Application of User-Centered Design for a Student Case Management System

Vincent Kahl
Abstract

Application of User-Centered Design for a Student Case Management System

Vincent Kahl

The student office and student counselors of Uppsala University’s IT Department need a new application for organizing and coordinating student cases. The aim of this thesis is to define a specification for a new system. A user-centered design (UCD) approach is taken to ensure that the new application will increase productivity, is usable, and is accepted by the people that will work with it. The employed UCD process is a custom adaption of the ISO 9241-210 standard’s UCD process proposal. Following the activity cycle of the ISO standard for user-centered design, this specification will understand and specify the context of use, specify the user and organizational requirements, and produce design solutions that are evaluated against the requirements.
# Table of Contents

1 Introduction .............................................................................................................. 1

2 User-Centered Design and Usability ........................................................................... 2
   2.1 Usability ............................................................................................................. 2
      2.1.1 Goals and Context of Use ........................................................................ 2
      2.1.2 Usability Measures .................................................................................... 3
   2.2 User-Centered Design ....................................................................................... 3
      2.2.1 Key Principles ............................................................................................ 4
      2.2.2 Activities ................................................................................................... 5
      2.2.3 Benefits of User-Centered Design ............................................................ 7
   2.3 Design Guidelines and Principles ....................................................................... 8
      2.3.1 Guidelines .................................................................................................. 8
      2.3.2 Principles .................................................................................................. 9

3 Application of User-Centered Design ...................................................................... 12
   3.1 Specifying the Context of Use .......................................................................... 12
      3.1.1 Method ..................................................................................................... 13
      3.1.2 Current Email-Based System ................................................................... 13
      3.1.3 Future Web-Based Application ................................................................ 15
   3.2 Specifying the Requirements ............................................................................ 17
      3.2.1 Method ..................................................................................................... 17
      3.2.2 User Roles ................................................................................................ 17
      3.2.3 User Role Map .......................................................................................... 18
      3.2.4 Scenarios .................................................................................................. 18
      3.2.5 Use Cases .................................................................................................. 19
      3.2.6 Use Case Map ........................................................................................... 21
      3.2.7 Required Data ........................................................................................... 22
   3.3 Design Solutions ............................................................................................... 23
      3.3.1 Method ..................................................................................................... 23
      3.3.2 Aspects regarding the Overall Design ....................................................... 23
      3.3.3 Case Overview Screen .............................................................................. 25
      3.3.4 Advanced Search Feature ......................................................................... 26
      3.3.5 Add Case View .......................................................................................... 27
      3.3.6 Detail View of Case ................................................................................... 29
      3.3.7 Case Edit View .......................................................................................... 30
      3.3.8 Discussion .................................................................................................. 31

4 Limitations and Future ............................................................................................. 33

Bibliography .................................................................................................................. 34
Appendix ........................................................................................................................................36
Appendix A Interview Outline ........................................................................................................37
Appendix B Design Proposals ........................................................................................................38
List of figures

Figure 1: Usability Framework. .................................................................2
Figure 2: Activities of User-Centered Design. ...........................................5
Figure 3: Adaptation of the UCD activities for the project. .........................12
Figure 4: Workplace of a member of the Student Office. .........................16
Figure 5: User Role Map for the web-based Student-Case Management application. ....18
Figure 6: Use Case Map for the web-based Student-Case Management application. ....21
Figure 7: Dropdown-List with Fast Item Selection showing the suggestion feature. ....24
Figure 8: Case Overview Screen. ...........................................................25
Figure 9: Advanced Search Options. .......................................................27
Figure 10: Screen for adding a new student case. ....................................28
Figure 11: Add Case Period and Edit Case Period administrative options. ....28
Figure 12: Screen of a Detailed View for the Student Office of a Student Case. ....30
Figure 13: Case Edit Screen. .................................................................31
Figure 14: Feedback Message including Undo option for Case Deletion. ....31
Figure 15: Entire Screen when adding a new case period. .........................38
Figure 16: Screen of the Detail View of cases for the user group: Student Counselor. ..39
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HCI</td>
<td>Human-Computer Interaction</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>RUP</td>
<td>Rational Unified Process</td>
</tr>
<tr>
<td>UCD</td>
<td>User Centered Design</td>
</tr>
</tbody>
</table>
1 Introduction

Email as communication tool has been around for decades. Very often it is used to communicate, organize, and coordinate work. Also the student office\(^1\) and the student counselors\(^2\) of the IT Department of the University of Uppsala (Sweden) use email for performing those tasks. In order to fulfill the students’ requests that require the collaboration of the two university bodies, they use email to arrange their correspondence in administrative matters.

The use of email has many advantages like asynchronous communication but some of its advantages, like the very free and open possibilities to write and construct a message can turn into drawbacks in certain contexts. Due to the open form of message structures the reader has to search through the entire message for the relevant information. Furthermore, it requires a lot of user discipline to follow the agreed upon processes of the organization. This becomes evident in the case of introducing new co-workers until they have accommodated to the processes.

A dedicated application for the student office and student counselors may overcome the particular drawbacks of using email for task completion while keeping the advantages. The use of an application dedicated to the particular tasks allows all direct users to gain an instant summary over the current overall status of all student cases and an immediate overview of the status of a certain student case. To ensure that the system is usable, increases the productivity, and is accepted by the people working with it, the specification for that system is written employing a user-centered design approach.

This work presents a specification of a proposed new system. It explains the methods used to extract the relevant information and includes design solutions based on the established specifications. In order to demonstrate the chosen methods, the goals, aims and objectives of usability (section 2.1) and user-centered design (section 2.2), as well as basic guidelines and principles of designing user interfaces (section 2.3), are defined and explained in the next chapter.

In chapter 3, the specification for the prospective web-based application is presented. This includes the all steps from understanding and specifying the context of use (section 3.1), specifying the user and organizational requirements (section 3.2) and finally presenting design solutions (section 3.3) based on the previously specified requirements.

Chapter 4 highlights the limitations of the specification presented in this thesis, but provides steps to overcome them in the future. Additionally, the chapter discusses a fundamental idea to further improve the prospective application. The appendix holds the semi-structured outline for the interviews that were conducted for the thesis. The design solutions that are only presented partly within the text corpus of the thesis are provided in the appendix as full screen views.

---

\(^1\) Student kansliet in Swedish.

\(^2\) Studievägledning in Swedish.
2 User-Centered Design and Usability

User-Centered Design (UCD) is the approach that focuses on usability in the entire development process and life cycle of computer-based interactive systems (Gulliksen et al. 2003). In order to elaborate the UCD approach, the concept of usability will be described briefly in the following section 2.1. The section 2.2 covers UCD including its key principles in subsection 2.2.1 and in subsection 2.2.2 the activities of UCD – the process’s core. Guidelines and principles for user interface design are presented in section 2.3. They provide a good starting point for developing the interface of the prospective system.

2.1 Usability

*Usability* has two widely used definitions. For a long time, the probably best-known definition was by Nielsen: “Usability has multiple components and is traditionally associated with [the] five usability attributes: learnability, efficiency, memorability, errors and satisfaction” (Nielsen 1993). However, the ISO standard 9241-11 that defines *usability* as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (ISO 9241-11 1998) has now become the main reference (Jokela et al. 2003).

![Usability Framework](figure1.png)

**Figure 1: Usability Framework.**


2.1.1 Goals and Context of Use

Figure 1 shows the ISO standard definition’s usability framework diagram. The *goals* are derived by the intended objectives of the users, i.e. what the user is actually trying to achieve with the system. The users, tasks, equipment and environment affect the *context of use* of the system.

Crucial for the usability of a system are the people that are using it. Thus the *users* have to be specified and their relevant characteristics have to be described, such as knowledge, skills, age, physical, cognitive and perceptual abilities (Shneiderman & Plaisant 2009). In order to attribute differences among people and the roles they are performing, it may be necessary to define the characteristics of different type of users.
The activities that have to be performed to achieve a goal are described as *tasks*. The description of the *equipment* includes the hardware, software and other material required to perform the tasks. Workplace, temperature, organizational structure etc. compose physical, ambient, social and cultural *environment* of the usage situation.

### 2.1.2 Usability Measures

The *usability measures* are used to measure “the extent to which a product is usable in a particular context” (ISO 9241-11 1998) and hence builds the basis for comparing the relative usability of different products that are used for the same context (ISO 9241-11 1998). The measures provided by the ISO standard are:

**Effectiveness:** This measure is about determining whether the users are able to perform the tasks and achieve the specified goals within the system. The accuracy and completeness of the goal achievement are of interest. In a mnemonic sentence this measure could be expressed as doing the *right* things.

**Efficiency:** Efficiency is about measuring relating the effectiveness to the effort and resources the users have to spend to achieve the specified goals. Time is often used to measure the efficiency of a product. Articulated in a mnemonic sentence efficiency could be expressed as doing the *things* right.

**Satisfaction:** Satisfaction is of rather subjective nature as the users decide for themselves if they like using a product and if they are satisfied with it. Hence, this measure refers to the users’ opinion of the system expressed in the “freedom from discomfort and their general attitude to the use of the product” (Jokela et al. 2003).

For completeness, the measures mentioned in Nielsen’s definition (Nielsen 1993) will also be presented:

**Learnability:** This measure determines whether the use of a system is easy to learn.

**Memorability:** This measure refers to how easy the use of a product can be remembered over a long time of not using it.

**Errors:** This measure is about the amount of errors users make while using the problem and how easily they can recover from those errors.

However, these measures can be dissolved into the before mentioned measures. The choice of measures, their combination and their level of detail for each measure are not based on a general rule but should depend on their relative importance to the identified goals. If objective measures cannot be obtained, the users’ perception as subjective measures will give a good indication (ISO 9241-11 1998).

### 2.2 User-Centered Design

The term *user-centered* (system) *design* (UCD) was introduced by Norman and Draper (1986) in 1986. However, the meaning of the term has evolved: While the importance of having a
good understanding of the users is stressed, the users’ involvement in the design process is now also emphasized (Gulliksen et al. 2003). The ISO 9241-210 (2010) standard – formerly known as ISO 13407 (1999) – outlines UCD as a process for interactive system development with the focus to enhance usability of that system (Bevan 2009). Usability is defined in the standard as in the above-mentioned ISO 9241-11 standard. However, ISO 9241-210 does not provide a general accepted definition – only guidance for the planning and management of UCD. It also does not provide a detailed description of methods and techniques (Jokela et al. 2003). Nevertheless, UCD can be integrated into other established software development processes like the Rational Unified Process (RUP) and the family of Agile processes (Gulliksen et al. 2003).

2.2.1 Key Principles

Gulliksen et al. (2003) have identified twelve key principles that characterize a successful UCD process and that are not constrained to a particular development stage:

**User focus:** The focus should be on the users’ needs – instead of technical issues – by ensuring that the development is guided by the goals and the context of use early on.

**Active user involvement:** Users that represent the intended user group should be, at an early stage, directly and continuously involved into the whole lifecycle and development of the system.

**Evolutionary systems development:** The development of the system should be iterative and incremental as new and changing requirements occur.

**Simple design representations:** Users and all other stakeholders should be able to understand the language and representation of the design easily. Otherwise it will be difficult for them to understand the future use situation and their involvement would be sub-optimal.

**Prototyping:** Prototypes should be applied throughout the development in order to visualize and evaluate design ideas and solutions with real users.

**Evaluate use in context:** The development should be started at early stages and continuously guided by and evaluated against crucial usability goals and design criteria.

**Explicit and conscious design activities:** The development, especially of the user interface design and the interaction design, should be the consequence of design activities that are dedicated and conscious and not merely the byproduct of some coding or modelling activity.

**Professionalism:** Effective multidisciplinary teams should perform the development and system design as each activity and part of the process requires a different skillset and qualifications.
**Usability champion:** Experienced usability designers with enough authority to decide matters concerning the system’s usability should be included early and throughout the entire development process.

**Holistic design:** As software is interrelated with, amongst others, the activities, organization and practices of work, all facets that are affected by the system should be considered in the design process and modified and developed in parallel.

**Processes customization:** Since there is no UCD process that fits all situations, the contents of the process (used methods, order of activities etc.) must be adapted, customized and specified for each project.

**A user-centered attitude should always be established:** The development team and the client organization should be committed and aware to the value of usability.

### 2.2.2 Activities

![Diagram of Activities of User-Centered Design](source: ISO 13407 1999)

UCD consists of four design activities. These activities compose the core of the standard (stated explicitly in the ISO document). The illustration of the activities in Figure 2 clearly shows the iterative and incremental nature of the process. It starts with recognizing the users, their environment of use and the tasks they are using the system for. In the next step, the usability goals of the product have to be set. Furthermore, the specification of design guidelines, constrains and other requirements is included. The cycle continues by producing designs based on human-computer interaction (HCI) knowledge and the requirements. The following step evaluates those designs against the requirements established earlier. This may lead to the refinement or even creation of new user goals and requirements. The iterative process continues until the usability goals and requirements have been achieved (ISO 9241-210 2010; Jokela et al. 2003).
Specifying the Context of Use

The foundation of any UCD process is to understand the intended users of the product, their environment of use, and the tasks they are using the product for. Therefore, a typical UCD process starts with identifying the users. This includes secondary and indirect users. The emphasis is on identifying the characteristics of the users and user groups rather than individual people. Possible characteristics, following the ISO’s definition of usability (cf. section 2.1.1), are skills, education, age, physical abilities etc. (ISO 9241-210 2010; Jokela et al. 2003).

Following the identification of prospective users follows the identification of the tasks the users are to perform. “The description should include the overall goals of the use of the system” (ISO 9241-210 2010). The tasks should not be merely described as functions and features but include description of the characteristics that influence usability, such as frequency and duration of use.

For specifying and understanding the context of use, it is important to describe the environment in which the users use the product. This also includes used equipment – e.g. software, hardware and other material. Furthermore, the description should include relevant characteristics of the social and physical environment (ISO 9241-210 2010).

Several methods are available for gathering information about the context of use. They are not described or mentioned in the ISO standard but are introduced and recommended for UCD in textbooks by Constantine and Lockwood (1999), and Benyon (2010). The presented methods include: interviews, surveys, questionnaires, think aloud, observation and collecting samples of artefacts. The activity of understanding and specifying the context of use is very important as it is the basis for generating the requirements and the creation of the first design prototypes (Benyon 2010).

Specifying the Requirements

The second activity – specifying the user and organizational requirements – is basically about structuring the information collected in the previous step. Various methods to accomplish this task are presented by several authors. The most mentioned methods consist of the following:

**Personas:** A concrete description of a fictional person with the characteristics of the users. They use a narrative to describe the users’ groups behaviour, aims, physical abilities etc. Benyon (2010) stresses that it is crucial to develop several different personans as it is impossible to map all characteristics onto one person.

**Scenarios:** Scenarios describe the tasks people perform in a specified context. Their degree of abstraction can differ considerably. Starting from stories of real user experiences till conceptual scenarios (Benyon 2010).

**Use cases:** Use cases display the interaction of the users with the rest of the world – in particular, the system in development, of course. It is essentially a confrontation of the user’s intentions and the corresponding responsibilities of the system. Interrelated use cases can be mapped onto a use case map (Constantine & Lockwood 1999).
After those descriptions and diagrams have been created, they should be sorted based on importance and relevance to the users, the client, and business risks. Some measurable criteria should be produced at this stage of the UCD process to evaluate the emerging designs (ISO 9241-210 2010).

**Producing Design Solutions**

At this step of the cycle, the first designs are created. Sketches, mockups, simulations and other forms of prototypes are used to make ideas visible and facilitate efficient communication with the users. It prevents the possible need and the associated high costs for reworking the product in a later step of the lifecycle. However, “different representations of design ideas are useful at different stages for different people” (Benyon 2010). A doodle on a napkin might be sufficient to show an idea to a colleague but is inappropriate to show the client.

When the design solutions are presented to the users, they should also be allowed to carry out tasks. The users’ feedback gathered should be included in the creation of new design solutions. This should iterate until the design objectives have been met (ISO 9241-210 2010).

**Evaluating Designs**

In the next activity of the lifecycle, the design solutions are evaluated. This activity is closely coupled with the creation of design solutions but should occur in all stages of the system lifecycle. The aim is to generate feedback to further improve the product and to determine if the design fulfills the specified user requirement, usability goals and complies with general usability guidelines (Benyon 2010). The cycle of the UCD activities continues as long as the usability objectives have not been met. Consequently, there is no need to continue the iterative cycle when all usability goals have been satisfied.

Depending on the development stage and the type of design or representation of it, different evaluation methods will have varied effectiveness. Practically speaking, the choice of method depends often on the financial and time constrains of the project. Nevertheless, the evaluation methods can be divided into *expert-based* methods and participant-based methods. The first is about employing experts of usability to review the design solution. Examples of those methods: heuristic evaluation, cognitive walkthroughs and consistency inspection (Nielsen 1993; Benyon 2010).

The latter – *participant-based* methods – involves recruiting people (potential users if possible) to use the solution. This is most often more time consuming and more expensive but provides information about real-use situations (Benyon 2010). Examples of this kind of method include cooperative evaluation, cognitive walkthroughs and consistency inspection (Nielsen 1993; Benyon 2010; Dix, Finlay & Abowd 2004).

2.2.3 **Benefits of User-Centered Design**

Bias, Mayhew and Upmanyu (2003) state several benefits of employing a user-centered design process for the development of interactive systems and paying attention to the users’ needs besides an increased user satisfaction and system acceptability:
Reduction of production costs: Using a UCD process prevents the development team from over designing the product and, more importantly, helps the discovery issues early in the development – at a time when they are easy and cheap to fix.

Reduction of training and support cost: Usable systems require less user support since more users are able to perform their tasks without help. In addition, less user training and documentation material are required because carrying out tasks is more intuitive.

Increased productivity: A usable system will enable the users to have an increased work output due to a quicker workflow (e.g. less keystrokes), fewer errors committed, and faster recovery from errors. This creates higher quality work activities and decision-making. This, in turn, allows hiring less educated and less skilled employees for tasks that could previously only be performed by higher paid employees.

Making a usable system might increase the earnings due to higher conversation rates (e.g. e-commerce websites) and customer loyalty (Bias, Mayhew & Upmanyu 2003). According to Benyon (2010), ensuring a high usability can also have a significant impact on safety and therefore prevents catastrophes. For instance, the partial meltdown at the Three Mile Island nuclear power plant (USA) is partly attributed to usability issues.

2.3 Design Guidelines and Principles

Several practical guidelines and principles have crystallized over the recent years. They describe best practices and point out difficult and dangerous areas. Thus, taking the guidelines and principles in account when designing a new system will save time and result in overall better quality because errors and mistakes that others have committed earlier will not be repeated (Shneiderman & Plaisant 2009).

2.3.1 Guidelines

Guidelines have been around since the beginning of application development. Establishing specific guidelines for a project will help develop “a shared language” and promote “consistency among multiple designers in terminology usage, appearance, and action sequences” (Shneiderman & Plaisant 2009). The guidelines are records of best practices derived from previous practical experience or empirical studies and provide an excellent starting point for designers. However, guidelines can also be sometimes wrong, contradictory, incomplete or hard to apply (Shneiderman & Plaisant 2009). The following paragraphs provide a (very) brief overview to guidelines that address key topics.

Providing guidelines for navigating the interface is useful, as it will help users to find the things they are looking for in a large amount of information. Examples of guidelines that give advice on this topic (HHS 2006):

- **Standardize task sequence**: users should be able to carry out tasks in the same sequence and manner across similar conditions.
- **Ensure that embedded links are descriptive**
- **Use unique and descriptive headings**
- **Develop pages that will print properly**
• **Provide text alternatives for non-text contents**

Guidelines about *organizing the display* advise on presenting data to the user. Shneiderman and Plaisant (2009) summarized Smith and Mosier’s report (1986) of guidelines for data display to five high-level goals:

1. **Consistency of data display**
2. **Efficient information assimilation**
3. **Minimal memory load on the user**
4. **Compatibility of data display with data entry**
5. **Flexibility for user control of data display**

The display of time-dependent information or exceptional conditions may require *getting the user’s attention* to differentiate it from the information of the normal work performance. The following list (Shneiderman & Plaisant 2009) presents some techniques for attracting the user’s attention. Realizing these methods is relatively simple. Therefore, designers have to remember that less can be more. An excessive use of the techniques antagonizes their purpose and may disturb and annoy the users, thus reducing the overall acceptance of the system.

- **Intensity**: Use not more than two levels and the high intensity level only rarely to attract attention.
- **Marking**: Use techniques like underlining, enclosing in a box, pointing with an arrow or indicating with special symbols to attract attention to a particular item.
- **Size**: Use not more than four sizes and limit using larger sizes to grab attention.
- **Choice of fonts**: Use not more than three fonts.
- **Inverse video**: Inverse the colors.
- **Blinking**: Use with great care blinking color changes.
- **Color**: Use not more than four standard colors. Additional colors can be used for occasional uses.
- **Audio**: Harsh sounds should only be used in emergency conditions. Use soft sounds for normal positive feedback.

Users generally spend a significant fraction of their time on data input when using a system. Shneiderman and Plaisant state that the input of data “can be the source of frustrating and potentially dangerous errors” (2009). They have extracted five high-level objectives from Smith and Mosier’s (1986) collection of guidelines for facilitating data entry:

1. **Consistency of data-entry transactions**
2. **Minimal input actions by user**
3. **Minimal memory load on users**
4. **Compatibility of data entry with data display**
5. **Flexibility for user control of data entry**

### 2.3.2 Principles

In contrast to guidelines, which are narrowly focused, *principles* can be characterized as being more fundamental, widely applicable and enduring. A major principle is to *determine the users’ skill levels*. The designers should be aware and understand the intended users of the
interface they are designing. To be successful they must be aware “that people learn, think, and solve problems in different ways” (Shneiderman & Plaisant 2009). Being aware of the difference of people and groups of people may lead to different design goals (Shneiderman & Plaisant 2009).

Since the principles of identifying the tasks – similar to the principle of determining the users’ skill levels – totally overlaps with the description of specifying the context of use in section 2.1.1, the principle described in the next list briefly covers the choice of an interaction style. The primary styles of interaction are:

**Direct manipulation:** Direct manipulation is about providing visual representations of objects, actions, and feedback. Users can perform tasks quickly by pointing at the visual representations and can observe the results immediately. This style is appropriate for novice, intermediate as well as expert users because it is easy to remember and allows fast usage if designed well.

**Menu selection:** In menu-selection, the users are presented a list of tasks. They then select the most appropriate and observe the effect. It is important to carry through a thorough task analysis and carefully choose consistent terminology. If designed well, this style of interaction is appropriate for all three kind of skilled users.

**Form fill-in:** This interaction style is about letting the user fill in the blanks when entering data. It is good for tasks that require a lot of data entry. Since the users have to know and understand the field labels and permissible values, this interactions style is only appropriate for intermediate and frequent users.

**Command language:** Command languages allow users to declare complex possibilities quickly. However, it requires a lot of training, as the users have to learn the syntax of the language. Therefore, it is only appropriate for expert frequent users.

Shneiderman and Plaisant (2009) formulated *eight golden rules of interaction design* that provide a useful guide for designing interactive systems:

1. **Strive for consistency:** Similar situations require consistent action sequences. In prompts, menus and help screens identical terminology should be used.
2. **Cater to universal usability:** Recognize the differences of people and add appropriate features. People with disabilities should be able to use the system.
3. **Offer informative feedback:** There should be a feedback for every user action. Frequent and minor actions may merit modest feedback; unlike for infrequent and major actions more substantial feedback may be required.
4. **Design dialogs to yield closure:** Sequence of actions should have a defined beginning, middle and end. Provide feedback when a coherent group of actions has been completed.
5. **Prevent errors:** The design should prevent the user from committing serious errors. Simple instructions for recovery should be provided in case of an error.
6. **Permit easy reversal of actions:** Provide an undo function wherever possible.
7. **Support internal locus of control**: The user should initiate the actions the system responds to.

8. **Reduce short-term memory load**: Avoid interfaces where the users have to remember information from one screen to the next screen.

The presented principles have to be interpreted, refined and extended for each environment. They provide a good starting point for interface designers even though they have limitations (Shneiderman & Plaisant 2009).
3 Application of User-Centered Design

In the previous chapter the objectives, principles and activities of usability and UCD were presented. They emphasize a focus on the user and on the context of use of the system. Therefore, this chapter shall investigate users and the context of use for the new web-based student case application. As outcome it shall provide a thorough specification for the implementation of that system.

Since there is not a UCD process that fits to all projects and teams, a process, the used methods and the order of the activities have to be adapted to the particular needs of the project (Gulliksen et al. 2003). This chapter states how UCD was implemented in this project and the outcomes of the used methods.

![Figure 3: Adaptation of the UCD activities for the project.](Image)

This chapter is outlined in close congruence to the activity cycle of ISO 13407 (1999), as it is illustrated in Figure 3. Hence it starts with understanding and specifying the context of use (section 3.1), followed by specifying the user and organizational requirements (section 3.2) and ends with the production of design solutions (section 3.3), which is closely coupled with the evaluation of the designs. Each of the section (one for each UCD activity) starts with a subsection explaining the methodology of gaining the relevant information or respectively building the corresponding models. Subsections presenting the findings and results then follow that starting subsection.

3.1 Specifying the Context of Use

As described in the ISO standard the first step of the UCD process is about understanding and specifying the context of use (cf. section 2.2.2). Thus at first the users and their tasks have to
be identified and the environment in which they perform the tasks has to be described. However, before the users, the tasks and the environment for the prospective system are covered (section 3.1.3), the currently used email-based system is briefly described (section 3.1.2); including the problems the user’s face with it and the expectations they have for a new system.

### 3.1.1 Method
Since there is no preexisting explicit knowledge about the tasks performed by the student office and student counselors, a very explorative research approach had to be chosen. To gain the needed knowledge about the tasks and processes between the two university bodies, interviews with members of both teams were conducted. As the real users were accessible no ersatz (substitute) users had to be interviewed.

The interviews were open, semi-structured and qualitative interviews. Till the interview outline reached the form presented in Appendix A, it underwent several iterations to optimize its structure and phrasing. The partial standardization introduced by using an interview outline allows comparing the conducted interviews and ensures that all relevant topics are covered during the interview (Mayring 2002).

The interview questions were designed to be open to allow the participants to answer freely. This has according to Mayring (2002) the distinct advantage that the interviewees reveal their very subjective perspectives, views and interpretations and are able to develop independently bigger coherences. Since the interviewee shouldn’t feel squeezed, the questions were phrased to be friendly and unthreatening (Mayring 2002; Yin 2003). Additionally, a relaxed atmosphere was induced as much as possible by performing some small talk at the beginning of the interview (Bortz & Döring 2006). The interviews were conducted in person and recorded.

### 3.1.2 Current Email-Based System
Since the prospective system is supposed to replace the old system based on using emails to exchange the correspondence between the student office and the student counselors, the current context of use should be summarized before the context of use for the new system is defined.

**Context of Use**
Currently the email system has three users: students, people of the student office and the student counselors. If students want to (among other things) register or re-register a course they have to use the web-based system Studera.nu. However, due to several reasons – for instance, being an exchange student or taking distance study courses – a student might not have access to Studera.nu or the permissions to trigger that transaction.

In those cases, the student is required to talk to a student counselor about his request. The student initiates the request either by email or a personal meeting. If the student counselor agrees to the request, the counselor will write an email to the student office with the request and confirmation to grant the student’s request. An officer of the student office will then
update the Uppdok system manually to comply with the request. But not in all cases immediately.

Student counselors cannot update Uppdok themselves when they want to grant a request because they don’t have the permission to do so. The members of the student office cannot grant the permission to a student, for example, to take a certain course, as they don’t possess the knowledge of which prerequisites have to be met by the student like, for instance, knowledge of previous courses.

In the case that the student states the request to the student office first, the student office will refer the student to the student counselors. If the request came by email the officer will simply forward the request to the counselors. This triggers the counselors to also write an email to the student office if the request of the student cannot be granted (for example, because the student has not the required prerequisites to take a particular course).

The counselors and officers use the email client Thunderbird to access the emails. The email system is configured in the way that every student counselor has access to all the emails written by the other counselors regarding the group of task described above. The same is true for the student office: All student officers can view the emails written by the other officers as if they were written by themselves. However, the people of the student office use the email system to organize and group requests for later fulfillment.

For instance, all requests (already permitted by the student counselors) for the summer course registration will be moved into a special mailbox dedicated solely for that kind of requests. The actual registration will, however, be performed at a later time when the deadline of the official registration period on Studera.nu has been elapsed. The student office is not writing any confirmation of a successful request fulfillment to neither student nor counselor.

**Issues and Expectations**

Even though, the student office and the student counselors get the job done with the email system, both parts are very unhappy with it. One of the student counselors even started writing an initial (but very brief) specification due to his dissatisfaction with the current system. This specification is of a very functional character and has only been used as the basis for some functional requirements like the data needed to be saved for a request ticket (cf. section 3.2.7).

Anyway, the main source of dissatisfaction for the people of the student office and the student counselors is the immense lack of clarity and visibility of the email-based system. Users of both groups complained that it is very difficult to find the emails concerning a particular case. Especially, when there is an intensive email exchange associated with the case, it is very cumbersome to find and collect all the relevant information to either continue the conversation or fulfill the request by entering the information into Uppdok.

Furthermore, the student counselors hope and wish that a new system would increase the transparency of the whole process. They complain that they never receive any feedback from the student office if a request was successfully (or not) entered into Uppdok. The student
office is looking forward to a possibility to ease the grouping of cases and receiving an overview of all open cases with ease.

In essence, the new system shall increase the clarity, the visibility and the searchability of cases. In their unanimous opinion, members of the student office and student counselors regard the current email-based system as unnecessarily cumbersome.

### 3.1.3 Future Web-Based Application

This section (finally) states the results of the first UCD activity. As stated in section 2.2.2 on page 6 the foundation of any UCD process is to understand the intended users of the system, their environment of use and the tasks they are using the system for. This is not about an individual person as user but characteristics of the users and user groups.

**Identifying the Users**

The two user groups who will use the system directly are the people of the student office and the student counselors of the IT Department of Uppsala’s University. The users of both groups possess good or very good computer office and Internet skills. The age ranges from 25 to 60 years. All have at least a successfully finished Gymnasium education. No impairments are known.

The students can be regarded as indirect users of the system. They are not involved in the communication between the student office and the counselors. But their requests trigger this communication and they are naturally very interested in the outcome of that correspondence.

**Identifying the Tasks**

The student counselors must be able to create new requests that the student officers have to register with the Uppdok system. The request tickets should include all the necessary information for performing the fulfillment of the request. Thus, it must be possible for both department bodies to request additional information from the other body about a particular request ticket.

Both groups of users need to view the list of ticket requests. Not just view the open request tickets but also the tickets that have been marked as done. Sometimes they are required to view the email of the student that initiated the request. The student counselors need to be able to change the information associated with a request ticket.

When the request is completed or/and registered a student officer and a student counselor have to mark the request as done to show the other party that their part of the request is completed. Depending on the period of the academic year ten to a hundred requests might be issued per week.

Both user groups stated that they need to search and find request tickets; including already completed requests. For instance, in the case a student inquires the current status and progress of the course registration request issued by the student. The student officers need to be able to mark a request ticket for transfer to Uppdok at a later time. But it should be visible that the
ticket is accepted but due to internal processes scheduled for a transfer to Uppdok at a later time.

**Description of Environment and Equipment**

Both user groups have a dedicated office for their work as student officers and student counselors. The two offices are in different buildings but the hallways of both offices are connected to each other via a covered and closed bridge. Thus both offices are (kind of) on the same floor just 200 meters apart from each other. However, the officers might always work from their office, the counselors are not. As the counselors are also giving lectures they each have their own private offices. Only for the opening hours they will be in the student counselors’ office. Their own offices are scattered across the university building.

![Figure 4: Workplace of a member of the Student Office.](source: Own photograph.)

Each individual user has at least one computer available with at least one display (with the physical screen size of 24” and a resolution of 1980*1200), a mouse and a keyboard. The usual working computer runs the Windows 7 operating system. The members of the student office use a recent version of the Internet Explorer to access websites in the Internet and Intranet. A recent version of Firefox is also available but not used as much as the Internet Explorer. The student counselors use Firefox. The web browser is usually run with its window size not maximized but very large.
3.2 Specifying the Requirements

As presented on page 6 the activity of specifying the user and organizational requirements is basically about structuring the above-mentioned information. The following paragraphs display the outcome of employing methods for structuring the information.

3.2.1 Method

For the second activity – specifying the user and organizational requirements – the information gathered in the previous activity is structured. This is done by compiling user roles and, if appropriate, user role maps. Additionally, for a better understanding of the situation of use scenarios are created. Furthermore, use cases and use case maps are built for the specification of the tasks. However, even though each method will be shortly defined or explained, an explanation of how to employ them is not given.

3.2.2 User Roles

Constantine and Lockwood (1999) define a user role as “an abstract collection of needs, interests, expectations, behaviors, and responsibilities characterizing a relationship between a class or kind of users and a system.” A single user can take on several roles. Each role can be played by any number of users. Following are the two identified user roles:

- **StudentOfficeStaff**
  - frequent and regular almost daily use; periodic high volume (up to hundreds) of requests per week; view request tickets; postponement of requests for later processing; need to group and search request tickets on arbitrary criteria; access to Uppdok; criteria: efficiency in use; reliability in use; learnability

- **StudentCounselor**
  - frequent and regular almost daily use; periodic high volume (up to hundreds) of request per week; view request tickets; add request tickets; search ticket on arbitrary criteria; receive feedback of Uppdok entry; criteria: efficiency in use; reliability in use; learnability

These two user roles can be regarded as the so-called focal roles. “Focal roles are those few user roles judged to be the most common or typical […]” (Constantine & Lockwood 1999). Even though, it is merely an indirect user the role of the student, or respectively, requestor will also be presented, in order to provide an utmost holistic view upon all users of the system.

- **StudentRequestor**
  - seldom indirect use; inquiry of request progress and status; initiates request via StudentCounselor;
3.2.3 User Role Map

User role maps are a means to provide the big picture of the system’s user roles. The user role map displays the interrelationships between the roles a user can play in the system. The user roles can be interrelated by affinity, classification, and composition (Constantine & Lockwood 1999). Due to the limited amount of user roles for the student-case management system there is only one relationship of the type affinity (cf. Figure 5). This interrelationship exists between the **StudentOfficeStaff** and the **StudentCounselor** role. The roles resemble each other because their style of interaction is similar and they share a variety of common characteristics.

![User Role Map](image)

**Figure 5: User Role Map for the web-based Student-Case Management application.**
*Source: Own illustration.*

3.2.4 Scenarios

“Scenarios are narrative description of an activity or activities” (Constantine & Lockwood 1999). The scenarios depicted in the next paragraphs are quite abstract, as this lets them to be “particularly useful for generating design ideas and for understanding the requirements of the system” (Benyon 2010).

**Adding a request**

A student without access or permission to Studera.nu sends a request to the student counselors. A student counselor will check if the request is legitimate. In the case that the request is legitimate, the student counselor sends the request including all relevant information to the student office.

**Fulfilling a request immediately**

The student office receives a request. They check if all necessary information is available and correct. If necessary they inquire missing or correct information from the student counselors. If all information is available and correct, a member of the student office will incorporate the contents of the request into Uppdok.

**Fulfilling a request later**

The student office receives a request. They check if all necessary information is available and correct. If necessary they inquire missing or correct information from the student counselors. If all information is available and correct, a member of the student office...
office will move the request to a dedicated mailbox. Personnel of the student office will – a certain later time – incorporate the contents of the request into Uppdok.

### 3.2.5 Use Cases

A *use case* is a case of use and expressed as “a narrative description of interaction between a user – in some role – and some system” (Constantine & Lockwood 1999). They are often formulated in a linear continuous sequence. Constantine and Lockwood differentiate between use cases and *essential use cases*. Essential use cases focus more on the purpose or intentions of a user than the non-essential (or respectively, conventional) use case, which contain too many premature assumptions. Since they recommend the use of essential use cases, that kind of use cases will be employed for this work.

The following are the use cases in this application based on the findings presented in the previous subsection 3.1.3:

<table>
<thead>
<tr>
<th><strong>viewingRequestTicketList</strong></th>
<th><strong>System Responsibility</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Intention</strong></td>
<td><strong>System Responsibility</strong></td>
</tr>
<tr>
<td>ask for list of request tickets</td>
<td>present list of request tickets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>filteringRequestTicketList</strong></th>
<th><strong>System Responsibility</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Intention</strong></td>
<td><strong>System Responsibility</strong></td>
</tr>
<tr>
<td>ask for request tickets based on certain criteria</td>
<td>present all request tickets as list that satisfy the certain criteria</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>addingRequestTicket</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Intention</strong></td>
</tr>
<tr>
<td>indicate request ticket to add including all necessary and relevant information</td>
</tr>
<tr>
<td><strong>viewingRequestTicket</strong></td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>ask for a particular request ticket</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>addingCommentToRequestTicket</strong></th>
<th><strong>USER INTENTION</strong></th>
<th><strong>SYSTEM RESPONSIBILITY</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ask for or provide further information about a particular request ticket</td>
<td></td>
<td>add question or answer to the particular request ticket</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>readingCommentOfRequestTicket</strong></th>
<th><strong>USER INTENTION</strong></th>
<th><strong>SYSTEM RESPONSIBILITY</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ask for provided further information about a particular request ticket</td>
<td></td>
<td>present further information if available</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>markingRequestAsDone</strong></th>
<th><strong>USER INTENTION</strong></th>
<th><strong>SYSTEM RESPONSIBILITY</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>indicate that a particular request ticket has been fulfilled</td>
<td></td>
<td>make the particular request ticket marked as done</td>
</tr>
</tbody>
</table>
markingRequestAsApproved

<table>
<thead>
<tr>
<th>USER INTENTION</th>
<th>SYSTEM RESPONSIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>indicate that a particular request ticket has been approved for fulfillment but the fulfillment is scheduled for a later time</td>
<td>make the particular request ticket marked as approved</td>
</tr>
</tbody>
</table>

updatingRequestTicket

<table>
<thead>
<tr>
<th>USER INTENTION</th>
<th>SYSTEM RESPONSIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>indicate information changes of a particular request ticket</td>
<td>save the updated information of the particular request ticket</td>
</tr>
</tbody>
</table>

3.2.6 Use Case Map

Similar to the user role maps are the use case maps a means to provide the big picture of the system’s use cases. The use case map displays the interrelationships between the use cases. They can be interrelated by classification (specialization), by extension, by composition or by affinity.

![Use Case Map Diagram]

Figure 6: Use Case Map for the web-based Student-Case Management application.
Source: Own illustration.

Figure 6 represents the use case map for the web-based student-case management application. The position of the use cases reflects their importance based on the estimated quantity of use
or the estimated effect a good designed solution has on the use case. The order is from top to bottom. For instance, reading a comment of a request ticket is done more often than writing a comment.

The positioning of the elements in the map also represents the interrelationship between the displayed use cases. The overlapping use cases, for instance, markingRequestAsDone and markingRequestAsApproved signal an interrelationship of the type affinity, as both use cases are closely related to each other. Additionally, the position of use cases that are not overlapping hints to their relationship to the other use cases. The position of addingRequestTicket and updatingRequestTicket shall display that both use cases are somewhat related. The same is true for viewingRequestTicket and viewingRequestTicketList.

The use case filteringRequestTicketList is a specialized version of the use case viewingRequestTicketList because filteringRequestTicketList is basically representing the same user intention – getting a list of request tickets – but filtered by criteria specified by the user. In other words, viewingRequestTicketList could also be declared as filteringRequestTicketList but without specifying any filtering criteria.

Mapping the interrelationships as extension points out that in some situations the interaction of the extending use case may be inserted in the extended use case. For instance, the use case of reading the comments of a request ticket may be inserted into the use case of viewing a particular request ticket.

### 3.2.7 Required Data

The members of the student office need certain pieces of information to perform a request. These pieces of information have to be collected by the student counselors. The following list includes all information that is needed by the student office to fulfill a request:

- **Case type**: this states what actually has to be done.
- **Student's name**
- **Student’s personal identification number**: a number that uniquely defines every student of the university. The Swedish government usually issues the number but exchange student normally get one by the university.
- **Course code**
- **Course name**
- **Application code**: A code that defines each course for each semester uniquely. The code is issued by Studera.nu.
- **Study Program**

Additionally, the people of the student office and the student counselors asked for a possibility to save a reference to the original email that triggered the request. Especially, the student counselors need it in case they require more information from the student about the student’s case.

---

3 In Swedish: personnummer.
3.3 Design Solutions

As stated in chapter 2.2.2 the next activity after the user and organizational requirements have been defined is to create prototypes and design proposals. The following section will present the design solutions and proposals for the web-based student-case management application. The general method employed is described in section 3.3.1. If for a particular design solution a special method was used it will be mentioned in the section concerning that particular design solution.

3.3.1 Method

The process of creating a design solution started with very low-fi prototypes. For instance, the positioning of elements in an application screen was scribbled on pieces of paper before a more high-fi version of that particular application screen was produced. Using sketches and other forms of low-fi prototypes allowed a fast and quite inexpensive design of the interaction concepts for the product and initial determination of which information to present, its location of presentation and in which order the information is presented in respect to the other information.

Each design artifact was evaluated after its creation against its compliance with the guidelines and principles introduced in chapter 2.3 and its performance in a rather superficial heuristic evaluation. However, this chapter will only display the outcome of the designing activity and not its intermediary steps. The high-fi prototypes were thoroughly evaluated by a heuristic evaluation and in cooperation with the users. The results of the evaluation and the user feedback was incorporated into the design proposals and illustrations depicted in this subchapter.

The high-fi prototypes illustrated in this work, were created with OmniGraffle\(^4\) and Konigi’s Omni Graffle Wireframe Stencils\(^5\).

3.3.2 Aspects regarding the Overall Design

Look

The look of the design solutions were designed with respect to the – at the time of writing – new design of the Uppsala University’s IT Department website. The web-based student-case management application should be seen by the users as an integrated part of the department’s website (cf. Figure 8).

Navigation

The first two navigation levels – the horizontal bar on top and the vertical menu on the left – are static throughout the entire application. The horizontal navigation bar is not changed at all and remains as designed for the IT Department’s website. A link, or respectively, a menu

\(^4\) URL: http://www.omnigroup.com/products/omnigraffle/
\(^5\) URL: http://konigi.com/tools/omnigraffle-wireframe-stencils
entry is added to the menu on the left. This link will direct the user to the application (cf. Figure 8 – “Student Case MGMT”).

It was a conscious decision not to add any menu hierarchies to the application, as this would have cluttered the interface. Especially, considering that the mockups with a third menu do not add any usability to the application compared with the current design solutions, strongly supports this decision.

An explanation of more detailed navigation paths will be given in the following descriptions of each screen.

**Dropdown-List with Fast Item Selection**

This special kind of graphical user interface (GUI) element is used throughout the entire application. It combines the advantages of a normal text field with the advantages of dropdown lists. The combination even leads to significant synergy-effects. It combines the principles of *menu-selection* and *form fill-in* (cf. section 2.3.2).

![Dropdown-List with Fast Item Selection showing the suggestion feature.](source)

As in dropdown lists the user can click the downwards-facing arrow to see all valid and available items for this form field. This is very useful for new and occasional users as no memorization of all available options is required. However, to satisfy the expert users as well, the GUI element allows the user to type-in the requested item. While typing the element suggests options that match the typed-in string. The user can choose the preferred option by either completely typing it or by simply selecting it via the use of the arrow-keys on the keyboard. Of course, a selection by mouse click is still possible.

**Form of the URL**

The URL of each site is formatted in a way to include a very distinctive term of the site they are pointing to. For instance, the URL to the detailed view of a student named “Victoria Lindgren” will contain the name of that student. This supports the history-functionality of web browsers. If the user types in the name of that student into the browser’s address bar it will present that particular link if the user has already visited the page with detailed information about a case associated with that particular student.

**Introduction of Case Period**

A case period is a case’s position in the academic calendar (or whatever the users like to specify). It does not specify an absolute position in time but a position relative to the other case periods. The case period can be classified as a necessary meta-information required for
each student case. The rationale for its introduction is to ease the retrieval of cases. With help of the *advanced search* functionality (cf. section 3.3.4) the user can view all cases of a certain period, before or after a certain period.

Using a manually set period for a case instead of an automatic method, for instance, based on the course schedule, gives the users the freedom to set a time period for a case as they see fit. Additionally, using an automatic method would have traced issues caused by the selected case type and how that should be interpreted.

### 3.3.3 Case Overview Screen

The overview screen is the main screen for both user groups. It is the first screen a user sees when accessing the application via the (left) menu. It lists all cases registered with the application. However, the cases that are actually displayed depend on the specified search term, or respectively, filter settings (more on page 26).

![Figure 8: Case Overview Screen.](https://example.com/caseOverviewScreen.png)

*Source: Own illustration.*

At the top (of the part that is relevant to the application) of the screen resides the input text field for the *search* functionality. The user can either input a student’s personal identification number or a student’s name to find the cases associated with that particular student. The search should be triggered by clicking on the *Search* button or in case the search input field is still on focus by either hitting *Enter* or *Return* on the keyboard. A click on the
more Search Options button expends the search bar and presents more search options to the user. The search functionality is covered in the section 3.3.4 in more detail.

Underneath the search-bar resides a pagination element. In case more than a specified number of cases satisfy the display criteria, or respectively, search criteria, the cases should be split up to several result pages. The navigation between those pages is accomplished by the pagination element. A second pagination element resides underneath the table. Reasons for only displaying a specified number of cases on one page are to prevent clutter but more importantly prevent enormous load times. On the far right of the pagination element sits the Add new Cases button. It links to the page that allows the addition of cases to the system (cf. section 3.3.5). This button is only visible to the student counselors.

Between the two pagination elements resides the main element of this screen: the table of cases. It consists of eight columns that provide the users with enough information to select the case they were looking for. The column headers allow the user to sort the table ascending or descending for the particular column when clicked on the header. Each column should be self-explanatory from looking on the screenshot (cf. Figure 8).

However, the first column – case type – should be explained in more detail. It represents the case type by the string associated with the case type. But in front of the string the status of the case is depicted by a pictogram. The gray star represents cases that a student counselor has marked as done (concerning the student counselors parts of the process) and send to the student office. The check-mark on a gray background signals cases that have been put on the waiting list by a member of the student office. The check-mark on a green background depicts cases that have been marked as done by a member of the student office. Cases without any pictogram are cases that have only been saved by a student counselor but not yet marked as done and send to the student office.

Rows with an italic font style signal student cases the current logged-in user has worked on by creating it, marking it as done or commenting on it etc. This allows the users to easily identify cases they have worked on. The row, which is hovered by the mouse, is highlighted. In Figure 8, it is illustrated with the row highlighted in blue. A click anywhere on a row opens the detail page of the case of that row. A click on the View-button does the same.

3.3.4 Advanced Search Feature

A good implementation of the advanced search feature is of utmost importance. It ensures the retrieval of cases when the system is hosting large numbers of student cases. The extend search options are displayed when the user clicks on the more Search Options (cf. Figure 8; far right of the search bar). Extending the search bar pushes the content underneath the search bar further down the page. All GUI input elements are of the type discussed in section 3.3.2 to ensure a fast usage.
Except for **Case Status**, **Case Period** and **Last Update** all dropdown lists will offer the options to select *All* or one of that kind (like case type, or study program etc.) that is registered with the system and has at least one case associated with it. **Case Status** will offer the options *All*, *Not marked*, *Done by Student Counselor*, *On Waiting List* or *Done by Student Office*.

For **Case Period** and **Last Update** the system will offer the user for the first dropdown list the choice between *within*, *after* and *before*. The second dropdown list of **Case Period** will let the user choose between *All* and all case periods registered with the system that have at least one case associated to it. The second input field of **Last Update** will hold a date. If nothing is selected in the dropdown list nothing will be considered for that particular field.

The user can move between the dropdown lists by hitting the *TAB*-key on the keyboard. With the use of AJAX every input shall instantaneously affect the table, which lists the cases (cf. section 3.3.3). For instance, if the user selects *Done by Student Counselor* as case status, the website should immediately after the selection – without any further required interaction – only show student cases which have been marked as done by the student counselors.

### 3.3.5 Add Case View

The *add case screen* allows the student counselors to add new cases to the system. Users that belong to the student office do not get granted access to this page. The page is accessed by clicking the *Add new Case*-button displayed in the case overview screen (cf. Figure 8). This view makes heavy use of the special GUI element presented in section 3.3.2. Each data input field except for the field for the student’s personnummer, the student’s name and the reference to the student’s email filing the request employ the special GUI element.

If a proper connection to the university’s other database can be achieved only the form fields on the left have to be filled out because they uniquely identify the value for the fields in the right column. For instance, by providing the student’s personnummer the system should be able to retrieve the student’s name. Those database lookups should be performed with AJAX calls.
By pressing the **Add Case**-button the case will be added to the system’s database. As the student counselor will mostly possess all necessary information when creating the case, the case can be immediately marked as done and send to the student office. Thus the checkbox is checked by default. After adding the case the user is redirected to the detailed view of that case. This allows the user to easily review the case. However, when the **Cancel**-button is pressed the user is redirected to the overview screen.
The **plus on a gray background** and the **gray gear** besides the input field for the case type and case period depict further options. The former is for adding either another case type or case period. The latter is for editing already existing types and periods. A click on the button will open the corresponding inline screen. A semitransparent layer masks the elements in the background. An illustration of both for case periods can be seen in Figure 11 (Appendix B.1 illustrates it as a full screen view).

For adding a new case period the user must specify its name and its position in time. A new period can be either **before** or **after** an already existing period. Clicking on the **Add**-button obviously will add the new period. The newly added period will be automatically selected in dropdown list GUI element. A click on **Cancel** or the cross will close the inline screen and remove the semitransparent layer.

When editing a case period, the user has to select a particular case period. If a period is already selected in the corresponding dropdown list then that period will also be automatically selected in the inline editing screen. The user has three choices of how to edit the case period. Either **rename the period**, **change its position in time** or **delete the period**. In the case deletion is chosen, the **Save**-button will be renamed to **Delete**. As in the previously mentioned inline screen, a click on **Cancel** or the cross will close the inline screen and remove the semitransparent layer.

The inline screens for adding and editing case types, look almost and behave the same as the above-mentioned two inline screens. The only difference is that they don’t offer the functionality regarding the case type’s position in time, as this wouldn’t make any sense.

### 3.3.6 Detail View of Case

The detail view of a case is shown when the user clicks on one in the overview table (cf. 3.3.3). It shows all the information associated with the case. In this view the user can change the status of a case. The view illustrated in Figure 12 is for the student office. Hence, it offers the user to change the status of a case to **Mark as Done** or **Put on Waiting List**. There are two possibilities in the view to do this to avoid the user the hassle of scrolling all the way down. In other words, to give the user more control and flexibility.

In the second half of the screen the commenting, or respectively, inquiry feature can be seen. Each user can comment on a case, the user is granted access to. A comment is added by typing it into the text area at the bottom and hitting subsequently the button **Reply**.

It also acts as a timeline and allows the user to track the progress of a case. To illustrate who interacted with the case the user’s name (who performed the specific interaction) is shown next to the comment or the information describing the interaction. Underneath the name the accurate time and date that interaction had happened is displayed. To ease the recognition of who interacted with the case a little thumbnail picture of that user is show above the user’s name.
The detailed view of a case for the student counselors can be seen in Appendix B.2. It differs in the available buttons for interaction. A user belonging to the user group student office can either mark a case as done or put it on a waiting list. A user belonging to the student counselors user group can only mark a case as done and send it to the student office. However, a student counselor can also edit a case.

If a user has marked a case as done, put it on the waiting list or marked as done and send to the student office, that particular button will change its function to undo the corresponding action. For example, **Mark as Done** would become **Unmark as Done**. Obviously, after clicking on **Unmark as Done** the case should not be marked as done anymore.

### 3.3.7 Case Edit View

The edit view of a case essentially looks almost the same as the add view with the exception that all input fields (e.g. dropdown lists) have the values selected that have been saved for that case. However, they function the very same way as in the add screen. In addition, to the add view the edit screen will also show the timeline of the case. It also offers the user the option to delete the case.
Then the Save-button is clicked, the system will save the changes and return to the detail view of that case. In case, Cancel is clicked no changes will be saved and the user is returned to the detail view of that case. When Delete Case is clicked the user gets redirected to the overview screen. Also a feedback message is shown whether the deletion request was successfully executed or not. However, as can be seen in Figure 14, it also shows the option to undo the deletion.

3.3.8 Discussion

An application that follows the specification and proposals defined in the previous chapter will fulfill the aims of replacing the email-based system with a system that increases the productivity, is usable and will be accepted by the people working with it. It keeps the email-based system’s advantage of a fast, cheap and asynchronous way for communicating but overcomes its context dependent disadvantages. It provides a coherent structure for the relevant information of student cases. That makes it easy and efficient for the members of the
student office and the student counselors to grasp the status of a student case and its information as well as the overall condition of the student cases.

The proposed design solutions incorporate the usability heuristics of Nielsen (1994) and the eight golden rules by Shneiderman (2009). The use of dropdown lists with fast item selection satisfies the two heuristics recognition rather than recall and flexibility and efficiency of use. The former states that the users’ memory load must be minimized. This is clearly accomplished as the users do not have to memorize almost any input data because all valid inputs are displayed in the dropdown-list. However, since it is also possible to input the information by typing, the latter heuristic is also met.

Nielsen’s heuristics of error prevention and user control and freedom and Shneiderman’s golden rule of permitting easy reversal of actions is satisfied by providing undo functionalities throughout the entire application. For instance, an unwanted deletion of a case can be easily undone by clicking undo immediately after the removal has been ordered. Using such an unobtrusive mechanic eliminates the need for various intrusive confirmation dialogs. Integrating already existing databases into the application can even increase the error prevention further. For example, only course codes should be available to be selected (or written) that are inside the course database.

A critical point for the success of the web-based application will be the choice of the default values for the overview screen. Unfortunately, an appropriate set of default values can only be defined after a proper observation of the users’ usage patterns when using the new system. To derive values from the observations based on the current email-based system will probably not yield good results, as the change of usage patterns is too severe.

Another issue that can only be really successfully dealt with when it is possible to observe the users using the new system is the selection of columns for the table in the overview screen. Only after some time of usage the users are able to determine which information they need to properly select a case from a list of cases. Currently, they do not know because they do not have a standardized list of cases. The selection of the columns of the current overview table is based on their assumptions.

The implementation of the application itself can be considered quite difficulty due to the extensive usage of AJAX-based functions. The AJAX-powered instant (or respectively, live) search could be delayed to a later iteration of the software. However, the very advanced dropdown-list should be implemented in the first version deployed at the users’ workplace because they are widely used throughout the application and promise a huge yield in increasing usability.
4 Limitations and Future

Obviously, the application defined in the specifications has to be implemented. Only, this allows a final conclusion and evaluation if it really increases the productivity and satisfies the users. Additionally, to yield thoroughly valid results that satisfy scientific requirements the evaluations have to be performed by other independent experts than the author of this work.

To further improve the efficiency for expert and frequent users keyboard shortcuts should be introduced throughout the entire application. If done properly, the users should not need to lift their hand from the keyboard to the mouse-pointing device. Of course, the shortcut keys should be consistent throughout all views.

The application should use the university-wide CAS login system. CAS is used for most other web-based applications within the university software landscape. It is a system that provides single sign-on for all connected applications. Thus, if the user has already logged-in with CAS, the user is already logged-in for all other applications that incorporate CAS. This increases and simplifies the workflow between the applications significantly. As already the users’ names and even pictures are deposited with CAS, the new application can easily use them for, for instance, the functionality regarding the case commenting.

Furthermore, the inclusion of the students should be considered in the future. Currently, there is a disruption of media. The student sends the request by email and the student counselor has to transfer it into the student case management system. In case, the original request has to be viewed, it is always necessary to switch back to the email application. An inclusion of the students would prevent that disruption of media and reduce the work of the student counselors as the initial ticket is created by the students.

However, this inclusion clearly faces some issues that have to be resolved. For instance, that the student do not regard sending a request to the student counselors as the official progress. It has to be ensured that committing a request to the student counselors is only allowed if nothing else works. Furthermore, it has to be resolved what the students are allowed to see in the application. For example, a student should not be able to view all comments between student office and student counselors regarding the student’s case.

---

6 URL: http://www.jasig.org/cas
Bibliography


Appendix

Appendix A  Interview Outline .................................................................37
Appendix B  Design Proposals .................................................................38
  Appendix B.1  Add Case Period Fullscreen........................................38
  Appendix B.2  Detail Case View for Student Counselors.........................39
Appendix A  Interview Outline

Start

• Can I record the interview?
• Acknowledge anonymity to the interviewee.
• How would you rate your computer skills on a system of 1 to 10? 10 being on the level of computer guru.

Purpose of Student Office and Student Counselors

• What is the general job of the student office/student counselors?
• How would you describe the job of the student counselors/student office?

Tasks of prospective System

• Which student issues require the cooperation of the student office and the student counselors?
• What tasks need to be done with the system?
• How often do you perform those tasks?
• What are the goals for the system?

User Identification

• Are other people besides the student office and student counselors interested in task fulfillment? Who?

Rationale for prospective System

• What do you like about the email system?
• What don’t you like about the email system?
• What would you like to do with the email system but can’t?
• What do you wish the new system would do better?

End

• Did we miss a topic? Do you want to add something?
Appendix B  Design Proposals

Appendix B.1  Add Case Period Fullscreen

Figure 15: Entire Screen when adding a new case period.
Source: Own illustration.
Appendix B.2  Detail Case View for Student Counselors

Figure 16: Screen of the Detail View of cases for the user group: Student Counselor.
Source: Own illustration.