know when you have been successful?

    d. How would you describe the system?

        i. From an organizations viewpoint?

        ii. From a users viewpoint?

5. Conduct task analysis and prioritize task

    a. What do you do on the system? (Tasks, content, features and functionality)

    b. Which tasks are critical to the organizations success on the system?(Criticality)

    c. Which tasks are critical to your success on the system?

    d. Which tasks are most important to you? (Importance)

    e. Which features of the system do you use the most? (Frequency)

    f. Which features are prone to usability issues? (Vulnerability)

6. Measurable usability objectives

    a. Which tasks should you be able to accomplish easily with few errors?(Eficiency)

    b. Which tasks should you be able to _nish quickly and e_ciently?(E_ectiveness)

    c. What level of satisfaction should you have after using the system?(Enjoyability)

    d. What level of satisfaction do you actually have?

7. Purpose/vision of the new Personeel Online

    a. What is the purpose of the system?

    b. What are the goals of the system?

    c. What prompted the redesign?

8. Expectations, requirements and preferences regarding the new system

    a. What is your vision of what the system should do?

    b. Would you add, change or remove something to the new system?(Tasks, content, features or functionality)

    c. Are there any interface style that you prefer or feel very comfortable with?

    d. Give some real examples.

9. Any other comments

Thank you for your time !

Appendix A.2 Interviews outcomes

1. Payroll Specialist at Personeel en Zo

1.1. Open-ended questions

- Payroll administrator.
- From 25 to 45 years old.
- Several years as payroll administrator.
- Working with different systems as ADP, CMG, SAP CRM and SAP HR.
- Currently working with Cobra.
- ADP is more trustable than Cobra is.
- Cobra is less trustable but you can work on it at anytime.
- They use Excel for making calculations.
- They are used to work with computers at user level.

1.2 Talking about Cobra system

- Working with Cobra since 2006.
- They use Cobra to make all the pay slip's calculations, set everything about a new company, establish the social security settings, make corrections, proformas ...
- They access the system at work.
- The server is not in their office. They have to access the system through a remote desktop which makes this access slowest.
- They consider Cobra fast and friendly but they would like it to be more friendly and trustable.

1.3 Talking about PZ Web system

- They know how PZ Web works and although they only work with Cobra, some of them started working with PZ Web.
- PZ Web is the portal. The clients use it to make requests and Personeel en Zo have to process and answer these request. They use PZ Web to support the client walking with him throw the system.
- Personeel en Zo consider easier to see the XML corrections in PZ Web than in Cobra.
- They access the system mainly at work but also at home for doing some things as checking the email.
- They would like to have a Data Management System (DMS) in which they wouldn't have to make calculations anymore. The clients would make the requests and the system, let's say would automatically generate the outcome of that request (Not manual anymore).

1.4 Conduct task analysis and prioritize tasks

- They use the system to
  - Implement new clients.
  - Processing the salary.
  - Make changes in the setting where the client is not allow to access.
  - Make analysis.
  - Download reports in Excel and make analysis (statistics).
  - Make offers for a company when it is interested on the system.
  - Put back the payroll period. This is something that goes throw a manual process and can lead to errors. When they have to put back a payroll period the communication between the payroll administrators is throw email or telephone which is no secure enough.
- They consider critical tasks
  - The deadline of the payroll administration (critical dates).
  - What happened if the company grows? Maybe they would need more payroll administrator during the critical days.
  - Need of DMS.
  - Speed of the system. Problems with the system most of the time are due to updates or the connection with the external server.

    - The most important for them as users is that the system was fast and trustable. (Trustable pay slips on time)
- The most important task for them is to process pay slips.
- And the features on the system they use the most are the "batch" and the loon components.
- About the features which are prone to usability issues..
    - The Save button.
    - Don't repeat the same information several times.
    - Group tasks in different pages.
    - Like salary experts they like to see several tabs with all the functionalities on one page, but like users, it scares them.

1.5 Expectations, requirements and preferences regarding the new Personeel Online

- They found that the system should be easy to use without too much problems. The whole process shouldn't take too long (the monthly payroll, the payroll taxes, etc.)
- They consider the system quite slow sometimes. This is the biggest problem. If it works at normal speed, the system works fine.
- Another problem is the updates: after the updates, there are some errors in the system; for example, some documents can't be printed.
- If there is a mutation in the salaries of CAO (see CAO code), it would be good if you only need to change this in one place and the other companies would change it automatically.
- They would like to have a list in which you can see when a company is already implemented.
- To use DMS. Year Management Documentation is also manual.

2. Payroll Specialist of clients

2.1 Open-ended questions

- From 35 to 45 years old.
- Normal office applications.
- Payroll specialist.

2.2 Talking about PZ web system

- 3 to 4 years working with PZ Web.
- They use this application to provide the payroll administration to their customers.
- They access the site from their own office or when visiting clients.

2.3 Develop goals for the system

- Easy to use.
- Access for every location.
- Give their customers also access to the system.
- They want to see what a customer has done on the system.
- Speed in processing. Now we have to wait until a payroll specialist process their request and send them the outcome.
- Be able to process everything at every time.
- They describe the system like a Payroll processing system.

2.4 Conduct task analysis and prioritize tasks

- With the new system they want:
    - To see what a customer has done on the system.
    - Speed in processing. Now we have to wait until a payroll specialist process their request and send them the outcome.
    - Be able to create/modify companies/employees.
    - Be able to make proformas.
    - Process the pay slips themselves.
    - They would like to be able to have in only one file all the documents to print.
    - Manage signals.
    - Give their clients access to the system.
    - Enter/Modify companies/employees.

  - Make proformas.
- Critical tasks are:
  - Immediately payroll process.
  - Make less mistakes.
- The most important for them is the payroll processing system.
- And they also find very important to be able to make proformas.
- The feature on the system they use the most is managing loon components (Add/Modify)

2.5 Measurable usability objectives

- They should be able to add new employees and process the "out of service" tasks easily with few errors.
- The task they should be able to finish quickly and efficiently is Entering loon codes.
- To increase their level of satisfaction they would like to know the result of what they are doing immediately.

2.6 Purpose/vision of the new Personeel Online

- They see the new system like a very efficient payroll administration system in which they can make everything on their own.
- The goal of the system will be to allow the users processing pay slips themselves

2.7 Expectations, requirements and preferences regarding the new Personeel Online

- Login of the mutations.
- Give their clients the feeling that it is Their System.

3. Normal User

3.1 Open-ended questions

- From 22 years old.
- Grew up with computers and use it every day.
- Have all different kinds of other high Tec products like the iPod and mobile phone.
- Use the computer to get all kinds of information and to communicate.
- They learned to use computer programs like excel, word, internet, on their education.

3.2 Talking about PZ web system

- It took them like a month to really know this system.
- They mainly use this application to provide their employees the pay slips.
- It is very good to know that everything works out to the tax company.
- It is very handy to have access to the system everywhere and always can see all the need information like (verlof) when the employees are on holidays or for how long they were ill.
- It is also a good application because you can put all kind of information in the system: employees ID, contract, results of a meeting...
- They access the system mainly at work but also at home for doing some things as checking the email or have a quick view to employees information as possible mutations.

3.3 Conduct task analysis and prioritize tasks

- One of the things they do on the system is: when they are on the system they have a list where all the different mutations of the past month are listed. They copy these mutations in the PZ Web. When they finish this list they ask Personeel en Zo for providing them the pay slips. That is the main task they have.
- Beside this main task they also use the system to have some insight to all the different employees information like, spare holidays, illness, etc. They use it also as a way of communication with their employees.
- They consider critical tasks
  - That all the salary information were correct. It is very important for the employees to get a good information about their salary and that there was a lot of trust in the company. They need confidence in the company.
  - To pay the employees on time.

    - To give to the employees a correct pay slip.
    - For company success it is also very important that everything was correct to the tax company.
    - It is necessary that everything just works out. It is very important that the system provides them to focus on the core business of the company. If they need to spend lots of hours in the system with all kinds of bugs coming out, then they don't need the system. It is very important for the success of the company that all the given functionalities work well.
    - Speed is very important for their personal success. They need to work in a quick and handy way.
    - Usability is also very important. All the information needs to be well organized and if they want to find a specific piece of information they want to find it fast.
- The most important thing for them is to trust in the system. If there is a problem with the system they want it to be solved very quick. They don't want to spend a lot of time in the system. They want it to work.
- The features on the system they use the most are
    - The quick possibility to insert the monthly mutations.
    - To ask for a salary calculation.
    - To see the monthly results.

3.4 Measurable usability objectives

- The tasks they should be able to accomplish easily with few errors are the not so important tasks like, hanging a result of a meeting or a contract; the vacation questionnaires, etc.
- Personeel en Zo should do the important things like the salary related errors.
- About the salary related tasks, they don't want to spend a lot time on that subject.
- They would like to have good feeling about they did all the salary in a short time.
- They want to have a good feeling having the chance of asking the system for all kinds of support at any time. When they need some information they want to find it really fast.
- All the information should be good organized.
- They feel happy with the system but think there are too many irrelevant information on the main pages of Personeel Online. They don't use most of these information.

3.5 Expectations, requirements and preferences regarding the new Personeel Online

- They are satisfied with the Personeel Online like it is now. They trust in the system and it works out fine.
- They would hide all the irrelevant information better. They would like to have less options in Personeel Online.
- They would like to get the tabs ordered chronologically. As they use it.
- They think that the "snelkoppelingen" can be downside to a less important tab.
- They would like to give to each of their employees his own digital pay slip.
- They found the interface very direct. All blocks are very straight. They would like it to be a bit more childish.

4. Payroll Specialist of one client

4.1 Open-ended questions

- From 40 to 50 years old.
- Very long time experience.
- Book keepers.
- Office applications.

4.2 Talking about PZ web system

- The worked with Cobra at the beginning but they are not using it anymore.
- They use PZ Web for processing pay slips and always from his office.

4.3 Conduct task analysis and prioritize tasks

- Enter payroll processing mutations.
- Fill in loon components.

- Add and dismiss employees.
- They used to make Proformas when they were working with Cobra, but not anymore.
- They find a critical task to compare quickly the differences between the last month and the current one (Standard Register).
- Very important are, the task of entering loon components and making fast calculations.

4.4 Expectations, requirements and preferences regarding the new Personeel Online

- They would like
  - To have the import function
  - To be able to make Proformas.
  - To have an overview about all the changes made by someone else.

5. Human Resource Specialist of clients

5.1 Open-ended questions

- From 21 years old.
- They work as users and administrators.

5.2 Talking about PZ web system

- They find it very easy to work with and friendly.
- The thing they really miss on the system are Evaluations (Statistics).
- They would like to have the help telephone number somewhere visible in the main page.
- They want to have like a template that includes all the documents they can have about a person. So they have visible all the documents they already have and which ones they could still add if they wanted to. "Suggestions"
- Previous run so they can check if everything is going ok.
- Make the panels more clear. For example, when they add a bonus, they don't know if it is going to be positive or negative; so we should give to them explanations about what are they actually doing.
- Check lists when. for example, creating a new employee

5.3 Conduct task analysis and prioritize tasks

- The tasks they do on the system are tasks as:
  - Make contracts.
  - Make the EDM.
  - Salary changes.
  - Creating an employee.
  - The Verlof and Verzuim tabs.

5.4 Measurable usability objectives

- Missing the employee photo.
- Save several things at the same time.
- Save button should be always visible to not forget to save changes.
- Chance of combine several things, as salary, CV, address... in only one sheet in order to print it. We could add a check list to select the documents they want to print at the same time.
- Have the possibility of checking what you have changed or done.
- They don't use the tabs since the direct links are in the system.
- They would like to be able to change the salary in advance, as well as other features.
- Chance of have a button to "send a letter" in *Personal information*. And have a template letter in which the employee address is automatically copied.
- If an employee is not working anymore for a company...
  - Indicate that the employee is not working anymore so the system can do all the settings (They are always the same).
- The system could generate a signal if there are holidays left. What do you want to do with them?

6. Employee

- In the new system they would like to have the following features:
  - See/Ask for holidays (Verlof)

- See news.
- Handbook.
- See and fill in some kind of document as the declaration document.
- See/change his own personal information.
- Hour registration.
- See their own pay slip online.

Appendix A.3 Standard Roles, User Levels and Tasks

Standard Roles

(R1) Pay roll Specialist
(R2) Pay roll Specialist
(R3) Normal User
(R4) Payroll Specialist
(R5) HR Specialist
(R6) Employee
(R7) Debtor

User Levels

(L1) Administrator
(L2) Client
(L3) Group
(L4) Organization
(L5) Company
(L6) Employee
(L7) Debtor

Tasks per Role and User Level

Group settings

- Insert Group (example accountant office)
  - Bind organizations to a Group
- Insert Organization (example big company 100+)
  - Bind companies to an organization
  - Insert Departments (example system administration)
- Insert Debtor
  - Bind a Group to a debtor
  - Bind a organization to a debtor
  - Bind companies to a debtor
- Insert a System administrator (professional Personeel en Zo) … *L1*
  - Insert username
  - Insert password
  - Insert email address
- Insert a new group user (accountant) ………………………… *L2*
  - Insert username
  - Insert password
  - Insert email address
  - Bind to a Group
  - Bind to an authorization profile
- Insert a new organization user ………………………………… *L3*
  - Insert username
  - Insert password
  - Insert email address
  - Bind to an organization
- Insert a new company user …………………………………… *L4*
  - Insert username
  - Insert password

  - Insert email address
  - Bind to a company
- Insert a new department user ………………………………… *L5*
  - Insert username
  - Insert password
  - Insert email address
  - Bind to a company
- Insert a new employee user ……………………………………. *L6*
  - Insert username
  - Insert password
  - Insert email address
  - Bind to a organization / company / department
- Insert Debtor user ……………………………………………… *L7*
  - Insert username
  - Insert password
  - Insert email address
  - Bind to a debtor
  - Bind to an authorization profile
- Insert authorization profiles (none/read / write)
  - Rights for the pages

Login page

- Login
- Recover password by email

Start page

- Modify user information (login name / password / email address)
- Module Agenda (signals, birthdays, important dates)
- Module illness (pending illness cases)
- Module absence (people on holidays)
- Module wage costs (brief survey salary cost)
- Quick link to a module (list with activities)

System settings

- Insert a loon codes ………………………….. L1 R1
- Insert a pension plan ……………………….. L1 R1, R2
- Insert model company ……………………… L1 R1
- Insert a salary table ………………………… L1 R1
- Insert a SVW table (franchise / percentages . L1 R1
- Insert a SVW table (sector and risk group)…. L1 R1

Debtors

- Insert a new debtor ………………………… L1 R1, R2
- Insert a debtor contract …………………….. L1 R1
- Manage debtor NAW information ………… L7 R1, R2, R3, R4, R5, R7

Organization management

• Insert an organization	L2	R1, R2, R3, R4, R5
• Insert an organization contact person , email	L3	R1, R2, R3, R4, R5
• Insert departments	L3	R1, R2, R3, R4, R5
• Insert departments contact person / email	L3	R1, R2, R3, R4, R5
• Insert external suppliers (bijv. arbodienst) / email	L3	R1, R2, R3, R4, R5
• Insert a contact information for the external suppliers (*)/ email	L3	R1, R2, R3, R4, R5
• Insert the company free days (calendar)	L3	R1, R2, R3, R4, R5

Company management (salary registration)

• Insert a new company	L3	R1, R2, R3
• Insert period type (week/4weeks/months)	L4	R1, R2, R3, R4
• Insert a company contact person / email	L4	R1, R2, R3, R4, R5
• Insert a cost location	L4	R1, R2, R3, R4
• Modify the company SVW table (bvWGA premie)	L4	R1, R2, R3, R4
• Insert a Loonhefing nummer	L4	R1, R2, R3, R4
• Import new employees	L4	R1, R2, R3, R4, R5
• Insert a company Loonmodel	L4	R1, R2, R3, R4
• Bind the looncodes to a company grondslagen	L4	R1, R2, R3, R4
• Insert model employee(s)	L4	R1, R2, R3, R4
• Insert a accountant book / descriptions	L4	R1, R2, R3, R4
• Insert a reservation	L4	R1, R2, R3, R4
• Insert components to see on the standard register overview	L4	R1, R2, R3, R4
• Bind a company to a pension plan	L4	R1, R2, R3, R4
• Insert the company pension plan	L4	R1, R2, R3, R4
• Insert a salary table to a company	L4	R1, R2, R3, R4, R5
• Make available a salary table to an employee	L4	R1, R2, R3, R4, R5
• Insert employee functions to a company	L4	R1, R2, R3, R4
• Bind the looncodes to the accountant book	L4	R1, R2, R3, R4
• Copy a company	L2	R1, R2, R3, R4
• Copy a company with employees	L2	R1, R2, R3, R4
• Copy a company with employees and period type	L2	R1, R2, R3, R4
• Quick insert looncomponents	L4	R1, R2, R3, R4
• Import looncomponents	L4	R1, R2, R3, R4

Employee salary management

• Insert a new employee	L5	R1, R2, R3, R4, R5
• Insert Eerste Dag Melding (EDM)	L5	R1, R2, R3, R4, R5
• Out of service notification for an employee	L5	R1, R2, R3, R4, R5
• Modify Personal information	L5	R1, R2, R3, R4, R5
• Modify Bank account	L5	R1, R2, R3, R4, R5
• Modify the calendar	L5	R1, R2, R3, R4, R5
• Modify the Salary	L5	R1, R2, R3, R4, R5
• Modify the Contract	L5	R1, R2, R3, R4, R5
• Modify the Leasecontract	L5	R1, R2, R3, R4, R5
• Modify Pension plan (participation yes/no)	L5	R1, R2, R3, R4, R5
• Modify a Reservation (participation yes/no)	L5	R1, R2, R3, R4, R5
• Modify loon components	L5	R1, R2, R3, R4, R5
• Modify loonhe_ng.	L5	R1, R2, R3, R4, R5
• Modify components to see on the standard register overview	L5	R1, R2, R3, R4, R5
• Modify the SVW (deelname ja/nee) .	L5	R1, R2, R3, R4, R5
• Modify insurance policy	L5	R1, R2, R3, R4, R5

• View the current payslip	L5	R1, R2, R3, R4, R5
• Make a Proforma calculation.	L5	R1, R2, R3, R4, R5

Employee staff management

• Modify department / position	L5	R1, R2, R3, R4, R5
• Modify holidays	L5	R1, R2, R3, R4, R5
• Set the amount of holidays	L5	R1, R2, R3, R4, R5
• Modify absence	L5	R1, R2, R3, R4, R5
• Insert illness report	L5	R1, R2, R3, R4, R5
• Modify Dossier (action points)	L5	R1, R2, R3, R4, R5
• Modify Dossier (documents)	L5	R1, R2, R3, R4, R5
• Insert user information for employee login	L5	R1, R2, R3, R4, R5
• Send user information for employee login	L5	R1, R2, R3, R4, R5
• Modify supplies / loan information	L5	R1, R2, R3, R4, R5
• Merge MS word files (contract / bruikleenovk. Etc.)	L5	R1, R2, R3, R4, R5
• Module recruitment en selection	L5	R1, R2, R3, R4, R5
• Modify training plans (*)	L5	R1, R2, R3, R4, R5
• Modify Competences (*)	L5	R1, R2, R3, R4, R5
• Modify Insurances (*)	L5	R1, R2, R3, R4, R5
• Modify family / partner and children information. (*)	L5	R1, R2, R3, R4, R5
• Modify evaluation (*)	L5	R1, R2, R3, R4, R5
• Modify external contacts (family doctor / alarm number) (*)	L5	R1, R2, R3, R4, R5
• Modify phisical exams (*)	L5	R1, R2, R3, R4, R5
• Dashboard staff information (*)	L5	R1, R2, R3, R4, R5

Salary processing

• Proccess the salary run for a company	L4	R1, R2, R3, R4
• Roll back the salary run for a company	L4	R1, R2, R3, R4
• Proccess the salary run for a company t.b.v. TWK calculations	L4	R1, R2, R3, R4

Salary outputs View / generate

• Proccess Status overview/ niet verwerkte loonruns per company	L4	R1, R2, R3, R4
• View / Print pay slip per company	L4	R1, R2, R3, R4
• View a Historic overview per company / per employee	L4	R1, R2, R3, R4
• View / Print van payment list per company	L4	R1, R2, R3, R4
• View / Print standard register component per company	L4	R1, R2, R3, R4
• View / Print wage costs per company/employee/ cost location	L4	R1, R2, R3, R4
• View / Print Year statement	L4	R1, R2, R3, R4
• View / Print Journal overview per company/employee/department	L4	R1, R2, R3, R4
• View / Print deductions and payments of the pension plans (period type)	L4	R1, R2, R3, R4
• View / Print fiscal overview	L4	R1, R2, R3, R4
• Insert Clieop file for bank transfers	L4	R1, R2, R3, R4
• Insert wage declaration XML	L4	R1, R2, R3, R4
• Generate a .pdf with all monthly output	L4	R1, R2, R3, R4
• Dashboard financial (*)	L4	R1, R2, R3, R4

Signals per email:

• Manage of signals (who receives which signals)	L5	R1, R2, R3, R4, R5

Birthday	L5	R1, R2, R3, R4, R5
End contract	L5	R1, R2, R3, R4, R5
End probation	L5	R1, R2, R3, R4, R5
Expiration day for the passport	L5	R1, R2, R3, R4, R5
Action point of staff file	L5	R1, R2, R3, R4, R5
Action point of absence	L5	R1, R2, R3, R4, R5

Staff information overview / generate

View / Print vacation overview	L5	R1, R2, R3, R4, R5
View / Print absence overview	L5	R1, R2, R3, R4, R5
View / Print N.A.W. overview	L5	R1, R2, R3, R4, R5
View / Print birthday calendar (*)	L5	R1, R2, R3, R4, R5
View / Print Organigram (*)	L5	R1, R2, R3, R4, R5

Module wage declaration

View / Send wage declaration XML	L4	R1, R2, R3, R4
View / Send wage declaration EDM	L4	R1, R2, R3, R4

Overviews

Overview imported mutations per user	L4	R1, R2, R3, R4, R5
Overview imported mutations per company	L4	R1, R2, R3, R4, R5
Status overview (not) processed loonruns	L4	R1, R2, R3, R4, R5
Status overview of sent signals	L5	R1, R2, R3, R4, R5

Employee logins

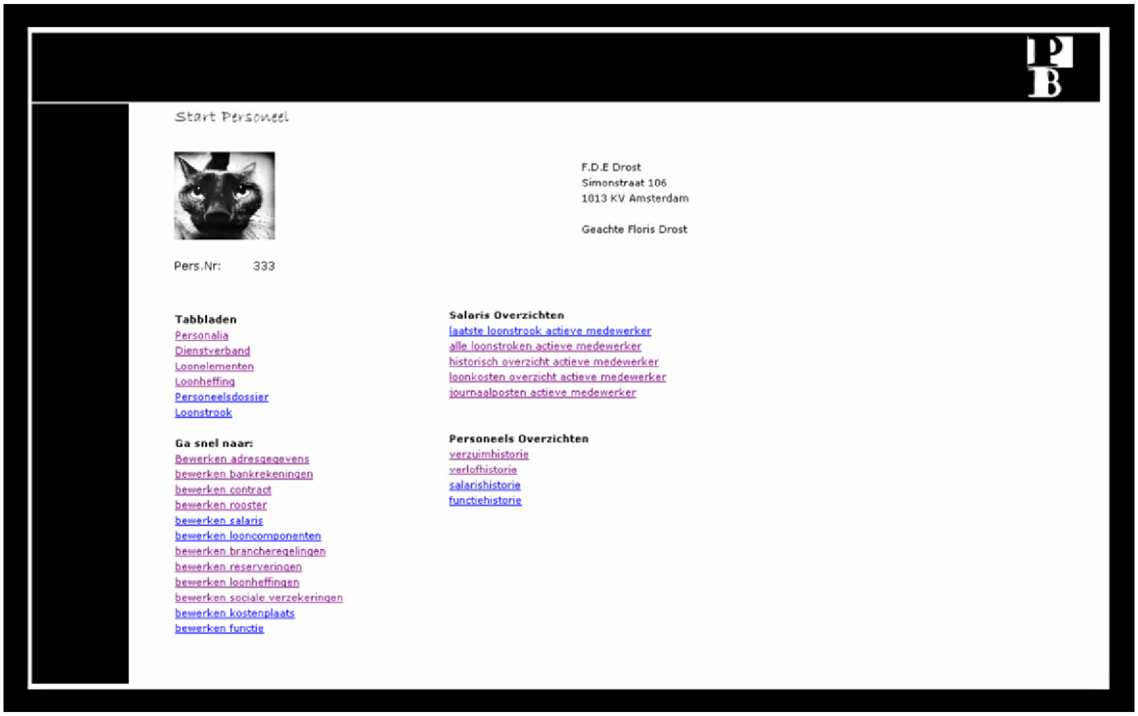
Insert company settings of the employee login	L4	R1, R2, R3, R4, R5
Insert (employee) user	L4	R1, R2, R3, R4, R5
Insert user profiles	L4	R1, R2, R3, R4, R5
View employee holidays overview	L6	R1, R2, R3, R4, R5
View / Modify employee personal information	L6	R1, R2, R3, R4, R5
View / Modify payslip individual employee	L6	R1, R2, R3, R4, R5
Leave card claim	L6	R1, R2, R3, R4, R5
Holiday planning	L6	R1, R2, R3, R4, R5
Hours registration justification	L6	R1, R2, R3, R4, R5
Insert Declarations	L6	R1, R2, R3, R4, R5

Support pages

- Getting started
  - Insert company
  - Insert employee
  - Insert looncomponent
- Quick help per tab.
  - FAQ

Appendix B. Prototype description

Appendix B.1 First Prototype



Start Personeel

F.D.E Drost
Simonstraat 106
1013 KV Amsterdam

Geachte Floris Drost

Pers.Nr: 333

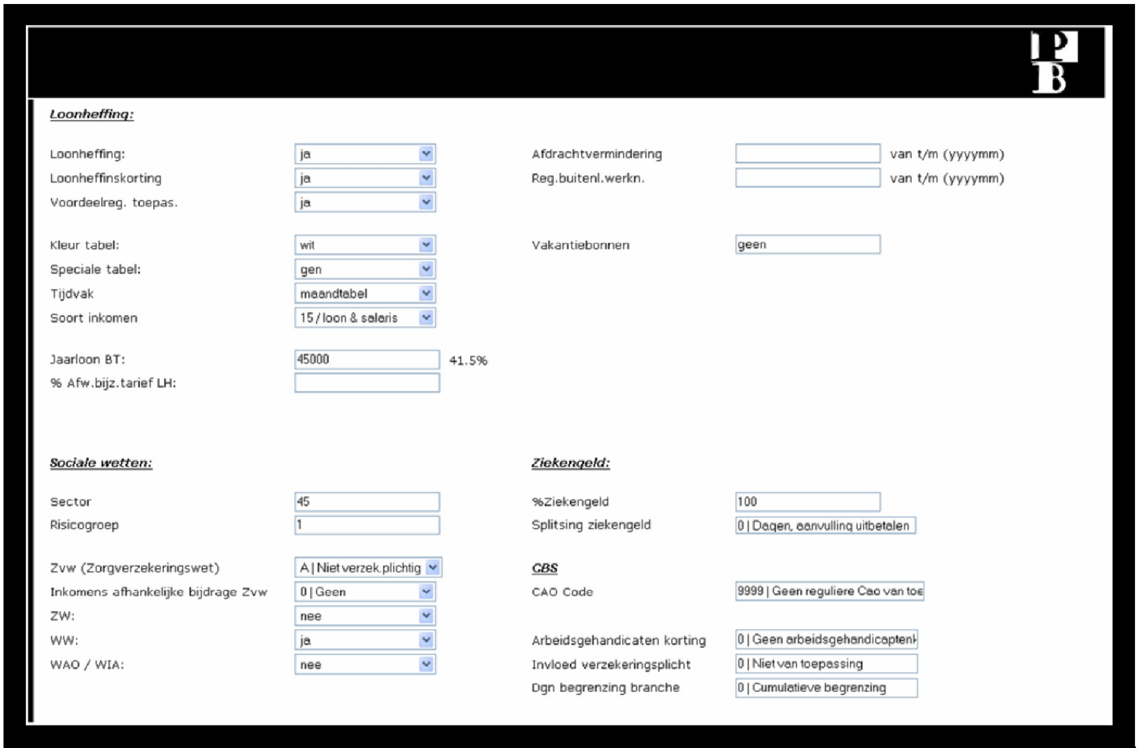
Tabbladen
Personalia
Dienstverband
Loonelementen
Loonheffing
Personeelsdossier
Loonstrook

Ga snel naar:
Bewerken adresgegevens
bewerken bankrekeningen
bewerken contract
bewerken rooster
bewerken salaris
bewerken looncomponenten
bewerken brancheregelingen
bewerken reserveringen
bewerken loonheffingen
bewerken sociale verzekeringen
bewerken kostenplaats
bewerken functie

Salaris Overzichten
laatste loonstrook actieve medewerker
alle loonstroken actieve medewerker
historisch overzicht actieve medewerker
loonkosten overzicht actieve medewerker
journaalposten actieve medewerker

Personeels Overzichten
verzuimhistorie
verlofhistorie
salarishistorie
functiehistorie

Start page with quick links and the company/employee most important information and photo.
On the left, the left menu to choose between companies and employees will be place.
The two buttons on the right top corner (*P*,*B*) stand for Personeel and Bedrijf (Employees and Company) and are used to switch between Employee and Company level.



Loonheffing:

Loonheffing: ja

Loonheffinskorting ja

Voordeelreg. toepas. ja

Kleur tabel: wit

Speciale tabel: gen

Tijdvak maandtabel

Soort inkomen 15 / loon & salaris

Jaarloon BT: 45000 41.5%

% Afw.bijz.tarief LH:

Afdrachtvermindering van t/m (yyyymm)

Reg.buitenl.werkn. van t/m (yyyymm)

Vakantiebonnen geen

Sociale wetten:

Sector 45

Risicogroep 1

Zvw (Zorgverzekeringswet) A | Niet verzek.plichtig

Inkomens afhankelijke bijdrage Zvw 0 | Geen

ZW: nee

WW: ja

WAO / WIA: nee

Ziekengeld:

%Ziekengeld 100

Splitsing ziekengeld 0 | Dagen, aanvulling uitbetalen

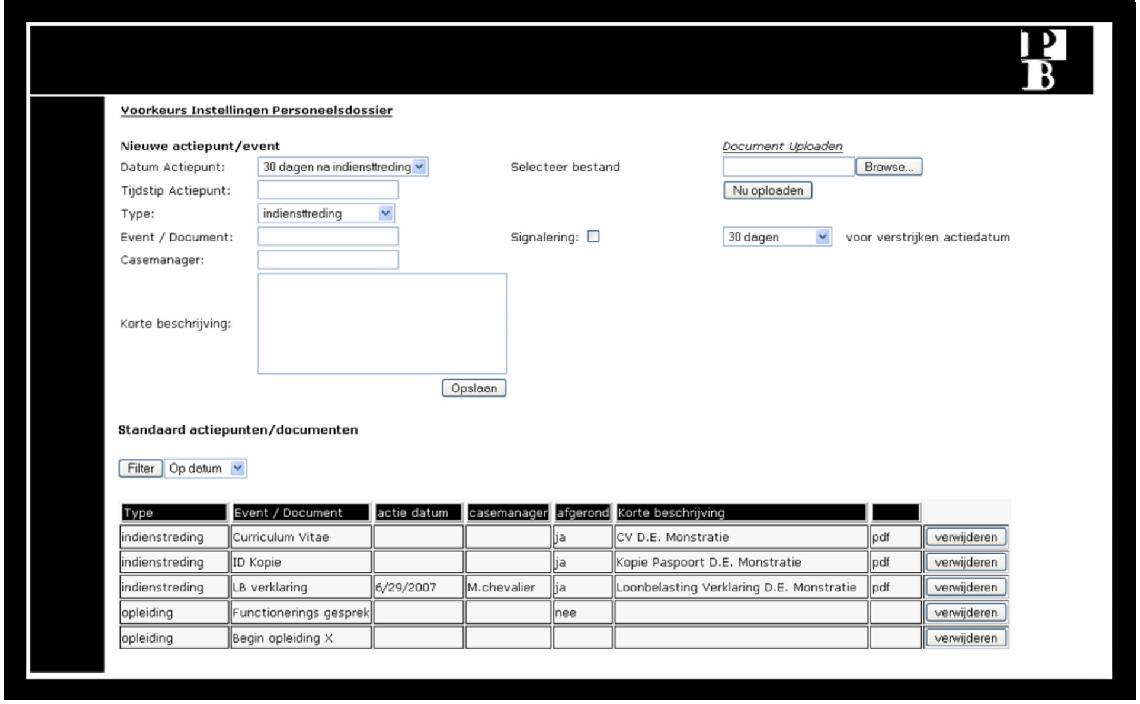
CBS

CAO Code 9999 | Geen reguliere Cao van toe

Arbeidsgehandicaten korting 0 | Geen arbeidsgehandicaptenk

Invloed verzekeringsplicht 0 | Niet van toepassing

Dgn begrenzing branche 0 | Cumulatieve begrenzing



Voorkeurs Instellingen Personeelsdossier

Nieuwe actiepunt/event

Datum Actiepunt: 30 dagen na indiensttreding

Tijdstip Actiepunt:

Type: indiensttreding

Event / Document:

Casemanager:

Korte beschrijving:

Selecteer bestand

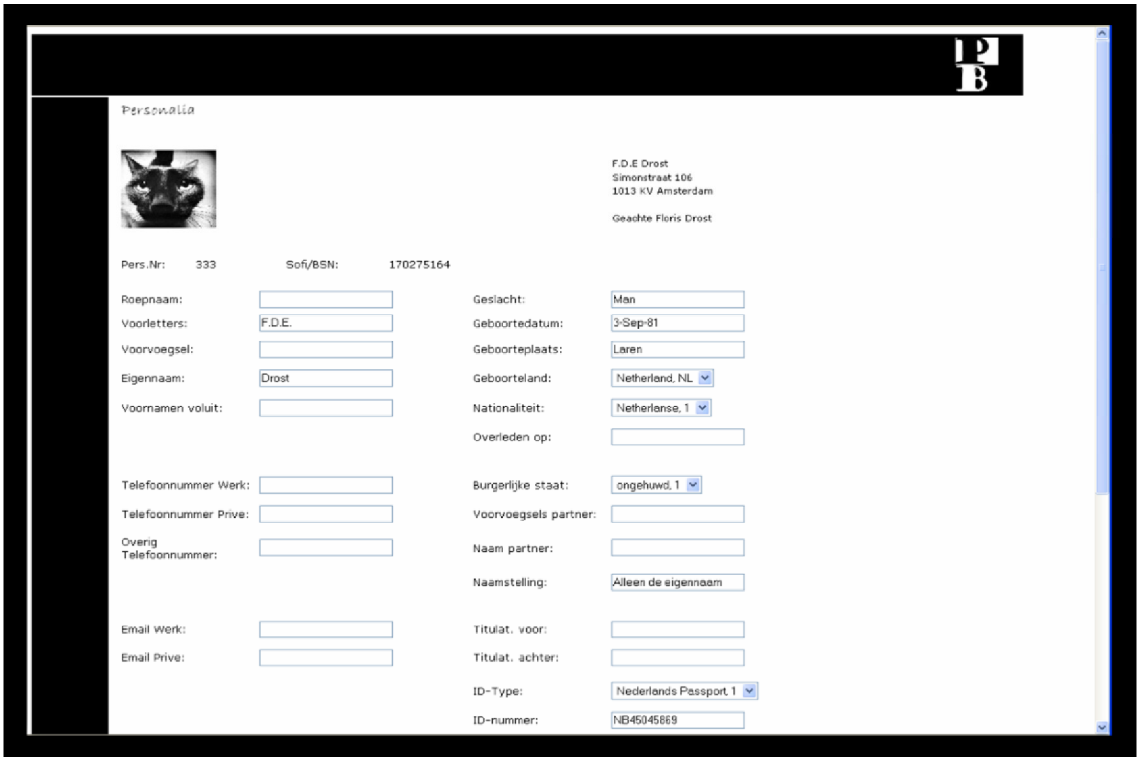
Document Uploaden

Signalering: □ 30 dagen voor verstrijken actiedatum

Standaard actiepunten/documenten

Type	Event / Document	actie datum	casemanager	afgerond	Korte beschrijving		
indienstreding	Curriculum Vitae			ja	CV D.E. Monstratie	pdf	verwijderen
indienstreding	ID Kopie			ja	Kopie Paspoort D.E. Monstratie	pdf	verwijderen
indienstreding	LB verklaring	6/29/2007	M.chevalier	ja	Loonbelasting Verklaring D.E. Monstratie	pdf	verwijderen
opleiding	Functionerings gesprek			nee			verwijderen
opleiding	Begin opleiding X						verwijderen

These kind of pages are shown either you want to change the current information or if you only want to see the information. The tables and forms are editable and already filled in with the current or default information.



Personalia

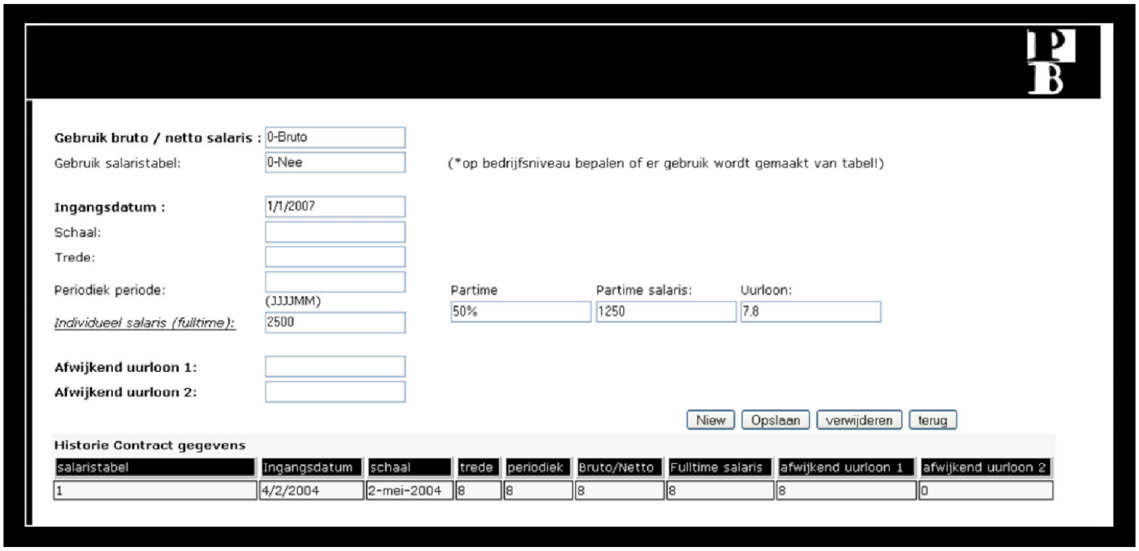
F.D.E Drost
Simonstraat 106
1013 KV Amsterdam

Geachte Floris Drost

Pers.Nr: 333 Sofi/BSN: 170275164

Roepnaam:		Geslacht:	Man
Voorletters:	F.D.E.	Geboortedatum:	3-Sep-81
Voorvoegsel:		Geboorteplaats:	Laren
Eigennaam:	Drost	Geboorteland:	Netherland, NL
Voornamen voluit:		Nationaliteit:	Netherlanse, 1
		Overleden op:	
Telefoonnummer Werk:		Burgerlijke staat:	ongehuwd, 1
Telefoonnummer Prive:		Voorvoegsels partner:	
Overig Telefoonnummer:		Naam partner:	
		Naamstelling:	Alleen de eigennaam
Email Werk:		Titulat. voor:	
Email Prive:		Titulat. achter:	
		ID-Type:	Nederlands Passport, 1
		ID-nummer:	NB45045869

Sometimes these pages occupy too much space and you have to scroll down to see all the information. Due to this if you only want to change something that you can see without having to scroll down, you can forget about saving because the save button is not visible.



Gebruik bruto / netto salaris :	0-Bruto			
Gebruik salaristabel:	0-Nee	(*op bedrijfsniveau bepalen of er gebruik wordt gemaakt van tabel!)		
Ingangsdatum :	1/1/2007			
Schaal:				
Trede:				
Periodiek periode:	(JJJJMM)	Partime	Partime salaris:	Uurloon:
Individueel salaris (fulltime):	2500	50%	1250	7.8
Afwijkend uurloon 1:				
Afwijkend uurloon 2:				

Historie Contract gegevens

salaristabel	Ingangsdatum	schaal	trede	periodiek	Bruto/Netto	Fulltime salaris	afwijkend uurloon 1	afwijkend uurloon 2
1	4/2/2004	2-mei-2004	8	8	8	8	8	0

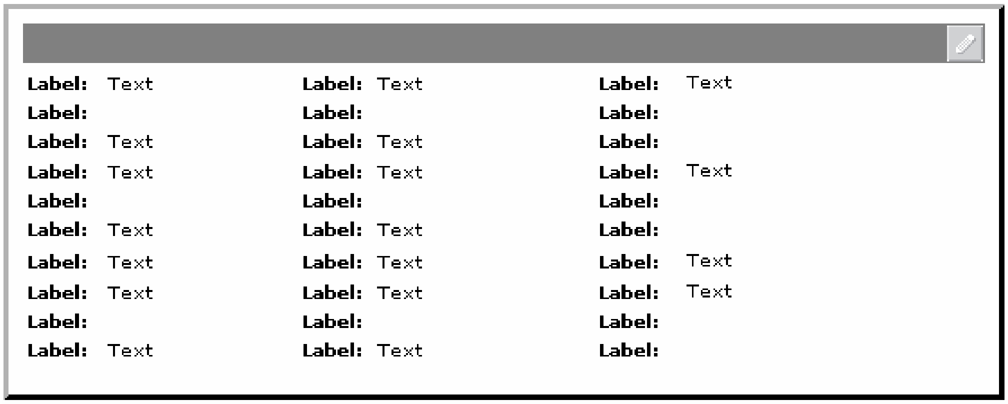
The historic information is always visible although if you don’t want to see it. Why to show always information that the user hasn’t asked for? It makes the site to take much more time to load.

Appendix B.2 Second Prototype

Patterns

Detail page areas patterns: These are the four patters that will be applied on the detail page areas. On them, fixed information is shown and the user will be able to carry out some actions as edit, make new or open the historic. All these actions will be triggered by clicking on one button, and a pop-up will appear. We can see the pop-up patterns on the next section.

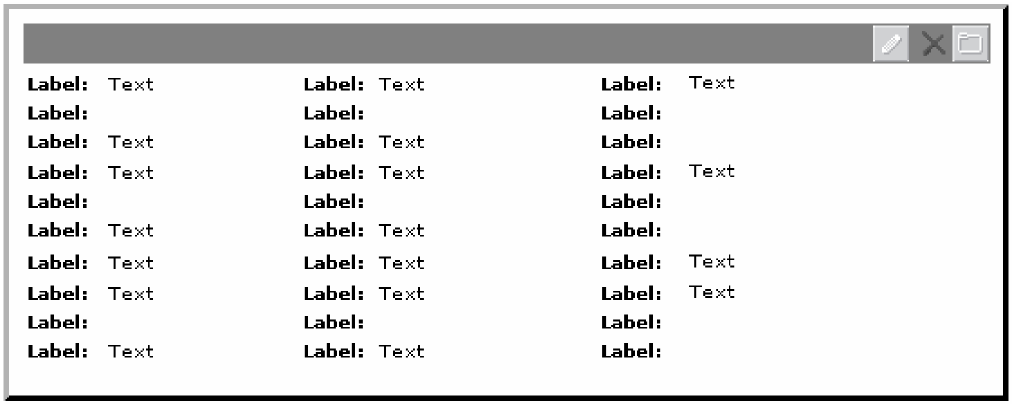
Pattern 1: Represents a form with fixed information on it and just one button to edit this information. A pop-up *pattern1* or *pattern2* will pop up.
We can find this type of areas for Personal Information, Loonheffing, Sociale wetten, Ziekengeld and CBS detail areas between others.



Label:	Text	**Label:**	Text	**Label:**	Text
Label:		**Label:**		**Label:**	
Label:	Text	**Label:**	Text	**Label:**	
Label:	Text	**Label:**	Text	**Label:**	Text
Label:		**Label:**		**Label:**	
Label:	Text	**Label:**	Text	**Label:**	
Label:	Text	**Label:**	Text	**Label:**	Text
Label:	Text	**Label:**	Text	**Label:**	Text
Label:		**Label:**		**Label:**	
Label:	Text	**Label:**	Text	**Label:**	

Detail page area: Pattern1

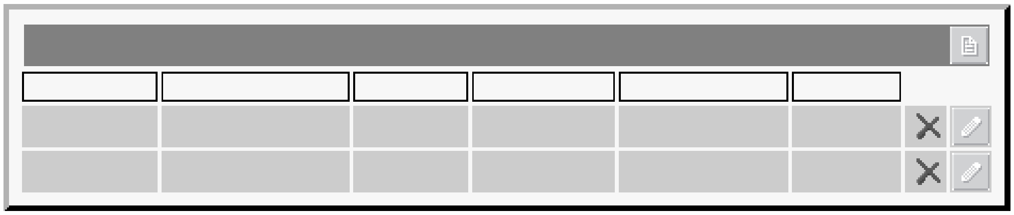
Pattern 2: Represents a form with fixed information on it and three buttons to edit or remove this information, and open the Historic pop-up. A pop-up *pattern1* or *pattern2* will pop up clicking on the *edit* button. Pop-up *pattern3* will be for the *historic* one. The remove button will remove the current information, which is the first row of the historic table.
We can find this type of areas for Salary, Contract, Rooster and Lease auto detail areas between others.



Label:	Text	**Label:**	Text	**Label:**	Text
Label:		**Label:**		**Label:**	
Label:	Text	**Label:**	Text	**Label:**	
Label:	Text	**Label:**	Text	**Label:**	Text
Label:		**Label:**		**Label:**	
Label:	Text	**Label:**	Text	**Label:**	
Label:	Text	**Label:**	Text	**Label:**	Text
Label:	Text	**Label:**	Text	**Label:**	Text
Label:		**Label:**		**Label:**	
Label:	Text	**Label:**	Text	**Label:**	

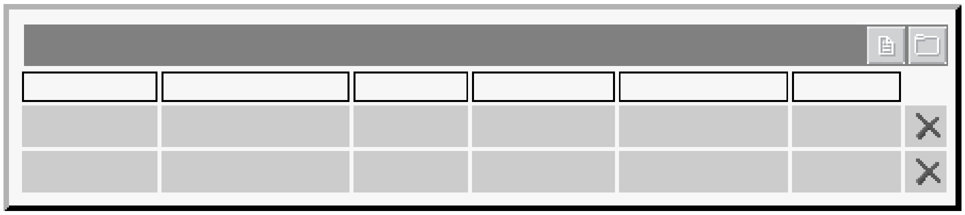
Detail page area: Pattern2

Pattern 3: Represents a table with fixed information on it. The user could delete and edit every table row. He will have the possibility of making a new one as well. A pop-up *pattern1* or *pattern2* will pop up clicking on the *edit* or *new* button.
We can find this type of areas for Adresgegevens, Bankgegevens, Action Points, Verlof, and Verzuim detail areas between others.



Detail page area: Pattern3

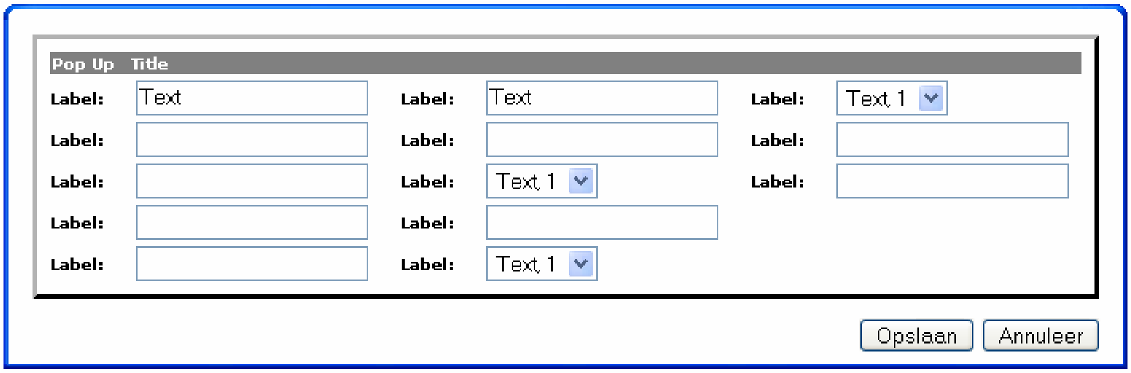
Pattern 4: Represents a table with fixed information on it. The user could delete every table row. He will have the possibility of making a new one or see the historic as well. A pop-up *pattern1* or *pattern2* will pop up clicking on the *edit* or *new* button. Pop-up *pattern3* will be for the *historic* one.
We can find this type of areas for Looncomponenten detail areas between others.



Detail page area: Pattern4

Pop-ups patterns: These are the three patterns applied for the pop-ups. All of them have the Save and Cancel buttons in common. The information on them is editable. This one can be the information that was already in a form and the user wants to edit, or default values when the user wants to create a new entrance.
To save the changes the user will have to click on Save, and by clicking on Cancel nothing will be saved. No matter the button the user press, the pop-up will be always closed.

Pattern 1: We can find this pattern for Salary, Action Points, Contract, Looncomponenten, Verlof, Rooster and Verzuim pop-ups between others.

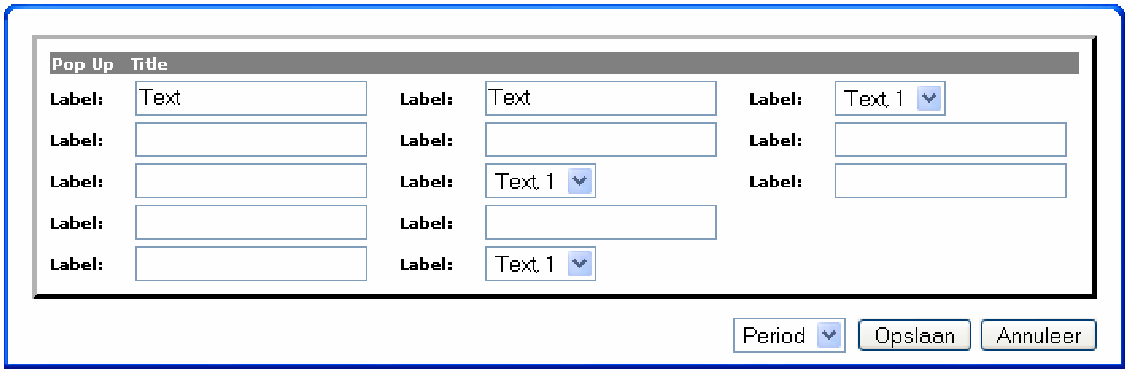


Pop Up Title

Label:	Text	Label:	Text	Label:	Text 1
Label:		Label:		Label:	
Label:		Label:	Text 1	Label:	
Label:		Label:			
Label:		Label:	Text 1		

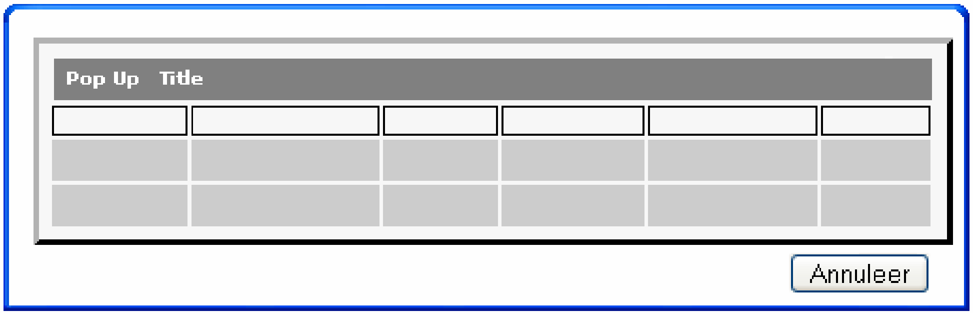
Pop-up: Pattern1

Pattern 2: This kind of pop-up includes a Period picker. All in this application is based on a certain period and sometimes the user needs to specified the period in which this information will be applied.
We can find this pattern for PersonalInfo, Address, Bank Account, Reserveringen and Loonheffing pop-ups between others.



Pop-up: Pattern2

Pattern 3: This pattern is mostly applied for the historic view.



Pop-up: Pattern3

Navigation Diagram:

Appendix C. Evaluation

Appendix D. Technical Design

References

This is the book I'm using as guideline during the project : Usability Engineering - scenario-based development of human-computer interaction - Mary Beth Rosson & John M. Carroll - Morgan Kaufmann publishers

Another book : Interaction Design - beyond human-computer interaction - Yvonne Rogers, Helen Sharp & Jenny Prece - John Wiley & Sons, Inc.

Learning ASP.net : Beginning.ASP.NET.2.0.pdf

Learning ASP.net : http://asp.net

Learning Usability : http://www.usability.gov :

Patterns in Interaction Design : http://www.welie.com

Screen recorder program : Camtasia Studio 5

New Web-Usability Evaluation Method : Scenario-Based Walkthrough: http://www.fujitsu.com/downloads/MAG/vol41-1/paper14.pdf

Sun Web Application Guidelines : http://developers.sun.com/docs/web-app-guidelines/uispec4_0/index.html#toc

Used in the Evaluation section : http://www.usabilityhome.com

Wikipedia

Ajax Control Toolkit : http://www.asp.net/ajax/ajaxcontroltoolkit/samples/

What is : http://whatis.techtarget.com

Microsoft Developer Network: http://msdn.microsoft.com/en-us/default.aspx