Design Project

Department of Computer Science

Design Report

Author:
Pieter Bos
Iris Heerlien
Kimberly Hengst
Sophie Lathouwers
Thomas Raaijen

Supervisor:
Klaas Sikkel

April 14, 2016

UNIVERSITY OF TWENTE.

Abstract

This report is written as part of the "Design project" course at the University of Twente. A system called "BOZPlanner" has been developed for the educational affairs department of the faculty of Behavioural, Management and Social sciences. The development of this system was motivated by the following goal: *"To develop a system that can be used by the educational affairs department for a resource planning of students for taking minutes during programme meetings."*. This report elaborates on the design process of the developed system which enhances awareness of the resource planning of the deployed students for a programme meeting. It describes all phases of the project work such as formulating requirements, elaborating on design choices, and discussing the architectural design. The system is designed with a strong emphasis on its usability in order to aid the work processes of educational affairs department employees. In the future this system will run a pilot at the faculty of Behavioural, Management and Social sciences as it might be deployed at other educational affairs departments at the University of Twente in the future.

Contents

Chapter 1

INTRODUCTION

To the University of Twente, knowledge is the most treasured asset. Knowledge can only be created through education and practice. In the academic year of 2013-2014, the university introduced a new educational model, the "Twents Onderwijs Model" (TOM). Therefore, the bachelor studies at the universities have changed drastically from that year onwards.

A high level of education is needed in order to attract talented scholars towards the university. In order to maintain this high level of education, its education should be monitored and evaluated thoroughly. A programme committee (OLC) is tasked to do just that, therefore the OLC gathers monthly to discuss current developments within the programme. During the meetings, minutes are taken which describe what has been said, which decisions have been made and what measures will be taken.

In the past, members of the educational affairs department (BOZ) took the minutes during the meetings. Last year (in 2015), students were asked to take over this task. Due to their experience and knowledge of their own programme, students are easily able to take minutes with the advantage for the university of paying lower (student) wages.

In the current situation, the process of finding students to take the minutes during a meeting is labour intensive and lacks instruments that provide information about whether students are going to take minutes at a meeting. In addition, after the students attend such meetings, they have to finalise their taken minutes. Students are fairly free to plan whenever they want to finalise

these minutes. Due to a lack of awareness of the different parties involved in these meetings, this can result in undesirable delays. As there is no overview, there is hardly clarity about whether minutes have been submitted or not.

IT can enhance these frail processes and puts forward opportunities of improving the minute taking process. From the perspective of BOZ, an information system can improve the overview of planned meetings, the overview of assigned students and it informs BOZ whether minutes are already uploaded or not. Moreover an information system can make it easier to schedule a meeting, because it can notify everybody involved in the yet to be scheduled meeting. Students can be aided by making it easier to enrol for a meeting and upload the taken minutes, as students do not have to manually notify everybody by mailing the involved parties.

In Chapter 2, the domain in which the system is situated is analysed. In Chapter 3, the proposal of this system is discussed, illustrating how this fits in Agile practices. In Chapter 4, the requirements specification methodologies are explained and the requirements are provided based on the needs of the involved stakeholders. In Chapter 5, the results of the requirement analysis are provided and elaborated. In Chapter 6, the global design of the system is discussed, an overview of the system is given, architectural design choices are explained, and the purpose of the system is elaborated. In Chapter 7, the detailed design discusses different components within the system, how design choices altered the implementation of the system and how these are justified. In Chapter 8, the approach of testing the system is described in an elaborative plan and its results are provided. In Chapter 9, the future plans for the system are discussed in terms of maintenance and wider enrolment at the university. Finally, in Chapter 10, this design project is evaluated and its final conclusions are provided and discussed.

Chapter 2

DOMAIN ANALYSIS

In this chapter, the process of identifying the domain of the system is discussed. The view of the client and users on a supportive system is issued from an 'outside-in' perspective [Vijverberg and Opdenakker(2013)], such that the existing problem is explained in more detail. By understanding the domain, the development will proceed more swiftly and it aids the planning for future development.

2.1 Introduction to the Domain

The domain in which the system is introduced concerns the procedure of finding a secretary to take minutes during meetings of programme committees and to provide the possibility to schedule meetings in the planning to which the secretaries can subscribe themselves and providing the opportunity to easily submit taken minutes. Thus, the system should support both BOZ and secretaries in the procedure of taking minutes for the programme committees.

2.2 General Knowledge of the Domain

The University of Twente has different BOZ departments for each faculty. Each faculty has its own field of expertise, meaning that the requirements for a supportive system may differ for each BOZ department. Understanding these differences in the domain can determine whether the system can successfully be deployed on the whole university.

2.3 Client, Users and interested Parties

The system is developed for the parties involved during the planning procedures of a programme committee meeting. The client of this project is BOZ as department within the faculty BMS, however the system will be used by employees of BOZ within the different faculties and the secretaries associated with these faculties after the pilot has been successful.

BOZ employees benefit from being more aware about the status of a planned meeting, while secretaries can easily subscribe to a meeting and easily upload taken minutes. These two users differ heavily in their available knowledge regarding the use of IT systems, this affects the requirements for the system and its design. Parties such as the programme committee and the IT department of the university (LISA) are interested in the development of the system due to respectively their involvement in the meetings or the maintenance of the system. Please refer to Figure A.1 in Appendix A for all stakeholders within this domain.

2.4 Software Environment

In the first meeting with the client, they made clear that they do not have any requirements concerning technologies used. Therefore, the choice was made to utilize technologies which are common to use in this kind of project. The server on which the system is running has Linux installed. Django is utilized as web framework [Django Software Foundation(2015-2016)] and the system is programmed in Python [Python Software Foundation(2016)]. More detailed information about these technologies and the choice to use these can be found in section 6.2.

2.5 Procedures of the current situation

In the current situation, BOZ receives a request from a programme committee to schedule a meeting and BOZ then needs to send e-mails to students who need to reply to subscribe as a secretary to these meetings. This procedure is labour intensive and lacks instruments to make sure that there actually is a student attending the meeting from the perspective of BOZ. Students do not have an overview of meetings which they attend or which they even have to be aware of, or they need to find older e-mails that have been sent by the BOZ. The current procedure is depicted in Figure 2.1.

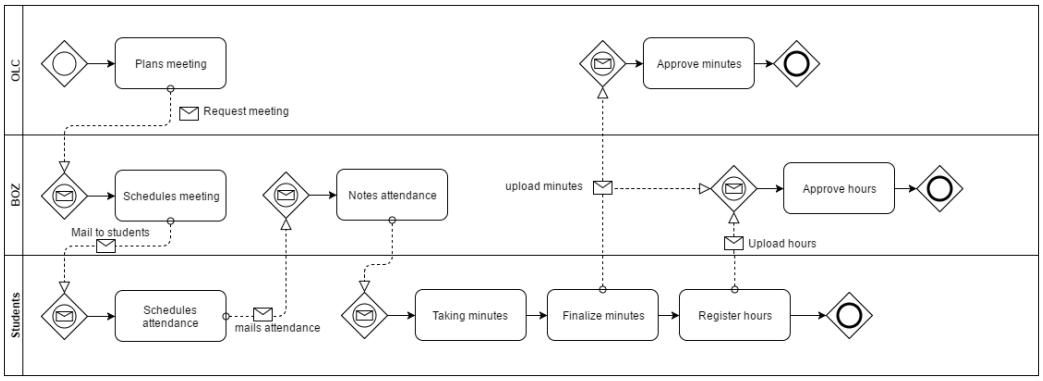

Figure 2.1: The current procedures of taking minutes during OLC meetings

2.6 Commonalities of UT Software

The users of the system also use other software deployed on the University of Twente. Therefore, the system bears resemblance to the systems Osiris and BlackBoard, by utilizing UT sign-on credentials and by using similar designs between these systems. These commonalities should make the system familiar for users to work with.

2.7 Conclusions

By analysing the domain in which the system is developed, the current processes were identified and the necessity of a supportive system is indicated. Moreover, the roles of all stakeholders are identified and taken into account in the development.

Chapter 3

SYSTEM PROPOSAL

In this chapter, the proposal of the system towards the client is explained by discussing different proposing phases during the project. In chronological order, these proposals are a requirement proposal, a mock-up proposal, a system introduction, a usability testing phase and a final presentation of the system to the client.

3.1 Requirements Proposal

At the start of the project, meetings were held with the client to determine the scope of the project. In these meetings, questions were asked to determine the requirements the client had for the system. The client did not really have a system envisioned, therefore there was room for own ideas. This also resulted in few requirements proposed by the client. For most questions different options were provided and the client had to express their preference. The requirements were used as a guideline for the following phase. Note that new requirements can emerge in every stage of an Agile development project and these were also proposed to the client in later stages.

3.2 Mock-ups Proposal

In this phase, the requirements were used to create different designs. These designs incorporated the requirements formulated in the first phase. The mock-ups were presented to the client, and the client was able to consider the different options depicted in the mock-ups. When the client finished considering the different options, a meeting was held in which the options were discussed. In this meeting, the opinion of the client was asked concerning the lay-out of the mock-ups and questions the client had about the designs were answered. The client also had the opportunity to set more requirements and ideas for possible improvement were explored.

More information about the received feedback can be found in Section 3.6. A selection of the different presented mock-ups can be found in Appendix B.

3.3 System Introduction

The system was introduced to the client halfway the development of the system. The client got to interact with the system on a laptop and had the opportunity to experience the system by clicking through all the elements. This allowed the client to get a feeling for the system and to test if the given system met their expectations. The client also had the opportunity to

express ideas for possible improvement which were included in the formulated requirements.

More information about the received feedback can be found in Section 3.6.

3.4 Usability Tests

The usability tests were performed with employees of the BOZ BMS. Test scenarios were created that the employees had to walk through. Each employee received a test scenario which they needed to follow while a tester observed the test process. The scenarios were created as little stories which incorporated use cases that could happen daily, therefore this was a valid way of testing the daily interaction with the system. Before testing, it was made clear to the employees that they were free to comment or criticize the system.

The tester observed the test and noted any comment or detail that stood out. Details could, for example, be confusion as to where an element was located.

The results of this test are described in Section 8.2.3.

3.5 Proposal Presentation

At the end of the project, the system was presented to the management of BOZ BMS. This presentation showed them the added value of the system regarding the process of planning resources for OLC meetings. It also demonstrated the resulting application to employees of other BOZ departments as the program will possibly be used for other departments.

3.6 Results of meetings

The general meetings resulted in useful feedback.

The mock-up meeting proved very useful as the client got a better impression of what the system could look like. It enabled the client to provide more detailed information about their requirements and if the mock-ups matched their expectations. In this meeting, the client expressed their preference for a combination of two given mock-ups. They also mentioned that they would like the design of the system to be similar to the design of Osiris.

The system introduction proved very useful as the client got to click through the system which gave them a good impression of what the system would look like. It enabled the client to give more specific feedback if they thought an element did not make sense or was confusing. The client was very content with the presented result and had no feedback. The only remark that was given was that the presented system matched their envisioned system.

Chapter 4

REQUIREMENT SPECIFICATION

In this chapter, the process of specifying the requirements of the system is discussed. The requirement specification is guided by Agile approaches and practices, the iterative loops of development also occur in the requirement specification. The requirements need to be clearly stated when specifying them, therefore several techniques are used in order to identify the requirements.

4.1 Agile Project Management Approaches for Requirement Specification

As previously mentioned, Agile practices are used in order to manage the project [Kent Beck, et al.(2001)]. Therefore, the requirements for the system are specified by iterative loops, ensuring that the system contains all functionalities that the users need. For this project, the framework or approach of SCRUM is chosen [Scrum.Org and ScrumInc(2004)]. This results in smaller development phases in which the requirements are validated, formulated and implemented.

4.2 Requirement Formulation

The requirements are formulated according to the SMART guidelines [Mannion and Keepence(1995)], this method guides the formulation of the requirements such that they are specific, measurable, acceptable, reasonable and time bound. This results in clearly identified requirements such that interpretation biases do not exist. The formulation of the requirements is essential when testing the developed functionalities, due to the clear definition.

4.3 Requirement Prioritization

The SMART formulated requirements are prioritized based on the importance of the associated functionality and its impact for important stakeholders, whom should be managed to prioritize the requirements of the project, as illustrated by construction projects in Sweden [Olander and Landin(2005)]. The prioritization of the requirements is guided by the MoSCoW method. This method prioritizes requirements based on the importance of their existence in the system. MoSCoW has been proven to be a useful tool to prioritize requirements [Qiao(2009)].

4.4 Stakeholder Requirements and System Requirements

In order to build upon the analysis of the stakeholders, as mentioned in Chapter 2, the requirements are formulated according to the demands of the stakeholders. These requirements are called the 'stakeholder requirements'. In order to fulfil these requirements urged by the stakeholders, 'system requirements' are identified in order to formulate which functionalities the system should contain.

4.5 Conclusions

These techniques as part of the system engineering practices help to identify the requirements of the system. A grasp on Agile practices is not lost in the process due to the practices that facilitate a structural approach for project execution. Please refer to the Appendix C for the specified requirements.

Chapter 5

REQUIREMENT ANALYSIS

In this chapter, an analysis of the specified requirements is provided. The two discussed levels of requirements for the system are; user requirements and system requirements. The measures to which these requirements are analysed are whether they meet the user's needs, or its accuracy or speed of expediting requests of the containing functionalities.

5.1 Stakeholder Requirements

1. As secretary I want to log in and out of the system
   A secretary has to be able to log in and out to use the system.
2. As secretary I want to sign up for an OLC meeting
   The most important action a secretary has to perform in the system is that secretaries have to sign up for an OLC meeting for taking minutes at that meeting. When this fails, the system is not fulfilling its purpose anymore.
3. As secretary I want to sign out of an OLC meeting
   A secretary is able to sign out of a meeting. If the meeting is within ten days, this is not possible anymore to make sure that BOZ does not have to look for a secretary on short notice.
4. As secretary I want to upload the taken minutes
   When the secretary has finished the minutes, (s)he has to be able to upload them in the system. BOZ will check if the minutes are available and if so, they will approve the hours worked to make sure that the secretary will get paid.
5. As secretary I want to get overview e-mails about planned meetings for my programme
   Secretaries are able to receive overview e-mails by activating this option on the preferences page.
6. As BOZ I want to log in and out of the system
   BOZ employees have to be able to log in and out to use the system.
7. As BOZ I want to schedule a meeting in the planning
   BOZ has to be able to add a meeting in the system. When a meeting has been added, secretaries are able to sign up for it.
8. As BOZ I want to see whether secretaries can still sign up for a meeting
   To preserve all the aspects of the overview, BOZ has to be able to know whether secretaries are still able to sign up for a meeting. This is done by making sure that secretaries can sign up until the meeting takes place, to get the highest chance of having a secretary at the meeting. When the meeting is within 10 days, secretaries are not able to sign out

anymore.

9. As BOZ I want to see which secretary has signed up for a meeting
   To keep the overview BOZ also has to know which secretary will be taking minutes at a meeting. This is done by showing the secretaries' name in the meetings overview.

10. As BOZ I would like to receive reminder e-mails about scheduled meetings
    BOZ employees can choose to receive reminder e-mails about scheduled meetings on the preferences page.

11. As BOZ I want to get notified when minutes are uploaded
    When minutes are uploaded, BOZ has to confirm the hours declared by the student and make sure they get paid. Therefore, a notification could be sent to BOZ.

12. As BOZ I want to get updated about all meetings (overview mailing)
    To help BOZ to get a better overview, BOZ will be updated every week about the upcoming meetings of the next three weeks.

13. As BOZ I want to be able to change the secretary attending the meeting
    When a secretary is not able to attend a meeting and this is noted less than 10 days before the meeting, the student is not able to sign out themselves. He has to ask BOZ to do so, to make it less attractive to sign out in short notice.

14. As BOZ I want to be able to download the taken minutes
    When minutes are uploaded, BOZ has to be able to download them to check and verify the minutes.

15. As BOZ I want to be able to change or cancel a meeting
    When a meeting is cancelled or will take place at another location or time, BOZ has to be able to change this in the system.

16. As a user I would like to change my preferred page size
    Users have provided feedback about the small size of the pages, making it difficult to read them. Therefore, the users desire a zoom-in functionality which can be activated on the preferences page.

17. As a user I would like to export my scheduled meetings to my preferred calendar

Users desire to export planned meetings to their personal calendar in order to have a better overview of the meetings planned. They can be exported to Outlook, Google Calendar and Apple Calendar.

5.2 System Requirements

1. The system must have a mechanism which enables secretaries to log in and out of the system
   The mechanism which is used is the University of Twente login. This is easy to understand and the users do not have to remember another username and password. The login may take a maximum of 30 seconds from the moment the user clicks the login button. The accuracy has to be very high, if someone logs in and gets to see another profile, the system made a big mistake at the security level.

2. The system must have a mechanism which enables secretaries to sign up for an OLC meeting
   On the meetings page the secretaries are able to sign in by clicking on the toggle button. If the button has turned green, the secretary has signed up for the meeting belonging to the toggle button. When the toggle button is clicked it may take a maximum of 30 seconds to switch colours. The accuracy has to be very high, otherwise the user gets frustrated and the system misses its goal.

3. The system must have a mechanism which enables secretaries to sign out of an OLC meeting for which they were supposed to take minutes
   If a secretary is not available anymore they are able to sign out by themselves ten days or more before the day of the meeting. This can be done by clicking the toggle button as described in the previous requirement. The accuracy, speed and priority are also the same as the previous requirement. If the meeting is within ten days, the secretary has to ask a BOZ employee to sign them out.

4. The system must have an option for secretaries to receive overview e-mail about planned meetings
   The preferences page contains a check-box which enables secretaries to receive overview mails about planned meetings. The priority in terms

of MoSCoW of this requirement is a must.

5. The system must have a mechanism which enables BOZ employees to log in and out of the system.
   This requirement has the same procedure, priority, accuracy and speed as the first system requirement.

6. The system must have a mechanism which enables BOZ to schedule a meeting in the planning
   BOZ has to click on the add meeting button and fill in the form shown to add a meeting. It may take up to 5 minutes to schedule a new meeting, so from the moment the add meeting button is clicked until the meeting is added. The accuracy is moderate, because when something goes wrong, BOZ is able to edit the meeting.

7. The system must show whether the sign up for a meeting is open or closed
   Meetings are always open to sign up for, but within ten days before the meeting secretaries are not able to sign out anymore. The speed is ten seconds for the time it takes for a user to see whether the sign up is still open.

8. The system must show BOZ which secretary signed up for a meeting
   BOZ has to keep an overview of which secretary is signed up for a meeting. This can be seen on the meetings page for every meeting. The time it takes for the user to see who will take minutes may take up to 10 seconds. Without this requirement, the overview is reduced and thus the goal of the system is partly missed.

9. The system must enable BOZ to change the secretary for a specific meeting
   If a secretary is not available anymore within ten days before the day of the meeting, the secretary has to ask a BOZ employee to sign them out. This can be achieved in the edit form of a meeting. The process to understand the mechanism may take up to 30 seconds.

10. The system must have a mechanism which enables BOZ to change or cancel a meeting
    BOZ is able to change a meeting by clicking on the pencil icon next to a meeting on the meetings page and is able to cancel a meeting by clicking on the cross icon next to the pencil icon. The time it takes to

change or cancel a meeting may take up to 5 minutes.

11. The system must have a check-box which can be checked to zoom-in the pages
    The users of the system did not find it easy to read the pages of the system, this stressed the need of having a functionality to zoom-in the pages.

12. The system should have a mechanism to upload minutes
    The secretary can click on the upload button next to a meeting at the minutes page to upload minutes. This may take up to one minute for the user to understand the mechanism.

13. The system should save uploaded minutes in its internal storage
    The time it takes to upload minutes may take up to one minute as well. The accuracy is moderate, if the user uploaded a wrong file, a BOZ employee is able to delete it and the secretary can upload a new file.

14. The system should only show meetings for the specific faculty I am associated with
    The systems shows only the meetings for the specific faculty the secretary is associated with on the meetings - and minutes page. This is done to keep a better overview. The criterium is the successfulness of handling which has to pass to let this part of the system work.

15. The system should send weekly email updates to the BOZ employees informing them about the planning
    The system automatically sends the BOZ employees an email which lists the meetings in the next three weeks. This mail has to be received every Monday morning.

16. The system should have a mechanism which enables BOZ to download uploaded minutes
    BOZ has to take a look at uploaded minutes to determine if they are good enough. The minutes can be downloaded on the "Minutes" page using a menu on the right side of each meeting. The process of understanding the mechanism may take up to 30 seconds.

17. The system should have three options which can be chosen to export your calendar to (Microsoft Outlook, Google Calendar and Apple Calendar)

The system provides the possibility to select the preferred calendar.

18. The system could notify the BOZ that the OLC has requested a meeting
    There was no time left, therefore this requirement is not implemented.

19. The system could send e-mails to remind BOZ employees about upcoming meetings
    The system enables downloading a calendar file which can be added to the calendar, this functionality aids the usability of the entire system. The functionality does not have criteria, however tests have indicated that these calendars can be downloaded implying that this requirement is met.
    This requirement is also implemented using the weekly e-mails with the upcoming meetings.

20. The system could send a notification to BOZ when minutes are uploaded
    There was no time left, therefore this requirement is not implemented.

21. The system could have a mechanism which consistently asks BOZ whether they are sure of proceeding with the changing or cancellation of a meeting
    In case BOZ makes a mistake in changing or cancellation of a meeting, he or she has to have the possibility for a rollback. This is achieved by showing pop up screens if BOZ clicks the cross icon which asks if the user is sure about the cancellation.
    The response time of the pop up asking if the user wants to proceed may take up to five seconds. Although the priority of this requirement in terms of MoSCoW is a could, the requirement is implemented.
    The mechanism is not implemented for editing, because by clicking the editing button again, the error can be resolved.

22. The system could notify all parties of a change considering a meeting
    This requirement is not implemented in the system such that it happens automatically. However, BOZ has the possibility to select a particular group to send an e-mail when there is a change to a meeting.

Chapter 6

GLOBAL AND ARCHITECTURAL DESIGN

In this chapter, the global and architectural design choices are explained, justified and the system structure is discussed on a high level. Moreover, an overview of the system is provided elaborating on the system pages and overall functionalities.

6.1 Global Design Choices

The engineering of a system such as this system for BOZ is meant to reduce demanding labour-intensive work processes, it also re-designs the flow of the work processes and its associated procedures. In Chapter 2, the work processes and procedures are identified and analysed, unveiling directions in which the process can be enhanced. Moreover, analysing the client provoked to search for the key design pillars or the system's Unique Selling Points (USPs) in order to adapt the system to the end user's needs.

6.1.1 Key Design Pillars

The designing of the system serves to enhance work processes and procedures while making the system as intuitively usable as possible for the end user. To that extent, the system is primarily designed to enhance its usability and ease of use. In order to develop an easy-to-use system, the users of the system are preferred to be closely involved. The practice of involving the users is also known as Participatory Design [Greenbaum and Loi(2012)]. During the project the users were provided mock-up designs [Brandt(2007)], they were asked to walk-through use scenarios [Österman et al.(2016)Österman, Berlin, and Bligård] and they were interviewed in order to receive fruitful feedback from the users and strengthen the relationship with the client - this is also known as Customer Intimacy. Furthermore, the system is developed with regard to future enhancements of the system. The modular design serves to simplify the process of adding new functionalities to the existing system.

Therefore the USPs of the system are; its usability, ease of use, the customer intimacy and its modular design.

6.1.2 Revised Work Processes

The development of the system serves to re-design the work processes for the employees of BOZ, by automatising work procedures of these employees and providing an overview. The revised work processes are depicted in Figure 6.1.

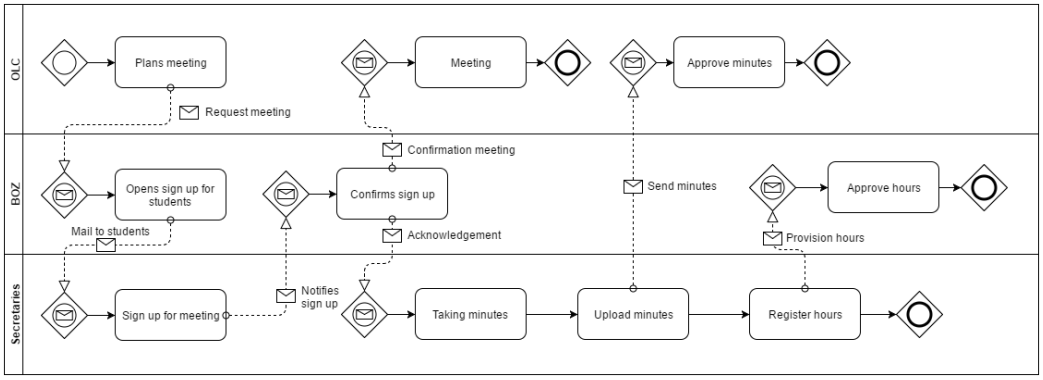

Figure 6.1: The revised procedures of taking minutes during OLC meetings

6.2 Preliminary Design Choices

Conducting a project starts with making well-advised preliminary design choices. These choices consider the expertise and skills of the project team. Hinge on the expertise and skills the programming environment has been contrived, considering the programming languages, used frameworks and libraries.

6.2.1 Programming Language

Python has been chosen as the programming language for this project. Python is known as an easy to learn programming language and is known to be used often for the back-end of a web application. Examples of famous web applications which have been made with help of Python are Dropbox, Pinterest and Instagram. Several team members already had more experience with developing applications in this language which is preferable to a language that no one is familiar with. Another option was to use Java however this has not been chosen as it is verbose compared to Python. Python is very concise which enables quicker development, something considered important as time is a constrained factor of this project.

6.2.2 Frameworks and Libraries

Django has been chosen as high-level web framework. Django is known to be well-documented, has a large support community and is exceedingly scalable. Scalability was very important considering the system might be deployed throughout the whole university. Django encourages rapid development and clean, pragmatic design due to the many features that are already available. Several team members were already familiar with this web framework which was preferable to a framework that no one in the team was familiar with. Django also provides great security features, preventing common attack methods such as SQL injection, CSRF and XSS by default.

The library DjangoSAML2 was chosen to facilitate authentication for students and employees via the university, since SAML is the authentication method that is offered by the university for external software projects. The library integrates with the Django authentication system with little configuration, which makes checking for permissions and group membership easier.

6.2.3 Architectural Design Choices

An important design decision to make was whether to use function-based or class-based views. Class-based views enable the programmer to implement views as Python objects rather than in function-based views where they are implemented as functions. Class-based views allow the separation for different HTTP methods (GET, POST, etc.) in different methods instead of using conditional branching which is needed in function-based views. Class-based views can also inherit from each other, thus code can become a reusable component. Django also comes with several generic views which abstract common patterns in view development, an example is that Django includes views for all CRUD (Create, Read, Update, Delete) actions, which prevents the programmer from writing similar code. Therefore the choice has been made to use class-based views.

The next design choice deals with naming convention of urls. The standard naming of urls is “app:verb_model”. The prefix “app” refers to the name of the application in the Django project, this can for example be “meetings” or “members”. This naming convention is enforced by Django. The choice for the suffix “verb_model” has been chosen because the pages are often

made for an action. In English you would describe these actions like “change a meeting” or “add an organisation”. Therefore urls will be named like “change_meeting” and “add_organization”.

6.3 System Overview

Now that the architectural structure of the system is defined by the previous section, the different components contained in the system can be discussed based on the requirements discussed in Chapters 4 & 5 and stated in the Appendix. These components serve to aid the user experience and are designed to enhance the ease of use. In the following paragraphs, these components are identified and elaborated.

6.3.1 Meetings Page

Firstly, it is needed to create or plan a new meeting. For this purpose, the meetings page is created. On this page, the scheduled meetings are listed in a table in which users are able to search and change preferred number of items shown in the table of meetings. Users with the correct permissions can add meetings, delete meetings, edit meetings or can subscribe to meetings. This page is meant to provide an overview of upcoming meetings and to develop awareness of their current state.

6.3.2 Minutes Page

After meetings have taken place, the meetings are transferred to the minutes page. Minutes can be added to these meetings by the secretary who has attended the meeting. The system allows multiple files to be uploaded for a single meeting, meaning that minutes can be assessed, revised and approved. Minutes can also be deleted by a BOZ employee if asked by the secretary when the uploaded file was incorrect or did not suffice. When one or multiple minutes are uploaded for a meeting, these can be downloaded by the secretary themselves and BOZ employees.

6.3.3 Users Page

The system considers four unique groups of users. These are 1) administrators, 2) planners, 3) secretaries and 4) user managers. These groups of users enjoy different permissions in the system, such as creating meetings and deleting users. When the user is allowed to, the users page provides the possibilities to add users to the system by simply providing their employee number or student number. Moreover, this page provides an overview of all users of the system, including their email address and their associated organizations - please refer to the Paragraph 6.3.4 for more information on organizations. The users can be selected for contacting purposes such as mailing them.

6.3.4 Organizations Page

BOZ would like to deploy the system throughout the whole university, therefore it is necessary to distinguish different organizations for which minutes need to be taken. These organizations are 1) faculties of the university and, 2) programmes within these faculties. The organizations page enables users with correct permissions to add organizations to the system. Organizations can also have a parent organization such as the faculty of BMS is for International Business Administration, these parent organizations should be provided in order to correctly identify, for example, whom may take minutes for certain programme committees.

6.3.5 Help & Preferences Pages

The system is designed with an emphasis on usability and user experience. Respectively, the help page enhances the usability of the system by helping the users recover from problems when utilizing the system, such as not knowing where needed information can be found. The help page provides information about the different pages discussed above and helps to indicate which functionalities can be found on these pages. The preferences page supports adaptation of the system to the preferences of the users and therefore, enhancing the user experience. It provides options to change the size of the content on the pages, to set preferred notifications settings and to export

the agenda to your personal Microsoft Outlook, Google Calendar and Apple Calendar.

Chapter 7

DETAILED DESIGN

In this chapter, the technicalities of the system and its descriptions are provided and explained. Furthermore, the design choices on a lower level are identified, explained and their justification is given.

7.1 System Description

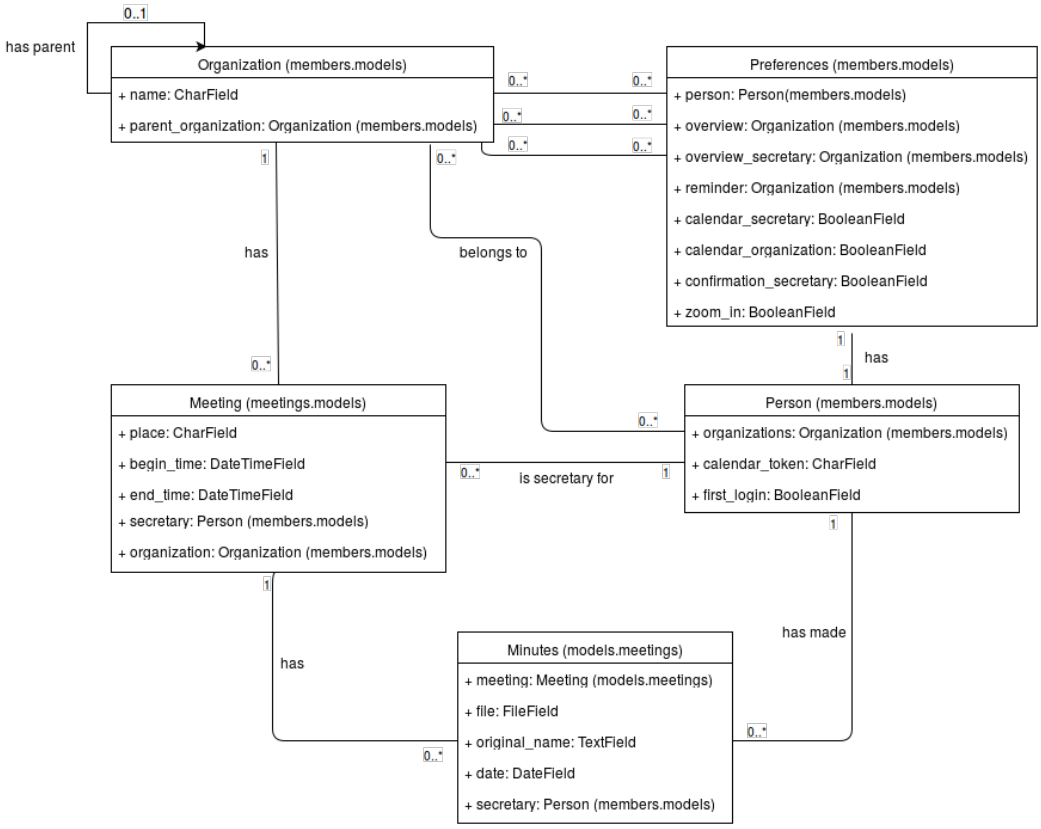

Figure 7.1: Class diagram representing classes made for this project

In figure 7.1 five classes can be observed which represent part of the data that is stored in the database. These are specific classes developed for this project. The class "Organization" represents a body of the university for example the study European Public Administration or the faculty of Behavioural, Management and Social sciences. An organization has a name and may have a parent organization. For example "European Public Administration" can be the name of an organization of which the parent organization is "BMS" (faculty of Behavioural, Management and Social Sciences).

The class "Meeting" represents a programme meeting (OLC meeting). This

object is characterised by a begin date & time, end date & time, location, organization and possibly a secretary. For example a programme meeting for "European Public Administration" takes place on the 8th of April from 10:30 until 12:30 in Ra 3223 (Ravelijn 3223 is a room in a building of the university). This example has no secretary yet, this will be entered in the database when a secretary subscribes to take minutes for the meeting.

The class "Person" represents the users of the system. This class is characterized by a calendar token, a boolean representing whether it is the first time that the user logs in and possibly one or more organizations. The calendar token is used to give each user a unique token to import their own meetings to their calendar application. The boolean which represents whether it is the first time that the user logs in, is used to determine whether to redirect the user to the preferences page, something which will only happen the first time that the user logs in. Organizations are used to determine to which organization(s) a user belongs, this influences which meetings the user can subscribe to and which meetings the user can view on the overview page "Meetings".

The class "Minutes" represents minutes that have been taken for a specific meeting. Minutes consist of a file which are the actual minutes, the original name of the uploaded document, the date & time of uploading and the secretary, the person who has made the minutes. The original name is stored because Django renames files if a duplicate name occurs; storing the original name enables the system to show the user the original name. The minutes can be downloaded via the "Minutes" page.

The class "Preferences" represents the preferences of a user. The following preferences are stored:

- "overview" is a preference for non-secretary users, user can use this to indicate from which organizations they want to receive an overview mail
- "overview_secretary" is a preference for secretaries, a secretary can use this to indicate from which organization they want to receive an overview mail, this is different from the overview as secretaries may not be shown all meetings
- "reminder" is a preference for non-secretaries so that users can indicate whether they want to receive a mail in case no secretary has subscribed

to a meeting which takes place within 10 days

- "calendar_secretary" is a preference which can be used to indicate that the user wants to include meetings for which (s)he is a secretary in his/her calendar
- "calendar_organization" is a preference which can be used to indicate that the user wants to include meetings of the organization(s) that (s)he belongs to in his/her calendar
- "confirmation_secretary" is a preference that secretaries can set if they want to receive a confirmation mail when they (un)subscribe for a meeting
- "zoom_in" is a preference which users can use to indicate that they want to zoom in on the page, it makes the text bigger and thus enhances readability

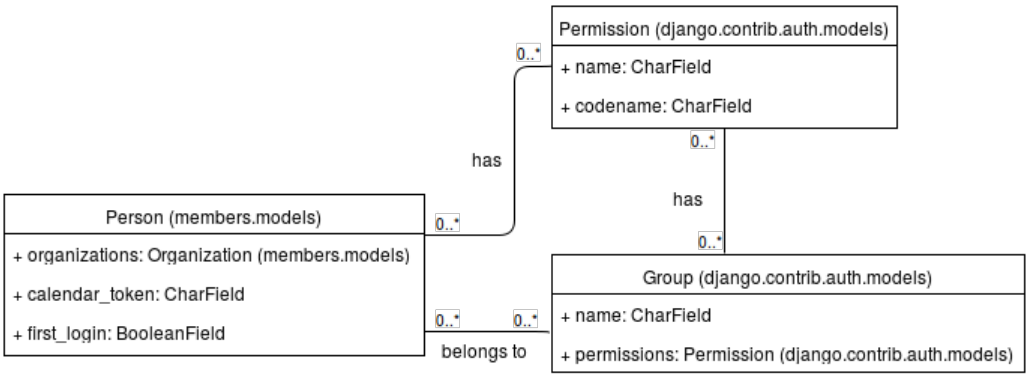

Figure 7.2: Class diagram representing users, groups and permissions

Figure 7.2 contains three classes, "Group", "Permission" and "Person". The class "Person" is the same as in figure 7.1. The other two classes are not classes developed for this project but they are included by Django, they are discussed because of their important role in this system. A "Person" can have multiple permissions which are used to determine which actions (s)he may perform. A "Group" is used to make a distinction between users with different roles. It also enables administrators to easily give a new user the correct permissions. This application has four main group of users: "Administrator", "Planner", "User Manager" and "Secretary". An administrator is

allowed to perform all actions and view all information. A planner is allowed to add new meetings, change existing meetings, delete meetings, view users and may view all meetings but may not view the "Organizations" page. A user manager is allowed to add new users, change users, delete users, add organizations, change organizations and delete organizations. A user manager may not view the "Meetings" and "Minutes" pages. A secretary is allowed to view meetings which belong to their own organization or sub-organizations of their organization. They may also (un)subscribe for these meetings and upload/download minutes for meetings where they took minutes. For example a BOZ employee can be a planner or user manager and a student can be a secretary.

7.2 Design Choices

Several design choices have been made during the project in consultation with the whole project group. These choices have always been well-advised in order to be able to elaborate on their justification.

7.2.1 System Look & Feel

The first design choice concerns the system's look and feel. For this system it is very important that the usage is intuitive and resembles the other University of Twente systems to make it easier for the users to use the system.

7.2.2 Upload Minutes

The second design choice concerns whether a functionality should be added to make it possible for users to upload minutes. This solves the problem that when a BOZ employee has to approve the hours a student has worked, that the employee is unsure whether the student has actually taken minutes. Those minutes will be shown on the minutes page together with their time of uploading grouped per meeting. This contributes to a clearer view of which minutes belong to which meeting as well as what is the newest version of the minutes. When there are a lot of meetings, and so a lot of minutes, the

clarity will help to find the right minutes fast. This choice has been deemed favourable and therefore the functionality is implemented.

7.2.3 Uploading multiple files for a single meeting

The third design choice concerns the ability to upload multiple files for one meeting. It has been decided to allow the secretary to upload multiple files so that either multiple versions of the minutes can be uploaded or to allow diagrams or pictures to be uploaded alongside the minutes.

7.2.4 Download Minutes Interface

The fourth design choice deals with the way minutes are displayed in the download menu. The original name is displayed followed by the date and time of uploading. Another option was to rename each file with the date and time of the meeting, parent organization and organization. However this resulted in very long file names due to which this option was rejected. The date and time of uploading has been chosen to enable to see what the latest version of the minutes is if multiple minutes are uploaded. Parent organization, organization and date & time of meeting was deemed superfluous since this is already shown in the rest of the table.

7.2.5 Delete Minutes Interface

The fifth design choice concerns the ability to remove minutes and how this option is displayed. The first option was to disable secretaries to remove minutes as this means that they can remove data from the system which might be important to for another user. The second option was to enable secretaries to remove minutes such that they can remove a file when they accidentally uploaded a wrong file. The options have been discussed with the client whom indicated that secretaries should not be able to delete minutes however planners should be able to remove the files if necessary. This option could be shown in different ways such as a new menu made to delete minutes or in the same drop-down menu in which the download buttons for minutes are shown. The latter option has been chosen to make the interface less

cluttered. The first option was to show a link with the file name and besides that a small red button with an icon to delete these minutes. This has been shown to a possible user and it was pointed out that the button was confusing. Therefore, this design has been changed by adding an icon to the download link as well as removing the red colour from the button.

7.2.6 Modal based templates

The sixth design choice is whether to show edit and delete forms on new pages or in modals (also called pop-ups). Showing forms on new pages has the advantage of less information on one page. The advantage of modals is that the user is able to see some part of the page on which (s)he is editing. The choice has been made to use modals.

7.2.7 Link Placement for Organizations & Users

The seventh design choice concerns the placement of the links to the pages "Organizations" and "Users". The first option was to place both links in a drop-down in the menu. The second option was to place the items separately in the menu. The second option has been chosen because this results in less clicks needed to navigate to the correct page. Moreover, it is easier to implement that the links are only visible to those with the correct permissions.

7.2.8 Link Placement for Scheduling Meetings

The eight design choice deals with the placement of a link to the page where users schedule a meeting. The first option was to place a link in the menu if the user is allowed to schedule meetings. The second option was to place a button on the overview page of meetings. The second option has been chosen to de-clutter the menu as well as to enable consistency throughout the design because this design can also be used on the users and organizations page.

7.2.9 Preferences Page

The ninth design choice deals with the preferences page. This page has been made to enable users to choose which mails they want to receive by using an opt-in system. The other option was to use an opt-out system, however this was deemed unfavourable as it would result in everyone receiving all mails, even the ones they should not be able to receive.

7.2.10 Manual & Help Placement

The tenth design choice concerns the placement of the manual on the website. The first option was to make a section containing short explanations for the most important functionalities of the system. The second option was to place the manual on the website. The choice has been made to place short explanations on the help page for the most important functionalities and to also put a link to the manual on this page. This choice will make sure that the help page is not extensively long which discourages the user and if necessary, the user can still look through the manual, please refer to the User's Manual, which is more detailed than the information on the help page.

7.2.11 Time zone Awareness

The eleventh design choice concerns time zone awareness of the application. Time zone awareness is initially enabled in Django however this resulted in several difficulties combined with the date & time picker as this plug-in handled time zone awareness differently. To avoid these integration issues, it has been decided to disable time zone awareness. This has no great influences on the system as all users are expected to use the system in a time zone similar to the time zone of the Netherlands.

7.2.12 Button Designs for Subscription

The twelfth design choice deals with the design of the button used to sign up for a meeting as a secretary. The first option was to make one button namely "Sign up". Second option was to make a second button as well to

"Unsubscribe". The third option was to use a toggle button. The first option has as the advantage that students need to make sure that they will be able to take minutes, something which can also be a disadvantage as students are not able to unsubscribe. The second option enables students to subscribe and unsubscribe however always showing two buttons when the options may not even be available is superfluous. The last option enables students to subscribe, unsubscribe as well as only showing the available action. This toggle also uses colour which makes it clear to the users in one glance for which meetings they are subscribed. Therefore the last option has been chosen as it was considered the best one.

7.2.13 Mailing

The thirteenth design choice considers sending mails to secretaries when a new meeting has been added. This option has been addressed by an employee of BOZ and has been carefully considered. BOZ employees mentioned that they often add meetings in large bulks which would result in secretaries receiving bulks of mails if a mail is to be send for each meeting. This seemed unfavourable for the secretaries however the option to mail an organization has been made such that a BOZ employee can still mail all possible secretaries at once if wanted.

Chapter 8

TESTING THE SYSTEM

In this chapter, the test plan and the test results are provided. The different functionalities which are tested are indicated, the test approach is explained, test criteria are provided, the schedule is discussed and, the risks and contingencies are indicated.

Furthermore, the test results are provided of the unit tests, integration tests, results of general meetings and usability tests. The design has been changed or bugs have been found in the code and solved based on the results of these tests.

8.1 Test Plan

8.1.1 Approach

The project will focus on providing sufficient unit tests to achieve maximum coverage of the code paths in the project source. External dependencies will not be extensively tested using unit tests, since it can be assumed that for example the login functionality provided by the university will work as expected. In order to test the system, the system is tested according to three testing strategies. These strategies are a) Unit testing, b) Integration testing and c) Usability testing. While the first two strategies focus on testing the back-end of the system, the latter tests the interaction with the end-users without regarding the processing 'in the back'. In the following subsections, these three testing strategies are discussed.

8.1.1.1 Unit testing

Unit testing will focus on writing automated tests. The unit tests will use the Django unit test framework. The unit tests will attempt to test as many edge cases as possible.

8.1.1.2 Integration testing

Integration testing goes beyond the scope of unit testing. Each functionality within the system can operate as expected, however when integrating these functionalities within a containing system, problems can occur. Integration testing attempts to address bugs during the later phases of the development, assuring a fully functional system. The approach to integration testing will be similar to the one in system testing in which the system will be treated as a black box.

8.1.1.3 Usability testing

In this subsection, the usability testing strategy is discussed. It is based on the ten heuristics for user interface design by Nielsen [Nielsen and Molich(1990)].

A subset of these heuristics are considered for the testing of the design, containing the following heuristics:

- **Visibility of system status**: the users should always be informed by the system about its current situation via feedback.
- **Consistency and standards**: it should be clear to the user whether different words, situations or actions mean the exact same thing.
- **Aesthetic and minimalist design**: the system should not contain information which is irrelevant for its use purposes. Every extra snippet of information in the system competes with the relevant snippets of information and diminishes their relative visibility.
- **Help and documentation**: it may be necessary to provide help and documentation when delivering the system. All information should be easy to find by searching for it, elaborate on the tasks for the user, list concrete steps to be carried out and it should not be extensive.

The user interface designs can only be accepted when they meet the expectations that these heuristics provide. The masters of the acceptance are the users of the system. Therefore, the system can only be accepted when the system satisfies the demands of these users.

The interface designs are tested through user based testing. The employees of the BOZ will be asked to perform a series of tasks defined by use scenarios as part of a cognitive walk through. Furthermore, the employees were asked to say what came to mind to receive even more feedback from them, which is also known as the think-aloud protocol [Preece and Rogers(2011)]. The interface passes the test when these employees can complete these tasks.

8.1.2 Software risk issues

There are several critical areas of the software to be tested. The first critical area is the authentication with University of Twente credentials as permissions are needed to be able to support this. The second critical area is the ability to use and understand new packages and tools. Several team members have not worked (a lot) with Django and/or Python thus it may cost significant time to understand this language and framework.

The project also makes use of multiple interfaces, as both BOZ and students will use the program. Therefore, the two different interfaces should overlap regarding appearance but should still provide the necessary functions.

8.1.3 Functionalities to be tested

In this section a list of what should be tested from the users' viewpoint of what the system does will be presented. These are functionalities derived from the current requirements of the project, and may change in the future. The level of risk for each functionality is rated using High (H), Medium (M) or Low (L). This level of risk indicates how important it is to test a certain functionality. Refer to Table 8.1 for these risks indications.

Functionality to be tested	**Level of risk**
Add a new meeting	H
Delete a meeting	H
Change location of a meeting	H
Change begin/end time of a meeting	H
Add new student as secretary	H
Delete a student as secretary	H
Add a new administrator	H
Delete an administrator	H
Add secretary to meeting	H
Remove secretary from meeting	H
Set availability for meeting	H
Login with University of Twente credentials	H
Log out	H
Upload minutes	M
Delete minutes	M
Integration with Google Calendar	L

Table 8.1: The functionalities that should be tested

Furthermore, the front-end of the system is tested based on its usability for the end-users.

8.1.4 Item pass/fail criteria

All test cases identified as high risk in Table 8.1 will need to be completed without defects. Test cases identified as medium or low risk will allow minor defects if it does not hinder the use of any features. Otherwise the feature will not be included in this release.

8.1.5 Schedule

In this project an agile approach was used thus requirements were added and changed during the development of the system. Unit tests were made each time a new feature was added to the project, existing unit tests had to be changed according to the requirement changes. The unit tests had to pass before being merged into the code base of the project. Integration tests should be done any time a feature is added as well, though this process is more informal. Finally, usability testing will be used towards the end of the project. Since this testing method will primarily be used to tweak user interfaces it is not imperative to do this while implementing a feature.

8.1.6 Risks and contingencies

The project is managed by applying Agile practices [Pressman(2009)], implicating that the original requirements and designs are not fixed throughout the process of the development. The challenge is to manage newly defined requirements and designs without endangering the scope of the project.

The consequences of a change in requirements is that it leads to a change in functionalities for the system. There are three cases to which these changes can lead. The first case is that a functionality is deleted and therefore, this leads to less tests that need to be done. Secondly, newly defined functionalities lead to an increase of tests that should be done. Finally, a requirement can be changed, resulting in changing the corresponding tests that need to be done. Thus it is possible that the test schedule can draw out an appropriate number of tests or days. By keeping track of the progress of the project and therefore, managing the requirements and designs, time constraining difficulties can be avoided.

8.1.7 Approvals

Approving whether the process of testing is complete can only be done in consultation with the whole project group. Considering the end-users of the system, it should work as they request in their requirements. Therefore, the system as a whole should also be approved by the end-users.

8.2 Test Results

8.2.1 Unit tests

All views have been tested to assert that the correct information was shown and to assert that a correct HTTP response was given for administrators as well as students.

One bug has been discovered on the overview page of meetings. The unit test tested whether the student could view all expected meetings. A student and meeting belonging to the same organization were instantiated, the student was logged in and the student should therefore be able to see the meeting. Two queries were applicable to this student, it should see the meetings for which it is the secretary and the meetings of the organization to which the student belongs. The results of these queries were combined with a '&'(and) instead of a '|'(or) due to which the expected meeting was not shown. To solve this bug '&' has been replaced by '|'.

8.2.2 Integration tests

The login/logout with University of Twente credentials is a service provided to the developers by LISA and therefore has only be tested on the surface during development and usability tests. No problems have arised during integration testing as expected, since this service is used by many students and professors at the university and thus it is already extensively tested.

8.2.3 Usability tests

The usability tests have provided a considerable amount of feedback which served to enhance the system design. Please refer to Subsubsection 8.1.1.3 for more details on the methodologies used to conduct these usability tests. These employees were asked to fulfil a variety of user tasks listed below, please refer to Appendix D in the Appendix for the test scenarios.

1. Log in to the system with UT log in credentials.
2. Schedule meetings in the system.
3. Change a meeting in the system.
4. Delete meetings in the system.
5. subscribe and unsubscribe secretaries for meetings in the system.
6. Download the most recent minutes for a meeting.
7. Sort and search through the list of meetings in the system.
8. Consult the help page of the system.
9. Add new organizations to the system.
10. Add new users to the system.
11. Log out of the system.

As a result of these test scenarios, the employees provided rich feedback on the usability of the system. For example, the date & time selector contained in the schedule a meeting form has been adjusted based on the feedback of the users. The usability issues that are solved by means of the test scenarios are:

1. The pages were too small for the employees to read.
   Solution: *The size of the pages can be adjusted in the preferences page by making use of the zoom-in functionality.*
2. The functionality of the date & time selector is not clear enough.
   Solution: *The date and time selectors are positioned next to each other instead of behind each other.*

3. Missing submit button in the date & time selector.
   Solution: *A submit button is added below the selector in the same pop up screen.*
4. The end date & time field does not change when the begin date & time are changed.
   Solution: *The field changes according to the begin date & time field.*
5. Forms are not consistent in their ordering of the different fields.
   Solution: *The forms are changed such that the ordering is the same for each form.*
6. The labelling of the buttons is not clear enough.
   Solution: *The labels of the buttons are rephrased and tested.*
7. Colour indication (such as green and red) are not clear enough for the employees.
   Solution: *The indication of colours is more clearly defined in the system.*
8. The scroll bar in the list of meetings is not necessary and it messes up the list.
   Solution: *The scroll bar is deleted and any pop-ups in the lists are placed in front of the list.*
9. The help page is not clear and employees cannot find relevant information.
   Solution: *The headers and the help texts are reformulated and tested.*
10. The meaning of parent organizations is not clear.
    Solution: *The parent organization tags are provided with help texts explaining their meaning.*
11. There is no consequent design in pop-up and other forms.
    Solution: *The forms are updated and consistently ordered.*
12. The definition of username is not clear to the employees.
    Solution: *The username field is provided with help text explaining its meaning.*

The usability tests serve to enhance the Nielsen's Heuristics which are improved by the solutions to the issues, please refer to Section 8.1.1.3 for these heuristics.

Chapter 9

Future Planning

In this chapter, the future planning after this quartile is discussed. The development of the system comes to an end during this quartile, however the usage of the system will then start. Therefore, a planning for the future is of utmost importance due to the fact that the system will actually be used by BOZ employees. In the remainder of this chapter, the plan of supporting the system and its users will be discussed. A need exists for wider deployment in all faculties of the university, therefore this deployment should be anticipated during development and planned in advance.

9.1 Utilization and Support of the System

After the development of the system, the BOZ responsible for the faculty of BMS will start to use it. Therefore, it is necessary to offer them the support they need from the beginning onwards. It is safe to assume that they will initially have many questions. The system is developed to take away the most severe difficulties. It is also safe to assume that the system could still contain bugs that need to be solved during its operation. The plan is to go in production on a server hosted by the university and maintained by the employees over at the Library, ICT-services and Archive (LISA) department. On these servers, all other services from the university are hosted such as Osiris. Due to the fact that the system for this design project is developed in other tools (Python and Django) than the existing services, the employees at LISA lack the expertise to maintain the system. Therefore the choice has been made to support the software with members of the development team, these members will have enough knowledge to resolve problems occurring during deployment.

9.2 University wide Enrolment

During the development of the system, measures are taken to easily support broader deployment of the system for all faculties. Other faculties can easily be added to the system and different users can be added by taking just a few steps. This results in opportunities to deploy the system without too much effort and in smaller steps if necessary. The administrators of the system can add other employees and students by filling in their employee-number or student-number as username, then the added user can log-in with their university credentials, which they already have.

Chapter 10

Evaluation

In this chapter, an overall evaluation on the project is provided by means of elaborating on its planning, identifying the different responsibilities within the project team, evaluating the project team and discussing the final result. Finally, this chapter concludes this report by stating the conclusion on this project.

10.1 Planning

Th planning of the project has been very strict from the beginning, almost every week there were deadlines scheduled by the project team. This implies that the workload has been distributed throughout the entire module. The progress of the project and its deadlines were monitored by utilizing a SCRUM-board (Trello [Trello, Inc.(2016)]).

Every two weeks, a meeting with the client was scheduled to discuss the progress of the project and to request feedback from the client about iteratively specified requirement, designs and prototypes, please refer to Chapter 3 for these meetings with the client.

At the later phases of the project the deliverables were needed to be finalized and handed-in. The deadlines of these deliverables were set far in advance of the deadlines towards the supervisors, enabling the team to read and rewrite the documents to improve the quality of these deliverables. This planning is respected during the project which ensured an equally distributed workload and high quality of the handed-in documents.

10.2 Responsibilities

The team was composed of team members with different interests and expertises, meaning that the project was well divided into different tasks and responsibilities. These responsibilities were assigned to the different team members primary based on their interests, because working on something that has your interest motivates to do your utmost best. Moreover, due to the different interests and expertises, the assignment of responsibilities was quite straightforward from the beginning. The assigned responsibilities are listed below:

- **Iris:** Interface Designer & Developer, Requirement Analysis, Poster Design and Presentation Slides.
- **Kimberly:** Database Designer, Interface Developer, Head of Mailing Services and Preferences, Poster Design and Presentation Slides.

- **Pieter:** Database Designer, Head of Back-end Development and Log in Services, Head of System Development.
- **Sophie:** Database Designer, Requirements Specification, Back-end Developer, Interface Designer and Head of Testing.
- **Thomas:** Project Leader, Communication Manager with the Client, Interface Designer & Developer, User's Manual Editor and Editor in Chief.

Note that it is a group effort to write the project deliverables, meaning that every single team member contributed to it. Likewise, the personal communication with the client has been a responsibility of the group as a whole.

10.3 Team Evaluation

A difference in work attitude led to different team members having an argument in the beginning of the project. To solve and further prevent this, two team evaluation sessions were planned to discuss the differences in attitude and to re-think the agreements agreed upon by the whole team. The team had also planned a team building activity, to strengthen team spirit, which was successful.

10.4 Final Result

The project resulted in a system for the users that is easy-to-use, and is created in collaboration with the client by involving them in participatory design sessions. In the process, there have been difficulties to involve the client is participating in designing the system. Nonetheless the feedback from the users has been taken to heart and the system has been changed accordingly. The system does not contain unnecessary functionalities, meaning that it solely contains functionalities which the client has requested or are approved by the client. It ensured that the system is exactly as the client has envisioned and it, which could only be accomplished by Customer Intimacy.

10.5 Conclusion

The strict planning of the project ensured that the system and deliverables were finalized and handed-in in time. At the start of the project, a team building activity was scheduled to aid the team spirit. It helped to meet on regular basis with the entire team, which was possible due to the new design of the design project course.

The aim of the project was to deliver a system that emphasizes on its usability and the ease of adding new users and organizations to the system with regard to UT wide deployment - please refer to Section 9.2. Furthermore, the aim of the project was also to deliver a bug-free and complete system which can be hosted after the design project has ended without running into trouble. Due to the future plans of deploying the system on servers hosted at the UT, delivering a bug-free and complete system was even more stressed for this project.

As stated in Section 10.4, the project resulted in a working system meeting the requirements and expectations of the client. After this module, the system will probably be taken into operation by BOZ BMS.

This design project served to gain insight into the development process and all its phases. This project also served to understand what difficulties may occur in terms of team work, due to its long development time.

Bibliography

[Vijverberg and Opdenakker(2013)] A. M. M. Vijverberg and R. J. G. Opdenakker, *Vitale organisatiestrategie: een ontdekkingstocht naar waarde, positie en identiteit.* Kluwer, 2013.

[Django Software Foundation(2015-2016)] Django Software Foundation, "Why django?" 2015-2016. [Online]. Available: https://www.djangoproject.com/start/overview/

[Python Software Foundation(2016)] Python Software Foundation, "Python," 2016. [Online]. Available: https://www.python.org/

[Kent Beck, et al.(2001)] Kent Beck, et al., "Agile manifesto," 2001. [Online]. Available: http://agilemanifesto.org/

[Scrum.Org and ScrumInc(2004)] Scrum.Org and ScrumInc, "The scrum guide," 2004. [Online]. Available: http://www.scrumguides.org/scrum-guide.html

[Mannion and Keepence(1995)] M. Mannion and B. Keepence, "Smart requirements," vol. 20, no. 2, 1995.

[Olander and Landin(2005)] S. Olander and A. Landin, "Evaluation of stakeholder influence in the implementation of construction projects," vol. 23, no. 4, 2005, pp. 321–328.

[Qiao(2009)] M. Qiao, "The effectiveness of requirements prioritization techniques for a medium to large number of requirements: A systematic literature review," 2009.

[Greenbaum and Loi(2012)] J. Greenbaum and D. Loi, "Participation, the camel and the elephant of design: an introduction," *CoDesign*, vol. 8, no. 2-3, pp. 81–85, 2012.

[Brandt(2007)] E. Brandt, "How tangible mock-ups support design collaboration," *Know Techn Pol*, vol. 20, pp. 179–192, 2007.

[Österman et al.(2016)Österman, Berlin, and Bligård] C. Österman, C. Berlin, and L. Bligård, "Involving users in a ship bridge redesign process using scenarios and mock-up models," *International Journal of Industrial Ergonomics*, vol. 53, pp. 236–244, 2016.

[Nielsen and Molich(1990)] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," 1990, pp. 249–256.

[Preece and Rogers(2011)] J. Preece and H. Rogers, Y. Sharp, *Human-Computer Interaction*, 3rd ed. West Sussex, UK: John Wiley And Sons Ltd., 2011.

[Pressman(2009)] R. Pressman, *Agile Development*, 7th ed. Software Engineering: A Practitioner's Approach, 2009.

[Trello, Inc.(2016)] Trello, Inc., "Trello," 2016. [Online]. Available: https://trello.com/

[Alexander(2004)] I. F. Alexander, "A better fit-characterising the stakeholders." CASE Workshops, 2004, pp. 215–223.

Appendices

Appendix A

Stakeholder onion model

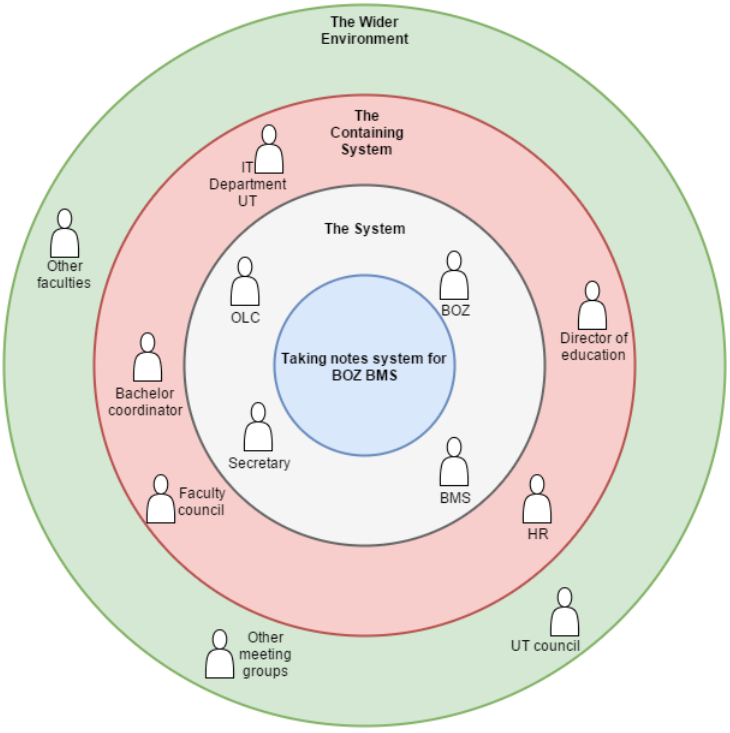

Figure A.1: Stakeholder onion model [Alexander(2004)]

Appendix B

Mock-ups

A selection of the different presented mock-ups.

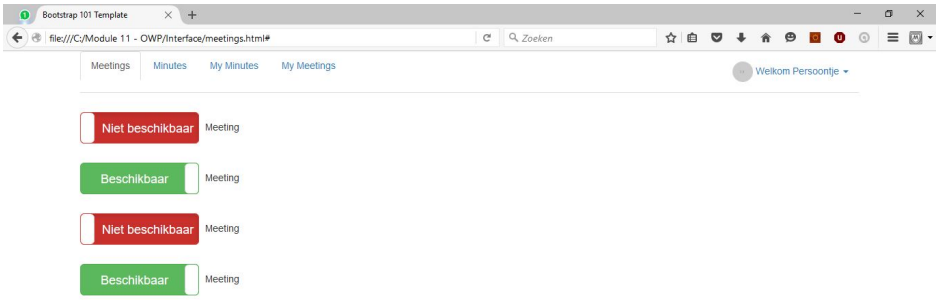

Figure B.1: Mock-ups showing the meeting page

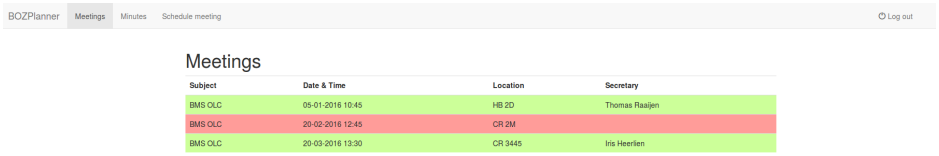

Figure B.2: Mock-ups showing the meeting page

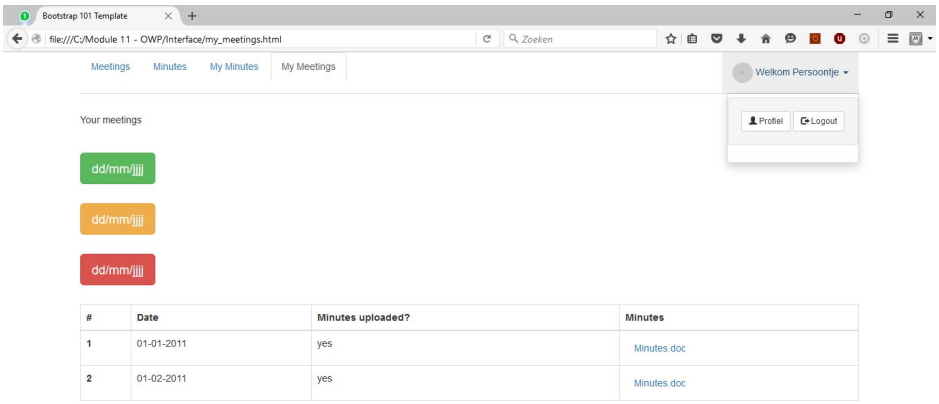

Figure B.3: Mock-ups showing the meeting page

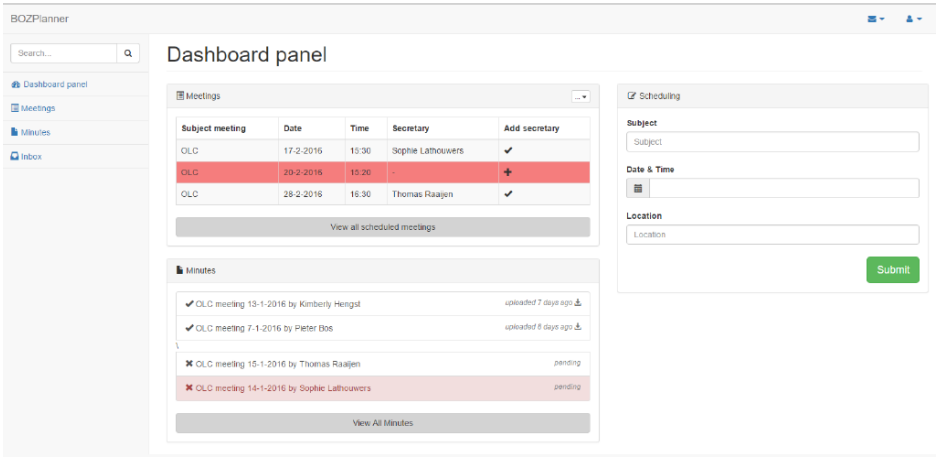

Figure B.4: Mock-ups showing a dashboard homepage

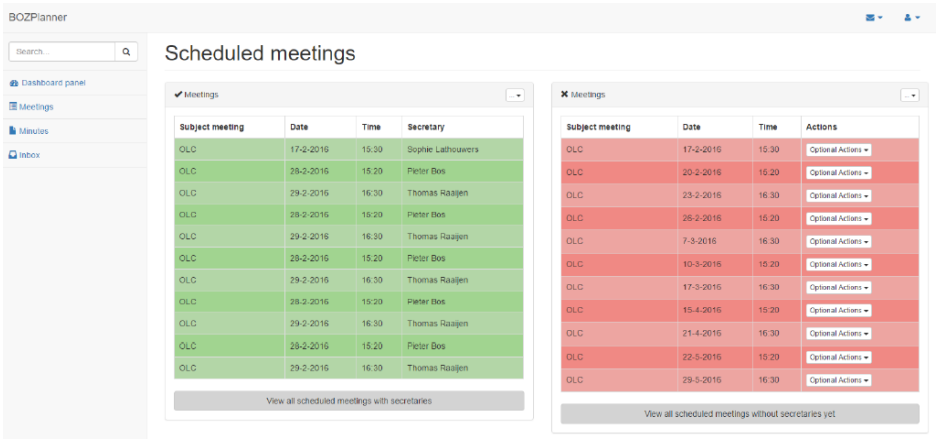

Figure B.5: Mock-ups showing the meeting page in a dashboard view

Appendix C

Requirement Specification

ID	Description	Criterium	Performance	Bandwidth	Priority	Source
1.	As secretary I want to log in and out of the system					
1.1	The system must have a mechanism which enables secretaries to log in and out of the system	Time it takes to successfully log in or out	Maximum of 30 seconds	+/- 5 seconds	Must	Developer
2.	As secretary I want to sign up for an OLC meeting					
2.1	The system must have a mechanism which enables secretaries to sign up for an OLC meeting	Time it takes to successfully sign up	Maximum of 30 seconds	+/- 5 seconds	Must	Developer
3.	As secretary I want to sign out of an OLC meeting					
3.1	The system must have a mechanism which enables secretaries to sign out of an OLC meeting for which they were supposed to take minutes	Time it takes to successfully sign out	Maximum of 30 seconds	+/- 5 seconds	Must	Developer
4.	As secretary I want to upload the minutes taken					
4.1	The system should have a mechanism to upload minutes	Usability, time it takes for the user to understand the mechanism	Maximum of 1 minute	+/- 15 percent	Should	Developer
4.2	The system should save uploaded minutes in its internal storage	Time it takes to upload the minute	Maximum of 1 minute	+/- 15 seconds	Should	Developer
5.	As BOZ I want to log in and out of the system					
5.1	The system must have a mechanism which enables BOZ employees to log in and out of the system.	Time it takes to successfully log in or out	Maximum of 30 seconds	+/- 5 seconds	Must	Developer

Figure C.1: First set of specified requirements

6.	As BOZ I want to schedule a meeting in the planning					
6.1	The system must have a mechanism which enables BOZ to schedule a meeting in planning	Time it takes to plan the meeting in a timetable	Maximum of 5 minutes	+/- 30 seconds	Must	Developer
6.2	The system should only show meetings for the specific faculty I am associated with	Successfulness of handling	Pass or fail	-	Should	BOZ
6.3	The system could notify the BOZ that the OLC has requested a meeting	Time it takes to get notified	Maximum of 1 hour	+/- 30 minutes	Could	Developer
7.	As BOZ I want to see whether secretaries can still sign up for meeting					
7.1	The system must show whether the sign up for a meeting is open or closed	Usability, time it takes for the user to see whether the sign up is still open	Maximum of 10 seconds	+/- 5 seconds	Must	Developer
8.	As BOZ I want to see which secretary has signed up for a meeting					
8.1	The system must show BOZ which secretary signed up for a meeting	Usability, time it takes for the user to see who will take minutes	Maximum of 10 seconds	+/- 5 seconds	Must	Developer
9.	As BOZ I want to get notified when a minute is uploaded					
9.1	The system could send a notification to BOZ when minutes are uploaded	Time it takes to get send notification	Maximum of 10 minutes	+/- 5 minutes	Could	Developer
10.	As BOZ I want to get updated about upcoming meetings					
10.1	The system should send weekly email updates to the BOZ employees informing them about the planning	Time bound	Be received on monday morning	+/- 1 hour	Should	BOZ

Figure C.2: Second set of specified requirements

11.	As BOZ I want to be able to change the secretary attending the meeting					
11.1	The system must enable BOZ to change the secretary for a specific meeting	Usability, time it takes to understand the mechanism	Maximum of 30 seconds	+/- 5 seconds	Must	Developer
12.	As BOZ I want to be able to download the taken minutes					
12.1	The system should have a mechanism which enables BOZ to download uploaded minutes	Usability, time it takes to understand the mechanism	Maximum of 30 seconds	+/- 5 seconds	Should	Developer
13.	As BOZ I want to be able to change or cancel meeting					
13.1	The system must have a mechanism which enables BOZ to change or cancel a meeting	Usability, time it takes to change or cancel a meeting	Maximum of 5 minutes	+/- 1 minute	Must	Developer
13.2	The system could have a mechanism which consistently asks the BOZ whether they are sure of proceeding with the changing or cancellation of a meeting	Usability, response time of the pop up asking the BOZ to proceed	Maximum of 5 seconds	+/- 1 second	Could	Developer
13.3	The system could notify all parties of a change considering a meeting	Percentage of received notifications by all parties	Minimum of 99 percent	+/- 1 percent	Could	Developer
		Time it takes to receive the notification	Maximum of 30 minutes	+/- 5 minutes	Could	Developer
14.	As OLC I want to log in and out of the system					
14.1	The system must have a mechanism which enables OLC members to log in and out of the system.	Time it takes to successfully log in or out	Maximum of 30 seconds	+/- 5 seconds	Must	Developer

Figure C.3: Third set of specified requirements

15.	As OLC I want to be able to download non-approved minutes					
15.1	The system should enable the OLC to open the minutes	Usability, time it takes to open the minutes	Maximum of 1 minute	+/- 30 seconds	Should	Developer
15.2	The system could have a mechanism which notifies the OLC about a newly uploaded minutes	Percentage of received notifications by the OLC	Minimum of 99 percent	+/- 1 percent	Could	Developer
		Time it takes to receive the notification	Maximum of 5 minutes	+/- 1 minutes	Could	Developer
16.	As OLC I want to be able to approve minutes					
16.1	The system should have a mechanism to approve the minutes	Usability, time it takes to approve the minutes	Maximum of 1 minute	+/- 30 seconds	Should	Developer
16.2	The system could notify all parties on the approving of minutes	Percentage of received notifications by all parties	Minimum of 99 percent	+/- 1 percent	Could	Developer
		Time it takes to receive the notification	Maximum of 5 minutes	+/- 5 minutes	Could	Developer

Figure C.4: Fourth set of specified requirements

Appendix D

Test Scenarios

Bling! Er komt een mailtje binnen van een docent genaamd Klaas van de OLC van de opleiding European Public Administration binnen de faculteit BMS. Er moet een vergadering worden gepland en deze moet in het systeem ingevoerd worden door BOZ in het systeem, want jullie hebben de rechten in dit systeem. Daarvoor moet jij eerst inloggen in het systeem met jouw account *[**.Laat de student inloggen.**]*. De vergadering moet plaatsvinden op 4 april van dit jaar , het zal plaatsvinden in RA 3114 en het zou moeten beginnen om 11:00 precies. Omdat deze vergadering niet langer duurt dan 45 minuten moet deze ook aflopen om 11:45 *[**.Voeg de vergadering toe in het systeem.**]*.

Na enige tijd komt de voorzitter van de opleidingscommissie langs om te vragen of de geplande vergadering verplaatst kan worden naar 14:00 vanwege geplande activiteiten die plaatsvinden in de Ravelijn van de opleiding waarbij sommige docenten gevraagd zijn om een presentatie te geven. Verder is de geplande kamer bezet voor 14:00 en daarom moet ook de locatie gewijzigd worden naar RA 5148. Verder komt aan het licht dat Klaas betrokken is bij meerdere opleidingscommissies, omdat hij meerdere vakken geeft voor European Public Administration en International Business Administration. Daarom moet ook de organisatie gewijzigd worden, want de vergadering is bedoelt voor het evalueren van International Business Administration *[**.Wijzig de bovengenoemde gegevens in het systeem.**]*.

Bling! Er komt nu een mailtje binnen van een docent genaamd Ron van

de OLC van de opleiding Psychology binnen de faculteit BMS. Ron vraagt of er een vergadering gepland kan worden voor 7 april om 15:00 uur en het moet plaatsvinden in CU 202B ***[.Voeg de vergadering toe aan het systeem.]***. Een uurtje later komt er weer een mailtje binnen van Ron waarin hij aangeeft dat door voortijdige afmeldingen de vergadering toch opgeschort moet worden naar een onbekend moment. Hierdoor moet de vergadering verwijderd worden ***[.Verwijder de vergadering uit het systeem.]***.

Het systeem heeft reeds een notification gestuurd naar de studenten dat zij zich in kunnen schrijven als notulist voor de vergadering aangevraagd door Klaas. Een student van International Business Administration genaamd Thomas heeft zich ingeschreven voor de meeting. Enige tijd is verstreken en het is reeds 2 april en hierdoor kan Thomas zich niet meer uitschrijven voor de vergadering. Echter, door privé omstandigheden kan Thomas niet meer bij de vergadering zijn en daarom neemt hij contact op met BOZ om zich uit te laten schrijven. Jij als medewerker keurt zijn reden goed jij geeft aan dat hij wordt uitgeschreven en dat er een vervanger kan worden geregeld ***[.Schrijf de student uit.]***. Jij stuurt onmiddelijk een mailtje naar de andere studenten met de vraag of er iemand kan notuleren in plaats van Thomas. Sophie mailt binnen enkele uren dat zij bereid is om bij de vergadering aanwezig te zijn ***[.Schrijf Sophie in als notulist.]***.

Om de uren goed te keuren van Pieter moet zijn notulen worden bekeken in het systeem. Daarom moet zijn notulen worden gedownload en bekeken worden. Pieter heeft genotuleerd voor Industrial Engineering and Management op 18 maart om 10 uur. Hij heeft meerdere versies geupload van zijn gemaakte notulen en jij wil de meest recente versie hebben voor het goedkeuren van zijn uren ***[.Download de meest recente notulen.]***. Het is voor jullie als BOZ heel vervelend als er een vergadering nadert die nog geen notulist heeft. Daarom wil jij graag weten wat de eerst volgende vergadering is zonder een notulist ***[.Probeer de eerstvolgende vergadering te vinden zonder notulist zonder het lijstje af te gaan.]***.

Vanuit de UT willen ze het systeem verder uitrollen over andere faculteiten zoals EWI. Daarom moet er een BOZ medewerker van de faculteit EWI worden toegevoegd aan het systeem. Maar eerst moet EWI worden toegevoegd als organisatie in het systeem. Jij twijfelt over hoe je een organisatie kan toevoegen in het systeem en daarom raadpleeg je de help pagina van het systeem om te erachter te komen hoe je een organisatie toevoegt ***[.Raadpleeg***

de help pagina om erachter te komen hoe je een organisatie toevoegt.]. Je bent erachter gekomen hoe je een organisatie toevoegt en gaat dat dan ook direct doen ***[.Voeg EWI toe als organisatie.]***. Nu EWI is toegevoegd aan het systeem, kan een BOZ medewerker binnen EWI worden toegevoegd als gebruiker. De medewerker heet Wietsje Herlaagd, haar gegevens zijn als volgt: medewerkersnummers is m1234567 en haar email is w.e.herlaagd@utwente.nl. Ze is een medewerker van BOZ en daarom krijgt zij de functie van planner in het systeem ***[.Voeg Wietsje toe als gebruiker van het systeem.]***.

Jouw taken zijn weer gedaan voor de dag en jij kan nu uitloggen van het systeem ***[.Log uit.]***.