

TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Informatics
Chair of Network Software

Angelina Kobeleva
103937IAPB

**User Centric Design and Usability Testing
of Study Information System Interface Prototype**

Bachelor's Thesis

Supervisor: Jaagup Irve

Tallinn
2014

Author's Declaration

I confirm that no part of this thesis has already been, or is being currently, submitted for a bachelor's degree or any similar qualification at any other university. The thesis is the outcome of my own work, except when explicit reference has been made to the work of others.

Angelina Kobeleva

.....

(Date)

.....

(Signature)

Autorideklaratsioon

Olen koostanud antud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud. Käesolevat tööd ei ole varem esitatud kaitsmisele kusagil mujal.

Angelina Kobeleva

.....

(Kuupäev)

.....

(Allkiri)

List of abbreviations

Abbreviation

TUT

SIS

UCD

Definition

Tallinn University of Technology

Study Information System

User-centered design

Abstract

User-centered design is becoming a common approach to building systems in the last years. This approach is focused on users' needs from the beginning, which helps to build a user-friendly system. User-centered design also suggests frequent usability testing rounds, and as a result major usability mistakes are found at earlier stages of project and fixing them costs less.

In this thesis I used user-centered design approach to create a new prototype of Study Information System used in Tallinn University of Technology. The current SIS has a lot of room for improvement and in this project I have tried to fix major usability problems students meet with when using SIS. As UCD suggests, I created personas to understand end users better, and created scenarios to describe their main goals. A prototype was then built using Twitter Bootstrap based on previous analysis. This prototype was tested by three usability test participants to find problems. The problems were fixed in the second prototype version.

Annotatsioon

Kasutajakeskne disain on viimastel aastatel veebilahenduste ehitamisel populaarne meetod. See meetod keskendub süsteemile kasutajate vajadustest lähtuvalt, mis aitab ehitada kasutajasõbralikke lahendusi. Kasutajakeskne disain kätkeb endas sagedaseid kasutatavuse teste (*usability tests*), mille tulemusena on suured kasutatavuse vead leitavad projekti varajastes staadiumites, muutes seega minimaalseks vastavate vigade eemaldamistest tingitud lisakulutused.

Käesolevas töös kasutasin kasutajakeskse disaini meetodit selleks, et ehitada Tallinna Tehnikaülikooli Õppeinfosüsteemile uus prototüüp. Praeguses süsteemis on palju probleeme ja käesolevas projektis proovisin parandada peamised kasutatavuse vead, millega tudengid õppeinfo süsteemi kasutamisel kokku on puutunud. Nagu kasutajakeskne disain ette näeb, lõin ma kasutajad (*personas*), et lõppkasutajate käitumisest paremini aru saada. Seejärel lõin kasutusjuhud (scenarios) kasutajate põhieesmärkide kirjeldamiseks. Prototüüp on ehitatud Twitter Bootstrapi abil käesoleva analüüsi põhjal. Prototüübi kasutatavust on testitud kolme inimese abil ning leitud probleemid on kõrvaldatud prototüübi teises versioonis.

Table of contents

| | |
|--|-----------|
| 1. INTRODUCTION | 8 |
| 2. SHORT INTRODUCTION TO USABILITY | 10 |
| 2.1 WHAT IS USABILITY?..... | 10 |
| 2.2 WHY IS USABILITY IMPORTANT? | 13 |
| 3. USER-CENTERED DESIGN AS AN APPROACH TO IMPROVING USABILITY | 15 |
| 3.1 WHAT IS USER-CENTERED DESIGN: A DIVE INTO HISTORY | 15 |
| 3.2 ISO 9241-200 AN INTERNATIONAL HUMAN-CENTERED DESIGN STANDARD | 18 |
| 3.3 THE GENERAL APPROACH OF USER-CENTERED DESIGN | 19 |
| 3.3.1 UCD, part one: learn about people | 19 |
| 3.3.2 UCD, part two: learn about users..... | 20 |
| 3.3.3 UCD, part three: work with users..... | 21 |
| 3.4 TOOLS OF USER-CENTERED DESIGN (UCD WORKFLOW) | 23 |
| 3.4.1 Personas..... | 24 |
| 3.4.2 Scenarios..... | 25 |
| 4. BASIC PRINCIPLES OF USABILITY TESTING | 28 |
| 4.1 METHOD OF USABILITY TESTING | 31 |
| 4.1.1 Choosing Test Participants..... | 31 |
| 4.1.2 Choosing Tasks and Creating Scenarios | 32 |
| 4.1.3 Usability Test Workflow..... | 33 |
| 4.1.4 Debriefing Meeting | 35 |
| 5. APPLYING UCD TO DESIGNING A NEW OIS PROTOTYPE..... | 37 |
| 5.1 RESTRUCTURING OF MENU | 37 |
| 5.2 CREATING PERSONAS | 40 |
| 5.3 CREATING SCENARIOS..... | 43 |
| 5.3.1 Daily-use scenarios..... | 44 |
| 5.3.2 Necessary-use scenarios | 45 |
| 5.4 CREATING PROTOTYPE USING TWITTER BOOTSTRAP..... | 51 |
| 5.4.1 Prototypes created with Twitter Bootstrap | 52 |
| 5.5 USABILITY TESTING OF THE CREATED PROTOTYPE..... | 61 |
| 5.5.1 Results of the Usability Test..... | 63 |
| 5.6 CREATING A REVISED VERSION OF PROTOTYPE..... | 64 |
| 6. CONCLUSION..... | 67 |
| RESÛMEE..... | 68 |
| LITERATURE CITED | 69 |

1. Introduction

This work is focused on usage of user-centered design and early-phase usability testing in the process of development a new Studies Information System interface prototype. This thesis also shows how useful is usability testing in early stages of development process, by allowing to find out and fix major usability problems as early as possible, when the cost of changes is relatively low.

Normally, if a website or a system is badly designed, there is always a possibility of leaving the page or finding a better analogue. In this particular case, however, students don't have any choice and are compelled to use a poorly designed system. This leads to frustration among students, when they can't efficiently perform key tasks related to their studies: submission of declaration, registration to exams, registration to lab works, looking through the timetables. Tutors have to come to first-year students and show them step-by-step, how to do those things. Students tend to remember a sequence of clicks needed to be done to achieve some goal, because the process of just figuring it out on spot is too complicated due to well-known Studies Information System major usability problems, which include illogical menu structure, incomprehensible icon usage, obscure reminders of unpaid invoices. Also the design of system scales poorly for handheld screens, which leads to problems when using the system on a mobile device.

I investigated how using user-centered design approach and a simple usability test can help to build a user-friendlier system prototype. Starting by creating personas, scenarios and user stories as a basis for further work, I move forward to building a prototype using Twitter Bootstrap, which from the beginning is based on the user needs found out in the first stage as any user-friendly system should be. Then I carry out usability tests of created prototype to find out major problems and to get some material for further improvement. The latest prototype fixes usability problems which usability test found.

Further enhancement can be made by testing the second prototype for usability and changing it according to the results. However, to create a drastically improved from usability point of view

mockups of Studies Information System a one round of usability testing and problem fixing is enough. This is because the initial prototype is already based on the needs of users, and the only usability test already reveals and allows to fix major usability problems.

The first version of prototype can be found here:

- <http://geekkery.com/diploma/firstyear/login-page.html> - version of a first-year student;
- <http://geekkery.com/diploma/regular/login-page.html> - version of a regular student.

Revised prototype can be found here:

- <http://geekkery.com/diploma/revised/firstyear/login-page.html> - version of a first-year student;
- <http://geekkery.com/diploma/revised/regular/login-page.html> - version of a regular student.

In order to login to any of the above fill in the login form fields with any text and click “Log in”.

2. Short Introduction to Usability

Before further analysis, it is necessary to define vocabulary we are going to work with.

2.1 What Is Usability?

The term ‘usability’ refers to multiple concepts taken together: user experience (which itself can include different kinds of experiences depending on the context), user satisfaction with the performance of the system, and learnability (i.e. ‘ease of learning how to use the system’) [1]. There are a lot of possible definitions of ‘usability’, as the desired meaning of this word is changed depending on the context it is used in. The homogeneity of the definition of this term is further smoothed because some people tend to use it as an attribute of quality of a system, and others think of usability as of a process, a set of techniques using during making of the project. The second definition, however, is more accurately described as user-centered design.

Let’s then stop on the definition provided in the ISO 9241-11 standard, is a multi-part standard from ISO covering ergonomics of human-computer interaction (which suits the topic of this thesis) and describes usability as a quality of a system:

"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" [2]

This definition includes four elements that are necessary to create a usable system [3]:

There are specified users of the system.

The users have a set of specified goals.

1. The system should allow user goals to be met (effectively) in an efficient manner and the users will be satisfied with the process or outcome.
2. The system will be used in a particular context (e.g. within a physical location, a business process).

It also includes a hint for five characteristics which must be met for the users of a product (also known as five E's of usability):

1. Effective
2. Efficient
3. Engaging
4. Error Tolerant
5. Easy to Learn

A usable system is one that is designed to consider all of these aspects. Again, in the first part about the interaction between users and the system we shall dive further in the section concerning user-centered design. Before we do that, we need to establish criteria by which we could say that users are using the system effectively, efficiently and with satisfaction.

Effectiveness in the context of usability is the completeness and accuracy with which users achieve specified goals. It is determined by looking at whether the user's goals were met successfully and whether all work is correct.

Efficiency is very close to effectiveness, but they are by no means the same, because efficiency is mostly concerned about the speed with which user can achieve the desired result. ISO 9241 defines efficiency as the total resources expended in a task.

The most obvious element of **engaging** is visual design, because this characteristic means that the interface of a system must be pleasant to the eye and satisfying to use: colors should match, all texts should be readable, information should be presented clearly.

Of course there is almost no way to build a complex system, which will have no errors at all (but this is always a goal, of course). An **error tolerant** system prevents errors arising from user's interaction with the system, and helps the user to recover from errors that occurred. This can be achieved through designing a system where it is very hard or impossible to take an incorrect or invalid action, and as much actions as possible are reversible.

Easy to learn systems are, well, easy to learn. User should not need to read a manual to start using the system. It should be easy for the user to start using new features of the system after they were added. The system must be designed in a predictable way. This means that terminology does not change, that design elements and controls are placed in familiar locations and that similar functions behave similarly.

Now that we have established clear criteria for rating the user-friendliness of a system, we can move further to discuss how to achieve them. But first an excerpt about why should we at all be concerned about usability problems.

2.2 Why Is Usability Important?

A good usability of a system is important for users, because it means they will achieve their goals without problems. Among a couple of almost the same systems users will always choose the one using which they will not feel any frustration, ie a more user-friendly system. If the system is ineffective for users - they leave. If it is inefficient - they leave. An error occurs every second step - they leave, the design features red text on green background - they leave, system requires a manual to be used - they leave.

Usability goals are business goals [4]. Badly-built websites are hard to use, which results in growing users' frustration [5]. Eventually the user just leaves, and every user that left means loss of profit for the business. In case if user cannot leave (for example, the system is a kind of intranet and she has no way to find an alternative) this leads to the loss of user productivity and growing rate of frustration.

It also notable that it takes less time to teach an employee how to use a usability-engineered system than a similar system, which had no usability work involved [4]. Under ideal circumstances a training won't be needed at all if the system is well-build.

In short, all the benefits of applying usability best practices to a product can be summarized as following:

1. Increased sales and customer satisfaction
2. Competitive advantage (if features are similar, users will choose the system which is easier to use)
3. Advertising benefits: companies can emphasize that they follow the guidelines of usability Standard and articulate differences in usability between theirs and competitors' products.
4. Reduced development and maintenance cost. Studying user needs allows to "design it right the first time" reducing the cost of development and maintenance. Usability engineering techniques (usability testing in particular) allows to find out major mistakes

in the design of product on early stage when fixing them is easier and cost of that fixing is much lower.

5. Reduced training costs. Even if training couldn't be completely eliminated, the time used to train new users is reduced greatly if the product itself is user-friendly and easy to use.
6. Lower support costs. Creating a product with user in mind reduces the risk that user will not understand the developer's intended way of operating the product.
7. Lower documentation costs. A well-designed product doesn't need much documentation.

One of my goals when creating a new Study Information System (SIS) prototype was to make SIS easier to use and through this decrease users' frustration. Currently tutors have to show to new students, how to use SIS and how to make simple tasks such as declaring subjects. Usually this takes about 15 minutes of some other lecture, because otherwise it is hard to gather all the students together. A user-friendlier SIS would remove the necessity of training and will make study process a bit more productive.

3. User-Centered Design as an Approach to Improving Usability

3.1 What Is User-Centered Design: a Dive into History

Accordingly to authors of “User-Centered Design” [6], user-centered design (UCD) is a design process in which end-users influence how a design takes shape. This means that users or at least user-centered activities are included throughout the design and development process in a form of creating personas, user stories, and scenarios, analyzing use cases, and doing usability tests. Again like usability, the term user-centered design combines in itself both philosophy of approach to design and methods to achieve the desired result. One of the sections of “IS Management Handbook, Seventh Edition” [7] suggests another good definition for UCD: a framework of principles, methodologies, and techniques that maximize the likelihood that users will find the application solution useful and effective.

A goal of a designer working accordingly UCD is to understand the user and create an easy-to-use product, which user will be able to use to achieve her goals. An interface built with UCD in mind shows all the necessary information in way that fit in with the goals, desires and needs of the users, which helps to improve performance of a human-machine system as a whole [8]. A result of user-centered approach is reduction of errors and improvement of productivity because users now spend less time frustrating and thinking what do they need to do to achieve some result. Along with this grow user satisfaction and acceptance of the system.

The term ‘user-centered design’ was firstly introduced by Donald Norman in the 1980’s. In his book entitled “User-Centered System Design: New Perspectives on Human-Computer Interaction” he writes: “user-centered design and community participation in design became major attributes of design process” [9].

D. Norman continues his research of UCD in his next book, “Design of Everyday Things” [10]. In this book he talks about UCD as about a “philosophy based on the needs and interests of the user, with an emphasis on making products usable and understandable”. He also states the main features of a good design:

1. Design should make it easy to understand which actions are possible at any given moment
2. Design should make conceptual model of the system visible, as well as alternative actions and results of actions
3. Design should make it easy to evaluate the current state of the system
4. Design should follow natural mappings between intentions and required actions, actions and results, and between visible information and current system state.

Simply stating what a design must be is not enough. To support his suggestion of a good design Norman provides also seven principles of “transforming difficult tasks into simple ones”, by following which a designer will be able to achieve the desired result (i.e. a usable and understandable product):

1. Use both knowledge of the world and knowledge in the head. The designer should build an appropriate for the user conceptual model, which user will be able to understand based either on externally or internally available knowledge; and those two kinds of knowledge should be easily combinable, and the user should be able to use whichever is more readily available at the moment without interfering with the other.
2. Simplify the structure of the tasks to minimize the amount of planning or problem solving they require. The designer should be aware of limitations of human short-term memory (which we use to store information that will be useful for us only for a short amount of time, for example, navigation items): a person can only remember up to five different unrelated items at one time. On the other hand, information stored in the long-term memory can be kept there for a long amount of time, but it is more easily acquired if it is structured and makes sense. Design should minimize the amount of interruptions the user might face, and it also should provide mental aids to help user recover the exact state of operations that were interrupted.
3. Make things visible: bridge the gulfs of Execution and Evaluation. Make things visible on execution side to let users know what can be done and how and what will be this action’s outcome; make things visible on evaluation side to let users see effects of their actions. Provide indications of current system state that are easily distinguishable and match intentions and expectations of the end user.

4. Get the mappings right. Make sure the user understands relationships between her intentions and required actions, her actions and results, between visible information and current system state, and between the system state and her intentions, needs and expectations. The easiest way to make things understandable is to use some sort of graphics, for example, well-known icons.
5. Exploit the power of constraints, both natural and artificial. Make the user feel that she has only one possible thing to do - and that thing must be the right one.
6. Design for error. Any error that can be made will be made. Plan for it, and give the user a way to recover from the error, understand what and why happened, and let him reverse any unwanted result. Make irreversible actions hard to do.
7. When all else fails, standardize. When it is impossible to design something without arbitrary mappings - create or follow an international standard. Any standard mechanism needs to be learned only once, after what it can be used by users effectively in any situation.

3.2 ISO 9241-200 An International Human-Centered Design Standard

Aside from books, user-centered design approach is described in ISO 9241-200 Standard, which came as a replacement for an old ISO 13407. This newer Standard provides “requirements and recommendations for human-centered design principles and activities throughout the life cycle of computer-based interactive systems” and covers both software and hardware [4]. This standard includes 6 design principles of human-centered design:

1. The design is based upon an explicit understanding of users, tasks and environments. Users may behave differently from how designers think they would; and the environment the product will be used in can have great impact on how the product should be designed.
2. Users are involved throughout design and development. The involvement can be different on each stage of design process, but the feedback from users is always important as it may provide new information and affect the shape of the product.
3. The design is driven and refined by user-centered evaluation. Product features should be affected mostly by users, and not by developers.
4. The process is iterative. “It is impossible to specify every detail of every aspect of the interaction in the beginning of development”, says the Standard. Iterations can occur on *micro-level* (when a particular feature is tested, modified, and evaluated) or on *macro-level* (when a system as a whole is tested, changed, and evaluated until it is deemed to be satisfactory).
5. The design addresses the whole user-experience. This principle is new, it was not included in the Standard’s previous version. It emphasizes the fact that usability has more aspects than just ease of use: for example, user’s satisfaction from job, elimination of monotony, user expectation.
6. The design team includes multidisciplinary skills and perspectives, ie all appropriate skills and perspectives are presented during the design process when their input and interactions are most effective.

3.3 The General Approach of User-Centered Design

As said before user-centered design (UCD) is not only a general philosophy about how design should be created, but also provides certain methods and techniques that help to achieve the desired result; however none of them are mandatory. Among those techniques are creating personas, scenarios, user stories and/or use cases, and constant usability tests.

As said before, the focus of UCD is user, her abilities and limitations, her desires and goals that she wants to achieve using the system. Therefore the main aspects of UCD are [2]:

1. learning about human abilities and limitations,
2. designing for a particular group of people,
3. working with these people throughout the design process.

3.3.1 UCD, part one: learn about people

As long as UCD is focused on the user, it is highly important to understand human abilities and limitations. What exact limitation should be taken into account depends on which kind of users will be using the system which is being in development. For example, while building a website for a school for children with disabilities it is important to take into account possible blindness or low motor skills. Some major characteristics are:

1. Memory. A human being is capable of easily remembering only about 5 unconnected things at a time, for everything more than that we need to use our long-term memory which is harder and more energy-consuming. Therefore arise certain limitations to the design of system: menus should not contain more than 5 items (an average amount of unconnected things that person can easily remember), password hints should be available, breadcrumbs should be present.
2. Errors. People make errors: they press wrong buttons, they write undesired data into forms, they click everywhere. System should be error-tolerant, which means that the majority of errors should not be doable, and the remaining part should be easily reversible.

3. Perception: sight, hearing, smell, taste, touch. The most important of the senses when developing a computer-based system is, of course, sight; because we use our eyes to scan the computer screen for information. The rules arising from this concern mainly visual or web design: the fonts should be readable, the page should be contrast enough, there should be harmony in the colors used.
4. Fine motor skills. Fitts's law predicts that time required to move to a target area is a function of the distance to the target and the size of the target. This means that clickable area should be as big as possible and located in a easy to access place (a good example of such an area is a corner of a computer screen: user does not need to aim there, she can just move mouse rapidly somewhere in that direction and cursor will get in place). The size of clickable area is even more important on touch interfaces, because people mostly use their fingers when using touch screen, and fingers are bigger than a mouse cursor.
5. Mental models. One way that people make sense of the world is to form a mental model of how things work. Designing a system in a common mental model's way is a great approach, because it means that the user will not need any exceptional training to start using the system - she already has a model of the system in his head.

3.3.2 UCD, part two: learn about users

Every system is being built with some certain target audience in mind. When doing user-centered design it is important to learn as much as possible about who your users are. It can be tricky sometimes (for example when possible audience is wide like an audience of a government website). In this case some analysis of statistics, search logs and referrer logs is required.

There are a lot of possibilities to learn more about the users once you know who they are. The main ones include:

1. One-on-one discussions of the system with one of its users. Such interviews help to identify the key needs. It is important to remember that interviewer should ask about the tasks of user as opposed to her needs.
2. Task analysis, during which goals of the tasks, the steps involved, inputs, outputs and information needs are identified.

3. Focus groups. Small groups (6-8 people) who are using or will be using the system. Discussions inside focus groups elicit attitudes, feelings and ideas about a particular topic.
4. Surveys and questionnaires. As opposed to focus groups which cover only a small amount of people, a survey or a questionnaire can be used to gather data from a large amount of people.

After some data from users has been gathered it is vital to communicate it through the whole team in a simple form so that everyone has something to rely on during her work. UCD suggests a couple of ways to arrange gathered information in easy-to-use form. Here I shall provide only a short overview of them since more information will follow in the next chapter:

1. Personas. Personas are fictitious characters created to generalize the needs of specific user groups.
2. Scenarios, or short stories describing how the system will be used by personas to complete their tasks. Scenarios are used also in usability testing.
3. Task lists of all the tasks users will want to do with the system.
4. User stories, each of which states one of goals user wants to achieve. I shall not use this in my practical work.
5. Use cases, or step-by-step descriptions of the process which leads the user to her result. I shall not use this in my practical work.

3.3.3 UCD, part three: work with users

As stated before user-centered design implies that users should be involved throughout the whole design process. In the early stages the techniques described above are the most appropriate, as they help to understand what the system needs to do.

However the further into the design process we go, the more possibilities to involve users into it we have. Of course, the most important one is *usability testing*. It is a good practice to start usability tests as soon as possible, preferably already on a paper (or early computer) prototype stage. This is because the earlier usability test is made, the earlier major usability problems will

be found, which makes it much easier and cheaper to fix them. Small but frequent usability tests provide more benefit than one large one [11].

Although usability testing is maybe the most popular way of involving users into design process in the later stages, it is by no means the only one. Another simple approach is to make regular *design walk-throughs* to collect feedback about the design from various users. This may be useful if the prototype is still too raw to do a proper usability test.

Card sorting technique will be especially helpful when building a website with a lot of information. Users are provided with a list of potential content written on different cards and asked to sort them in groups. Based on the results the structure of the website is born.

Having a long list of desired features or requirements may be frustrating. *Prioritization sessions* will help to choose the most vital of them. During those sessions users are provided with lists of new features, functionality or content and are asked to sort them in some way, which can be according to usefulness, how often they would use the function or how quickly they need to access it.

3.4 Tools of User-Centered Design (UCD Workflow)

As said before, user-centered design combines in itself both philosophy of approach to design and techniques helping to achieve the desired result. All of those tools are simple in concept, but extremely helpful when it comes to the process. Common UCD tools and techniques include [12]:

1. Personas
2. Scenarios
3. User stories and use cases
4. Prototyping and testing as much as possible.

Personas, scenarios, and user stories will be described in greater detail further. Usability testing, however, is so important that it will be described in the next chapter.

3.4.1 Personas

The concept of a *persona* was created by Alan Cooper and popularized via his book “The Inmates Are Running the Asylum” [13]. The underlying idea is very simple: to create a design that user will find comfortable to use, we need to know what and how she wishes to achieve; the simplest way would be just to ask a real user, but it does not work for a couple of reasons. So, let’s just make up some fictitious users and design for them. Personas are fictitious, specific, concrete descriptions of target users (or user groups) who are going to use your service. They are not real people, but representations of hypothetical archetypes of users.

Using of personas has many benefits [14]:

1. Personas are valuable communication tool as they make assumptions and knowledge about users specific, allowing to talk about users meaningfully avoiding abstract “the user”.
2. Personas provide a set of specific users to focus on, preventing the team from trying to design for every possible user, which leads to better design decisions.
3. Personas feel like real users to the team, and help team members to feel empathy towards users.

There is no single guideline how to create a persona and what information should be present, however there are several things that bring sense to creating a persona and therefore should always be present:

1. Name and age, maybe even a picture to make persona more humane. Those details are important to produce a concrete individual in someone’s mind.
2. Persona’s background: who is she, how is she related to the website.
3. A brief story of the persona.
4. Persona’s goals, which is the most important part. Persona exists to achieve her goals, and goals give meaning to a persona. Goals are what would she like to achieve using the system which is being designed, ie a reason why she performs specific tasks. Those goals should not contain any information about *how exactly* is she going to achieve this; ie no design decisions.

When creating personas it is important to remember that the more specific personas are, the more effective design tools they will be. Until the persona is fully described there is always danger that someone of the team will imagine himself on spot of the persona, which may lead to incorrect design decisions. Personas must be precise, not accurate: in other words, it is more important to describe the persona in as much detail as possible, than to create a persona that will be one hundred percent correct one.

Personas can be combined into *casts*. Casts are created for each project and can contain 3-12 personas. It does not mean, that team is designing for all of them; sometimes a persona is a description of a user the team is definitely *not* designing for. However, it is always useful to remember that user population is great. To avoid spreading among that much personas, every cast should have a *primary persona*, who will be the main focus of the design. If it is hard to determine a single primary persona, it means that something went wrong: the problem set is too large, and too much is trying to be accomplished at one time. This is the right time to come back a little and rethink the main idea of a system or service in design.

3.4.2 Scenarios

A scenario concisely describes how the product or system would be used by personas in real life; in other words, it describes how personas (and through them real people they represent) interact with the product. Scenarios are constructed upon the information gathered during investigation phase when user observations and interviews occur. When writing a scenario it is more important to describe an action from start to finish than to describe it in great detail; also it is vital not to deepen into edge cases [14].

Generally all scenarios include goals, motivations, actions, and reactions. Even the longest of the scenarios have to be short, but at the same time they have to answer to the following questions:

1. Who is the user?
2. Why the user is coming to the website/using the system?
3. What is the user's goal?

There are three big groups of scenarios: daily-use scenarios, necessary-use scenarios, and edge-case scenarios. The most useful and important are, of course, daily-use scenarios, which describe the main actions the user will perform with the greatest frequency. Usually the number of daily-use scenarios is 2-3. All the actions that must be performed sometimes, but not frequently, are included into necessary-use scenarios. Every product will have its own number of necessary-use scenarios, but this number will generally be bigger than the number of daily-use scenarios. All the actions that are neither necessary nor frequent are combined in edge-case scenarios. Those tasks do not require careful design, but at the same time cannot be completely omitted from the software.

There are also a couple of types of scenarios, which are divided into four levels: brief scenarios, vignettes, elaborated scenarios, and complete task scenarios; starting from a brief description of a user's goal the product must handle and finishing with the specific stories of a user trying to reach her goal with details of tasks and interactions. The next diagram shows the main types of scenarios:

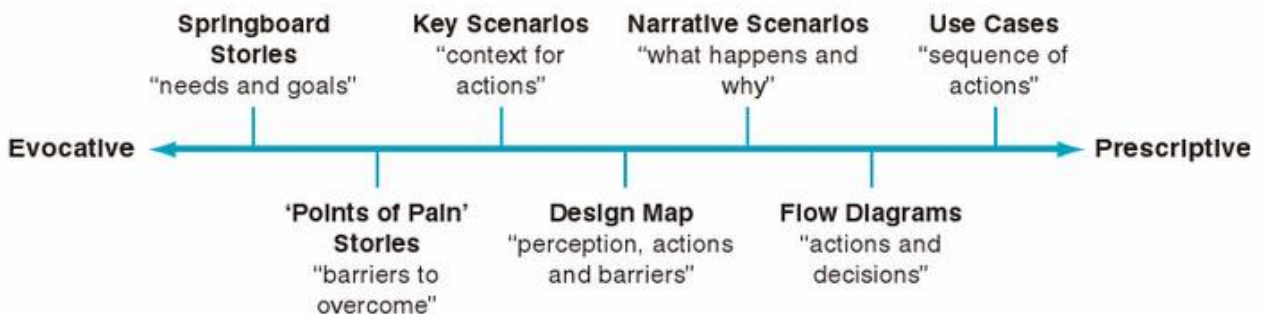


Figure 1: types of scenarios, taken from [14], page 537

Springboard stories are the shortest of all, illustrating a dilemma user is facing and a way out in a couple of sentences.

A *'points of pain' story* creates a vivid view of the problem from a point of view of a persona. The 'pain' here stands for gaps in the process of reaching the goal, usability problems, or new needs.

A *key scenarios* are, as the name implies, already closer to scenarios than to simple stories. They are more like concrete illustrations of how personas interact with the product. This description goes further in a *narrative scenario*, which explores how a given task is designed and implemented from beginning to end in a narrative format. On this stage it is important to carry the context and other environmental information along to make sure that proposed design or use case actually makes sense from the point of view of a persona.

To deepen into details even more *design maps* and *flow diagrams* are created. They are quite similar to use cases, only focusing on human side, not on how the system functions. Both design maps and flow diagrams help to explore the details of a process or a scenario, and walk through the logic of a process.

After main scenarios are examined and discussed, it is time to create *use cases* - sequences of actions of parts of the system (which can include people, processes, or even machines) that interact while using the system, stating what happens during a transaction. Use cases serve as a bridge between scenarios and implementation details.

4. Basic Principles of Usability Testing

Usability testing is a technique used in UCD to evaluate a product by testing it on users, or, in other words, to measure a product's capacity to meet its intended purpose. A little bit more verbose definition provides Steve Krug in his book "Don't Make Me Think" [15]:

“Watching people try to use what you're creating/designing/building (or something you've already created/designed/built), with the intention of (a) making it easier for people to use or (b) proving that it is easy to use.”

The aspect that during usability testing people are actually using the product is what makes it an irreplaceable usability practice. As opposed to interviews, questionnaires, and surveys, when people can only give their opinion about things or their past experiences about existing things, usability test gives a direct input on how real users are using the system.

It is important to start doing usability tests as soon as possible, because the longer is the delay, the more will be the cost of making changes. Some major problems that were detected too late cannot be corrected at all (because fixing them basically means rewriting the whole system from scratch).

There are four types of usability testing: exploratory, assessment, validation, and comparison tests [16]. The first three types have a place in the lifecycle of the product, when they will be the most useful; the comparison test can be used as an integral part of any of the first three types of testing. The difference between types of tests is in its emphasis on qualitative (ie *improving something* by finding things to make better) vs quantitative (ie *proving something* by *measuring things*) measures, and in the amount of interaction between the test operator and participant. The following diagram shows the exploratory, assessment, and validation tests next to their approximate place in the software lifecycle.

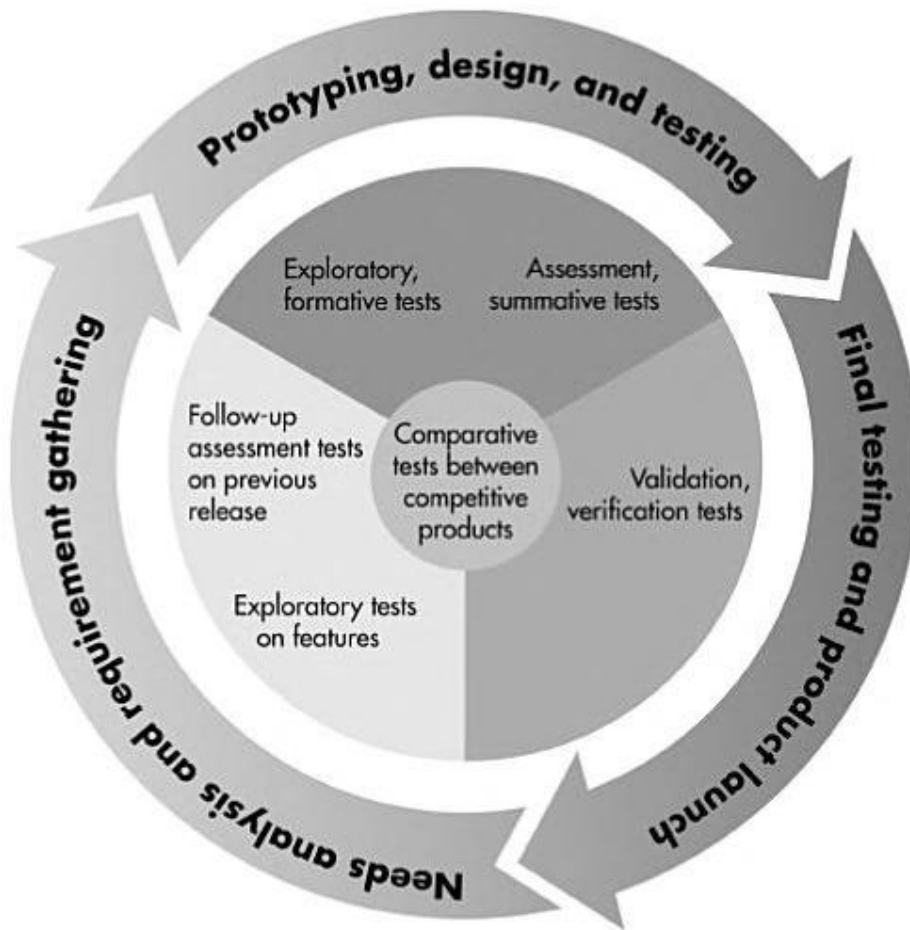


Figure 2: different kinds of usability tests, taken from [16]

Exploratory usability test is most useful on the early stages of product development when it is still in process of being defined and designed. The main objective of this test is to examine the effectiveness of the preliminary design concepts: does the interface support users' tasks within a goal, does it allow users to navigate inside the screen and between the screens of application, does the basic functionality have value for user, etc. This early analysis is highly important because this is when critical design decisions are made that lay the foundation for everything else to lay upon. Wrong assumptions and faulty premises about the user lead to great usability problems in the future. Exploratory tests are often conducted as comparison tests, when a couple of prototypes are being tested on different participants to choose the one which worked the best.

Assessment usability test is probably the most typical and simple for a usability novice type of test to conduct. Usually they are held somewhere between the early and middle stages of the

development cycle, after the fundamental design of the product has been established. The goal of assessment test is to expand the findings of exploratory test by evaluating the usability of lower-level operations within the product. In other words, assessment test helps to evaluate how effectively the concept of the product was implemented. This type of test is focused on how efficient can user perform various tasks by using the software. Again, elements of comparison test can be used here to test the effectiveness of a single element (for example, are textual buttons or pictograms better).

Validation or verification test is conducted in late stages of the development cycle when all the major work is already done. The goal of this type of test is to make sure that discovered in the previous tests problems are solved without introducing new ones. Validation test evaluates how well the product meets the usability standard (which comes from earlier usability tests, user surveys, interviews, etc), and is concerned about how efficiently and effectively can user achieve her goal. Another use of validation testing is to make sure that all components of the product (i.e., the software itself, documentation, hardware) work together well. As a part of verification test, a comparison test can be conducted to check how the product looks compared to competitor's product.

In my work I used assessment testing after I have done the first iteration of new OIS prototypes. This helped me to find the major problems so I was able fix them in the second iteration. My usability testing was in a great degree based on Steve's Krug book 'Rocket Surgery Made Easy', where author provided a detailed description of a simple do-it-yourself usability test; however, it was also enhanced with the techniques of other sources. Description of the used method of testing is provided in the next section.

4.1 Method of Usability Testing

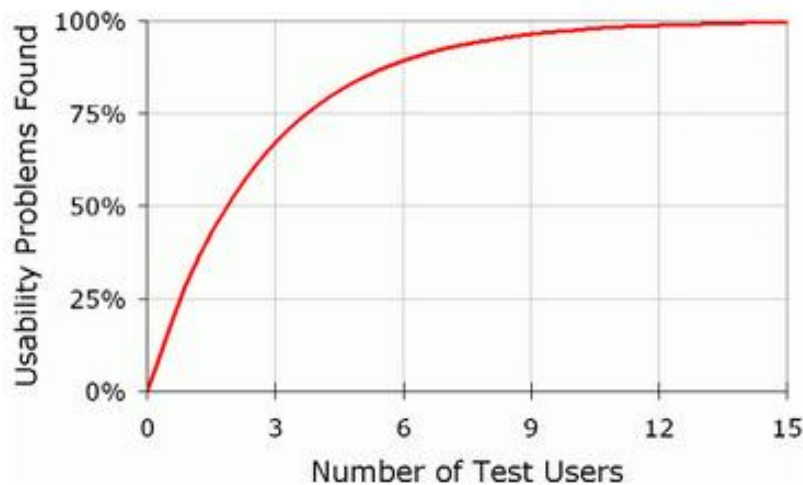
Setting up a usability test involves choosing test participants, choosing the tasks to test and expanding them into scenarios, and also some backstage work of setting up testing environment like installing and testing screen recorders and other necessary software. After the testing was done, a debriefing meeting is run to discuss detected problems, to choose the most serious among them and decide how to fix them.

4.1.1 Choosing Test Participants

Jakob Nielsen popularized the concept of doing small usability tests with the amount of test participants of 5 people at various stages of the development process iteratively. He explained this by saying that if you find two or three people struggling to complete some task, then it is already clear that you have problem in that place, and there is no need to look how many more people will suffer through the same problem. J. Nielsen suggested a mathematical model [17], which shows that the number of usability problems found in a usability test with n users is

$$U = 1 - (1 - p)^n,$$

where N is the total number of usability problems in the design, and L is the proportion of usability problems discovered while testing a single user. He also suggests that the value of L is 31%. This formula results in the following curve [17] (image taken from [17]):



Steve Krug follows the path of J. Nielsen, but reduces the recommended number of test participants to 3 people and gives a number of reasons for doing so: the first three users are very likely to encounter many of the most significant problems, three people is easier to find than five or more, testing with three users does not take a lot of time making it possible to do debriefing in the same day [11].

After deciding on the number of test participants it is necessary to define who those people should be. S. Krug suggests to recruit people who are actually using the system/software/website, or are going to use it in future. The reasons are simple:

1. There is no need to care about would people who are not going to use the product be able to use it or not - they are not going to use it anyway;
2. It is more likely that representative users will encounter the same problems during the testing as the real users;
3. People who are not the target audience can experience problems that actual users would not encounter;
4. Some kind of domain knowledge may be needed for successful use of the product, and non-representative users may lack that knowledge.

4.1.2 Choosing Tasks and Creating Scenarios

The first step is to write down a list of the most important tasks that people need to be able to do using the system. On this stage it is important to remember, that those tasks must reflect representative users' goals, not your idea about what they want to do. A task, as well as arising from it scenario, should be real and meaningful [18].

There can be a number of possible tasks to complete, and before the usability testing it is needed to choose a couple of most critical of them (for example, the ones that users absolutely must be able to accomplish or the ones about which you have based on some kind of research information that people are experiencing problems with them) for actual testing. It can be a one long task, or 10 smaller ones. Some tasks may be really simple, and it is reasonable not to use them in the test,

as too simple questions for too complex systems may provide misleading usability evaluation results.

As an alternative to simple making tasks for test participants to accomplish it has been suggested that a potential user brings her own task to the test session. This way user's perspective is also taken into account, as opposed to evaluating usability only from point of view of the system creators. K. Tarkkanen and his colleagues describe in their article "Back to User-Centered Usability Testing" [18] a modified version of user-defined task approach, which they have developed "in order to ensure that users' needs have better possibility to emerge in a usability test". Accordingly to their proposal, test participants are given an open-ended task that covers not a specific function, but the work process as a whole. This way users can follow their usual workflow during the test, which gives an idea about how they conceptualize their work and what kind of information and functions they really need.

In case if not using user-defined task approach, after a list of tasks that need usability testing was compiled, those tasks are expanded into scenarios. Accordingly to Krug [11], the scenario gives test participant a character, her motivation, what she needs to do, and a few necessary details (information, that user needs to know, but does not due to it is testing; for example, username and password). The most important thing here is not to give any clues about how to perform the task in the scenario. Some restrictions can also be put in the scenario, for example, "Do not use search" or "Stay on this site".

After scenarios are ready, a quick pilot test of them is required to make sure that they do not contain any disambiguations and uncertain points. There is no need to engage real users here, as it is not a product test, but a scenario test - anyone will work fine.

4.1.3 Usability Test Workflow

A quiet room with a desk and two chairs is required to conduct a usability test. In this room there should be a computer with Internet access, installed screen recording software, and if necessary

screen sharing software (to allow people in the observation room to see and to hear the test). It is also better to use a mouse during the test, as touchpads or trackballs can be difficult to use for some people.

Before test session starts, facilitator's task is to check that everything works fine and make some adjustments: test a screen recorder and screen sharing, make mouse cursor a little bit larger so that it would be easier to see, turn off any software that can disrupt the flow of the test, bookmark all the pages that are needed during the test to avoid typing URLs, and reset everything (ie load a clean set of test data if necessary, and delete browser history so that visited links would not be clues for test participants).

Test sessions begins with explaining to the participant how the test is going to work. In my work I used an example script by S. Krug, which can be found in appendix 1. The explanation should be clear in all senses: it should not arise much questions, and it should be done in a clear voice, without mumbling.

If the participant does not have any questions (or after those questions were answered), it is time to ask them a couple of questions. This part is intended to get the participants comfortable talking and show them that you are listening to what they are saying. Of course, questions also give an idea about who participant is, how comfortable she is with the computer, and whether or not she has some domain knowledge - in other words, all the information necessary to decide how this person compares to the target audience.

After that the real test starts. Usually participant is asked to look at the home page and tell what they can make of it. The goal is to see if users can figure out what the thing they are going to use is. After this 'tour', one of the tasks is read to the user (and a printed copy of a task is given to her). While participant is doing the task, she can be interrupted only if necessary. User is supposed to think aloud while doing the task. When the task is ready, it is time to move to the next one.

When all of the planned tasks are done, there is some probing time. During this time the participant can be asked about whether she had noticed some thing or not, why she made certain

choice, does she have any suggestions, or ask her to make a task in other way or from a different starting point.

After this stage usability testing session ends with some polite closing remarks. Before starting the next session, it is necessary to make some notes about the previous one, and to reset the computer to its previous clear state. Some adjustments can be made now, also. For example, if one participant was not able to complete a task and the reason is obvious, this task can be changed now or even excluded from further testing process. S. Krug suggests, that after each session three most serious usability problems noticed during this session should be written down.

4.1.4 Debriefing Meeting

The purpose of this meeting is:

1. To compose a list of most serious usability problems that participants encountered during the test;
2. To compose a list of the problems that are going to be fixed until the next round of testing.

It is best to conduct the debriefing just after the testing is over, so that everything would be clear in people's minds.

A list of usability problems can be done by concatenating smaller lists that each of the observers and the facilitator composed earlier during the testing sessions. The most serious problems can be later extracted by answering following questions to determine their severity: will a lot of people experience this problem? and will it cause a serious problem for people who experience it? If the answer to both questions is "yes" - then by no means this problem is serious and should be fixed in the first place. The toughest decisions involve corner cases, ie when a problem affects badly a few users or when a problem affects a lot of people but in fact is just some annoyance.

S. Krug suggests to write down ten major problems, and then during the discussion arrange them with a priority from 1 to 10. Also team should decide how to fix every of this problems as easy as possible.

After the debriefing, facilitator's job is to compose a short report to summarize this round of usability testing. The report should not be long, and should contain the following information:

- What was tested;
- The list of tasks the participant were asked to do;
- The list of problems to fix until the next round of testing;
- Some information about how people can watch the recordings of testing process and when the next round of testing will be.

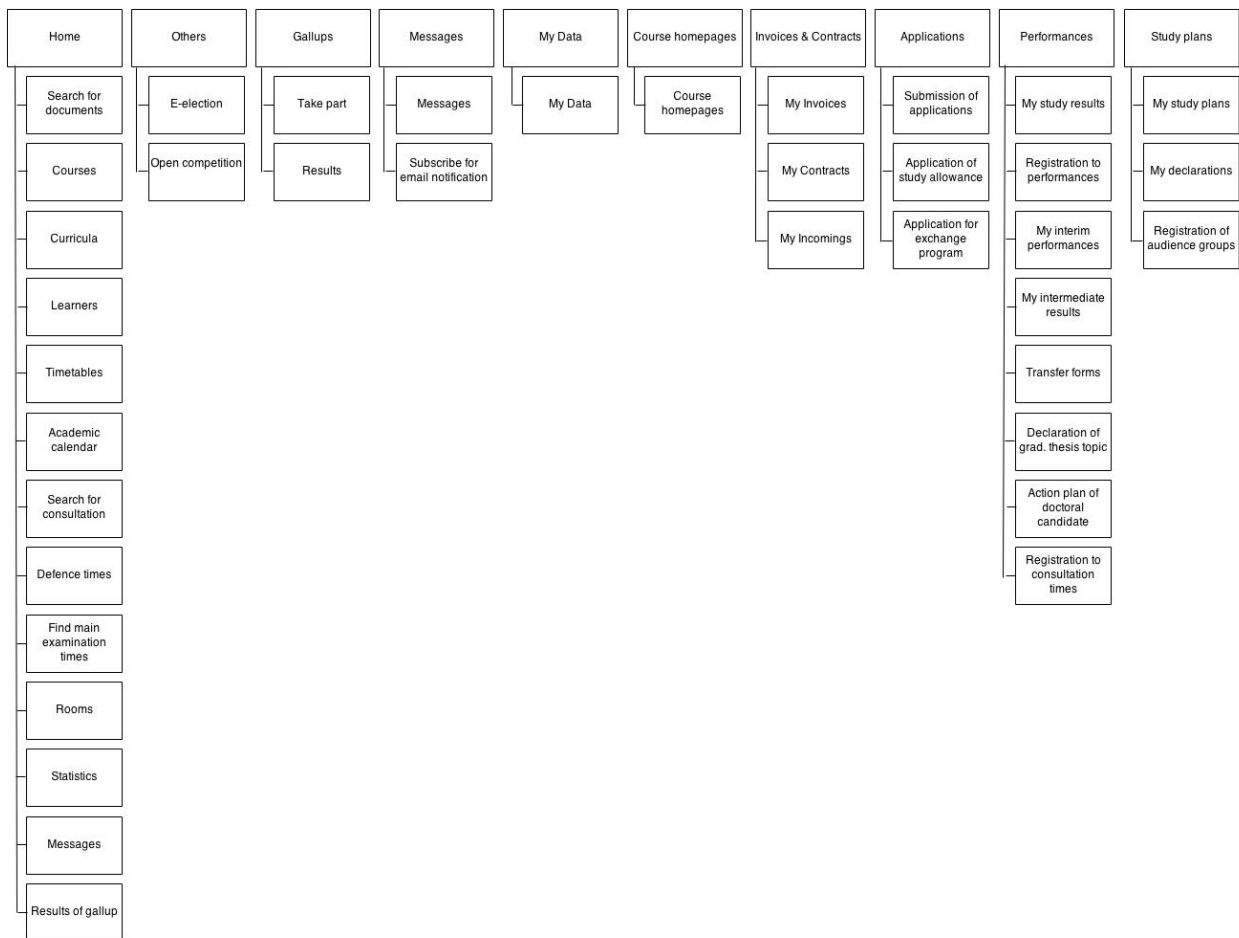
After the meeting is over, it is time to fix the problems. The fixes should be as simple as possible. Tweaking is preferred to redesigning things, as it costs less, requires less work, and can be made sooner. It easy to 'play' with tweaks: make one, observe whether it solved the problem or not, if yes - everything is great, if no - it is possible just to try another tweak. Another point Krug makes here is that a lot of people feel need to *add* something when there is a usability problem; for example, add more instructions. But often the best way to fix a problem is to *take something away*, clear the page from all visual noise that the user does not actually need but that is still there because of some abstract reasons.

5. Applying UCD to designing a new OIS prototype

As a practical part of this thesis I applied principles of user-centered design to creating a new, refined and more user-friendly, version of OIS.

5.1 Restructuring of Menu

Before creating personas I have decided to restructure the menu. Illogical menu structure was one of the most common complaints I have heard about OIS. In order to do this I started with sketching current menu structure:



It is clear that current menu structure has following problems:

1. The names of the sections are too abstract;
2. Submenus are too long and contain items of different sorts;
3. Titles of sections are all in different format;
4. There are unnecessary submenus (like 'My Data' dropdown of 'My Data' menu item);
5. The most used items are hidden in the menu structure.

As one of the most important tasks users do via OIS is submitting different applications, I started with this menu item and its submenu. A lot of applications were hidden under abstract "Submission of applications" menu item, so I put them in the submenu so that they would be clearly visible.

Another important thing is student's profile. In the current menu structure of OIS there is no way to get all the information about one student in one place, because personal data is in "My Data" menu, "My study results" - in "Performances", "My Study Plans" - in "Study plans", etc. I collected here all the data that concerns this particular student.

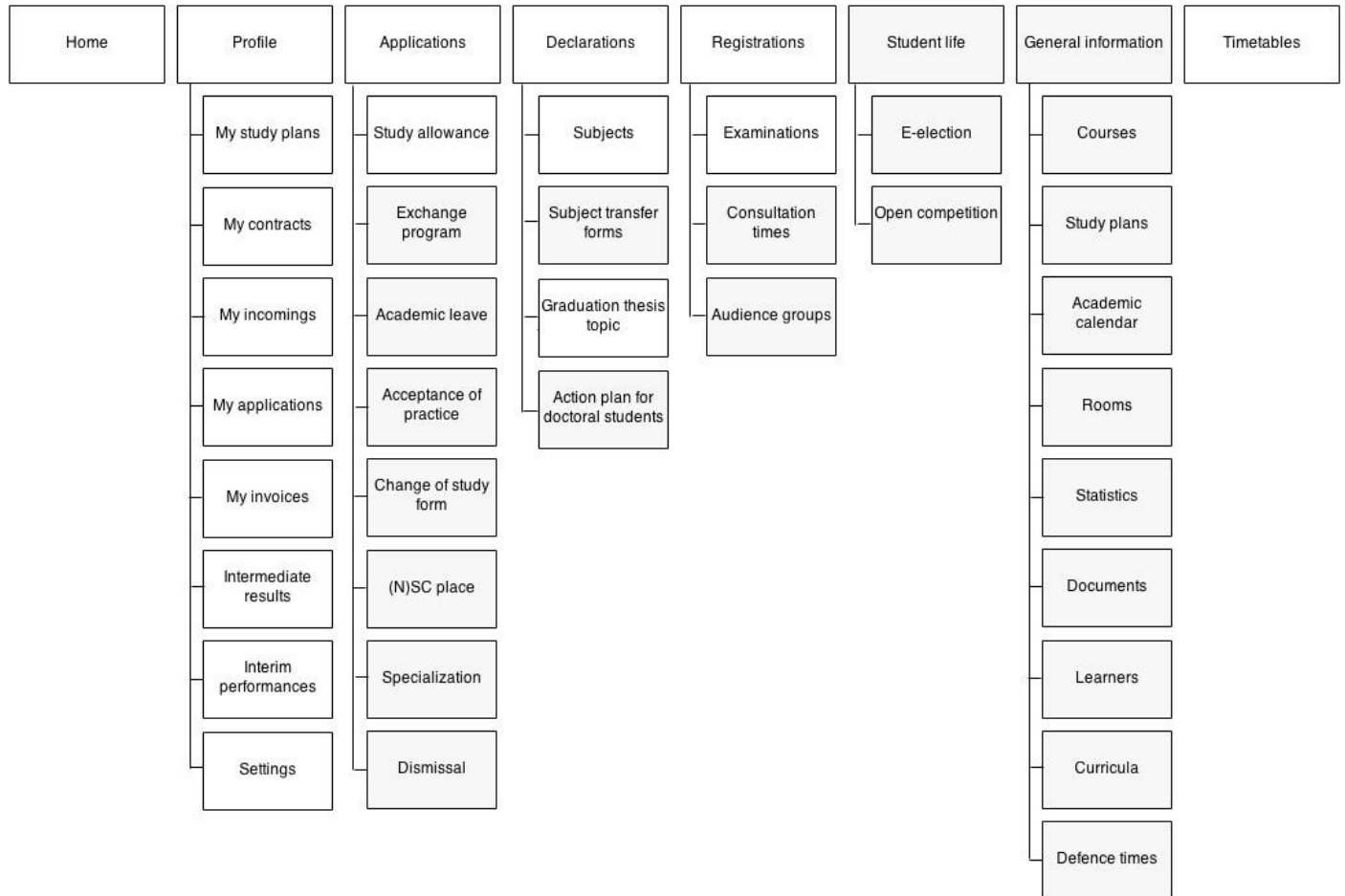
I also collected all declarations in one place. This includes declaration of graduation thesis topic, declaration of subjects in the beginning of new semester, and action plans for doctoral candidates.

As one of the most frequent actions on OIS is browsing of timetables, I decided to bring "Timetables" as a separate item in the menu to make it faster to find.

In my menu structure version there is "General information" submenu, where I put the majority of items from previous "Home"; and "Student life" submenu instead of abstract "Others".

Another important thing for making OIS faster to use is introduction of a dashboard on the home page. Student can see the most recent information about her study process: latest grades and reminders about deadlines of all sorts, as well as important news. Student can choose what kind of reminders to show in "Settings" section of his profile. He can also turn on e-mail notifications there, so that all reminders about deadlines would come also to his inbox.

Further is my version of restructured OIS menu. As OIS is a big and complicated system, I did not make prototypes of the whole system, as it would take much more time and effort and could not be done on a decent level without a team. The parts of OIS that I have not prototyped are shown in light grey color.



5.2 Creating personas

The following qualities were chosen to be present in the descriptions of personas:

1. Name, age, and picture;
2. A brief story telling who this persona is and how is she related to OIS;
3. Persona's goals.

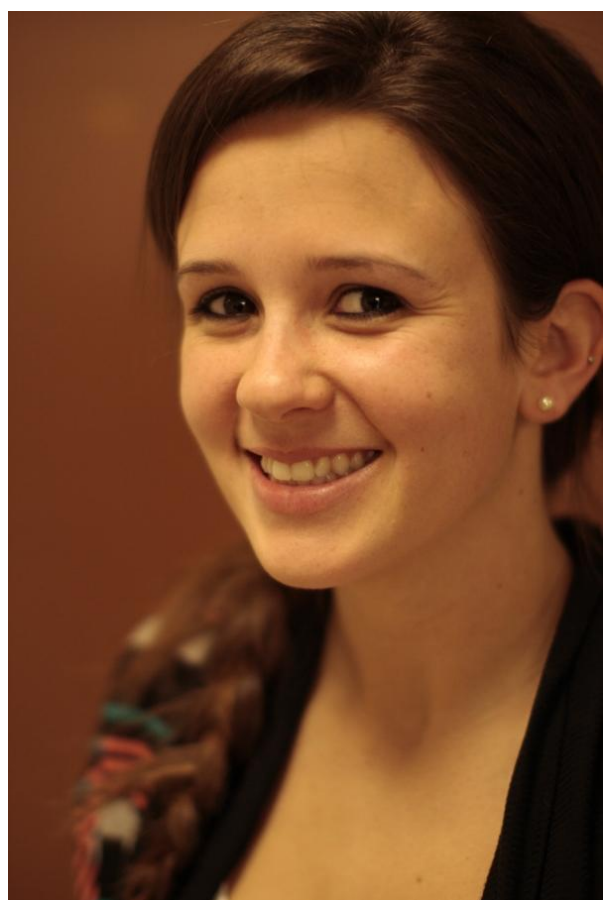
A total number of personas was chosen to be three to avoid spreading and getting lost among a huge cast. In addition, three personas represent four major types of users who are using OIS: freshmen, students on their last year and regular students who are not finishing their degree but already have some experience in academic life. The data for creating personas is based on personal experience of using OIS and such experiences of classmates. Information about each persona is represented as a table for a better clarity.

Name: Agnes Kowalski

Age: 19

Story: Agnes is 19. Although her surname is Polish, she was born in Tallinn and lives here since then. She just entered the University of Technology, and is going to study Informatics. She is really good with computers, and has been using them since age 10. Agnes always has her smartphone with her to ensure that she would not miss any information. However, she is nervous because it is her first year in the university and she is afraid she can miss something important because she might be not yet aware of all the inner rules.

Goals: Agnes wants to find out major information about her studying: the group she belongs to and the corresponding timetables. She also wants to choose subjects to study in this semester and declare them as soon as possible, because she heard that OIS may work badly in the last days before deadline for



submitting declarations. Agnes also wants to know whether she can receive any money allowances and if yes, then apply for them.

Name: Ivonne Tamm

Age: 22

Story: Ivonne has been studying Informatics in Tallinn University of Technology for three years. She was not that good with computers in the beginning, but she quickly got educated and found a good job as a programmer on her third semester in the university. After that she gets more knowledge at work from her colleagues than at lectures. This led to lack of interest in formal studying and drop of academic performance. Ivonne even had to take one of the subjects the second time because she was too lazy and keen on her work to pass it the first time. Ivonne want to get out of the university as soon as possible, preferably with a diploma. She does not visit lectures anymore, and does not care about her grades as long as her average is above 2.0.

Goals: Ivonne almost finished her thesis, so she wants to know whether she has any debts left. She knows that she has to pay for the subject she studied twice, but is not sure how and how much, so she wants to find this out. Ivonne also does not want to miss the deadline for submitting the thesis.



Name: Ivan Vassiljev

Age: 26

Story: Ivan was always interested in chemistry, so he had no doubts when choosing a faculty. He always was a great student, and finished his bachelor studies with *cum laude*. After getting his Master degree, Ivan went on and now he is a doctoral student. He is already familiar with university rules and deadlines; however he is a little scattered and can easily forget to do something which has to be done rarely. He has a laptop, but he is not an advanced computer user: he knows how to use some field-specific software and how to open Facebook.

Goals: Ivan cares a lot about his grades, and wants to trace his study results easily and on time so that he could have a chance to rewrite a work and get a better grade. He also interacts closely with his professors, so he wants to easily find out where to find them in non-lecture times.



5.3 Creating scenarios

While writing this thesis my main attention was focused on daily-use scenarios, as those kind of actions are the most common among OIS users and cause the majority of problems. I also decided to use key scenarios as a basis, and extend them if to narrative scenarios in necessary. I also omitted the step of creating use case diagrams, as they are closer to technical implementation than to mock up design and would not provide any new information for purposes.

I have firstly extracted actions the users perform in OIS. They all come from personas' goals:

1. Checking out own timetables to find out which lectures, when and where to attend;
2. Checking out timetables of professors to find them in non-lecture time;
3. Declaring subjects in the beginning of semester;
4. Submitting application for monthly allowances;
5. Finding out information about payments for repeatedly studied subjects;
6. Registering to examinations;
7. Submitting thesis theme;
8. Checking out grades and other study results.

After that I have chosen the most frequent actions OIS is used to perform. Those actions appear in the most personas' goals and shall be basis for daily-use scenarios:

1. Checking out own timetables to find out which lectures, when and where to attend;
2. Checking out timetables of professors to find them in non-lecture time;
3. Checking out grades and other study results.

Following actions form a base for necessary-use scenarios:

1. Submitting application for monthly allowances or applying for a stipend;
2. Declaring subjects in the beginning of semester;
3. Building a study-plan (which is a version of the previous scenario when a student wants to change the default set of subjects to declare);
4. Finding out information about payments for repeatedly studied subjects;
5. Registering for the exams;

6. Submitting thesis theme.

5.3.1 Daily-use scenarios

Checking out own timetables to find out which lectures, when and where to attend.

Agnes goes to a small room in the X building and gets her login information. On her way back she decides to check her timetable to visit the library and borrow necessary textbooks in advance. Agnes uses her smartphone to log in to OIS in browser. She has never seen the system before, but her experience suggests to tap on her name in the top right corner. And really, this navigates her to her profile page, where she checks out her group number.

After that Agnes clicks on the ‘Timetables’ link in the navigation, and sees a list of groups divided by faculties and form of studies. She clicks through the navigation to her own group, and her timetable shows up. She spots “Save timetable as PDF” button in the corner, and decides to go for it in case if she will not have Internet access. Agnes taps the button, and a .pdf is downloaded to her smartphone.

Checking out timetables of professors to find them in non-lecture time.

Ivan is at home working on a paper for a scientific journal. He has spent a couple of hours on it, and now he feels like he needs some help from someone with more experience. Ivan decides to contact one of his professors. As he lives near the university, Ivan decides to meet him after one of his lectures.

Ivan opens a new tab in browser, navigates to OIS page, and uses his username and password to log in. He clicks on ‘Timetables’ link in navigation menu. By default students’ timetables are shown, but he changes the tab to timetables of professors. He sees an alphabetically ordered list of lecturers with a search textbox upon it. Ivan starts typing the name of the professor he wants to meet in the textbox, and the list of the

lecturers gradually decreases. After Ivan spots his professor's name in the list, he clicks on it. Professor's timetable shows up.

Checking out grades and other study results.

Ivan had an exam a couple of days ago, and now he wants to check whether or not the exam is already graded. He opens a new tab in browser, navigates to OIS page, and uses his username and password to log in. On the frontpage he can see a couple of lists with recent grades, deadlines, and news.

a) At a glance Ivan can see that there are no recent grades.

b) At a glance Ivan sees his latest grade.

1) Ivan is satisfied with this grade, so he leaves the page happily.

2) Ivan does not like the grade and wants to improve it. He clicks on the name of the subject in the first column, and additional information from the professors slides down. He carefully reads the text, and finds out that he can have another try in three days. Ivan closes the page and opens the textbook.

5.3.2 Necessary-use scenarios

Submitting application for monthly allowances or applying for a stipend.

Agnes has heard that she can apply for a monthly allowance via OIS. She uses her computer's browser to navigate to OIS and to log in. The first page she sees after logging in contains a couple of lists on it: with recent grades (which is currently empty as she had no exams lately), news, and deadlines. The column with deadlines has two reminders: one about declaring subjects for the following semester, and one about applying for a study allowance.

Agnes clicks on the second reminder, and it brings her to "Application for study allowance" page. The page is divided in two columns, the left one describing the process of getting an allowance for regular students who already have some grades, and the right

one describing process of applying for special allowance for students from poor family. Both columns have “Apply now” buttons in the bottom, but the left one’s button is inactive and has an explanation next to it, why: because Agnes does not yet have any study experience in university.

Agnes decides to try to get special allowance. She clicks on “Apply now” link in the right column, and a popup window with a small form shows up. The form contains a field with students name (which is already filled in), student’s bank account number (which is also already filled in), and an optional file input field, where Agnes can upload some documents confirming that her family budget is relatively low. Agnes does not have such documents now, but as this field is optional, she decides to go for it anyway and clicks on “Submit application” button. The popup closes, and Agnes can now see that the button she clicked on previously is inactive, and has a green success message next to it, saying that she had already submitted an application for this kind of allowance.

When Agnes comes to the frontpage the next time, she sees that a reminder to apply for an allowance is gone from the deadlines section.

Declaring typical subjects in the beginning of semester.

On one of the intro lectures Agnes heard that the deadline for choosing subjects for this semester is in four weeks. She also heard that as she is a first-year student she can not yet choose which subjects to study, so she decides just to declare the typical studying program and not to think about it anymore.

Agnes uses her favorite browser to navigate to OIS page, and she logs in. As usual, she sees the frontpage with grades, deadlines and news shown on it. Agnes clicks on “Submit your declaration” link in the deadlines section.

This leads her to one of the “Declarations” subpages - “Declare subjects for this semester”. The page is divided in two columns, each containing only one of two button-style links with brief descriptions: “Declare typical program for this semester” and “Declare a revised list of subjects”. The second link is inactive, and a text next to it says

that it is due to the fact that Agnes is a freshman and can not choose subjects to study yet. Agnes clicks on the first link, and a this opens a popup where she can see a table with default for her semester and specialty subjects list. The table has 5 columns: subject code, title, exam form, number of ECTS credits, and a delete button (which is inactive).

Agnes clicks on the “Submit declaration” button below the table. “Your declaration has been successfully submitted” text shows next to it, and the button becomes inactive. There also is a “Withdraw” link if Agnes for some reasons wants to cancel her declaration.

When Agnes comes to the frontpage the next time, she sees that a reminder to declare subjects is gone from the deadlines section.

Building a custom study-plan.

One of the friends reminded to Ivan that he has to declare some subjects for the following semester. Ivan goes to OIS page, logs in, and clicks on “Declare subjects” in “Declarations” dropdown. He already has study experience, so he can choose which subjects he wants to study this semester.

The page where the system brought him is divided in two columns, each containing only one of two button-style links with brief descriptions: “Declare typical program for this semester” and “Declare a revised list of subjects”. Ivan is passionately curious, so he clicks on the right link to add more subjects to the default study plan.

This leads him to a new page, where he can see a table of default subjects for his semester and specialty. The table has 5 columns: subject code, title, exam form, number of ECTS credits, and a delete button. First of all, Ivan deletes all computer-related subjects by clicking on “Delete” button in the fifth column. The corresponding row changes its text color to lighter grey and button changes to “Undo” link, showing that this row is inactive. Ivan accidentally removes a wrong subject, but clicking on “Undo” link brings it back in a second.

After deleting all boring subjects, Ivan clicks on “Add other subjects” link below the table. A popup with search field shows up. Below the text field there is a list of chosen subjects which is now empty, and a list of disciplines available for studying, including Organic and Non-organic Chemistry, Programming, Math studies and etc, ie all kind of disciplines available in TUT. Ivan has decided to add some math to his timetable, so he clicks on “Math studies” link, which opens a long list of math-related subjects in table form, each of which has an “Add” button next to it. Ivan chooses one of the subjects by clicking on the button. He then chooses another subject of another discipline. All the subjects he chooses are in real time added to his chosen subjects list.

After he finishes, Ivan clicks on the “Add selected subjects” button in the corner of the popup. The popup disappears, and all subjects he selected are now visible in the table. Ivan scans the list once more, and clicks on the “Submit declaration” button. “Your declaration has been successfully submitted” text shows next to it, and the button becomes inactive. There also is a “Withdraw” link if Ivan wants to cancel his declaration.

Finding out information about payments for repeatedly studied subjects.

On one of the days at work Ivonne has some spare time, so she checks her e-mail. There she finds a letter saying that she has not yet paid the university for her repetitive studying of one of the subjects. Ivonne clicks on the link in the letter, which leads her to OIS login page, and uses her username and password to log in.

After login Ivonne is redirected to “My Invoices” section of her profile. There is a table of subjects that require payment: subject name, sum, and payment deadline. Ivonne has only one subject to be paid for. She clicks on it, and additional information opens in a popup. There is small text explaining how the sum was calculated, a link to the invoice in *.pdf* format, and two bank-links. Ivonne clicks on the link of her bank, follows the steps appearing on the screen, and after she clicks on “Back to the merchant” link on the last step, she gets back to OIS system. She sees a success message, and after going back to

the frontpage she notices that reminder about missing payment is now gone from the deadlines list. Ivonne logs out using a link in the top right corner and proceeds to work.

Registering for the exam.

Near the end of the semester Ivonne remembers that she has to pass her last exam in the university before she will be able to defend her thesis. She logs in to OIS in a usual way, and chooses “Examinations” link from “Registrations” menu item in the navigation bar. A list of subjects that end with an exam and that she had declared in the beginning of semester is shown in table, with a ‘Register’ button in the last column. She has only one such subject, so she clicks on the button in the corresponding row.

A calendar-style view opens in a popup. The days when there is an exams are slightly highlighted, and have time of the exam in it and the number of available seats. Ivonne thinks a little bit, and chooses one of the times she finds the most suitable by clicking on it. After that the button after the calendar saying “Register” becomes active. Ivonne clicks on it, popup disappears, and “Register” button in the table row became a “Remove registration” link.

When Ivonne goes back to the frontpage, she sees that a new sections called “Exams” appeared. She can see there the name of the subject, date and time of the exam, and a “Remove registration” link. Out of curiosity Ivonne clicks on this link, the row becomes greyish and “Registration deleted” text appears instead of “Remove registration”. Fortunately, it has also “Cancel” link next to it, and after Ivonne clicks on it the registration and representing it row in the table ‘come back’.

Submitting thesis theme.

Near the end of May Ivonne remembers that she has to register her thesis theme and the name of her supervisor. She goes to OIS, logs in, and sees a “Submit a thesis theme” link in deadlines section on the frontpage. Ivonne clicks on it, and it leads her to another page, where she can see a text field for typing in the topic, and another text field for typing in

her supervisor's name, as well as a "Save" button underneath. Ivonne fills in the text field: there she simply writes in, and a supervisor she chooses out of a dropdown list which appeared after she started typing, and clicks "Save". The button becomes inactive, and an information box appears next to it saying that the professor she chose now received a notification and has to confirm her as his ward.

The next time Ivonne logs in to OIS, she sees that she has a new notification in her inbox, which is shown next to her name as an envelope with a red number '1' next to it. Ivonne clicks on the envelope, and this leads her to her inbox. She sees a list of the letters she had received, latest on top. The subject of the notification is "Professor you have chosen has confirmed you as his ward". Ivonne clicks on the subject, and the rest of the notification opens in a popup. Ivonne reads it, finds nothing interesting, and closes the tab.

5.4 Creating Prototype Using Twitter Bootstrap

My tool of choice for creating prototype was Twitter Bootstrap - a front-end framework for developing responsive, mobile first projects on the web. There were a couple of reasons why I chose Bootstrap, and the main ones were:

- It has good documentation and tutorials both on its website and on the Internet;
- It is free;
- It allowed me to build the prototype in the way I am used to - by writing code.

Before starting coding, I made a couple of quick sketches on paper:



After I had a general idea about what I want to create, I moved on to coding part.

5.4.1 Prototypes created with Twitter Bootstrap

Mostly SIS looks the same for both first year students and other students, however there are certain differences in “Applying for a study allowance” and “Declaring a list of subjects” sections. Those differences shall be mentioned.

The home page for an anonymous user is extremely simple and features only a login form. Anonymous users have access to some general information and to timetables of both students and professors.

The timetables section features a step-by-step filtration of groups: user chooses her faculty, then her group, and then her group number. This helps to avoid eye searching through a huge list of groups. On the other hand, such system requires three clicks; but none of test participants had something against it.

Study Information System Home General information Timetables

Students Professors

A - faculty E - faculty H - faculty I - faculty K - faculty M - faculty R - faculty T - faculty Y - faculty

AAAB AAAM AAGB AAGM AAHM AAVB AAVM

AAAB21 AAAB27 AAAB41 AAAB61

| Time | Even week | Odd week |
|------------------|--|---|
| Monday | | |
| 08.00 | IDU0200 - Web-based Applications - Architecture, Desing and Technology lecture by Raul Liivrand in VIA-201 | |
| 10.00 | ITV0130 - User Interfaces lecture by Jaagup Irve in V-103 | ITV0130 - User Interfaces exercise by Jaagup Irve in V-103 |
| Tuesday | | |
| 10.00 | ITV0130 - User Interfaces exercise by Jaagup Irve in V-103 | |
| Wednesday | | |
| 10.00 | TET3070 - Micro- and Macroeconomics lecture by Raissa Kokkota in X-308 | |
| 12.00 | TET3070 - Micro- and Macroeconomics exercise by Raissa Kokkota in X-209 | |
| 12.00 | IDU0200 - Web-based Applications - Architecture, Desing and Technology exercise by Raul Liivrand in ICT-501 | |

Save timetable as PDF

After logging in student sees her homepage where are collected the most important things concerning her studies: deadlines, examinations (if there is present at least one registration to an examination; otherwise this section is not shown), recent grades, and recent news. In the grades section student can click on the subject name to see additional information about how to fix the grade:

Study Information System Angelina Kobeleva ✉ Logout ↗

Home Profile ▾ Applications ▾ Declarations ▾ Registrations ▾ Student life ▾ General information ▾ Timetables

Deadlines

| Action | Due date |
|--|------------|
| Declare subjects for this semester | 03.06.2014 |
| Apply for a study allowance | 15.06.2014 |

Exams

| Subject | Date and time | Actions |
|---|------------------|-------------------------------------|
| Introduction to system administrating | 03.06.2014 16:00 | Remove registration |
| Introduction to infosystems | 04.06.2014 12:00 | Remove registration |

Recent grades

| Subject | Grade | Date |
|--|-------|------------|
| Databases II ▾ | 4 | 03.02.2014 |
| Introduction to astrophysics ▾ | A | 13.02.2014 |
| Try again: To try again please write a 800-word essay about why should people study the Universe and bring in to the room II-202, Mon - Fri 13-15PM. | | |
| Mathematical analysis ▾ | 3 | 12.02.2014 |

Recent news

| News | Date of publication |
|---|---------------------|
| The work on improving SIS interface has started | 03.04.2014 |

The profile page is the longest of all. There is collected all the information concerning this particular student: personal information and notification settings, study results with a clearly visible GPA, all study plans, any of the applications she ever submitted, contracts, invoices, and interim performances.

Personal information

Angelina Kobeleva

| | | |
|---------------|------------------------|----------------------|
| Group | IAPB47 | |
| Personal code | 49110090277 | |
| Home phone | 6344129 | Edit |
| Mobile phone | 580 14 48 4 | Edit |
| Address | Mahtra 36-36, Tallinn | Edit |
| E-mail | angie.badfox@gmail.com | Edit |

Notification settings

Notify me if:

- A new grade is received
- A deadline is in 5 days
- A deadline is in 3 days
- A deadline is in tomorrow

[Save](#)

Study results

Current GPA = 3.78.

| Subject | Grade | Date |
|--|-------|------------|
| Databases II | 4 | 03.02.2014 |
| Introduction to astrophysics | A | 13.02.2014 |
| Mathematical analysis | 3 | 12.02.2014 |
| Role of Bible in Modern Society | A | 12.02.2014 |
| Mathematical Statistics and Probability Theory | 3 | 12.02.2014 |

[View all grades](#)

My study plans

| Semester | Date of submission | Status | Number of courses | Number of ECTS credits | Actions |
|------------------|--------------------|-----------|-------------------------------------|------------------------|----------------------|
| 2013/2014 spring | 03.01.2014 | Submitted | 2 | 10 | Edit |
| 2013/2014 autumn | 13.09.2014 | Submitted | 5 | 18 | Edit |
| 2012/2013 spring | 08.01.2013 | Submitted | 3 | 12 | Edit |
| 2012/2013 autumn | 10.09.2013 | Submitted | 8 | 40 | Edit |
| | | | Total ECTS credits: 80 / 180 | | |

My applications

| Type of application | Date of submission | Date of decision-making | Status |
|--|--------------------|-------------------------|------------|
| Acceptance of practice | 13.03.2014 | | Processing |

My contracts

| Contract number | Contract type | Concluding date | Status |
|-------------------------|-----------------------|-----------------|--------|
| C141090 | Student contract (OU) | 11.02.2014 | Valid |

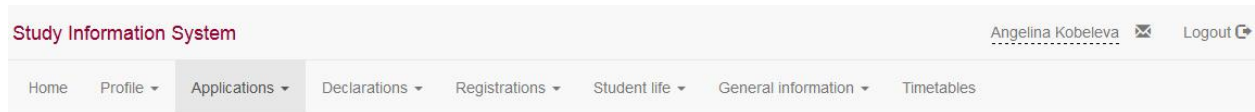
My invoices

| Invoice number | Subject name | Sum | Payment deadline | Status |
|---------------------------|---------------------------|-----|------------------|--------|
| OIS140987 | Introduction to Databases | 215 | 11.06.2014 | Unpaid |

Interim performances

No performances found

All applications are collected in one place. In current version of prototype only application of study allowance is present. New students do not have any study results yet, so they can only apply for a special allowance:



Apply for study allowance

Regular allowance

To apply for a regular allowance you have to:

- Be at least on the first semester of studies;
- Have GPA at least 2.0.

Apply now

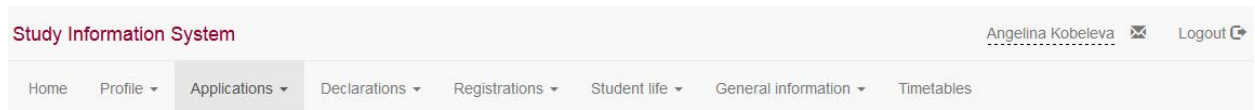
You can not yet apply for a regular allowance, as you have no study experience.

Special allowance

Special allowance is in the first place for students from poor families to help them during their study process.

Apply now

Other students can choose whether to apply for a regular or a special allowance:



Apply for study allowance

Regular allowance

To apply for a regular allowance you have to:

- Be at least on the first semester of studies;
- Have GPA at least 2.0.

Apply now

Special allowance

Special allowance is in the first place for students from poor families to help them during their study process.

Apply now

The popup modal window which shows after clicking on one of the “Apply now” buttons is the same in both cases. It features three fields: student name, her bank account number, and an optional field to attach some documents for a special allowance.

Apply for study allowance

Regular allowance

To apply for a regular allowance you have to:

- Be at least on the first semester of studies;
- Have GPA at least 2.0.

Apply now

Special allowance

Special allowance is in the first place for students from poor families to help them during their study process.

Apply for a special study allowance ✕

Name

Bank account

Documents optional No file selected.

Page for declaring subjects is built exactly as study allowance application's page. It features two choices: a quick one for declaring a typical list of subjects, and another one for declaring a revised list of subjects.

Declare subjects / submit a study plan

Declare typical program for this semester

Choose this if you are fine with following typical program of studies. You will finish university in 3 years if you will follow this program.

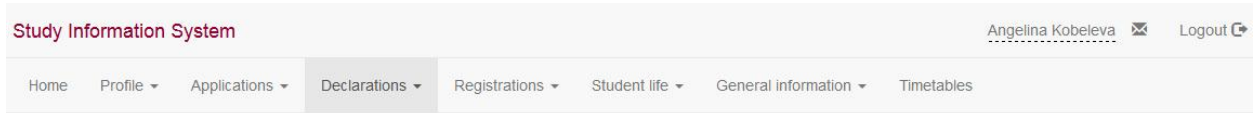
Declare typical list of subjects

Declare a revised list of subjects

Any student who has completed more than one semester of studies can choose which subjects to declare.

Declare a revised list of subjects

Again, new students cannot change their study plans yet and have to declare a typical list of subjects. So they have “Declare a revised list of subjects” button disabled:



Declare subjects / submit a study plan

Declare typical program for this semester

Choose this if you are fine with following typical program of studies. You will finish university in 3 years if you will follow this program.

Declare typical list of subjects

Declare a revised list of subjects

Any student who has completed more than one semester of studies can choose which subjects to declare.

Declare a revised list of subjects

You cannot revise your list of subjects before you finish first semester

Clicking on “Declare typical list of subjects” shows a modal popup with an overview of subjects to declare and a “Submit declaration” button:



Declare subjects / submit a study plan

Declare typical program for this semester

Choose this if you are fine with following typical program of studies. You will finish university in 3 years if you will follow this program.

| Subject code | Title | Exam form | Number of ECTS credits | Actions |
|-----------------------|---------------------------------------|-----------|------------------------|---------|
| ITV0050 | Introduction to system administrating | Exam | 5 | Delete |
| IKV0050 | Micro- and macro economics | Exam | 5 | Delete |
| KIT4013 | Introduction to infosystems | Exam | 6 | Delete |
| TJF9834 | Databases I | Exam | 6 | Delete |
| ITV0110 | User Interfaces | Exam | 5 | Delete |
| Total amount of ECTS: | | | 27 | |

Cancel Submit declaration

When submitting a revised list of subjects, it is possible to add other subjects outside typical for this semester program. There is a special page for this, where student can see her typical subjects, can delete them, and can also add other subjects by clicking on “Add other subjects” button. This opens a popup, where student can use a search field with auto complete to find a specific subject; or can browse through divisions using a filter. Student can select any amount of subjects by clicking on “Add” button in the corresponding row, and then add all of them at once to her declaration by clicking on “Add selected subjects”.



Create a revised list of subjects

Choose subjects to add to your declaration

Search for a subject

Chosen subjects:

Physics Math IT Languages Economics Humanities

| Subject code | Title | Exam form | Number of ECTS credits | Actions |
|--------------|------------------------------|-----------|------------------------|---------|
| ITN9745 | Introduction to astrophysics | Exam | 5 | Add |
| JUY0946 | Science of Risk and Safety | Arvestus | 3 | Add |
| IKT0921 | Electrical Materials | Exam | 4 | Add |
| IKT0922 | Electrical Machines | Exam | 5 | Add |
| IKT1922 | Basics of Automatic Control | Exam | 4 | Add |

Cancel Add selected subjects

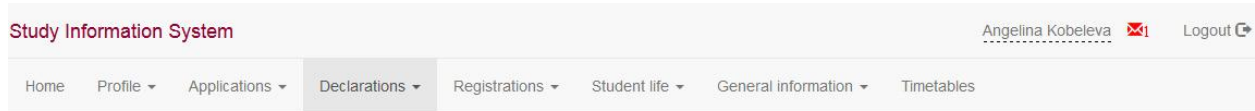
The declaration is submitted by clicking on “Declare this list of subjects”, after what a success message is shown as well as a link to withdraw the declaration and continue editing:

| | | | | |
|------------------------------|-----------------|------|----|--------|
| ITV0110 | User Interfaces | Exam | 5 | Delete |
| Total amount of ECTS: | | | 27 | |

Add other subjects Declare this list of subjects

Your declaration has been successfully submitted / Withdraw

Another declaration present in the prototype is declaration of graduation thesis topic. Student has to fill in the topic, choose her supervisor from a dropdown list with autocomplete, and click on “Save” button. After that a message appears that says that chosen professor now has to confirm you as his ward. Here is also visible how an inbox link looks like when there are unread messages.



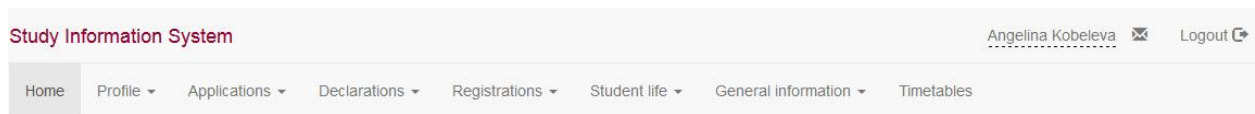
Declaration of graduation thesis topic

Graduation thesis topic

Supervisor

The chosen professor now has to confirm you as his ward.

Message about approval can be found in student’s inbox:



Inbox messages

| Subject | Sender | Date |
|--|----------------|------------|
| Professor you have chosen has confirmed you as his ward <p>The deadline for first draft is on 31 of May. The final version should be submitted until 5 of June. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p> | Administrative | 13.05.2014 |
| Please don't forget to vote | Administrative | 10.02.2014 |
| Happy Birthday! | Tutti el Fiore | 09.10.2010 |

An important part of student life is registration to examinations. This can be found in a separate section. In the current version of SIS available examinations are displayed as a huge table with a lot of unnecessary at this stage information. This makes finding a suitable time harder than it should be. The most natural way of thinking about dates is a calendar, so I have used a calendar

view to display available times. Again, after making a registration student can delete it from the same page in a click and then register again if needed.

Register to examinations

Choose a suitable time of the exam ✕

Subject code

ITV0050

IKV0050

KIT4013

TJF9834

ITV0110

June

| Mo | Tu | We | Th | Fr | Sa | Su |
|-------------------------------------|-------------------------------|----|--------------------------------|----|----|-------------------------------|
| | | | | | | 1 16:00 / 6 places left |
| 2 | 3 14:00 / 4 places left | 4 | 5 16:00 / 14 places left | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | | | | | | |
| ← Today → | | | | | | |

Register to examinations

| Subject code | Title | Exam form | Number of ECTS credits | Actions |
|------------------------------|---|-----------|------------------------|---|
| ITV0050 | Introduction to system administrating | Exam | 5 | <input type="button" value="Register"/> |
| IKV0050 | Micro- and macro economics | Exam | 5 | <input type="button" value="Register"/> |
| KIT4013 | Introduction to infosystems | Exam | 6 | <input type="button" value="Register"/> |
| TJF9834 | Databases I | Exam | 6 | <input type="button" value="Register"/> |
| ITV0110 | User Interfaces | Exam | 5 | Remove registration |
| Total amount of ECTS: | | | 27 | |

5.5 Usability Testing of the Created Prototype

When the prototype was ready, I asked a couple of TUT students to participate in usability testing. A few of them refused, but I got three people who agreed to participate and to have their actions and voice comments recorded. I decided to stop at three people as Steve Krug recommends, as three people is more than enough to fix the major problems. Later I will refer to them as Participant A, Participant B, and Participant C.

All of participants are student of Tallinn University of Technology (TUT), which means that they are part of Study Information System (SIS) target audience. Two people are now finishing their studies, and one of them had studied in TUT for a couple of months, and then left.

After participants were chosen, I moved on to choosing tasks to test and to creating short scenarios out of them. I decided to test the most important cases of using SIS, which were earlier described as personas' goals. The tasks to be tested were following:

1. To check your own timetable to find out which lectures, when and where to attend;
2. To check timetable of a professor to find him after lectures;
3. To declare subjects in the beginning of semester;
4. To submit an application for monthly allowances;
5. To pay for a repeatedly studied subject;
6. To register for an examinaion;
7. To submit a graduation thesis theme;
8. To checking out latest grades and current GPA.

The tasks were later expanded to short scenarios to give participants some necessary information (for example, requirements for login):

1. You are a student of Tallinn University of Technology (TUT). Your group number is IASB43. You were invited to a party on Thursday evening, and before you accept or decline the invitation want to know whether or not you have lectures on Friday before 13PM.

2. You are a student of Tallinn University of Technology (TUT). You are now writing your bachelor thesis, and want to hear some advice from your supervisor, professor Eugenia Ivanova. Find out where and when you can find her in the university.
3. You are a first-year student of TUT. On one of the lectures you have heard, that you need to declare subjects for this semester. Do it. To login to current version of prototype, you need to fill in the fields of login form with any text you like.
4. You have studied at TUT for a couple of years. It is the beginning of semester now, and you need to declare subjects to study. Besides typical subjects, you also want to study subject “Role of Bible in Modern Society”. Submit your revised accordingly to this information study plan. To login to current version of prototype, you need to fill in the fields of login form with any text you like.
5. You are a TUT student. The last semester was hard, so you have failed one of the subjects and had to study it again. Find out how much does it cost, and pay for it. To login to current version of prototype, you need to fill in the fields of login form with any text you like.
6. You are a studying at Tallinn University of Technology. The semester is almost over, so you want to register to an examination on “Databases I”. You do not have much spare time in May, so you’d prefer to have this exam in June. Please register to this exam.
7. You are a student of Tallinn University of Technology (TUT). You are hoping to finish your bachelor degree this year, and you are now working on your bachelor thesis. Submit your graduation thesis topic.
8. You have studied at TUT for a couple of years. You did not pay much attention to studying, so some grades were poor. Now that you are close to finishing the university you are worried whether your GPA is over 2.0. Check out your GPA.

The usability test itself was conducted by Steve Krug's book as described above, without modification to the process.

5.5.1 Results of the Usability Test

As expected conducted usability tests revealed a couple of problems of the new prototype. Two of the test participants had problems with finding invoices, which they explained was because invoice section was placed too low on the page. This means that menu structure has to be revived once more.

Another problem was with success, error, and information messages, which turned up to be not as noticeable as expected.

A couple of other things are more inconveniences than problems, for example, missing link to home page on a logo, not working form submission by hitting 'Enter' key. One of the test participants also noted, that he would like to have a back link somewhere on the page to move back to the previous section. This participant also said, that he can use the browser's built-in back button then. However, this lead me to an idea to add breadcrumbs (ie a series of links which step-by-step show to user where she is on the website) to the prototype.

As a conclusion, these are the problems that were found and that need to be fixed in the next prototype version, arranged by their priority:

1. Student cannot change typical program of subjects though he expected this;
2. Invoices are hard to find;
3. Back links or breadcrumbs are expected on the page;
4. Success messages are missing from some places, such as submitting a thesis topic;
5. Login form submitting does not work by hitting enter;
6. Link back to home page is missing from the logo.

5.6 Creating a Revised Version of Prototype

After usability testing I created a second, revised version of Study Information System prototype, which fixes found problems.

Student cannot change typical program of subjects though he expected this

I did not want to completely get rid of a fast way to declare just the typical subjects list, because a lot of students do not make changes to their study plans and just submit a typical declaration every semester. However, in the previous version user had to close the current modal window and choose another link. I solved this by placing a link to declaring a revised list of subjects in the modal window of declaring a typical list of subjects along with some explanatory text. I also removed unnecessary “Delete” buttons from typical subjects list as they were useless:

The screenshot shows a modal window titled "Declare typical program for this semester" with a close button (X) in the top right corner. The window contains a table with the following data:

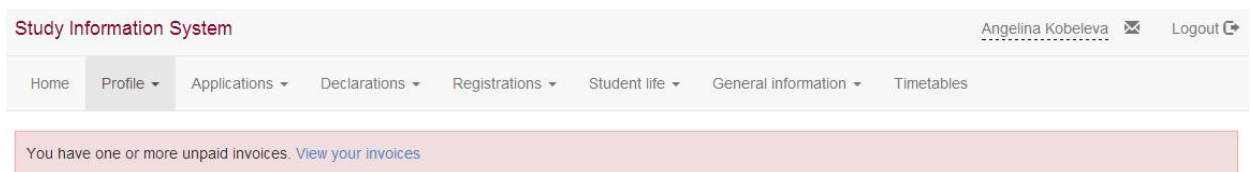
| Subject code | Title | Exam form | Number of ECTS credits |
|------------------------------|---|-----------|------------------------|
| ITV0050 | Introduction to system administrating | Exam | 5 |
| IKV0050 | Micro- and macro economics | Exam | 5 |
| KIT4013 | Introduction to infosystems | Exam | 6 |
| TJF9834 | Databases I | Exam | 6 |
| ITV0110 | User Interfaces | Exam | 5 |
| Total amount of ECTS: | | | 27 |

NB. This is a typical program for a quick declaration, you can not make changes here.
Please [declare a revised list of subjects](#) if you want to make changes.

At the bottom right of the modal, there are two buttons: "Cancel" and "Submit declaration".

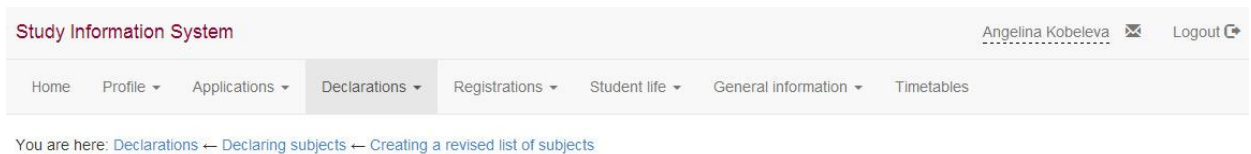
Invoices are hard to find

As information about payments is important, it has to capture the attention of user. So I placed a highlighted notification that user has unpaid invoices to the top of page. Notification has text explaining that user has to pay for something, and a link to the “Invoices” section of user’s profile where she can make payment. The same kind of notifications can be used, for example, to notify user about new items in her inbox (in addition to change in the inbox link in the header), or about new grades.



Back links or breadcrumbs are expected on the page

One of the test participants said that he expected a back link after he has submitted his study plan. I fixed this issue by adding a row of breadcrumbs on top of each page:



Success messages are missing from some places, such as submitting a thesis topic

As became clear during the test, people expect also a success message that thesis topic was saved and a simple information message is not enough. So a success message was added. I also added a link to student’s inbox, where she will find professor’s decision when he will give one:

Declaration of graduation thesis topic

Graduation
thesis topic

Supervisor

Your thesis topic was successfully saved.

The chosen professor now has to confirm you as his ward. You will find the decision in [your inbox](#).

Login form submitting does not work by hitting enter and link back to home page is missing from the logo

Those problems were fixed on a coding level, without visual representation.

6. Conclusion

In this thesis I have illustrated how to apply user-centered approach to creating a prototype of a system, starting from creating personas and writing scenarios for them and proceeding to creating a prototype and testing it in order to find problems and fix them. My main goal was to show how user-centered design (UCD) helps to build a more user-friendly interface. The user-friendliness of a prototype created using UCD was proven by a usability test which has revealed just a couple of problems.

The need for designers and developers to understand target audience is a necessity, as this is the way to creating an easy-to-use system, website, or application. The work showed that the most common user-centered design techniques such as personas and scenarios do not take much time, but are extremely helpful when it comes to understanding the end user. Conducted usability test was also helpful as it showed usability problems I have not noticed in the first time and allowed me to fix them before it was too late.

The resulting prototype is by no means complete and should be considered only as a starting point for further enhancement. First of all, missing in the current versions pages should be done. For this new personas and scenarios should be introduced to understand users whose needs were not covered by this thesis. After that, there should be another usability testing round to find out a new portion of problems to fix.

Resüme

Viimastel aastatel pööratakse süsteemide kasutatavusele üha rohkem tähelepanu. Selleks, et muuta süsteeme kasutajatele lihtsamini mõistetavaks, on arendatud spetsiaalne meetoodika, mille nimeks on kasutajakeskne disain. Käesolevas töös kasutasin kasutajakeskse disaini meetodit Tallinna Õppeinfosüsteemi uue prototüübi tegemiseks. Meetoodika eeldab kasutajate peale mõtlemist juba töö esimestes etappides. Enne töö alustamist on vaja koguda informatsiooni lõppkasutajatest – mida nad ootavad süsteemist ja mis on nende eesmärgid. Seejärel näeb kasutajakeskne disain ette luua olemasoleva informatsiooni põhjal kasutajad (*personas*) – lühikesed väljamõeldud inimeste kirjeldused, mis esindavad süsteemi lõppkasutajaid. Kasutajatel on olemas eesmärgid ja nende eesmärkide põhjal on kirjutatud kasutusjuhud (*scenarios*) – lühikesed kirjeldused erinevatest süsteemi kasutusjuhtudest, mida peab olema loodava süsteemi abil võimalik ellu viia.

Pärast ettevalmistavat etappi on aeg teha prototüüp. Käesolevas töös kasutasin prototüüpimiseks Twitter Bootstrap raamistikku (*framework*). Kui prototüüp on valmis, on vaja teha selle prototüübi kasutatavuse testid. Viisin testimise läbi kolme inimese abil ning parandasin leitud vead prototüübi teises versioonis. Sel viisil lõikasin kasu kasutajakeskse disaini põhieesmärgist – leida kasutatavuse vead võimalikult vara ning parandada need kohe projekt esimestes etappides.

Käesoleva töö jätkamiseks tuleks lisada puuduvad lehed ning rakendada nende valmistsamisel samuti kasutajakeskse disaini meetoodikat.

Literature cited

1. **Alain Abran, Adel Khelifi, Witold Suryn, Ahmed Seffah.** Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal*. 2003, Vol. 11, 4.
2. **Quesenbery, Whitney.** What Does Usability Mean: Looking Beyond 'Ease of Use'. *Proceedings of the 48th Annual Conference, Society for Technical Communication*. 2001.
3. **Spencer, Donna.** Papers: Step Two Designs. *Step Two Designs Pty Ltd*. [Online] November 1, 2004. http://www.steptwo.com.au/files/kmc_whatiusability.pdf.
4. **Reed, William A.** *Texology Sciences: Human Behavior & Technology Design*. [Online] 2014. <http://texology.com/wp-content/uploads/2013/07/User-Centered-Design-with-ISO-9241-Texology-Sciences-2014.pdf>.
5. **Nielsen, Jakob.** Usability 101: Introduction to Usability. *Nielsen Norman Group: Evidence-Based User Experience Research, Training, and Consulting*. [Online] January 4, 2012. <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
6. *User-Centered Design*. **Chadia Abras, Diane Maloney-Krichmar, Jenny Preece.** s.l. : Sage Publications, 2004.
7. **James J. Kubie, L. A. Melkus, R. C. Johnson Jr, G. A. Flanagan.** User-Centered Design. [book auth.] Heikki Topi Carol V. Brown. *IS Management Handbook, Seventh Edition*. s.l. : CRC Press LLC, 2000.
8. **Mica R. Endsley, Debra G. Jones.** *Designing for Situation Awareness: An Approach to User-Centered Design, second edition*. s.l. : CRC Press, 2004. 9781420063554.
9. **Donald A. Norman, Stephen W. Draper.** *User Centered System Design; New Perspectives on Human-Computer Interaction*. s.l. : L. Erlbaum Associates Inc., 1986. 0898597811.
10. **Norman, Donald A.** *The Psychology of Everyday Things*. 1988. 978-0-465-06710-7.
11. **Krug, Steve.** *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. s.l. : New Riders, 2010.
12. **Valeria Gasik, Zahhar Kirillov, Daria Tokranova.** *IxD, Personas, Scenarios, User Stories*. [Presentation] s.l. : TLU / HCI Summer School, 2013.
13. **Cooper, Alan.** *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity*. s.l. : Sams - Pearson Education, 2004. 0672326140.

14. **Tamara Adlin, John Pruitt.** *Persona Lifecycle : Keeping People in Mind Throughout Product Design.* s.l. : Morgan Kaufmann , 2006. 9780080455730 .
15. **Steve Krug, Roger Black.** *Don't Make Me Think! A Common Sense Approach to Web Usability.* s.l. : New Riders Publishing, 2000. 0789723107.
16. **Jeffrey Rubin, Dana Chisnell.** *Handbook of Usability Testing: Howto Plan, Design, and Conduct Effective Tests.* s.l. : Wiley Publishing, 2008. 978-0-470-18548-3.
17. **Nielsen, Jakob.** Why You Only Need to Test with 5 Users. *Nielsen Norman Group: Evidence-Based User Experience Research, Training, and Consulting.* [Online] March 19, 2000. <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
18. **Kimmo Tarkkanen, Pekka Reijonen, Franck Tétard, Ville Harkke.** Back to User-Centered Usability Testing. *Human Factors in Computing and Informatics.* 2013, Vol. 7946.