In the following research a usability evaluation of the Foleta web-application for use in secondary education is performed. Since 2008 many features have been added to the application. Foleta management suspects usability problems may occur. The aim of this research is to find and document as many usability problems in the Foleta user interface so that the problems can be corrected in future versions.

We have identified user types, hierarchy levels and their tasks respective tasks. To find usability problems, we tested the application with users and performed a heuristic evaluation. We present nine aggregated issues found in the user test sessions and compare issues to ten heuristic principles. We conclude our research with an advice on improving structure and design aesthetics.
# Table of contents

1 Introduction
   1.1 Motivation & problem definition 3
   1.2 Methodology 3

2 Foleta
   2.2 Foleta 4
   2.3 The application 4

3 User Identification
   3.1 Hierarchy levels 7
   3.2 User types 8

4 Task Analysis
   4.1 Schoolgroup tasks 9
   4.2 School tasks 9
   4.3 Overall tasks 10

5 User Testing
   5.1 User characteristics 11
   5.2 Aggregated issues 12

6 Heuristic evaluation 15

7 Conclusion 19

8 References 20

9 Appendices 21
Introduction
Chapter 1: Motivation description, problem definition & methodology

1.1 Motivation & problem definition
In the following research we will perform a usability evaluation of the Foleta web application for use in secondary education. We will evaluate the application by conducting a heuristic evaluation and test with users of the application. The goal of this research is to find and document as many usability problems as possible in the Foleta user interface, so that the problems can be corrected in future versions. Correcting these issues increases usability, improving the ease-of-use of the Foleta web-application [1].

Usability is a quality attribute that assesses how easy user interfaces are to use and the extent to which a computer system enables users to achieve goals efficiently and effectively [2]. An easy to use system requires less training and enhances performance on the user side. On the business side it can decrease support costs.

The application discussed in this research was launched on the web in 2008. Five years later the amount of users has increased and a number of new features has been added on top of the existing application. Foleta management suspects usability problems may occur. Since the system’s developers are not its intended users, developers finding the system easy to use does not mean its end users will have the same experience. With researching the usability of the Foleta application we aim to find issues that may have arose during the adding of features.

1.1 Methodology
In this section we briefly describe the method used for evaluating the Foleta application. We will look at the selected research lifecycle and elaborate on the different steps involved.

The evaluation will be performed following a four steps lifecycle. These four steps are: user identification, task analysis, user test sessions and a heuristic evaluation. The steps in the lifecycle will lead to a recommendation on how to improve the usability of the Foleta web-application. The four step lifecycle this research will follow is derived from D.J. Mayhew’s usability engineering lifecycle [3] and shown in Diagram 1.

To increase the usability and improve the ease-of-use for users we aim to find usability problems in the Foleta application. In order to identify these issues we will first identify who is using the application as a part of the first step in our lifecycle. Secondly we identify which tasks these users perform in the application. These first two steps presented in the lifecycle allow us to identify different user characteristics and the tasks users perform.

In step three and four we use two methods for gathering usability issues. These methods are divided into a test method (with end users) and an inspection method (without end users). The reason for combining these two methods is that studies and other research have consistently shown that different methods have various strengths; the best evaluation of a user-interface comes from applying multiple evaluation techniques [4].

The first method used to find usability issues is testing the application with users. These user test sessions are the third step in the selected lifecycle. The user test sessions are performed in an informal analysis where a number of evaluators are presented with tasks to preform in the interface and asked to comment on it.

The second method for finding usability issues is performing a heuristic evaluation. The heuristic evaluation is done by looking at the application and trying to come up with an opinion about what is bad about the interface. The heuristic evaluation is the last step in the lifecycle.

Each step in the lifecycle is discussed individually throughout the different chapters in this research. The different methodologies of the four steps are discussed at the beginning of each sections. Before starting with the first step of our lifecycle we introduce Foleta and the application our research is based on.
In this chapter we first introduce Foleta and discuss the company’s various goals to determine the aim of Foleta for its users. In the second section we introduce the Foleta application for use in secondary education to provide background information on the application and introduce the market Foleta operates in. We discuss the various goals of the application we will determine two main components inside the application. These components are identified by elicitation with Foleta management staff. Due to the closed nature of Foleta we furthermore provide and discuss various screenshots of the application to visualize functionalities discussed in this research.

## 2.1 Foleta

Foleta is a Dutch company located in Arnhem and offers a budget and formation-planning tool that is used to schedule formation for optimal capacity alignment and determine the number of required staff positions with the competencies and skills required. The Foleta application is used by management staff in educational institutes. This includes primary, secondary and higher education.

## 2.2 The Application

Foleta offers a web-based application [5] released by Foleta in 2008. The application serves the management, operations, and planning levels of educational institutes in the decision making process. The application is therefore categorized as a decision support system or DSS [6]. The application was originally build for use in secondary schools, providing support in the decision making process. The application aims to immediately show the effect of decisions.

This research paper will be focused on the web application used by management staff in secondary schools, for this market Foleta has a market share of 45.5 percent. These secondary educational institutions in the Netherlands are provided with 7.381 euro [7] for each admission by the state. With this budget, management wishes to optimize the educational benefits. Foleta aims to provide a tool to support this optimization. Only registered users can enter the tool. To visualize different functions discussed in this research we therefore provide screenshots of the application. With these screenshots we briefly discuss four different parts of the application.

Foleta management staff determines two main components in the application: the ‘G’, Group, function and the ‘S’, School, function. The G function is used for oversight of one or multiple schools. The S function is used on school level. These different user levels are discussed further in the user analysis. The different components have different functional properties that will be discussed in the task analysis.

All users start at the login screen shown in Figure 1.1. From here users with administrative rights enter the application in the ‘G’ component shown in Figure 1.2. From here users can enter the ‘S’ component, where also users with school rights enter the application. The ‘S’ component is shown in Figure 1.3. Foleta processes the data inputted by users to create reports. An example of such a rapport is shown in Figure 1.4.

On the web, usability is a necessary condition. If a website is difficult to use, users leave [2]. The goal of the Foleta application is to support management staff in the decision making process. Good usability is essential to this process. This research focuses on increasing the usability of Foleta. By evaluating the application with users and performing a heuristic evaluation we aim to gather the usability problems in the Foleta application. In the next chapter we will perform the first step in our lifecycle by identifying the application’s users.
Figure 1.1 The Foleta login page

Figure 1.1 shows the login screen for all Foleta users with the Foleta logo [1] and the login field used to access the application [2].

Figure 1.2 The Foleta schoolgroup screen

Figure 1.2 shows the Foleta application at the ‘G’ group level. The breadcrumb navigation [1], the school selection [2], year selection [3], notifications banner [4] and different tasks [5] are shown.
Chapter 2: An introduction to Foleta and the application

Figure 1.3 The Foleta school screen

Figure 1.3 shows the Foleta application at, ‘S’, school level. The school selection [1], top-navigation [2], year selection [3], notifications banner [4] and different tasks [5] are shown.

Figure 1.4 A task in Foleta

Figure 1.4 shows an example of a task. The task, in which courses can be appointed to sections, shows the different filter options [1], legend [2], shortcuts [3] and the data used in the task [4].
User identification
Chapter 3: Analysing different hierarchy levels and user types

In this section we identify different user levels and determine their hierarchical function. Identifying users is the first step of the lifecycle presented in section 1.2. This section furthermore provides information about different types of users that use the application.

User hierarchy and types were identified in a 90-minute interview with Foleta management and support staff. Foleta support staff has years of experience with identifying customer issues. Knowing the type of user is essential for providing support to customers and support staff has extensive knowledge in this area. Therefore we have interviewed Foleta staff to determine different hierarchy levels and user types.

3.1 Hierarchy levels
The different types are defined below as acquired from elicitation with Foleta management. Higher user hierarchy levels contain lower level types.

Schoolboards
The schoolboard, or board of education, is the highest level in the Foleta user hierarchy. Schoolboards operate as contractor and leave the functional use of Foleta to lower levels. Schoolboards are responsible for one or multiple schoolgroups; these schoolgroups contain one or multiple schools. The schoolboards is a higher administrative level and is often responsible for many schools within a district. Schoolboards use Foleta mostly for financial oversight.

Schoolgroups
The schoolgroup level is the second level in the Foleta user hierarchy. Schoolgroups often represent a partnership of individual schools combining resources, and is a high administrative level. A schoolgroup contains one or multiple schools. Schoolgroups use Foleta to schedule formation for optimal capacity alignment and determine the number of required staff positions with the competencies and skills required. Schoolgroups use the application at the ‘G’, Group, level discussed in section 2.2.

Schools
Schools are the third level in the Foleta user hierarchy. At school level users plan lessons, manage tasks and perform formation. Four different types of users are categorized within schools: Autonomous user, support staff, sections and autonomous teams. The different types of users are discussed further in 4.2. Schools use the application at the ‘S’, School, level discussed in section 2.2.

3.2 User types
The different user types are defined below as acquired from elicitation with Foleta management.

Autonomous user
The autonomous user is a one-man team. All tasks in Foleta are performed by one individual. Having one person handle all formation-planning leads to a low level of independency in the organization.

Support staff
The support staff uses Foleta to handle the formation planning. Multiple individuals are involved to process data in Foleta. Having a support staff handle formation-planning leads to a mid-level of independency in the organization.

Sections
Different sections operate independently to complete tasks in Foleta. This leads to a higher level of independency in the organization.

Autonomous teams
 Autonomous teams. This level is described by Foleta staff as a utopian user level. At this user level different teams work fully independent and each section performs tasks independently. This level is a view rather than a realistic, practical occurrence.
In this section we have identified three different levels of hierarchy within the Foleta application and four user types as a part of the first step in our lifecycle. The schoolboard, operating as a contractor, uses Foleta for financial oversight. The schoolgroup, administrating the application for one or multiple schools, uses Foleta for capacity alignment and the positioning of staff at ‘G’, Group, level. Schools use Foleta for lesson planning, formation and task management at ‘S’, School, level. Several types of users characterize these schools. These different types use the application in four different ways: as one autonomous user, with the help of support staff, in different sections or with multiple autonomous teams.

In the next chapter we will look at the functional tasks performed on the different hierarchal levels and by the several user types.
In this section the user tasks are identified and described. In this task analysis we decompose the application into high level tasks. The aim of ‘high level task decomposition’ is to decompose the high level tasks and break them down into constituent subtasks. This will show an overall structure of the main user tasks [8]. The task analysis provides us with knowledge of the tasks users wish to perform in the application. We gather user tasks in two steps:

- First we observe the application and study support documentation on http://support.foleta.nl. By observing and walking through the application we have identified different structural differences in the application. These differences were compared to the support documentation and have provided us with several high level tasks.
- Secondly, we have asked users about the tasks they use. These interviews are a part of the user elicitation we will discuss in Chapter 6. The questions used for this interview can be found in Appendix B.

We have combined both the tasks identified in the first step and those identified by users to create a list of high level tasks. We have identified two main components in the application: schoolgroup level and school level. All tasks belong to one of the components. The two components are the same as Foleta staff have identified in section 2.2. In the following sections we will first list and describe the high level schoolgroup, or ‘G’; tasks after which we will list and describe the high level school, or ‘S’; tasks.

### 4.1 Schoolgroup tasks

Schoolgroup level is an administrative level where the tasks that effect all schools within the group are preformed. The schoolgroup hierarchy level discussed in section 4.1 performs the following grouped tasks, which can only be performed by users with schoolgroup access:

<table>
<thead>
<tr>
<th>Administration:</th>
<th>User management for schoolgroup level and up. The different user levels are discussed in section 2.1.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrange schoolgroup:</td>
<td>Tasks for modulating information about employees, functions and several other data to model and fit Foleta to the organization.</td>
</tr>
<tr>
<td>Arrange school year:</td>
<td>The Foleta application uses information concerning a specific year, such as the number of school weeks in a year. With these tasks Foleta is fitted to a specific year.</td>
</tr>
</tbody>
</table>

### 4.2 School tasks

The school hierarchy level, as discussed in section 4.1, performs school tasks at school level. School level contains the following grouped tasks, which can only be performed by users with school level access:

<table>
<thead>
<tr>
<th>Administration:</th>
<th>User management on school level for users who only have access to functionalities on school level.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrange school:</td>
<td>The Foleta application uses specific school information such as departments and courses. With this task Foleta is modeled and fitted to a specific school.</td>
</tr>
<tr>
<td>Employees:</td>
<td>In this task employees are appointed. Furthermore they are categorized by competence and financial scale.</td>
</tr>
<tr>
<td>Lessons and tasks:</td>
<td>Lessons and tasks that employees can be appointed to are defined within this task.</td>
</tr>
<tr>
<td>Groups:</td>
<td>With this task different educational tracks are defined and appointed.</td>
</tr>
<tr>
<td>Finance:</td>
<td>In this task users gain financial insight. Financial data is inputted such as school budget.</td>
</tr>
<tr>
<td>Forecast</td>
<td>Data from previous years is combined in these tasks. With this data the application can generate forecasts for upcoming year(s).</td>
</tr>
<tr>
<td>Appointment</td>
<td>After tasks, groups, employees data are placed in the application tasks and lessons can be appointed to employees via this task.</td>
</tr>
</tbody>
</table>
4.3 Overall tasks

After schoolgroup and school level tasks are completed reports can be created to gain insight into the data and create overviews of the data. These reports are an important functionality and are generated by both user levels. The following tasks are therefore applicable to both user levels:

<table>
<thead>
<tr>
<th>Reports</th>
<th>Foleta uses the information from all tasks described previously to create reports about the data. For instance the total amount of budget spend on a specific task. Different reports are generated by the application to gain insight and obtain an overview of the data. These reports are used at school level as well as schoolgroup level.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differences</td>
<td>Employment data of external HRM-systems can be compared to equivalent data in the Foleta application.</td>
</tr>
</tbody>
</table>

In this section we have identified the different tasks in the Foleta application for the different hierarchical levels as part of the second step in our lifecycle. We have listed the different high level tasks that users of the Foleta application use to achieve their goals. Because users performing these tasks may experience usability issues we will test the application on the tasks we have identified in the next chapter. We will be testing the tasks with users, and identify the issues that arise.
User testing

Chapter 5: Testing the Foleta application with users

So far in this research we have seen different types of users and their respective tasks. To evaluate use of tasks and the usability of Foleta we have performed user test sessions. This is the third step in selected lifecycle. In this chapter we discuss the different usability problems that surfaced during these sessions.

To perform the usability evaluation we have interviewed five participants. Having five participants participate in the study is sufficient for identifying most usability problems [9]. These five users were interviewed in their regular working environment behind their own working station in order to test the application in a real user environment. Session took up 60 minutes on average in which the interviewees were asked to think aloud. With this technique we asked users to think out loud as they performed a task or navigated through the system, without the observer interpreting the user before task completion. By verbalizing their thoughts, the test users enable us to understand how they view the system and this allowed the observer to identify the usability problems users encountered. The aim of the described technique is to have users perform a task as they would usually do, and detect problems in the process of performing different tasks.

The think-aloud sessions were audio-recorded and the usability problems that were encountered noted. Interviewees were asked to perform several tasks in the Foleta application and comment on their actions. Users were also asked to comment on several other usability attributes such as readability, help functions and color use. The interview questions for the user testing can be found in Appendix A.

The following issues were discussed with users and will be discussed further individually:

- **User characteristics.** To determine the user type and hierarchy level we asked users about their role in the organization and the role of the Foleta application in the organization. To identify the user type and level of hierarchy we compared the users to the user identification in chapter 3, as a part of step one in the presented lifecycle.

- **Task analysis.** To identify tasks performed by the users we asked users which of the task offered by Foleta are using. To identify the tasks used we compared each high level task identified in chapter 4, as part of step two in the presented lifecycle.

- **Application walkthrough.** Users were asked to perform a task in each of the identified high level tasks to gather usability problems. Walking through the application with users is part of step three in the presented lifecycle.

- **General usability problems.** We asked users open questions about the current user-interface. Topics such as readability and use of color were discussed. These open questions about the application are also a part of step three in the presented lifecycle.
5.1 Participant characteristics

The five participants interviewed have different characteristics, matching different user types as defined Chapter 3. Before the think-aloud session we asked the interviewees several questions to determine several user characteristics. The questions are stated in Appendix B. Table 1 presents the different users and their characteristics. The table shows the number of years a user has been working with the application, whether the person shares usage of the application, which hierarchal level the user operates on and the type of user category, as discussed in Chapter 4.

Table 1 shows that we have interviewed users with different characteristics to find the usability problems in the Foleta application. We have interviewed users that have only recently started using the application up to users that use the application since its release. We have also interviewed users with and without shared responsibility of the application. The spread of these user characteristics allow us to detect usability issues for possible different levels of experience in using the application.

We have only interviewed users who have access on both the Group level and the School level. We do not consider this an issue as we have interviewed an autonomous user. Autonomous users use all levels of the Foleta application. We have furthermore not found the autonomous team user type, which might confirm the statement of Foleta management staff that this is a utopian user level. We have however, not tested with all users and therefore cannot confirm this for all users.

<table>
<thead>
<tr>
<th>User</th>
<th>Usage in years</th>
<th>Shared responsibility</th>
<th>Hierarchy level</th>
<th>User type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>4</td>
<td>No</td>
<td>Schoolgroup</td>
<td>Autonomous user</td>
</tr>
<tr>
<td>User 2</td>
<td>5</td>
<td>Yes</td>
<td>Schoolgroup</td>
<td>Sections</td>
</tr>
<tr>
<td>User 3</td>
<td>4</td>
<td>No</td>
<td>Schoolgroup</td>
<td>Support staff</td>
</tr>
<tr>
<td>User 4</td>
<td>2</td>
<td>Yes</td>
<td>Schoolgroup</td>
<td>Sections</td>
</tr>
<tr>
<td>User 5</td>
<td>1</td>
<td>No</td>
<td>Schoolgroup</td>
<td>Support staff</td>
</tr>
</tbody>
</table>

5.2 Aggregated issues

During the user testing sessions we encountered 34 usability problems. The issues were obtained by analyzing the notes made during the interview and by reviewing the audio recordings. All the issues are listed for each user in Appendix B. We have aggregated the usability findings to create a categorized list of usability issues. This list consists of nine aggregated usability issues that contain all usability issues that surfaced during the user test sessions. Each issue is applicable to one of the aggregated issues presented in this section.

Feedback and error messaging

Users have stated they sometimes do not know why they are unable to perform an action. The system does not provide the users with information why several actions are unable to be performed.

Example: Selection boxes for the (de)activation of sections are deactivated. The users cannot check or uncheck these boxes, as shown in figure 1.5. There is no feedback provided as to why this is the case.
Chapter 6: Testing the Foleta application with users

Update notification

4 out of 5 users stated that new functionalities and updates appear in the application without their knowledge or notification. They do not know what the new functionalities offer. One user was aware of changes due to the Foleta newsletter. Users also stated that support documentation via the helpscreen was not always present for new updates.

Overview

Two of the users used a spreadsheet along side the Foleta application. The users used duplicates of the data in an Excel spreadsheet in which they performed several tasks the Foleta application also offers. Both users felt that the spreadsheet gave them direct insight in the changes they make. In Foleta users need to navigate to a different task to see the changes take effect. In their own spreadsheet, they note, differences are visible immediately.

Removing content

User stated that different faculties, courses and other data couldn't be removed in the application. These records can be hidden if not active but not permanently removed. Users find the unused data redundant.

Users also found that not being able to hide tasks from the dashboard, they felt were not applicable, made the application feel more crowded and the tasks needed, harder to find.

Design inconsistency:

Users stated that within the application different functionalities such as save buttons are not consistently placed on the same location of the page. Furthermore the notification icon that indicates users need to save in order for changes to take effect is not presented consistently.

Flexibility in customization:

Educational changes in the organization need to be implemented in the Foleta application. Changing the number of minutes that is a lesson was found rigid. The Foleta application did not offer an easy solution to these educational changes according to 2 of the users.
None of the interviewed users for the usability evaluation is making use of the search function. Users did not give a clear reason as to why they do not use the search function. The majority of the users do use the help screen function that offers information related to the selected task. The help screen however did not show information in the help screen for several tasks, and users found the button linking to the help screen to be too small and not placed in relation to the content they needed information about.

Users are very positive about the Foleta helpdesk. All users said to have a positive experience with technical support via the telephone, which may explain the absence of search use.

The Foleta application offers several ways of navigating between tasks in the application. 4 out of 5 users say they often return to the dashboard overview, via the breadcrumb menu shown in figure 1.2, before entering a new task. Only 1 of the 5 users makes use of the top buttons, described in figure 1.3, to navigate to tasks and 1 user uses the shortcut navigation in the sidebar. Users found the many possible ways of navigating to be confusing.

The majority of the users find the appearance of the Foleta application not satisfying. 3 out of 5 users responded negative in regards to the application’s appearance. Users stated the application’s appearance is ‘outdated’ and ‘retro’. One user responded neutral about the appearance of Foleta, and one user was mildly positive about the appearance.

Via the user testing sessions we have encountered 34 usability issues, aggregated in this chapter. We have not found crucial problems in which the application failed or the user could not proceed in using the application. The issues found are however usability issues that decrease the ease-of-use for the Foleta application. The aim of this research is to improve the ease-of-use of Foleta. The usability problems found in user test sessions indicate where the system can improve this ease-of-use. In the next session we will be looking further into the usability issues by testing the application on several heuristics.
The last step in our presented lifecycle is the heuristic evaluation. During this evaluation, the evaluator goes through the interface several times and inspects the various dialogue elements. For each screen in the application we try to come up with an opinion about what is bad about the interface. What is bad will be noted as a usability issue. Heuristic evaluation exploits the evaluator’s knowledge of general problems and solutions. The usability issues found will then be compared against the ten usability principles presented by Jakob Nielsen [10], which we will discuss in detail in this chapter. Heuristic evaluations are a cost-effective way of detecting usability problems [4]. The aim of the heuristic evaluation is to identify usability issues. The used heuristics can also be found in Appendix D.

A downside to heuristic evaluation is that the method could be biased by the current mindset of the evaluator and normally does not generate breakthroughs in the evaluated design [11]. To avoid this bias this research performs a heuristic evaluation as well as test the application with users of Foleta, as discussed in Chapter 6.

In this chapter we will discuss the heuristic principles and present related usability issues found. Rather then present all the issues in a list form we present the issues in relation to the principles. For each principle we explain the heuristic and list the related problems found. Matching issues also found in the user test sessions will be discussed. Multiple occurring issues are listed once. The full list of usability issues can be found in Appendix C.

1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time [10]. When a user interacts with the application, they need a visual cue indicating that the system is working or that what they’ve done is either correct or incorrect. The Foleta application does not always provide appropriate feedback to the user. Issues we found in regard to the visibility of system status are:

- The application does not provide feedback for deactivated selection boxes. There is no information as to why these selection boxes are deactivated in the task.
- The system does not provide the user any feedback for being at the dashboard. There is no clear indication present. For all other pages we clearly see which task is selected.
- When downloading reports the application does not let the user know a rapport is being generated.
- Feedback is provided in different ways including alert messages, icons and appearing text. Inconsistency in providing feedback may confuse the user.

The visibility of system status is related to the ‘update notification’ issue found in the user test sessions. Users also said that they sometimes do not know why they are unable to perform a task.

2. Match between system and the real world

The system should speak the users’ language, with words, phrases and concepts familiar to the user, rather than system-oriented terms [10]. Foleta users are domain experts, who are expected to have knowledge of the used terminology in Foleta. Therefore we do not see use of terminology within the application as a usability issue.

Issues we found related to the match between the system and the real world are:

- The filename of reports downloaded are the task number in the application, not the name of the task. These are system oriented terms.
- Users are not able to delete data in school arrangement tasks. The data in the system and the data used by school(groups) does not match.

The match between system and the real world principle does not relate to any issues found in the user test sessions. The application’s users being domain expert may be the reason for this non-relation.
3. User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue [10]. The Foleta application offers many tasks. Navigating to and between tasks should be effective and efficient. There are several ways of getting from point A in the application to point B. The Foleta application offers five ways of navigating. The five ways are:

- **Shortcuts**: The sidebar inside tasks offers shortcuts to related tasks.
- **Top buttons**: The top buttons link to a category of related tasks, shown in the sidebar.
- **Breadcrumb**: The breadcrumb is an active trail, enabling users to go back in hierarchy. The breadcrumb menu provides a trail for the user to follow back to the starting or entry point [15]. Clicking on a school or a schoolgroup level leads to the dashboard. The dashboard shows all of the tasks for group level or school level.
- **Changing school**: By selecting a different school in the dropdown menu in the buttons at the top users can switch schools.
- **Changing year**: By selecting a different year in the dropdown menu right from the breadcrumb users change year.

Having multiple ways of navigation may confuse users and each place for navigation takes up a space inside the application. The top buttons, the breadcrumb trail and the sidebar are shown in each task. This means there is less space for the actual content. The top buttons take up a lot of space but do not lead to more information than the dashboard. Other issues we found in regard to user control and freedom are:

- **Sidebar menu headers have an icon indicating they are clickable whilst they are not**.
- **The breadcrumb link to the current page is active on several pages in the application**.
- **There are multiple ways to select schools on the dashboard. Unused ways of navigating may be redundant**.
- **Clicking top buttons only shows tasks in sidebar**.
- **The application has no back button**.
- **The application has no undo or redo button**.
- **The logo is not an active link. User expect the logo to be an active link [16]**.

The user control and freedom principle relates to two issues found in the user test sessions: removing content and navigation. Users have found they are unable to remove content and use several ways of navigating through the application.

4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing [10]. The consistency of an interactive system strongly affects how quickly its users progress from controlled, consciously monitored, slow operation to automatic, unmonitored, faster operation [12]. When an application is highly consistent, users learn how everything in the system works and its use quickly becomes habitual. User sessions have already shown that the Foleta application has consistency problems. Issues we found related to consistency and standards are:

- **The save and cancel buttons are presented in different places throughout the application**.
- **The style of support documentation on support.foleta.nl is different. The logo, colours and overall style differ from the application**.
- **The date selection calendar is presented in a different style**.
- **The user administration has a different UI**.
- **In several tasks the sidebar does not show**.
- **Several tasks have different style for tables and buttons**.
- **Clicking the print buttons starts a PDF download in some tasks. In all others tasks the browser print function appears**.

The consistency and standards principle relates to the ‘design inconsistency’ issue found in the user test sessions. Users stated that within the application different functionalities are not consistently placed.
Heuristic evaluation
Chapter 6: Comparing the Foleta application to heuristic principles

5. Error prevention
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action [10]. Issues we found in regard to preventing users from making mistakes are:

- Fields accept false and long input.
- Users are able to navigate to menu-items that are not (longer) in use.
- The application asks for a name in the login process while users use an email to log in.
- Users can change their password without confirming their old password.

The error prevention principle relates to none of the user test session issues. This relates to the errors we found. None of the errors we found will, in case of normal use, cause the system to fail. The errors found will only occur when mistakes are made. It is however important that the system prevents these mistakes from happening.

6. Recognition rather than recall
Minimize the user’s memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate [10]. It is easier for users when they are shown options and let to choose among them, rather than being forced to recall options. Instructions for use of the system should be visible or easily retrievable whenever appropriate. Issues we found in regard to recognition rather than recall are:

- Contact information is placed on the dashboard. There is no link to contact information in the application.
- The button linking to the help screen is small and placed unrelated to the task it provides information about.

The recognition rather than recall principle relates to the overview issue in the user test sessions. Two users use a spreadsheet next to the application to improve direct insight into changes.

7. Flexibility and efficiency of use
Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions [10]. Advanced users use systems different than novice users. As novice users get acquainted with the system and perform certain tasks frequently, the system should be able to let those frequently performed tasks be accessible more efficiently. Issues we found in regard to flexibility and efficiency are:

- The application does not offer the possibility of storing login data for a fast login process.
- The application has no undo or redo button.
- The application has no back button.
- The logo is not an active link. Users expect the logo to be an active link [16].

The flexibility and efficiency of use principle relates to the ‘flexibility in customization’ issue in the user test sessions. Implementing educational changes in the application was found rigid.
8. **Aesthetic and minimalist design**

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility [10]. A minimalist design draws the users to focus on the main subject at hand without being distracted by irrelevant images or text. Keep the information displayed on the application simple. Include only relevant information, otherwise the message will be lost in the “noise”. The issues we found in regard to aesthetic and minimalist design:

- The logo used in the Foleta application is an old version of the logo. The current logo, used in support documentation and company wide communication is different from the one used inside the application.
- Reportscreen legends are cramped at the top left of the screen. The margins are minimal. The appearance is not aesthetic.
- URLs and selection options show in the print version of pages. This is redundant information.

The aesthetic and minimalist design principle relates to the ‘design aesthetics’ issue in the user test sessions. The majority of the users found the appearance of the Foleta application not satisfying.

9. **Help users recognize, diagnose and recover from errors**

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution [10]. In regard to helping users with errors we found the following issues:

- The button for helpscreen is small. This issue is also identified as a usability issue in the user test sessions.
- Users who lose their password cannot retrieve this on the login page.

The help users recognize, diagnose and recover from errors principle relates to the ‘feedback and error messaging’ issue in the user test sessions. The system does not provide the users with information why several actions are unable to be performed.

10. **Help and documentation**

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out, and not be too large [10]. The issues regarding to help and documentation we found are:

- The button for helpscreen is very small.
- Contact information is placed on the dashboard. There is no direct link. Users need to navigate to the dashboard which is not the place where contact information is expected to be found.
- Several helpscreens to provide related information about a task were empty.

The help and documentation principle relates to the ‘help and search’ issue in the user test sessions. Users do not use the search function and found that several helpscreens were empty.

In this chapter we have identified several usability issues and related these issues to usability principles. By using the usability principles we have presented the issues in a structured form. However, we were not able to relate several issues to a usability principle. This is due to the fact that we also found functional issues that are not related to a usability principle. The heuristic evaluation did show issues for these principles. Users may not have encountered issues in regard to these principles during the user test sessions. This does not mean they never encounter issues regarding the principles. All usability issues found in the heuristic evaluation can be found in Appendix C.

In relation to the user test sessions we found that only error prevention and the match between system and the real world did not match with any of the issues identified by users. All other issues in the user test sessions were confirmed by the heuristic evaluation. This final step concludes the search for usability issues. We did not find any issues that would cause the application to fail or would not allow the user to continue with performing tasks. The issues we found however do decrease the ease-of-use for the Foleta application. The usability the problems stated in the heuristic evaluation indicate where the system can improve this ease-of-use.
Conclusion

Chapter 7: Concluding the usability evaluation

With this study we aim to improve the usability of the Foleta application. In this conclusion we make several recommendations for improving the usability. The results of this study have indicated that there are several usability issues in the application. If Foleta would implement the recommendations made in this conclusion the ease-of-use of the application will improve.

This study presented a four-step lifecycle for finding usability issues in the Foleta application. In the first step we identified three main levels of hierarchy and identified four user types. In the second step of the research we identified the thirteen categorized tasks users perform in the application. These tasks were related to the identified hierarchy levels in Foleta.

The third and fourth steps were aimed at finding usability issues in the application. In the third step we spoke with users about the application. By walking through the application with users we have presented nine aggregated usability issues. In the fourth step we performed a heuristic evaluation and compared the issues found to ten heuristic principles. The issues found in both steps are usability issues that decrease the ease-of-use for the Foleta application.

The usability issues we identified can possibly affect the user each time they use the application. We recommend Foleta to improve all the issues stated in Appendix B and C. Appendix B contains all the issues found in the user test sessions. Appendix C contains all the issues found in the heuristic evaluation.

We feel however that fixing the issues found would not benefit users of the Foleta application in the long-term. New features and updates would in all probability continue to be added in the same way. This would mean the same type of issues would surface, due to what is in our opinion ad-hoc development. Ad hoc development serves a particular purpose rather than a generic or pre-defined one. By improving structure, re- and predefining places for functionalities Foleta would prevent inconsistency in the application. The many ways of navigating through application show that there is no clear workflow for users in the application. Improving structure will solve design inconsistency and navigation issues.

In this new structure we recommend Foleta to implement specific places in the user interface for notifying users on new updates in the application, for users to obtain feedback form the application and for the help and contact information. The application is inconsistent in feedback to the user. Users say that they don’t always know why they cannot perform an action. The feedback the application does offer is placed differently. The new structure would solve the update notification issue and issues with feedback and error messaging. The new structure will improve the recognition for users; they will know where to look.

We also recommend Foleta to update the application’s design. User sessions have shown that users find the application outdated. Support documentation and the business website show that Foleta has upgraded its corporate identity but this has not been implemented in the application. Improving the design would solve the aesthetics issue.

A website’s appearance is important for the success of a website in attracting potential users’ attention in the first place, where they make an immediate judgment of the attractiveness [13]. For Foleta it is important to note that technology driven features may not be appropriate for information-oriented websites because they may ‘hurt users’ and undermine usability and thereby a business’s profitability. The type of aesthetics that is relevant depends on the application domain [14].

With improving structure and style Foleta would solve all non-functional usability issues. Improving current functionalities and the issues presented in Appendix A and B would also solve the functional issues; the removal of content, flexibility in customization and overview issues. The new structure and style we believe would offer users an easier to use system with improved usability. With that we have reached our goal of improving the usability of Foleta.
A Usability Evaluation of Foleta for use in Secondary Education

References

Appendix A
Research questions

User characteristics:
What is your function within your institution?
For how long is your institution using Foleta?
For how long are you yourself using Foleta?
How often do you use Foleta?
Do others use Foleta, and what is there function?

Task analysis:
[Perform login process]
Which tasks in Foleta do you use often?
Are there tasks you do not or sporadically use?
  Why do you not use this task?
[Walk through task process]
Can you access help?
  How do you access help?
  At what moment do you use help?
[Walk through task process]
Can you navigate to a task you often use?
[Walk through task process*]
  How did you navigate, and why this way?
  Are you running into any problems whilst executing this task?
Do you find any task more or less important?
  Can you separate these tasks?

* Walking through tasks is an iterative process.

Open user input:
What do you think of the colors used in Foleta?
Are you able to read all text, easily?
Do you use the search function?
[perform search task]
Do you find working with the application appealing?
What is your overall opinion on Foleta?
Do you use other applications like Foleta for budget planning and other related activities?
Appendix B
User test sessions results

User 1:
- Step-by-step navigation feels natural.
- No usage of other software for formation planning.
- One person responsible for construction, management and using the application.
- Use of small screen not optimal. When generating reports the reports are not well visible. Pop-ups within reports are not always showing on small screen.
  - Use of help screen easy and helpful.
  - Sober appearance of application.
  - Notification are well visible.
  - No distinction between staff support and general support. In application (OOP and no AOP).
  - No distinction between staff support and board.
  - Colors are pleasant.
  - Icons not noticed. No positive/negative reaction.
  - Changing school years of vital importance for working in the application.

User 2:
- No use of help screen.
- Missing selection multiple staff members in tasks.
- New functionalities not well indicated.
- Misses possibility of hiding tasks.
- When adding temporary expansion of staff initial status preferably not ‘Bruto werktijd factor’.
- Problem with attaching mentor to classes. Mentor lessons depend on faculty.

User 3:
- No use of financial tasks
- No use of prognoses due to educational changes
- Changing lesson duration, application is ‘star’ and makes it difficult to implement these changes. Successfully implemented with ‘opslagfactor’.
  - The application froze temporarily using the top buttons for navigation.
  - ‘Montesori’ concept lasting te implementeren.
  - Old look-and-feel.
  - Never used search function.
  - Uses help screens.
  - Starts preparing for next year in October.
  - Is ‘used to’ the help information being at a task.
  - Imports lessontables from PKB.

User 4:
- Uses a spreadsheet next to Foleta. The spreadsheet provides direct result.
- Groups, courses and formation are not combined in one overview.
- Uses Magister.
- Not easy to instantly see how much formation is needed.
- Uses task costs often.
- No use of financial tasks.
- Shortcuts are new, and take getting use to.
- Deep in the navigation it’s hard to get back.
- Always returns to the dashboard.
- No use of search.
- No use of helpscreens.
- Used prognoses once.
- Great service provided by Foleta staff.
- Comfortable to use, easy to read.
- Descriptions are not consistent.
- Misses the ability to delete data.
Appendix B
User test sessions results

User 5:

- Uses a spreadsheet next to the application. Spreadsheet shows immediate results.
- Unable to delete groups.
- After merging groups redundant data in the application.
- Unable to delete courses.
- No WTF choice if task is relative.
- No use of teams.
- Misses shortcut from S3.1.3 to S3.1.2.
- Reports grouping is not organized.
- Positive about phone support.
- Application style is not appealing but straightforward.
Appendix C
Heuristic evaluation results

General:
- Small question mark for help screen.
- Inactive logo.
- Save in different places. Sometimes bottom of the screen, sometimes only static down, or static top.
- Logo not the same as identity.
- Things that can not or should not, clear explanation is lacking.
- Fields accept incorrect entry (example BAPO 99999).
- Date selection calendar different style.
- Question mark on floppy, indication change (FF, Chrome).
- Names of downloaded pdf documents are meaningless. Only menu name.
- Sub-headers have double arrow on the right side menu but are not clickable.
- No more existing menu items remain displayed.
- Position legend for reports.
- Breadcrumb link to current menu is active.

Login:
- Name should be email.
- No password forgot option.
- System does not remember login.

G:
- Unclear user is in the system (breadcrumb, menu, status)?
- Multiple options for school selection. Possibly redundant.
- Sub-header contains links while presented as a text paragraph.
- No clear hierarchy what needs to be done first (School > Schoolyear).
- Clicking on large buttons G1 also leads to G0.

G1.0: Difference in design. User administration jquery UI. Change password as rest of the system. User Administration menu double as sitemap as a tabs.
G1.0.6 Change Password without old password confirmation.
G1.1.1: Overflow text first column in print. Shows active functions "appears in print.
G1.1.2: Required field jumps. No delete option. Alphabet selection appears in print. 'Inactive' appears in print.
G1.1.3: No delete option. 'Inactive' appears in print.
G1.1.7: 'Inactive' appears in print.
G1.1.8: Date selection is not consistent with layout. No filter.
G2.1.3: Lacks side menu.
G2.1.4: Lacks side menu.
G2.1.5: Plus and remove icons in different style.

S:
- Clicking S.2.1 in breadcrumb trail leads to S2.1.1. Misses another S2.1 items.
- Save buttons presented below.
- Not clear why active items can not be made inactive.
- Items not removable.
S1.0.0: Difference in design. User administration jquery UI. Change password as rest of the system. User Administration menu double as sitemap as a tabs.
S1.0.4: See 1.0.0.
S1.0.5: Difference in UI design from rest of the system.
S1.1.1: Items not removable. Save at the bottom.
S1.1.2: Import Module button below. Save at the bottom.
S1.1.3: Items not removable. Save at the bottom.
S1.1.4: Import Module now top and bottom. Add box below. Save at the bottom. Button to s2.2.6. at bottom of the screen, and in the sidebar.
S1.1.7: No sidebar
S1.1.8: Buttons below.
Appendix c
Heuristic evaluation results

S.2.1.1: Baporecht in overview incorrectly positioned. Floppy hover shows questionmark. Calendar in different style.
S.2.1.2: Import module also contains the export functions.
S.2.1.3: Hover name shows questionmark. Shortcuts right unclear. Buttons close together. Exportation function unclear.
S.2.2.1: Date display periods unclear. Display minutes lessons per week, etc. after label do not support input.
S.2.2.3: School storage factor often repeated.
S.2.2.4: Totals above and below the table. Possibly redundant.
S.2.2.5: Opslagfactor shown many times.
S.2.2.6: Not clear that there should be a choice for a training made.
S.2.2.7: Correct while no longer applicable.
S.2.2.8: Exportation button unclear. Adjust s / t with double click. Changes red button are inconsistent.
S.2.2.9: See 2.2.8.
S.2.2.10: See 2.2.8.
S.2.2.11: Other layout table. Koppeling task, not clear first to first select existing task.
S.2.2.12: See 2.2.8.
S.2.2.13: Used in S3.1.3. not clear that list will appear, no ‘show’.
S.2.3.1 filter and filter on / off not grouped. Floppy if change from red to other functions.
S.2.3.2 Not visible in Chrome. Department (Nothing, everything) unclear. / * Faint * / Towing is unclear.
S.2.4.1 Som under green stripe. Print: Which report and date below.
S.2.4.2 Print buttons differently. All schools declared export to XLS, no notification.
S.2.4.3 Print button is export to PDF.
S.2.5 Other colors than similar portable screens. Shortcut top-right unclear.
S.3.1.2 Export functions are not clearly marked.
S.3.1.3: Print button is export. Table is not clickable, despite hover. Legend to report and filtering different from other reports. Different icons with new meaning.
S.3.1.4: Print report contains web data. New shortcut (S3.1.3).
S.3.1.5: Employees as links, colored blue.
S.3.1.7: Save differently formulated.
S.3.1.8: Design staff list other than outside reports. Show year job is export PDF.

S4:
- When generating not clear having to waited for exportation process.
- Names PDF and XLS are not consequent made. Sometimes with, sometimes without title.
<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visibility of system status</td>
<td>The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.</td>
</tr>
<tr>
<td>Match between system and the real</td>
<td>The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.</td>
</tr>
<tr>
<td>world</td>
<td>Users often choose system functions by mistake and will need a clearly marked &quot;emergency exit&quot; to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.</td>
</tr>
<tr>
<td>User control and freedom</td>
<td>Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.</td>
</tr>
<tr>
<td>Consistency and standards</td>
<td>Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.</td>
</tr>
<tr>
<td>Error prevention</td>
<td>Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.</td>
</tr>
<tr>
<td>Recognition rather than recall</td>
<td>Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.</td>
</tr>
<tr>
<td>Flexibility and efficiency of use</td>
<td>Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.</td>
</tr>
<tr>
<td>Aesthetic and minimalist design</td>
<td>Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.</td>
</tr>
<tr>
<td>Help users recognize, diagnose,</td>
<td>Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.</td>
</tr>
<tr>
<td>and recover from errors</td>
<td>Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.</td>
</tr>
<tr>
<td>Help and documentation</td>
<td>Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.</td>
</tr>
</tbody>
</table>
A USABILITY EVALUATION OF THE FOLETA APPLICATION FOR USE IN SECONDARY EDUCATION

BACHELOR PROJECT IMM