

EQ

Management System

Group B403a:

**Christoffer Hviid Poulsen
Daniel Neupart Hansen
David Rasmussen Fibiger
Maxymillian Adam Schou
Michael Kvist Nielsen
Simon Munk Sørensen**

**BAIT4-project 2010:
EQ Mangement System**

Supervisor: Xiufeng Liu

Aalborg University

**Department of
Computer Science**

BAIT & Informatics

BAIT / Informatics 4. Semester

Project group: B403a

Title: EQ Management System

Theme: Modelling IT Systems

Project period: BAIT4 / INF4
3rd of February - 28th of May
2010

Project group: B403a
Education: BAIT / Informatics

Members: Michael K. Nielsen
Christoffer H. Poulsen
Daniel N. Hansen
Maxymilian A. Schou
Simon C. M. Sørensen
David R. Fibiger

Supervisor: Xiufeng Liu

Synopsis:

This report describes the development of a system, designed to support administration and loan of equipment to the employees and students at the Institute of Mathematics at Aalborg University.

First it explores which project management method best suits the development.

Afterwards an analysis of the problem domain is conducted. This leads to a design document and a functional system. The system is then performance and usability tested.

It is concluded that the system works, but could have used a framework in order to be more flexible.

Pagecount: 107

Submission date: 28/05/2010

Preface

This project and the associated software is done by the project group b403a on the BAIT4 and INF4 educations at Aalborg University. The project period stretches from the 3rd of February to the 28th of May 2010. The theme of the project is “Modeling IT Systems“ and the project is based on the development of an IT System. This system should help the IT Administrator at The Institute of Mathematics at Aalborg University, with administrative tasks, an on line equipment system.

As part of the semester, knowledge from the following courses are included in the project: Software Engineering(SOE) and Databases (DB). But the rapport also draws on the course “System Analysis and Design“ (SAD) from 2nd semester and the “Usability and Usability evaluation of IT-systems” (UUIT) course from the 1st semester.

The first chapter of the report describes the context of the project, and specifies the problem domain. The second chapter describes the selected methods of the project, both for managing and developing. Chapter three and four contains the analysis of the problem domain, and the design of the system. The testing of both the underlying code, and the user interface is explained in chapter 5. Finally a reflection chapter will summarize the project in a discussion, a conclusion and a consideration for further research. These considerations will indicate what the next steps of the project should be. One could argue that this is actually the main result of the project, as this reflects the learning process of the semester.

Various parts of the project involves data collection or data generation. Since the analysis and the testing sections are based upon a rather large data set, only the data worth mentioning is included in the report itself. The complete data set can be found in the appendix.

Michael K. Nielsen

Christoffer H. Poulsen

Daniel N. Hansen

Maximillian A. Schou

Simon C. M. Sørensen

David R. Fibiger

1	Introduction	11
1.1	Problem statement	13
1.2	Delimitation	13
2	Project management	15
2.1	Introduction	15
2.2	Previous experiences	15
2.3	Considerations	16
2.4	Use of method	17
2.5	Risk analysis	19
2.6	Tools	20
3	System analysis	21
3.1	Introduction	21
3.2	Purpose	22
3.3	System definition	23
3.4	Problem area	24
3.5	Scope	25
3.6	Use	27
3.7	Functions	28
3.8	User interface	29
3.8.1	Dialog form	29
3.8.2	Overview	30
3.8.3	The platform	30
3.8.4	The usefulness and realizability of the system	31
3.8.5	Strategy	31

4	System design	33
4.1	Introduction	33
4.2	Design quality	33
4.3	Architecture	34
4.4	Database design	36
4.4.1	Conceptual design	36
4.4.2	Logical design	39
4.4.3	Physical design	42
4.5	System design	43
4.5.1	Interaction	43
4.5.2	Connecting to the database	44
4.5.3	Navigation	45
4.5.4	Design	47
4.5.5	Security	55
4.5.6	Validation	57
5	Testing	59
5.1	Introduction	59
5.2	Performance testing	59
5.2.1	Loadtime test	59
5.3	Usability test	63
5.3.1	Purpose	63
5.3.2	Setup	64
5.3.3	Usability problems discussion	65
6	Reflections	69
6.1	Introduction	69
6.2	Discussion	69
6.3	Conclusion	75
6.3.1	Introduction	75
6.3.2	Project management - method choice	75
6.3.3	Tools	76
6.3.4	Database system	77
6.3.5	Summary	77
6.4	Putting into perspective	78
	Bibliography	82
A	User stories	83
B	Planning game	87
C	Datatypes in database	89

D Usability test	93
D.1 The test	93
E Usability problems	97
E.1 Remaining problems discussion	100
F Product backlog	103
G CD	107

CHAPTER 1

INTRODUCTION

This section will introduce the problem domain and the difficulties in gaining access to it. The goal is to define a specific problem, on which the project could be based upon.

The system administrator at the Institute of Math at Aalborg University currently has a problem with the system holding equipment information. The system holds information about the current hardware and other equipment at the math facility, where it is placed and so on. Every single piece of equipment goes by the system administrator before it is created in the database.

The current database is placed on a local desktop computer, which needs to run before he can enter new equipment or access old. The problem is that the existing system is based on an old version of the Perl language, which unfortunately does not work with newer versions and therefore it cannot easily be reprogrammed and moved to a newer server. He has considered to learn how to setup a MySQL database himself, to handle the information in the future, but so far he has not had the time. As one of the group members heard of this problem, all agreed that it could make a suitable project for the semester, as an alternative to the usual projects, that are based upon hypothetical problems. This project could improve the administrators efficiency, by supporting him in his daily work.

The system administrator needs a system that can hold the same information as the old database. In addition the administrator have several new ideas. Therefore the new system should also be implemented with information about the equipment, where it is placed and so on. This will be one of the main subjects in the analysis, where the users requirements to the system will be defined. The new system will be based on a database which will run on the current Linux servers at the Institute of Math facility. The system administrator also has an idea of a web interface from which he can access the information. This will also give the opportunity to access the database from anywhere.

The task of this project is to analyze and design a new system for the the system administrator. In this context there are three challenges to overcome. The first one is the design of a database, that can hold information of all the equipment. Knowledge from the course “Databases“ will be included here. The second one is to design a web-interface so that he can access the database. The third and possibly the most important of all, is the project management. This is the part which structures the entire project and combines the other two challenges. Especially when choosing the development method, this is where the analysis once again steps in and aids the decision.

Even though there are no direct method for choosing the right development method, certain key factors can point one in the right direction. This process is described in section 2.3. The next section will clarify the exact problem on which the project is based.

1.1 Problem statement

As the previous section described, the system administrator needs a new system that can handle the same functionalities as the old system and in addition some new. The problem statement is derived from an interview with the administrator. It can be found on the enclosed CD, see appendix G. The problem statement is as follows:

How do we design a new system for the administrator at the institute of mathematics, that fulfils the users requirements?

1.2 Delimitation

The following section will describe what the project tasks will be. Since the system could be made in several different ways, we have chosen to delimitate it, also because of the time limit of the project. The following bullets will describe the delimitation of the project.

- Analysing and developing a system based on the criteria that was derived from the administrator.
- Creating a system with the knowledge from the course “Databases”.
- Choosing a system based on our level of technical skill. The group have experience with C# applications, which means that aiming towards languages on the same level. A language that has a lower abstraction level is therefore not preferable.
- Securing the system. Given that the system could potentially hold some sensitive information.
- Choosing a development method based on the course “Software engineering”. This mean that agile and traditional methods are considered.

These delimitations is used as guidelines for the project and these will hopefully reduce the risk of aiming the project to wide and also reducing the chance of ending up in an uncontrollable situation. It will also help to ensure that the time limit of the project is kept in focus. At first it was considered designing a system that would involve analysing both the administrator and the loaners, but because of the time limit it was decided to focus on the administrator only. The next chapter will discuss which development method should be used for this particular situation.

CHAPTER 2

PROJECT MANAGEMENT

2.1 Introduction

As of this point a problem have been selected, but before begining the analysis, an overview on how to manage the entire project is considered needed. This is due to the fact that the base for the analysis, simply is scoped to generally in the problem delimitation. The chosen methods for developing will derectly reflected in the analysis and design style. Therefore considerations on how to delevlop the system is the next logical step. It also describes the different tools used to manage the project.

2.2 Previous experiences

Our past experience with development methodologies have been few. Since we basically only have worked with traditional methods, we have no experience with agile methods. The traditional methods we have worked with (e.g. the waterfall model) were successful and we had little or no problems. The reality was that our past projects were fictive and therefore they were not subject to much change. This semester we have a real situation and an actual costumer that is in need of a working product. This means that the situation has changed and that we have a real person that awaits a working program.

2.3 Considerations

Before choosing a development method for the project a comparison between traditional and agile methods should be made. A comparison of traditional vs. agile can be seen in figure 2.1. One of the major things to notice is that traditional design is solved by thorough analysis. Some agile methods on the other hand solve the problem by trial and error e.g. XP. Another important thing to notice is that traditional methods see the world as consistent. Therefore it would make good sense not to test the product before the product is finished. Agile on the other hand sees the world as changing. Therefore the product should be tested at the end of each iteration to correct mistakes.

	traditional	agile
the world (user domain) is	stable and governed by rules (natural science)	changing and governed by relationships between people (social science)
development is the	application of an externally imposed algorithm (the design method)	stepwise refinement of a local design solution
a complex problem is solved by	rational analysis	informed trial and error/experiment
knowledge exchange is	formal and documented	informal by direct communication
.....		

Figure 2.1: *This model university [2010] describes the main differences between traditional and agile development, while traditional is waterfall strict and documented, agile on the other hand is characterized by trial and error and little or no documentation.*

Traditional methods require that you document and define the system before starting the programming. This documentation will end up in an analysis and design document which will work as basis for programming. Therefore, there may also be a defined time on the completion of the development work. A traditional method is preferable for a project where there is a low probability of change direction and therefore defined in advance.

However a lot of documentation will not benefit us because there may be changes in development e.g. that our customer changes his opinion about what he wants. This could potentially be characterized as a major change, thereby making the analysis incomplete. This would point in the agile direction, where the analysis is updated throughout the project, accordingly to the users requirements.

XP will call for a minimal amount of planning in the start, since XP begins the developing process right after the first requirements have been identified. Short sprints is a main aspect of XP. The outcome of a sprint would in the end of an iteration be

presented to the customer. It is therefore possible to catch any problems in the design before they become a big problem. This method is good if there is high uncertainty for the product and where the customer does not know what he wants. XP cannot give a precise deadline of when the development is completed because of the possible changes in the project. Therefore this method is not preferable because our project is limited to a certain period of time. This factor therefore speak for a traditional development methodology where the project period is planned in advance and therefore ended at a certain time Larman [2004].

The last method we have chosen to use is Scrum Larman [2004]. This method works in iterations, like XP, and recommend that you make little or some documentation before starting the development work and this could be a better method for setting a date for completion. Secondly the uncertainty for the project would be eliminated or at least minimized because of the iterations. Therefore this method combined with the short iterations from XP has been chosen.

2.4 Use of method

The use of XP and Scrum will be done by choosing certain parts from each method. From XP we have chosen to use the short iterations. This will help if the customer changes his mind about product development. This will hopefully catch problems in time in order to correct them. A Scrum-master will be chosen and according to Scrum it should be the one with the most experience in the group of developers. The task of the Scrum-master is to partial control and keep an overview of the development of the project.Larman [2004]

A meeting was arranged with the customer who is the system administrator at the Institute of Mathematics. At this meeting the customer gave a description of what the current situation was and what the new system should do. This includes functions and design. The analysis will therefore be build on an understanding of the current situation and an understanding of the future situation. These descriptions will be documented before the development itself begins. This will also give a rough estimate of exactly how much time is needed to develop the entire system.

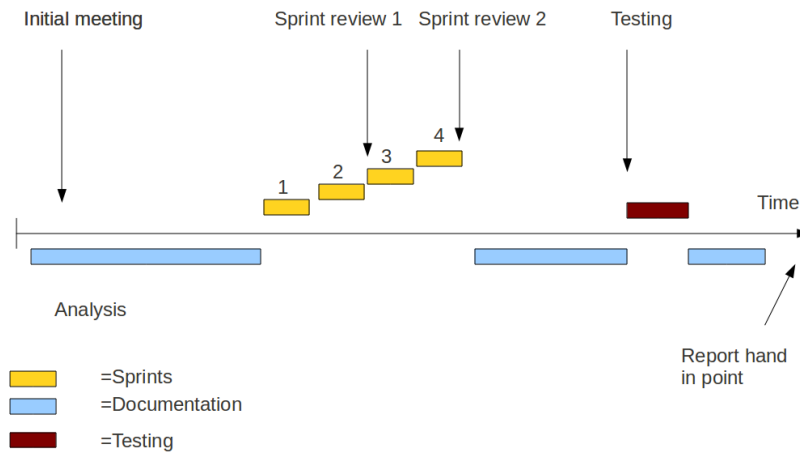


Figure 2.2: *The project period, different colors shows different stages.*

Development work will be divided into different tasks with prioritation and will be included in the product backlog. The backlog keeps track of the remaining tasks of each sprint. A time line of the project period can be seen in figure 2.2. The first task is to analysis the current situation. This will be done with an interview in the start of the project period. The analysis, see chapter 3, will also describe the future system with the wanted functions. Second task is the design, this will describe how the system is to be designed, see chapter 4, and includes aspects like technical platform, architecture and the user interface.

After that the sprint period will begin. There will be four sprints, due to the time aspect. The sprints will run over a period of one week. Because of us only having 4 weeks of implementation time some functions might not implemented, these functions can be seen in the appendix on figure F.3. Every second sprint we will have a review with our customer, to see if the solution we have implemented is good enough. The most optimal solution would be to have a meeting with Finn after each sprint but due the small sprints we feel that a meeting after two sprints will be sufficient.

After the sprint period some more documenting will be done and after that the testing period will begin. This will include a usability test and a performance test of the system, see chapter 5.

2.5 Risk analysis

Dealing with the many project risks, have been a growing concern. During the lectures the importance of this subject have been overwhelming. Especially when lecture holders from Atira and Netcompany, respectively, talk about their way to develop software. In both the traditional and agile thinking, dealing with project failures were a everyday concern. This stressed the importance off making a list of all possible scenarios in which this project could fail.

Possible project failures

As part of the chosen method Larman [2004], a risk analysis have been completed midway in the project. This was done to comprehend the many scenarios in which the project could come to an halt Wikipedia [2010c]. This is the list which were created by all the group members. All were assigned the task to list all the possible causes to failure. This was then combined to one list as listed below.

Project concerns

1. Requirement misunderstanding 8 ↑
2. Absence in the group 8 ↓
3. Lack of programming skills 7 ↑
4. Lack of planning skills 7 ↑
5. Lack of development time 7 ↑

This list summons up all the possible failure scenarios. It represents the scenarios, which the group sees as the most likely scenarios to fail, and those with the greatest impact on the project. These are the scenarios which have to be handled immediately, therefore escape scenarios were created to avoid project failures. The idea was that if a project faced any of the scenarios the risk analysis were to be consulted.

2.6 Tools

During the project, Dropbox has been used as a version control manager. Subversion were also considered, but because of complications with Windows subversion clients on earlier semesters, it was rejected in favor of Dropbox. Both the seamless integration in all operating systems used by the group, and the updates on the fly, in contrast to the manual commits of subversion, were reasons for this decision.

As development environment, Netbeans was chosen. At first, it was intended that Visual Studio should be used for development, but since Netbeans NetBeans [2010] was freely available as open source, and because it supports multiple operating systems, which makes it ideal for the group that uses both Windows, Linux and OS X systems, it was selected.

The report is written with L^AT_EX, as this was agreed by the majority of the group members. Though not all had experience with it, it was the general opinion that it was the most viable solution to use, when writing the project report in the group.

3.1 Introduction

This chapter describes the analysis phase of the project. The analysis phase will be described chronologically, accordingly to some of the object oriented analysis principles Mathiassen et al. [2001]. Especially some of the UML diagram has funded a baseline for the further analysis part. Before starting, a collection of data to perform the analysis on needs conducting. This is the main part of the next section.

Data collection

Different techniques were used to collect data for the system design, from the user. The process has involved both interviews and usability testing.

In the beginning of the project, a interview with the costumer was arranged in order to determine, whether the proposed idea of a on line booking and lending system for the institute of mathematics was a suitable project. The interview is available as a audio file on the enclosed CD-ROM in appendix G.

Before the first sprint in the development, an interview with the customer was arranged, in order to find out how the website design should be. This lead to some sketches of how the customer imagined the website should be designed.

To get feedback for the visual design of the website, along with the functionality, a usability test was designed. The test is described in appendix D. The results from the test are presented in section 5.3.

3.2 Purpose

The system administrator at the Institute of Math at Aalborg University has a problem with the current database he is using. The problem is that it is based on an old version of Perl, which does not work with newer versions of Perl, and cannot easily be changed/rewritten to fit it. The system holds information about the current hardware that the institute possesses.

The current database is placed on a local machine, that needs to run before one can access it. Therefore every single piece of equipment has to go by the system administrator before it is created in the database.

When a person wants to rent an item, he or she must come to the system administrator. One of the current problems is that the administrator then must fill out a piece of receipt, that he archives. When the item is then returned he discards the receipt.

The system administrator would like a new system that can hold the same information as the old one. Therefore the new system should be implemented with information of equipment, where its placed and so on. Furthermore the need for a function that can represent a list of hardware (that can be printed) is present when for example the administrator needs to do inventory of all the equipment

There will also be room for some features, in which the system also could be holding information about other types of equipment like software, furniture and so on. Though the primary concern is hardware. Also there will be room for the feature that students could rent equipment by themselves.

Conditions (Betingelser)	The system must be implemented so it will work on different systems(mostly Linux) and different browsers(mostly Firefox).
Scope (Anvendelsesområde)	An administrator that handles all data going into the system.
Technology (Teknologi)	A website running on a server. HTML for the website, PHP for data handling and MySQL for the database management.
Objects (Objekter)	Items, Persons, invoices and Locations
Functionality (Funktionalitet)	The system is supposed to replace an older system and will be a supporting the System administrator in his work
Philosophy (Filosofi)	Administrative tool

Figure 3.1: *The BATOFF criteria helps when developing a system. The purpose is that you can hold the project up against it and determine if you are on the right track.*

3.3 System definition

A system that can handle information about equipment(primarily hardware) that The Institute of Mathematics currently holds. The system will primarily be used as an administrative tool and is primarily used by the system administrator. The system can in the future be used as a system to lend out equipment to employees or students at the institute.

The system will be implemented as a website running on a server, so that it can be accessed from multiple systems. The system will mainly be created for Linux, and optimized for Firefox, since this is the browser the system administrator uses. The languages used is HTML for forming the web page and PHP to control the data. Furthermore MySQL will be used for database implementation and management.

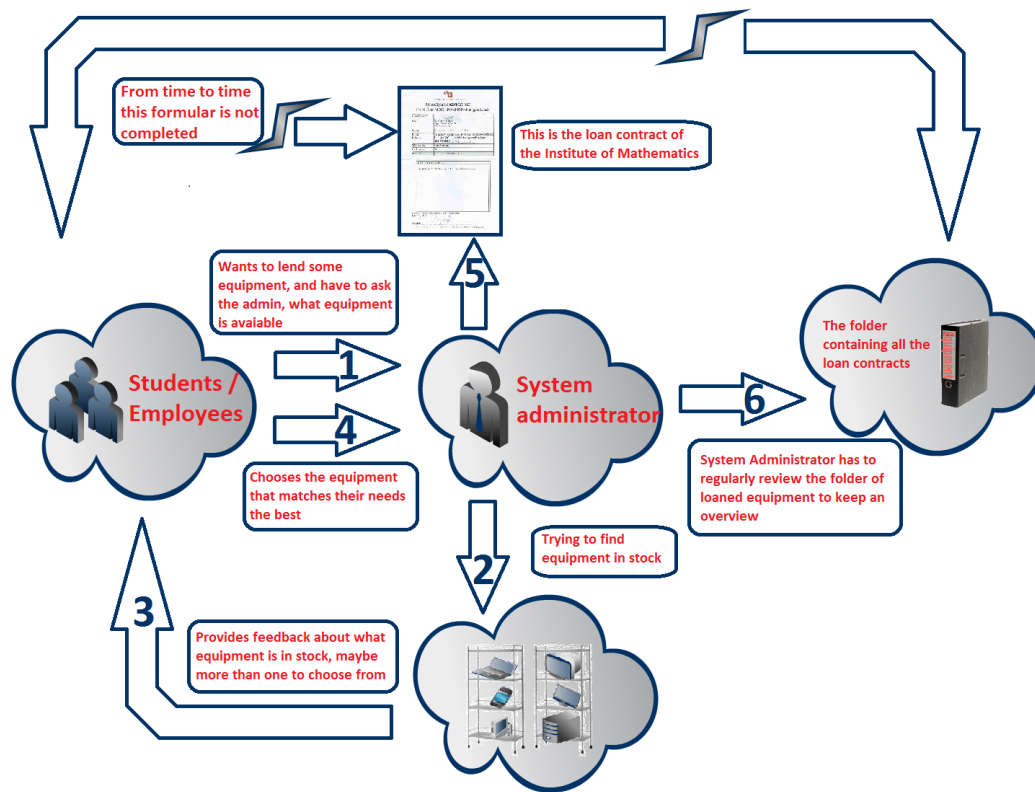


Figure 3.2: This picture describes the problemarea of the system. The lightning icons should be considered as possible problems in the problemarea. Two main concerns are identified. The first being that the lending forms not always are filled out and the second that the proces of lending equipment is not digital.

3.4 Problem area

The system must hold information about items that the institute possesses and different persons and locations. Figure 3.2 shows the problem area.

When new items arrive at the institute, it is the system administrators job to insert the information into the database. The only function of the database is that it holds information about the various hardware. The administrator can print a list of items to use for logistical purposes. The problem with this database is that it is old, unused, and only runs on a local machine.

When a student or employee asks if he or she can loan a piece of equipment, the administrator must first locate the equipment, then he must fill out a paper form that describes who or where has the item. The database does not hold any information about where the item is located. The system administrator must manually go through a folder with these paper forms to find out where it is. When the item is returned the form, if any, is discarded. The problem of the paper forms is that they are not consistent, that is, the administrator does not always remember fill them out.

3.5 Scope

The system must have certain functionalities, figure 3.3 describes these. These functions will be used by the System Administrator. When the administrator receives a new item he must enter it into the system. He must type in all the information related to the new item and then give it an AAU registration number, this number is unique for all equipment on the Aalborg University. Also he must take the invoice that came with the item and enter the related data into the system. For logistical purposes the administrator can then print a list of all items that are currently entered into the system and he must have the possibility to see detailed information about certain pieces of equipment. The Administrator must also have the possibility to edit and delete the equipment e.g if an item is lost or broken. Furthermore he must have the possibility to search the equipment in order to find exactly the piece of equipment that he is looking for.

Hold information about items
Hold information about where the items are located
Print list of items
Edit items
Search equipment
Lend equipment

Figure 3.3: *The most important tasks that should be implemented in the system.*

While logistics is one of the main concerns, the system must also hold a function that describes where the equipment currently is located. The items can be lent to a person or to a location. When a person asks if they can borrow an item, the administrator must access the system and register the loan. If the person is not currently in the system he must enter the persons information into it. In the case that an item is at a location but not rented to a person the administrator can also chose to lend the item to a location only. Figure 3.4 describes a use case for the system. The administrator can use the system full functionality of the system, and the students and employees that can see and request/book equipment from the system.

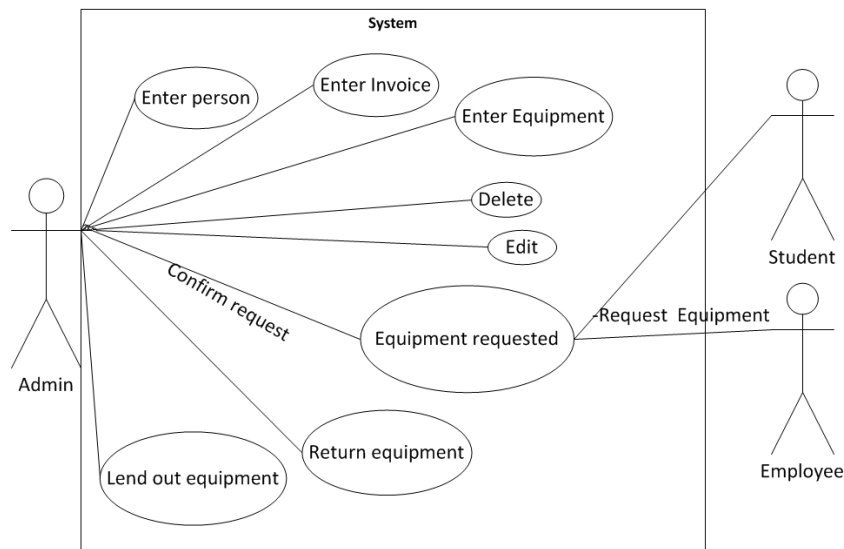


Figure 3.4: While the administrator can use all the functionalities of the system, students or employees can use the system to see what equipment that is in stock and there by book it.

3.6 Use

We have identified three user for the system. This first one is the administrator which will be the main user of the system but we also recognize that in the future could be used by students and employees but for now it will be used for administrative purposes. A use pattern for the administrator can be seen in figure 3.5.

Administrator	
Purpose:	The purpose for the administrator is to use the system for various purposes. This includes that he should be able to add, remove and lend out equipment. It is also important that he can print lists containing the equipment and all the information belonging to the equipment.
Characteristics:	The administrator has a good knowledge about computers and know how to operate it.
Example:	An administrative staff employed at the university who administrate all equipment at his faculty. He has worked and had the responsibility for this area in many years. He has been in the previous administration of the equipment and has a good understanding of how the system should work. He is an experienced PC Linux user.

Figure 3.5: *Definition of the use pattern for the administrator*

3.7 Functions

Based on the user stories, described in appendix A, the list of functions in figure 3.6 has been created.

Function	Complexity	Type
System login	Medium	Reading
Enter equipment	Medium	Update
Delete equipment	Medium	Update
Edit equipment attributes	Medium	Update
Search for equipment	Medium	Reading
Print list of equipment	Simple	Reading
List equipment details	Simple	Reading
Lend equipment to location	Medium	Update
Lend equipment to person	Medium	Update
Return equipment	Medium	Update
List booked equipment	Simple	Reading
Send return notice to loaners	Complex	Reading and calculation
Book equipment	Medium	Update
List equipment status	Simple	Reading
Cancel booking	Medium	Update
Repair	Medium	Update

Figure 3.6: *List of functions with complexity in the system and what type of system functionality it implements.*

The only complex function is the return notification, which will have to be automated to ensure timely notices, without relying on the administrator to be aware of upcoming return dates. This will involve automated checks of all equipment which is lent out.

All the functions listed as “simple” need to list tuples from the database, but does not involve any advanced functionality.

The functions listed as “medium” and “reading” need to do more advanced queries in the database, and involves more advanced functionality.

The functions listed as “medium” and “update” use relations between tables in the database, or need to alter existing data.

3.8 User interface

English is not the official language of the Institute of Mathematics. But due to several factors it has been chosen that English is the language of this web page (user interface). These reasons are, among others, that the project needs to be written in English, since the supervisor does not understand Danish.

3.8.1 Dialog form

Since the web page would be used as a work tool, it was chosen that it should be easy to navigate, and fairly easy to understand. Therefore a menu has been chosen to navigate through the different parts of the web page. When data has to be inserted, this is done via forms - A HTML wrap around tag for e.g several text fields.

The user interface should be kept in a simple/minimalistic design due to the fact that the web page is going to be used as a work tool. The menu contains links to the main functions of the web page these can be put into four different categories entering of data, viewing data, lend and return of equipment and searching. An overview which web pages should be implemented on the system can be seen in figure 3.7.

Page	Description
Login	Login screen
Home	Welcome screen
View all equipment	Shows all equipment + function to print all equipment
View all persons	Shows all persons in the system
Lend/repair equipment	Lends out equipment to either person or location
Return item from user	Returns items from persons
Return item from location	Returns items from location
Return item from repair	Returns items from repair
Log	Shows a log with a overview of changes made in the system
Search specific	Used to search specific data
Search all	Used to search after text in all data

Figure 3.7: *The different pages with descriptions*

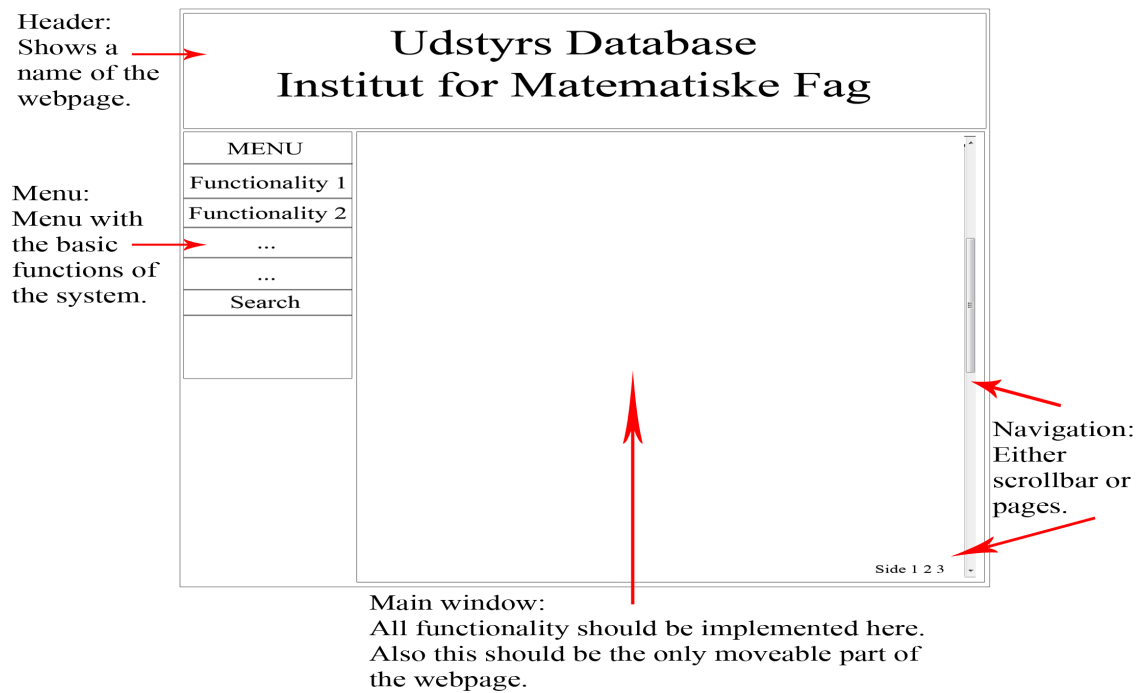


Figure 3.8: *The customers design idea*

3.8.2 Overview

The System Administrator had an idea of how he wanted the basic look of the web page the rest was up to the development team. The idea/drawing can be seen in figure 3.8.

3.8.3 The platform

The web page is going to be programmed in a mix of HTML, CSS and PHP, also it is going to use JavaScripts for some functionalities. The database is created using MySQL. The system will be built to run on an Apache server. The web page should support the most common web browsers, but most importantly it should support the Firefox [2010] since this is the primary browser that the system administrator uses.

3.8.4 The usefulness and realizability of the system

The functions of the system of been identified through an interview with the system administrator at the Institute of Mathematics (the customer), these will be reviewed further during the project. This is done because he has existing system and he knows what kind of information he would like to have in the database. Also the administrator was given an task to draw a picture of the user interface of the system he wanted.

The project team has various programming skills, but few has experience with HTML, CSS, PHP, databases and Java Scripts. So it is a bit uncertain if it is possible to implement all the functionality because there will be a learning curve for the different languages. Despite this learning curve it is estimated that a working prototype or releasable system will be possible to make during the span of the project. It should also be possible to expand the system on the chosen platform without much reconstruction.

3.8.5 Strategy

The system administrator needs a new database to keep track of his hardware since his old database was outdated so there was a strong inclination from both customer and development team. The development team works with Scrum Larman [2004] which will give possibility to evaluate progress with the customer. Also when using Scrum after each sprint there should be a working product. The system will be implemented using a program called Netbeans IDE 6.8NetBeans [2010] for the web page part and PHPmyAdminphpMyAdmin [2010] for manipulating the database.

It is estimated that the programming and implementation will take approximately four weeks. The total project would take from two to four months to complete.

4.1 Introduction

In the ending of the analysis 3.8.3, the basic for the homepage layout was set. But the underlying system, e.g. the database and PHP connection structure have yet to be defined. In this section both the architecture and the physical design of our implementation will be defined. The structure of the homepage includes navigation diagrams and screenshots of the different pages. This section also includes a description of the systems security settings.

4.2 Design quality

Before designing a system one must identify the overall goal of the system. What criteria does the system need to fulfil? Figure 4.1 shows how we ranked the design criteria. The table is designed to determine which values should require most attention, less attention, a little attention or no attention at all.

The criteria are derived from the customers opinions and are prioritized as well. The focus will primarily be on the administrative functionality. Concepts like usefulness, security, effectiveness and reliability are needed to give the customer the opportunity to work more efficient.

Things like correctness, easy to use, memorability are not prioritized as high because the only one using the system would be the administrator. If we were to include students or teachers into the system, these things would probably be prioritized higher. Also if the system were to have higher modularity, it would be good to prioritize this higher. Though the system will be placed locally on a server, one should be able to access it via the Internet. Therefore the mobility of the system is important.

Testing of the system should be considered important, in order to catch any mistakes

Criteria	Very Important	Important	Less Important	Irrelevant
Useful	X			
Secure	X			
Effective	X			
Correct		X		
Reliable	X			
Maintenance				X
Testing		X		
Flexibility				X
Understandable		X		
			X	
Mobility		X		
Integrate			X	
Easy to use		X		
Modular		X		
Back ups				X
Memorability		X		
Utility				X
Satisfying				X
Enjoyable				X
Motivating				X

Figure 4.1: *This model describes how the design criteria are prioritized. Notice that useful, secure, effective and reliable are prioritized high because the system is primarily going to be used as an administrative tool.*

that we might have made. The irrelevant parts of the criteria schema is not going to be discussed further besides the maintenance point. Since this is a one time delivery there is not going to be any maintenance.

4.3 Architecture

No architecture has been chosen though the different components should be divided into different parts. The first part is the HTML code, this is what the user can see and does not contain any vital information regarding database calls and so on. The second part works on the principle that vital information must be hidden from the user. This part contains information regarding the database name, user name and passwords and so on.

The philosophy regarding our chosen architecture is, that by keeping the user from being able to see any important code, there must be a some sort of code that lies behind what the user have access to. This code controls the input data and sends the data to the database. The user interface cannot communicate directly with the database, and must therefore send the data to the hidden component, that communicates with the database. The database then sends the data to the user interface, updating it for the user to see. As the PHP scripts are used at the server side, not all the information is send to the client. A model of the architecture can be seen in figure 4.2.

The system is implemented on an Apache Web Server version 2.2, that is running on an Ubuntu Server 9.10 Linux based machine. The Web Server provides access to the user interface from the local network. The Apache server also handles information floating between the user interface and the database.

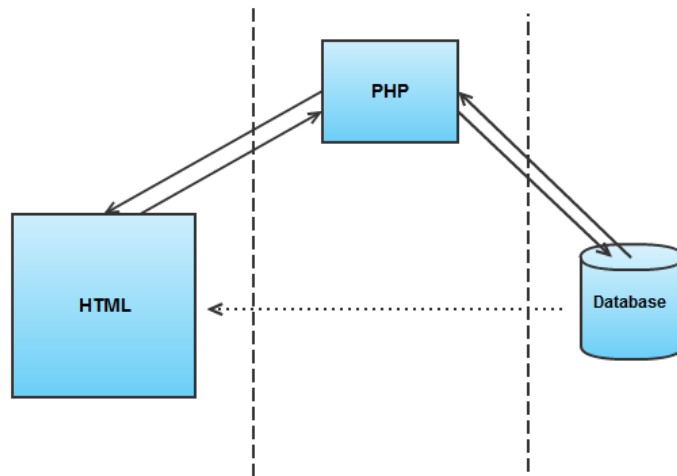


Figure 4.2: *The architecture model. The PHP scripts only runs on the server and can therefore not be seen by the user. The PHP sends information to the database and vice versa. The database can then send information to the HTML part and there by updates the information displayed on the screen.*

Since the system will be running on Linux systems, the choice has been made to make it specific to the standard browser running on them e.g. Firefox. This means that some other browsers may not be able to interpret the system exactly in the same way. The server hardware consists of a dual-core Intel Pentium E2140 Conroe processor with 2 GB ram.

The database has been implemented using MySQLMySQL [2010] and phpMyAdminphpMyAdmin [2010].

Standards

The system does not follow any standards regarding design. The feedback from the customer, see section 3.8.2, has however influenced the design.

4.4 Database design

4.4.1 Conceptual design

In software engineering ER or EER diagrams are viewed as conceptual representation of a data set Chen [1976]. Figure 4.3 shows the database structure expressed as an EER diagram Wikipedia [2010a]. Starting in the center of the diagram we find the entity “Item”. This is the entity which the whole system is centered upon. “Item” has 5 attributes in which one is a primary key, “AAU_Reg_No”. This key symbolizes the AAU registration number, which is present on almost all hardware found on the university. The item must also hold information about the “Type_ID” (what kind of item it is, e.g., software, hardware.). The “App_Index_No” (stands short for Apparatus index number, that is a number in which all item classes are given, e.g. a PC-Card has the number 2545). This number is not unique though. The item must also have a “date of purchase” and a “status” (can be that the item is lent out, in stock or sent to repair.)

The given item could be a number of things, though the diagram only displays hardware as a generalization. The item could also be software, furniture or other (Misc). Note that the last three are not present in the diagram. They could potentially share the attributes of the superclass “item” but holds some individual attributes. Why not just give hardware, software and so on the same attributes? The problem is that though they share some attributes, some of the individual attributes are only bound to a specific piece of hardware, software etc. If we give all the the entities the same attributes we would experience a lot of NULL values. This is certainly not desirable seen from a database performance view and therefore they became subclasses.

If we take a look at the “Hardware” entity, it must have some information regarding what kind of hardware it is. A “HW_ID” is needed to say what kind of hardware it is e.g., screen or a computer. Also it must have a serial number, a model number and a manufacturer. The comment attribute holds the specifications of the hardware e.g. that a memory module has 256 MB of ram and so on. Last but not least, the AAU registration number is needed to join “Hardware” with “Item”.

“Item” has three relations to three entities. The first relation is called “Belongs_to”, and connects “Item” and “Invoice”, because an invoice belongs to an item. The “Invoice” entity is representing an invoice, which holds 5 attributes. When the system administrator receives an item he also receives an invoice. This invoice has an invoice number, supplier and date information.

Furthermore the primary key of invoice is “Internal_Index_No” which is used to uniquely identify each different invoice. The reason why the invoice number is not used as a key, is that different suppliers might have the same numbers and therefore it would not be unique. When the information is entered the administrator is able to scan the invoice into the system and upload it. The Ref_Scanned_PDF attribute is a container that holds a path to the scanning of the invoice. The relation is at the left side an (1,n) that is, a given invoice can hold several items. At the right side it is an (0,1) relation that is, an item does not necessarily have to have an invoice.

When the administrator lends an item to a person he must enter the relevant information about a person or a location. A person has of course a name, a phone number and an e-mail address. A person is identified by a user name. Though this user name is unique for all students a problem might occur if an employee also wants to lend an item. Theoretically the same user name might occur both as a student and as an employee. Therefore the key is combined with the association attribute so that the key is uniquely identifiable.

The relation is on the left side a (0,n) relation, that is a person can rent zero to several items. The right side of the relation is a (1,1) this mean, that if an item is rented then it can only be to one person.

Every person has a location but it has no location attribute. That is because a "location" entity is linked to a "person" by the "item" entity and can be accessed by the "AAU_Reg_No" attribute.

The third relation from "Item" is a "Has_A" relation, This relation links the "Item" and the "Location" entities. If the administrator wants to register an item that is placed at a location he must fill in the relevant information for this location. This includes the address, the room number and the institute. Given that different institutes have the same room numbers and possible the same room numbers the key is a combination of institute and room number to make it uniquely identifiable. Also there is a cluster specification, that would be registered under room number so that the number for cluster 3 room number 413 would be 3.413. The left side of the relation is a (0,n) that is, a location can hold as many items as possible. The right side the relation is a (1,1) that is, that an item can only have only one location.

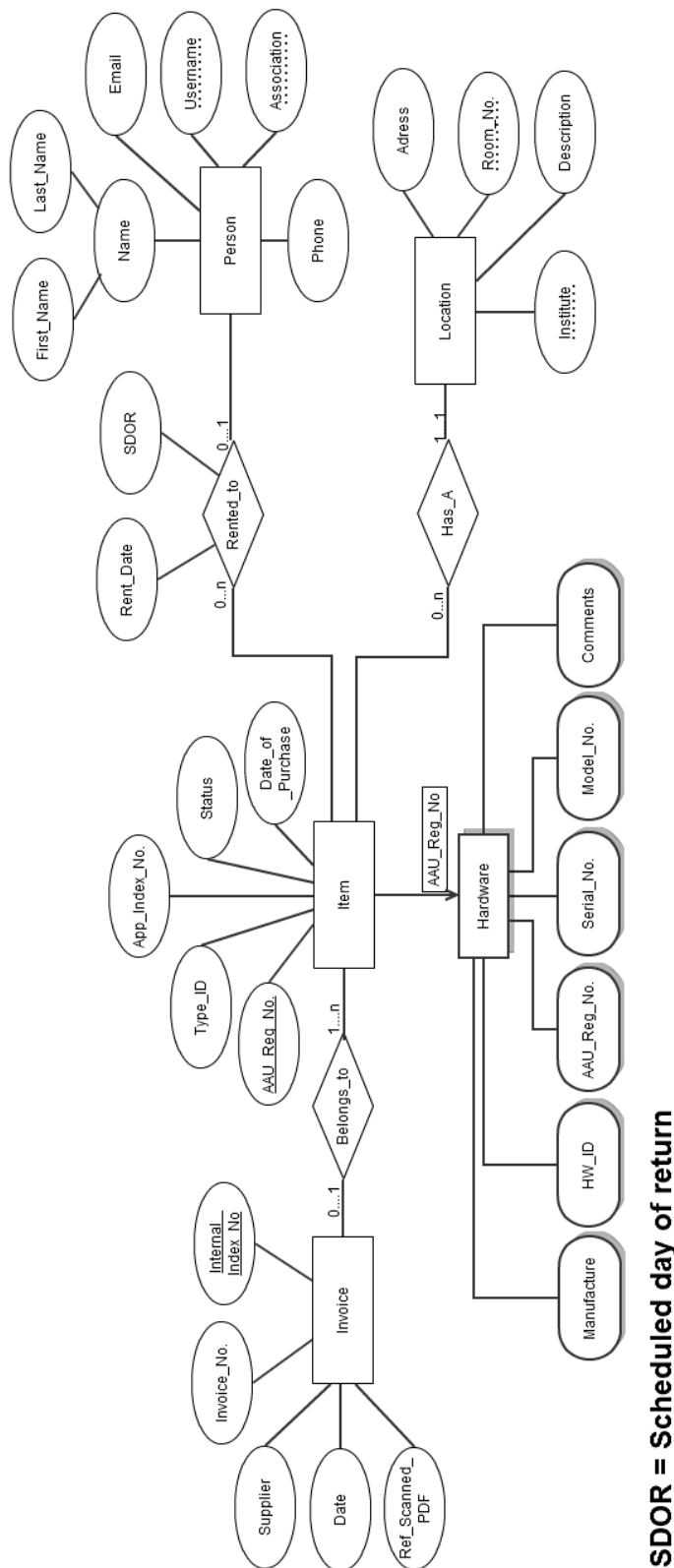


Figure 4.3: The EER diagram describes the conceptual design of the database

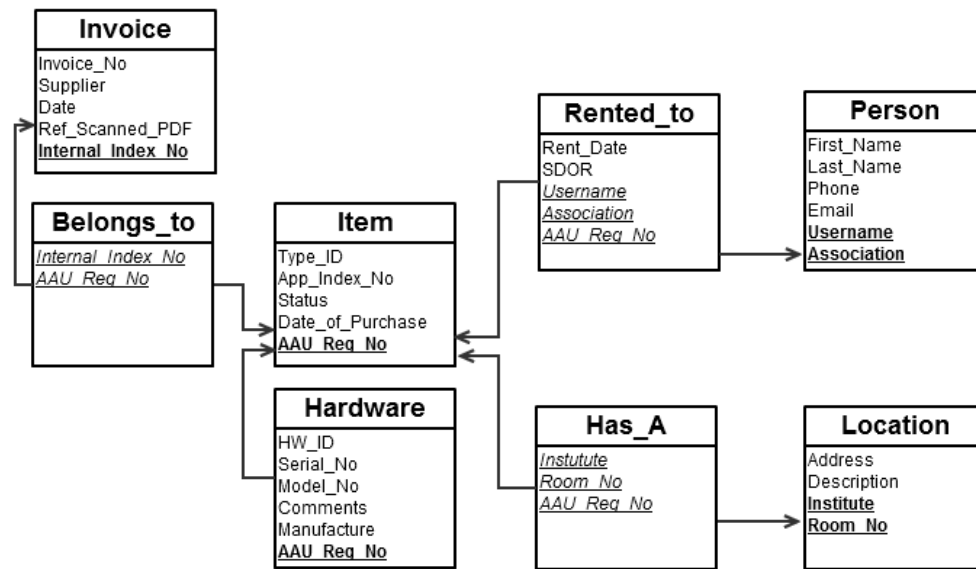


Figure 4.4: Logical structure of the database. Shows tables in the database with their respective attributes and keys. Keys are bold and underlined, Foreign keys are italic and underlined.

4.4.2 Logical design

The logical design describes how the data from the conceptual design is made into a logical structure, for example data storing object like tables. The database holds some tables. Each entity from the EER diagram was made into a table with their respective attributes. Also each relation has been made into a table. The bold and underlined character represents primary keys and the italic character represents foreign keys. An important thing to notice about this table is that instead of the intended relation between hardware and item (The EER diagram showed an inheritance), this is now a 1 to 1 relation. So there is really no point keeping the tables separated. This is done on purpose in order to be able to expand the system with other item types. Although there is no relation table between “Item” and “Hardware” at the moment, one is needed for the database to hold other types of equipment than just hardware, this is through an unimplemented feature. The Logical design is shown in figure 4.4. The tables are explained further in the following section.

Database tables

The following section describes the tables in the database with their respective purpose and attributes. Further explanation can be seen in section 4.4.1. Underlined attributes are keys.

Item	
Purpose:	Holds the information about general data for an item.
Attributes:	<u>Type_ID</u> , <u>App_Index_No</u> , <u>Status</u> , <u>Date_of_Purchase</u> , <u>AAU_Reg_No</u>
Operations:	Enter equipment, Lend equipment

Hardware	
Purpose:	Holds more specific information about a piece of hardware. Specialization of “Item”.
Attributes:	<u>Hw_ID</u> , <u>Serial_No</u> , <u>Model_No</u> , <u>Comments</u> , <u>AAU_Reg_No</u> , <u>Manufacturer</u>
Operations:	Enter equipment, Lend equipment

Invoice	
Purpose:	Holds information about an invoice. An invoice can be related to an item.
Attributes:	<u>Invoice_No</u> , <u>Supplier</u> , <u>Date</u> , <u>Ref_Scanned_PDF</u> , <u>Internal_Index_No</u>
Operations:	Enter invoice

Person	
Purpose:	Holds information about a person. Can be related to an item.
Attributes:	<u>First_Name</u> , <u>Last_Name</u> , <u>Phone</u> , <u>Email</u> , <u>Username</u> , <u>Association</u> (Can be student or employee)
Operations:	Enter person, Lend equipment

Location

Purpose: Holds information about a location. Can be related to an item.
Attributes: Address, Institute, Room_No, Description
Operations: Lend equipment

Rented_to

Purpose: Relation that holds information about a loan. Is related to both an item and a person.
Attributes: Rent_Date, SDOR(Scheduled day of return), AAU_Reg_No, Username, Association(Can be student or employee)
Operations: Lend equipment

Has_A

Purpose: Relation that holds information about a loan. Is related to both an item and a location.
Attributes: AAU_Reg_No, Institute, Room_No
Operations: Enter equipment

Belongs_to

Purpose: Relation that holds information about which invoice an item belongs to. Is related to both an item and an invoice.
Attributes: Internal_Index_No, AAU_Reg_No
Operations: Enter equipment

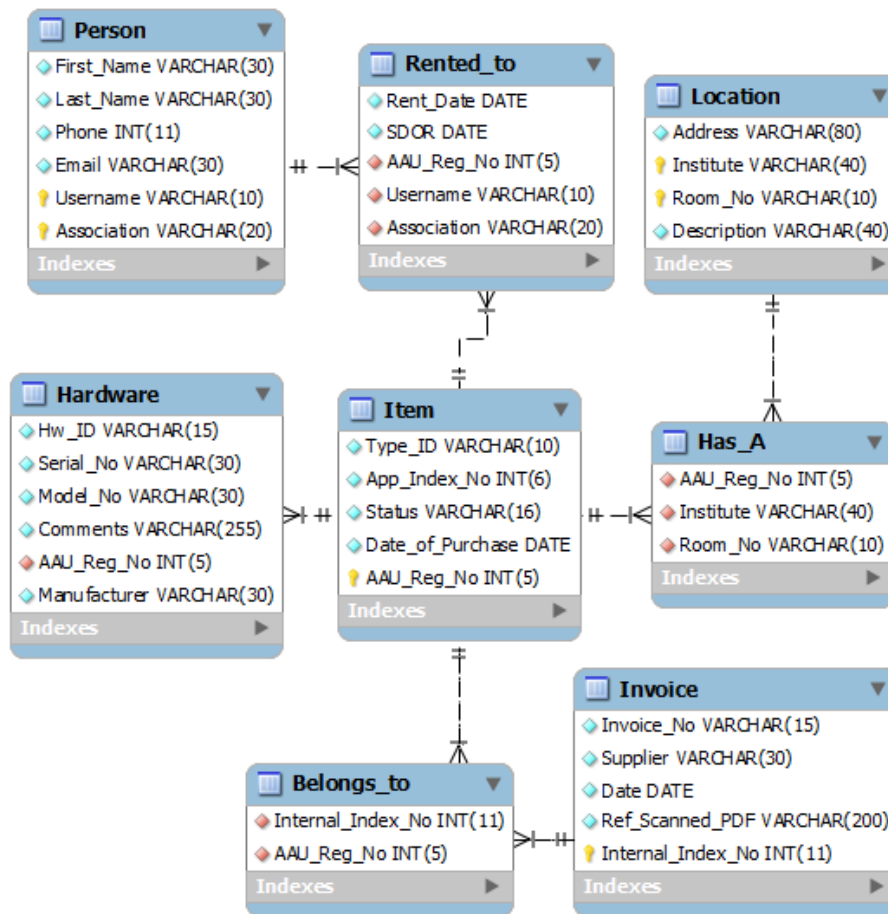


Figure 4.5: *Physical structure of the database. Shows the tables, attributes, indexes and constraints.*

4.4.3 Physical design

The physical design is a specification of each data element with their data types and indexes. The physical design is the last step of defining a database. Below a list of each table is provided with their attributes, keys and their constraints. From the physical design the database “booking system“ has been created.

Each of the tables in the database holds attributes. Each attribute has a data type. These data types are described in figure 4.5. Further description of each attribute can be found in appendix C.

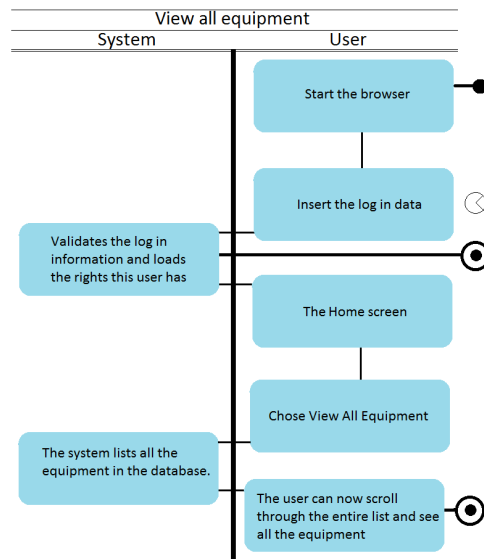


Figure 4.6: *This model describes the interaction with the system when the administrator wants to lend an item to a person. The bold black line in the middle is to separate the actions of the user and the system. The black dot indicates where the starting point of a session begins. The starting point in this case is that the user needs to log in to use the system, notice the “Pac man” in the right side of the figure, this signals the user filling in some sort of data into the system.*

4.5 System design

4.5.1 Interaction

The system must be able to handle multiple data input from the user. The input is handled by the system, or more specifically its sent to PHP scripts that sends it to the database. The following section will describe when the system handles the situation and when the user handles the situation. This is done via interaction diagrams.

The first diagram 4.6 is based on a situation where the user wants to look at all equipment in the database. The diagram explains what is done by the system and what is being done by the user.

When a user clicks the “log in” button the system switches to the “system” side of the model, this is where the validation is done. If the system accepts the user name and password, the user continues. Alternatively the session ends and he must start over. The user can then proceed to choose the “view all” tab in the menu bar. The system runs the request and returns a list of equipment for the user to see, as illustrated in figure 4.6.

We have two examples, the first we have already explained, the next will not be explained as thorough, but is a good example of a session where the user interacts even more with the system. It can be seen in figure 4.7.

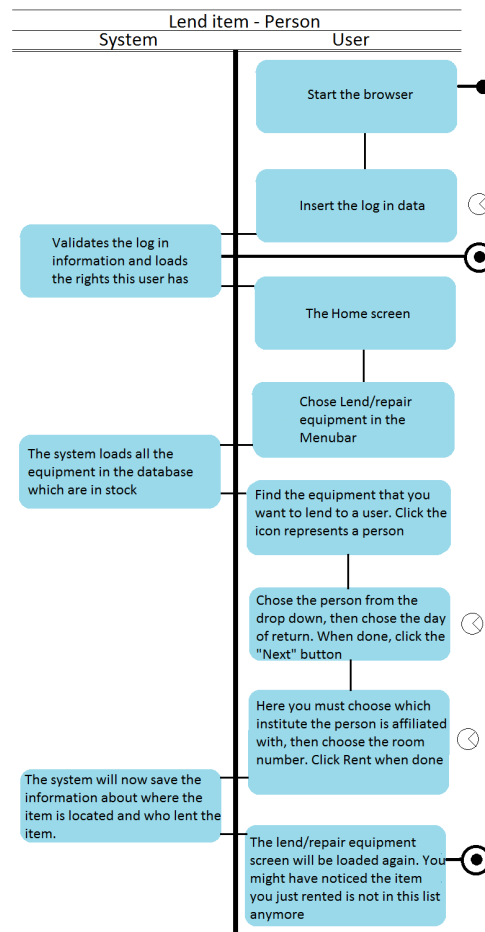


Figure 4.7: This model describes the interaction with the system when the administrator wants to lend an item to a person.

4.5.2 Connecting to the database

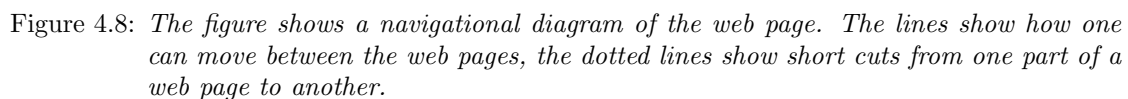
Basically we connect to the database every time we need to see, add or delete anything. An example of such a connection to the database can be seen in section 4.5.1. We have put the connection string, containing the IP address, user name and password for our database in another file. This file is called `bookingsystem_dao.php` which can be found on the reference CD, see appendix G. All other scripts mentioned in this section can also be found there. Every time we need to make a change in the database we include this file in a PHP include statement. If we did not do this we would have to write the connection string every time we made a database call. That way the login info about the database would be in almost every file in the system, and it would be a potential security risk. Say if someone for example out commented the connection string for testing purposes and forgot to uncomment the line. This way any person could see the information when watching the source code. Ideally we would have all our different queries in a file for

themselves, and then call the different functions instead of writing the queries in the different PHP files. This is tried in the `view_all_eq.php` file which includes a file called `queryscript.php`. This is not critical for the system, although it would be a nice feature to implement later in the development for security and code consistency purposes.

4.5.3 Navigation

This section will show how the web page is build. To show this a navigational diagram has been build, see figure 4.8, that shows the main pages of the website and how these are (inter)connected.

- **A:** Shows the login screen of the web page, you have to be logged in order to use the web page.
- **1:** Shows the welcome screen of the web page.
- **2:** Shows the data entering screen from here one can choose which data one wants to enter, these are hardware, invoice and person.
- **3:** Shows all the equipment that is entered in the system, these can be manipulated with delete, edit and the details can be shown.
- **4:** Shows all persons who has been entered into the system, these can be either deleted or edited. Also if the user has Skype, one can click the hyper link telephone number to call this person.
- **5:** Shows the lend and repair screen from here one can lend to either person or location and send the item to repair.
- **6:** Shows the return equipment from user screen here one can change the status from lend out to person to stock, by returning the item.
- **7:** Shows the return equipment from location screen here one can change the status from lend out to location to stock, by returning the item.
- **8:** Shows the return equipment from repair screen here one can change the status from repair to stock, by returning the item.
- **9:** Shows the log screen, here one can see lend out and return information. The log can also be cleared.
- **10:** Shows the search specific screen here one can choose to search in hardware, invoice, person and location.



4.5.4 Design

This section will describe how the various parts of the system, is put together visually, and why we chose to do it this way. We did the best we could within our abilities to keep consistency throughout the system, and we have thought of options and features, which we hope will make it easier to use the system.

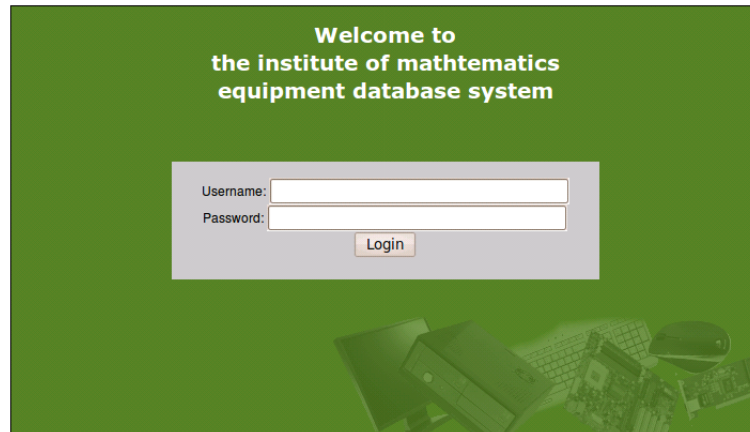


Figure 4.9: *This is the Log in screen. As you can see, it contains 2 input fields and a “Log in” button.*

Let us start by looking at the first screen, as seen in figure 4.9. There is not much to see, only the login screen. When you use the login system, the system itself will recognize which user type logs in. The existing user types at the moment are the administrator and users. As we wrote in some of the previous sections of the rapport, we chose to focus on the administrator part, so the user part has not been build. The user part will be used in this section to illustrate our basic idea for some of the functions the user should have in a later version. But as you can see, this screen is very simple, with two fields for input of data and a submit button that allows you to login.

If the user name and password do not match, the following message will show up in the browser:

Sorry, the username or password were incorrect.
Please try again. [click here to log in again](#)

When you are logged in the first thing you will see, is the “Home” page or the main page. There are two different home pages. One is for the Users and the other is for administrators. First of all, this section will show how the general system is put together, what should be expected to be consistent, and how to navigate in the system. Then we will go into more detail with a few screen dumps. The main reason that the login screen was described before this paragraph, is that it stands out from the rest of the system design.

The design is built up around three main elements. We have Top Bar which do not change at any given time throughout the system, it is the one marked with black frame.

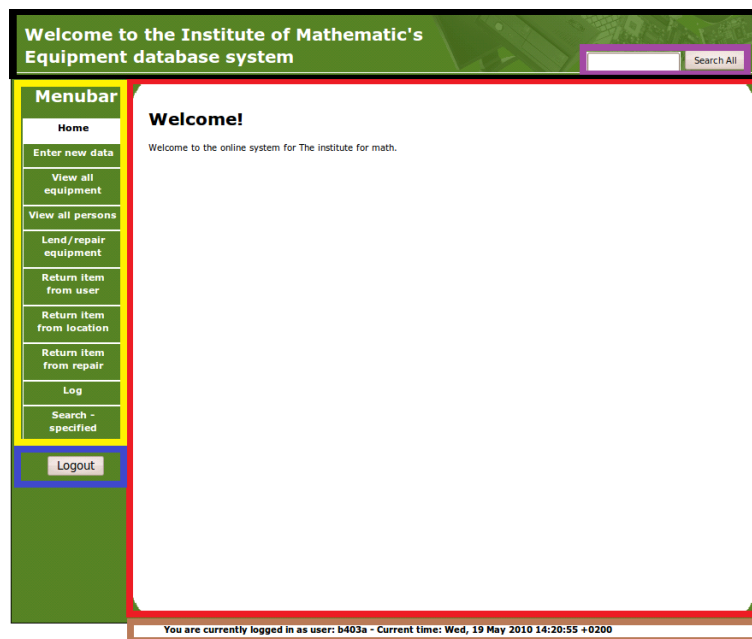


Figure 4.10: *This is the Home Screen with colored frames all around.*

There is always the same brief welcome, and the name of the system. We also have our “Search All” function in this top bar. The function is marked with purple frame. Search all does what the text says, it searches through the database for the string you put in the text field and lists the results.

To the left is a menu bar, this is marked with a yellow frame in figure 4.10. This is also consistent throughout the system. It appears the same place, so it is easy to navigate through the system. When the mouse is hovering over one of the fields, it will turn the field white and text black, so you do not have any doubts about what you are about to click. Below the menu bar is a Logout button, it is marked with the blue frame, also in figure 4.10. This is used if you want to log out of the system.

Within the big red frame the features will be loaded. The screen shot being used is the welcoming screen at the Home section. However, as seen on for example at the screen shots in figure 4.12 and 4.13, it is within this area functions are loaded into.

Below the function form, is a short info about which user one is logged in as. In addition, there will be some info about the date and time when you last updated the page you are currently looking at. This field is also consistent throughout the system.

If we compare the screen dumps, you see depending on whether you are logged into the system as administrator or normal user, you are given different privileges. We had to consider which functions would benefit the various user types. To the left is the administrator screen, and to the right is what the ordinary user can see. Both sides are built up equally, but the difference lies in what features users are entitled to. The user can only see all the equipment and search for specific equipment. The administrator,

however, may use all functions which are already mentioned in the report. Therefore, the menu bar list is smaller for the user than the one the administrator gets, but apart from that there is no difference in the visual design.

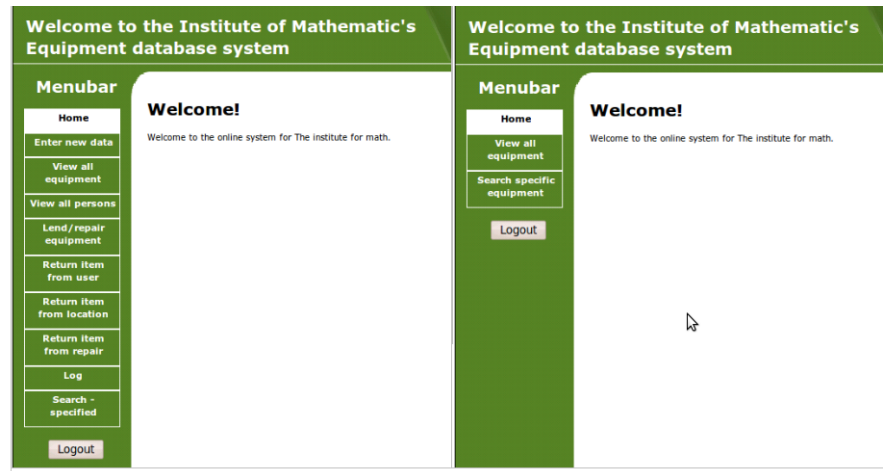


Figure 4.11: These two screens show the home screen for the Administrator and the User.

Another notable feature is the “enter new data” in the Menu Bar. When you enter the page, a form that only contain the text “What do you want to enter” and then you have to choose what to enter from a drop-down list. You can choose between: “Enter new hardware”, “Enter New Person” or “Enter New Invoice”. When you choose one of them, another form will be loaded. When you have selected what you want to enter, the text will change to “You are about to enter,” and in the drop-down list the text will be the form you told the system to load. In this example we are about to enter “A new piece of Hardware”. As you can see, the drop-down list shows that this was what we had chosen. In the field beneath it is possible to use previously stored data by entering an “AAU Reg No” and pressing “Load Data” button. Otherwise, this page consists only of text fields where you can enter the data being requested. At the bottom it is possible to reset all data that is typed into the form by clicking the “Reset formula” button, or if you are satisfied with the data, you can add the equipment by clicking the “Add Item”.

Welcome to the Institute of Mathematic's Equipment database system

Menubar

- Home
- Enter new data
- View all equipment
- View all persons
- Lend/repair equipment
- Return item from user
- Return item from location
- Return item from repair
- Log
- Search - specified
-

You are about to enter:

A new piece of hardware ▾

Load existing hardware data by AAU registration number(optional):

AAU registration number:	<input type="text"/>
Apparatus index number:	<input type="text"/>
Manufacturer:	<input type="text"/>
Hardware type:	<input type="text"/>
Serial number:	<input type="text"/>
Model number:	<input type="text"/>
Date of purchase:	19 ▾ / 05 ▾ / 2010 ▾
Comments:	<input type="text"/>
Add item to invoice number:	Choose ▾

You are currently logged in as user: b403a - Current time: Wed, 19 May 2010 14:24:14 +0200

Figure 4.12: The figure shows the “Enter new hardware” screen. As you see, there is an option to load data. Beside that there are some fields to fill out. The data that needs to be inserted is described to the left of the input fields. At the bottom there are two buttons, one to reset all the fields and one to add these data to the database.

Another place where you can highlight some features would be in View All Equipment. Here you will, once again, be reminded about where in the system you are. There are also two buttons which are designed as shortcuts to other functions in the system. The “Enter New Equipment” button will lead the user to the form where you enter the new hardware, the form that we have shown in the previous screen dump. The second button is to print the entire list. This feature was inspired by the list that was given to us by the customer early in the development process. This list had all the equipment that the customer had in the system, so we would make it easier for him to be able to print it whenever he needs it.

Welcome to the Institute of Mathematic's Equipment database system

Menubar

- Home
- Enter new data
- View all equipment
- View all persons
- Lend/repair equipment
- Return item from user
- Return item from location
- Return item from repair
- Log
- Search - specified
- Logout

All Equipment

Enter new equipment Print

AAU res. no.	Hardware type	Manufacturer	Model no.	Room no.	Status	Details	Edit	Delete
1	Laptop	HP	Pavilion dv7-3117eo	1-101	lent_to_person			
2	Laptop	HP	Pavilion dv7-3117eo	1-110	lent_to_person			
3	Laptop	HP	Pavilion dv7-3117eo	1-110	lent_to_location			
4	Laptop	HP	Pavilion dv7-3117eo	1-115	lent_to_person			
5	Laptop	HP	Pavilion dv7-3117eo	4-117	lent_to_person			
6	Laptop	HP	Pavilion dv7-3117eo	1-101	lent_to_location			
7	Laptop	HP	Pavilion dv7-3117eo	1-104	lent_to_location			
8	Laptop	HP	Pavilion dv7-3117eo		stock			
9	Laptop	HP	Pavilion dv7-3117eo		stock			
10	Laptop	HP	Pavilion dv7-3117eo		stock			
11	Laptop	HP	Pavilion dv7-3117eo		repair			
12	Laptop	HP	Pavilion dv7-3117eo		stock			
13	Laptop	HP	Pavilion dv7-3117eo		stock			

You are currently logged in as user: b403a - Current time: Wed, 19 May 2010 14:35:01 +0200

Figure 4.13: The “All Equipment” tab shows all equipment in the database. You get an option to enter new equipment or print the list. Beside that you can see the details, edit or delete equipment from the list, and you can list the equipment as you like via the hyperlinks at the top of the table.

But the most important thing in this form is the listed tables with the most important information about the different equipment. You can choose to list the equipment as you wish by clicking on the hyper linked text at the top of the table. To the right of data are three icons you can click on. The first will lead the user to a page where all details of this equipment can be found. Next to the details icon there is a pencil icon, this will let you edit data about this piece of equipment. Finally, there is a delete function; this icon is a red circle with a white X in it. This will delete the item from the system.

In comparison you can see that the normal users also have this “View all Equipment” function, so they can browse through all the equipment in the system. But one does not have the rights to edit the equipment or to delete them.

Lend / Repair Equipment also has some features which are not in any of the previous

Welcome to the Institute of Mathematic's Equipment database system

Menubar

- Home
- View all equipment
- Search specific equipment
- Logout

All Equipment

Enter new equipment Print

AAU res. no.	Hardware type	Manufacturer	Model no.	Room no.	Status	Details Edit Delete
1	Laptop	HP	Pavilion dv7-3117eo	1-101	lent_to_person	[person] [house] [hammer]
2	Laptop	HP	Pavilion dv7-3117eo	1-110	lent_to_person	[person] [house] [hammer]
3	Laptop	HP	Pavilion dv7-3117eo	1-110	lent_to_location	[person] [house] [hammer]
4	Laptop	HP	Pavilion dv7-3117eo	1-115	lent_to_person	[person] [house] [hammer]
5	Laptop	HP	Pavilion dv7-3117eo	4-117	lent_to_person	[person] [house] [hammer]
6	Laptop	HP	Pavilion dv7-3117eo	1-101	lent_to_location	[person] [house] [hammer]
7	Laptop	HP	Pavilion dv7-3117eo	1-104	lent_to_location	[person] [house] [hammer]
8	Laptop	HP	Pavilion dv7-3117eo		stock	[person] [house] [hammer]
9	Laptop	HP	Pavilion dv7-3117eo		stock	[person] [house] [hammer]
10	Laptop	HP	Pavilion dv7-3117eo		stock	[person] [house] [hammer]
11	Laptop	HP	Pavilion dv7-3117eo		repair	[person] [house] [hammer]
12	Laptop	HP	Pavilion dv7-3117eo		stock	[person] [house] [hammer]

You are currently logged in as user: johndoe - Current time: Wed, 19 May 2010 16:39:58 +0200

Figure 4.14: Comparison of user and administrator rights. As you can see, the user can not edit or delete equipment.

screen dumps. As you can see, we once again have a table in this form. But this table only contains all the equipment that is in stock. To the right side you can see three icons. A person, a house and a hammer. These three icons indicate who or where you choose to lend the equipment to. If you choose to lend to a person, another form will load where you type in some information about the person, and the system will then know who has borrowed the equipment. Location has the same lending function, but you will have to enter a location in place of personal data. Last but not least, we can send the equipment for repair if it is broken.

In the screen shot to the right shows how to return an item. You have to choose from the menu bar whether you must return the equipment from a person, location or repair. Here we have selected location, and all equipment that is loaned to a location will be listed here. To tell the system that it is returned, simply click the blue arrow on the right side.

The last part that will be shown and explained, are the search functions. These will be explained and described, including how we wish the users to experience them. We have got a Specific Search and a Search All function, where we will explain the search results.

As previously mentioned, the function Search All can be found in the top bar. This feature is a quick search function that will search through all data in the entire database. Therefore, this also shows much useless information and if the system is very large, this function also takes long to complete the search. We have therefore chosen to make a specific search function as well.

Search Specified allows you to cut down search time and the idea is that you should

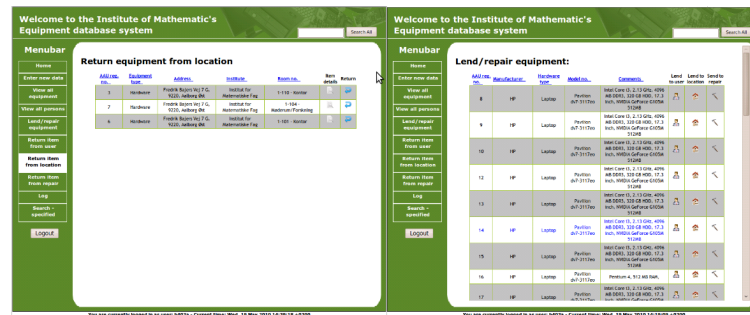


Figure 4.15: The left side of the figure shows the return screen. This is rather simple. Click the blue arrow and the item is returned. The right side shows the function to lend out the equipment. An icon is used to represent the Person, Location and Repair. By clicking on the icon, you will get another form to input more details.

get more useful search results. Once you have clicked on Search Specified in Menu bar, you will load a page which again is very clean and only the most necessary is shown. A text asks “What you want to search for?” and a Drop-Down list shows what one can search for. In the example being shown here you will find a piece of hardware. It is possible to search for various information you have about his piece of hardware.

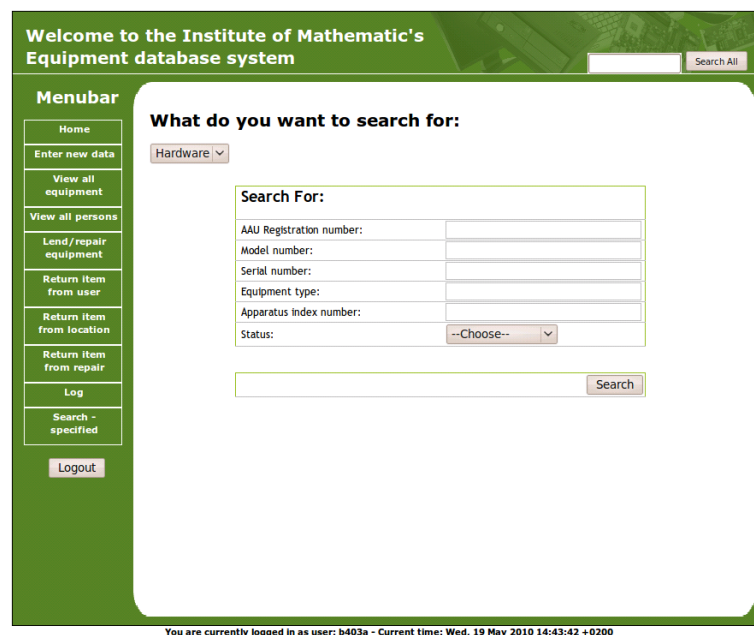
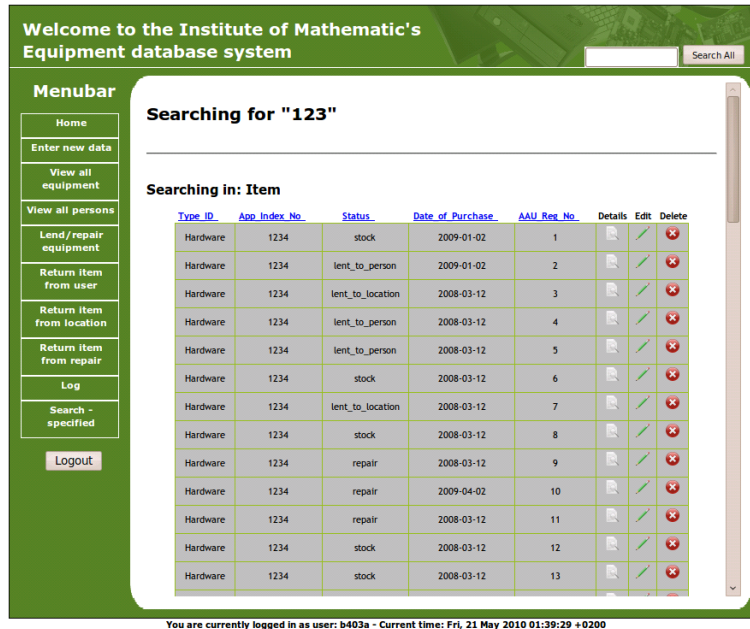


Figure 4.16: Search Specified equipment lets you search for equipment that fulfills the desired data. A model number and a serial number might look a lot alike, but by searching specified you will only get the information that live up to the requirement.

Both Search all and Search specified shows the results in the same way. Here you can both choose to edit and delete individual equipment. Again, you can sort the results as desired by clicking on hyperlinks.



Welcome to the Institute of Mathematic's Equipment database system

Search All

Menubar

- Home
- Enter new data
- View all equipment
- View all persons
- Lend/repair equipment
- Return item from user
- Return item from location
- Return item from repair
- Log
- Search - specified
- Logout

Searching for "123"

Searching in: Item

Type ID	App Index No	Status	Date of Purchase	AAU Reg No	Details	Edit	Delete
Hardware	1234	stock	2009-01-02	1			
Hardware	1234	lent_to_person	2009-01-02	2			
Hardware	1234	lent_to_location	2008-03-12	3			
Hardware	1234	lent_to_person	2008-03-12	4			
Hardware	1234	lent_to_person	2008-03-12	5			
Hardware	1234	stock	2008-03-12	6			
Hardware	1234	lent_to_location	2008-03-12	7			
Hardware	1234	stock	2008-03-12	8			
Hardware	1234	repair	2008-03-12	9			
Hardware	1234	repair	2009-04-02	10			
Hardware	1234	repair	2008-03-12	11			
Hardware	1234	stock	2008-03-12	12			
Hardware	1234	stock	2008-03-12	13			

You are currently logged in as user: b403a - Current time: Fri, 21 May 2010 01:39:29 +0200

Figure 4.17: Here you see the search function, search the whole database for data that matches your search.

4.5.5 Security

This section aims to describe the login screen and its underlying code and security features. So why is this even worth mentioning since the system runs on the math department's intranet. Well, since the system would probably have to fetch sensitive information from the university's user database, in this case, the security risk plays an essential role. So in short terms, since the system potentially could hold sensitive data, this project simply cannot ignore the security aspect.

Login Screen

There is nothing extraordinary about the login screen, one can enter user name and password. Handling this information however, needs a lot more consideration. The login screen itself does not follow the same design as the rest of the system. This is due to the significant change in functionality. Now let us take a closer look on how the system registers and remembers which users are logged into the system.

```
1 <div class="loginfield">
3
5     <form action="checklogin.php" method="post" >
6         Username:
7         <input type="text" id="myusername" name="username" size="32" value="" /
8     >
9     <br/>
10    Password:
11    <input type="password" id="mypassword" name="password" size="32" value=
12        "" />
13    <br/>
14    <input type="submit" name="register" value="Login" />
15 </form>
16 </div>
```

This is a simple HTML form call, which post both the user name and password to the “checklogin.php” script. So no work is actually done here, besides posting the two.

So instead, let us take a look at the underlying code, which conducts the actual work.

Underlying code

Basically the system relies on session variables. These are unique for every user, and this functionality combined with AES Encryption, should make it fairly safe Wikipedia [2010b]. This is however only as strong as the way it is implemented. As the group is all novices to programming in both PHP and HTML, the high level programming skills needed to keep a top level security are simply not present. But this does not in any way remove the importance of this subject.

But how to handle sensitive information, such as password etc. Well this project focuses on MD 5 hashing. This is due to the fact that most DBMS support this functionality, so in the case that DBMS needs changing, the login procedure will not be subject to much change. So how does this look in PHP, let us take a look.

```

2  <?php
4  //This script is done with the help from the tutorial from http://www.phpeasystep.com/phptu/6.
    html
6  include("userdb_login_dao.php"); //Including the DAO connecting to the DB. This uses
    userdb_login_dao, which should connect with a read only user.
8
8  // username and password sent from form
$myusername=$_POST['username'];
10 $mypassword=$_POST['password'];
12
12 // To protect MySQL injection
//Remove the slashes from both username and password
14 $myusername = stripslashes($myusername);
    $mypassword = stripslashes($mypassword);
16
16 //This is to avoid the user can write SQL commands in the user and password field.S
18 $myusername = mysql_real_escape_string($myusername);
    $mypassword = mysql_real_escape_string($mypassword);
20
20 /*Here the password is MD5 hashed*/
22 $mypassword = md5($mypassword);
24
24 /*Here a search is done in the database.
    The database is checked up against the data received from the index side*/
26 $sql="SELECT_*_FROM_Userdb_WHERE_username='$myusername' _and _password='$mypassword' ";
    $result=mysql_query($sql);
28
30 // Mysql_num_row is counting table row
32 $count=mysql_num_rows($result);
    // If result matched $myusername and $mypassword, table row must be 1 row
34 if($count==1) {
36     // Register $myusername and redirect to the main page"
36     // At the same time, a session is started.
38     // This session will uniquely identify the user.
38     // The usersession is identified by the username
38     // Here the permission level is stored in a session variable.
40
40     $userrights=mysql_fetch_array($result);
42
42     session_start();
44
44     $_SESSION['username'] = $myusername;
46
46     $_SESSION['permission_rights'] = $userrights['Permission_level'];
48     mysql_close();
50
50     header("location:main.php");
52 }
52 else {
54     echo "Sorry, _the_username_or_password_were_incorrect._<br/><br/>_Please_try_again._";
56     echo "<a_href='index.php'>click_here</a>_to_log_in_again";
58 }
58 //This clears the objectbuffer
60 ob_end_flush();
    ?>

```

There are several things to notice here. First of all the connection here are not done via the regular “dao” files. This is due to the fact that it would make sense only to use a user which has the rights to read in the database. If one would want to check for a user name in the database it is only needed to make a read of the entries. This would also raise the level of security against injection in the database. This has become a growing concern Wikibooks [2010], since the PHP handles the SQL in a quite interesting way. The seamless use of variables in queries, could potentially give access to executing command that were not intended. If for instance one would login with a SQL statement, instead of a user name, the PHP and DBMS will simply interpret this as a SQL query. This would make it possible for a user to login to the system

in several ways. However, some countermeasures have been made to comprehend such a scenario. Of course the read-only user, but *stripslashes* and the *mysql_real_escape_string* are especially important to notice PHP.net [2010], as they format the input, so that the most common way of injecting access to the system will be denied. However this way of breaking in can be done in so many ways that new methods are found ever so often, so a constant check of the systems security is necessary to keep a top-level security.

4.5.6 Validation

Validation of data is critical in most applications. There are two types of validation, client side and server side. It is important to distinguish between these two. Client side validation runs in the browser to check that the form values that are submitted are of the correct type. Server side validation also checks the code that is submitted to see if it is correct. It is possible to use them both for extra security. The advantage of using client side validation is that the value check can be performed before the actual data is sent to the server. This means that the server does not get burdened with extra things to do. An important aspect of validation is to get rid of possible garbage data that someone could accidentally put into the system.

To prevent mistakes when entering data or injection of data, we have found a script that validates text input for us. We have chosen client side validation. The choice to have this type of validation was primarily to catch any mistakes when typing in data.

We have used JavaScript for the validation. An example in our code is when a user wants to enter a person. The user enters the data that he thinks matches the appropriate text fields and presses submit. While the user thinks that the data he entered was valid, it turns out that some of the data does not match the required values. So the solution is to make a validation of the data before it is sent to the browser.

When the submit button is pressed it calls a script that checks if any of the forms is blank, if the phone number is a phone number and the date is a date. The way this is done is by including Java Scripts. It checks each of the fields and if one of the values are empty or does not correspond to the correct data type it simply returns an empty field. If all the data types are correct, it jumps to the next script that sends the data to the server. If the data is not correct it displays a message saying "Some fields need correction:" + the reason why the error occurred.

So how does the script know what the error is? The script can be seen below. Notice that it is a function, it is the name of the function we call when we press the submit button.

```
function validateEmpty(fld) {  
2     var error = "";  
4     if (fld.value.length == 0) {  
        fld.style.background = 'Yellow';  
6         error = "The_required_field_has_not_been_filled_in.\n"  
    } else {  
8         fld.style.background = 'White';  
    }  
10    return error;  
}
```

When the function is called it initiate a variable called “reason”. For each of the text fields in the form that was submitted the function the function calls different functions from another script. Take for example the validateEmpty function that is called on the first name. This function can be seen below.

```
1      function validateEnterPerson(theForm) {  
2          var reason = "";  
3          reason += validateEmpty(theForm.First_Name);  
4          reason += validateEmpty(theForm.Last_Name);  
5          reason += validateEmpty(theForm.Username);  
6          reason += validateEmail(theForm.Email);  
7          reason += validatePhone(theForm.Phone);  
8  
9          if (reason != "") {  
10             alert("Some_fields_need_correction:\n" + reason);  
11             return false;  
12         }  
13         return true;  
14     }  
15 }
```

The important part of this function is that if the length of the inserted text is 0 characters long it returns an error message. This error message (text string) is then put in the “reason” variable from the first function. Finally the first functions returns the “Some fields need correction:” + the reason why the error occurred (the “reason” variable) and displays it for the user. Also to quickly observe where the mistake was made, it colors the input field yellow.

5.1 Introduction

In the design chapter, the design, navigation, security and validation settings have been included in the project. Now a comparison on how the different features works together. That will be the goal of this chapter. Here a description on the different kinds of testing will be made, the focus of course being, what have been applied to the project. This includes a load time test, and a usability test. The results of these tests will be discussed in section 6.2.

5.2 Performance testing

The overall idea with our tests is to see how well the system performs under different loads, but also to measure the load time. The test measures how long it will take for the website to load, concurrently with how many items that is inserted into the database. The following sections will describe performance testing of the system.

5.2.1 Loadtime test

In order to check how long time the system would take to load we made a small code snippet to insert multiple records into the "Item" and "Hardware" table, in the database. The idea of the test was to see how much data we could insert before the load time of a webpage in the system became unacceptable.

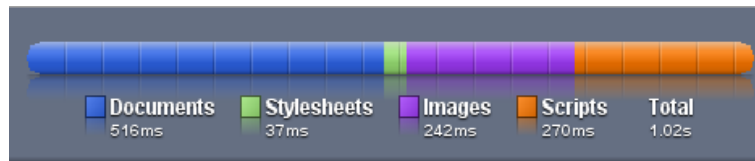


Figure 5.1: *The model shows an example of the kind of output you would get from Chrome. The “Documents” part is the time it takes to load the contents from the database into the page.*

Tools

Measuring how long the page takes to load is relative simple. We used the developer tool included in Google Chrome - One of the fastest browsers at the moment cnet.com [2010]. This tool measures the load time for the different objects on the page and makes a chart of it. The output looks something like in figure 5.1

Setup

We could now design a test, to tell us something about what the maximum amount of records would be, before the time to load our page would become unacceptable. The code used to insert data is shown in figure 5.2

```

1 <html>
2   <head>
3   </head>
4   <body>
5
6   <?php
7       //Connects to the database
8       mysql_connect("172.25.24.201", "b403a", "google")
9       or die("No connection to the databasesystem at the current time");
10      mysql_select_db("bookingsystem")
11      or die("No connection to the databasesystem at the current time <br/>");
12
13      $nr = 801;
14      $max = 801;
15      $no = 1;
16      $i = 1;
17      $nr2 = 801;
18      $no2 = 1;
19      $i2 = 1;

```

Figure 5.2: *The model shows how we connect to the database and declare the needed variables.*

```

21 while ($max > $i)
22 {
23     $query_item = "INSERT INTO Item(Type_ID,
24         App_Index_No,Status,Date_of_Purchase,AAU_Reg_No)
25     VALUES('Hardware','2222','stock','1234-12-12','$nr')";
26
27     mysql_query($query_item) or die(mysql_error());
28
29     $i++;
30     $nr++;
31     $no++;
32 }
33
34 while ($max > $i2)
35 {
36     $query_hardware_laptop = "INSERT INTO Hardware(Hw_ID,
37         Serial_No,Model_No,Comments,AAU_Reg_No,Manufacturer)
38     VALUES('Laptop','$no2','Test Laptop','Testing','$nr2','HP')";
39
40     mysql_query($query_hardware_laptop) or die(mysql_error());
41
42     $i2++;
43     $nr2++;
44     $no2++;
45 }
46
47 mysql_close();
48 ?>

```

Figure 5.3: The model shows how the while loop is created, and the query is executed x amount of times. When done, the query for the "Hardware" table begins.

First normal HTML tags are created in order to show a receipt at the end of the page. We then connect to the database the same way as our bookingsystem_dao.php file normally does. Then the variables needed, are declared.

- "nr" is the starting "AAU_reg_no" of the items needed.
- "max" is the number of entries we want created.
- "no" is like "nr" a number we are able to use in the query if we want a increasing number e.g. "App_index_no" This is not necessary for the page to run though.
- "i" is used in the following while loop, to compare to the "max" variable.

While "max" is larger than "i", the loop will execute the query where data is inserted into the item table. When the wanted amounts of queries have been run, the loop closes, and a new one begins doing the same. This time, however, into the Hardware table, which is related to the Item table. This is done so the data is consistent, and reflects the system in normal daily use.

Finally when both loops have been executed it show a receipt.

```

49 
50 <?
51 echo "Done! ".$max." rows changed  "
52 ?>
53
54 </body>
55 </html>

```

Figure 5.4: The figure shows how a receipt is shown when the loop is done.

Results

We decided to test the database with the intervals below. The max test was with 1600 items. This is pretty unlikely to happen since our customer presently has approximately 600 items. After we inserted the number of items, we loaded the view_all_eq.php file. The results can be seen in figure 5.5. Taking the graph into considerations it would take approximately 12 seconds to load the page with our customers 600 existing items.

Items	Loadtime(seconds)
50	1,02
100	2,11
200	3,45
400	7,63
800	17,03
1600	40,42

Figure 5.5: The figure shows the load time in seconds for the corresponding number of items in the database.

The result is pretty much linear as expected, but it gives a good idea of how critical a page splitting script is a needed feature.

5.3 Usability test

As part of the system for our customer, we needed a usability test. This kind of test is used to identify any problems, or errors in the system which we may have overlooked and is also used to see if the customer accepts the design of the project. The following section will describe how the usability test would be performed.

5.3.1 Purpose

The actual test was going to be done at the customer's office. The test was performed as a "think-out-loud test". This means that the user, had to explain everything that he/she did in the program aloud. This was then documented with audio and video. In addition, the person had to say aloud, what seemed confusing, annoying, and illogical or what expectations the person has had for each function. Our tester is the System administrator at The Institute of Mathematics. He relies on administrative tools for his everyday use and will also be the primary user of the system once it is finished. Therefore we felt it was natural to test the system on him.

There will be focus on testing the administrator part of the system, as it is the most important part according to our customer. The actual test is constructed upon a scenario to make the test more realistic. The scenario is that he receives some new equipment that he has to enter into the system. We have chosen to test the whole system and not certain parts, to get an insight into how the customer understands and interprets our design and the language. The test was performed in Danish. We wanted to test the customer in his familiar surroundings or where the system would be used - mostly his office. This would have meant that he was not distracted by the laboratory-like setup that the usability laboratory at our university could offer. We tried to keep it at a level that would feel as natural as possible for the customer, when he had to work with the system.

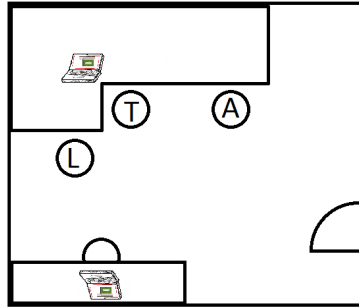


Figure 5.6: *This model describes the usability test setup. “T” is the test person, “L” is the test leader and “A” is the test assistant.*

5.3.2 Setup

In order to document the usability test we were going to use a PC that meets the requirements we have described in our “System definition” section 3.3. For our test we chose to use a laptop with a built-in webcam. We will use the webcam for recording the users facial expressions, when either something strange or illogical occurs. In order to record from the webcam we used the program “Debut Video Capture Software” NCH [2010]. This software is also used to record the sound through the built-in microphone in the laptop. While this program recorded the user we would also need to document his actions on the computer. In addition to “Debut Video Capture Software” we also used a program called “CamStudio” CamStudio [2010]. This program was used to record everything that was going on at the screen. In figure 5.3.2 the setup can be seen. There will be a test leader, leading the interview, and introducing the test person to the test. There will also be an assistant logging the interview. The test can be seen in Appendix D.1 and the found problems can be seen in Appendix E. Furthermore some of the problems will be discussed in the next section 6.2

5.3.3 Usability problems discussion

One remaining issue is the problems derived from the usability test. This section will discuss some of the problems stated in appendix E. The problems mentioned in this section discusses problems related to each other in four categories. The nature and the categorization of the problem will be discussed, and there will also be given an idea of how the problem could be solved. Lastly the discussion of the rest of the problems can be seen in appendix E.1.

Consistency

Problem 4: The user enters another invoice, and chooses a file he wants to associate with the data. After he presses the submit button, he is again greeted with a blank screen with text that explains that he has done the task correctly. But there is no back button. This is categorized as serious to critical because the user does not know what to do exactly, and is forced to use the browsers back button. To prevent this problem one could add a back button to the page with the explanatory text. Another way to handle it would be that it automatically went back to the page after a few seconds.

Problem 14: The problem refers to that the search data is not consistent overall in the system. For example the "AAU registration number" is to the far left in the "view all equipment"-screen but when you search for specific items you find that the registration number appears in the middle of the table instead. This is categorized to be a cosmetic to serious problem. This is due to the fact that the watchful user would discover that the registration has moved, but probably most of the users would not. A solution to this problem would be to make the output data consistent for all the search queries.

Problem 15: The user finds the serial number through the "Edit"-screen, but it seems he would have liked a details button as well. The problem has been categorized as cosmetic to serious, since the user could be misled not to use the "edit"-button to find all the information he is looking for and therefore not find the information he is looking for. The obvious solution would be to add a details button to the search query as well.

Problem 18: The user searches for a "AAU registration number", but he chooses a "apparatus index number" instead, which results in faulty data being loaded into the form in the "enter new hardware"-screen. This is categorized as a serious problem since it takes the user a while to figure out this problem. Also he gets help from the test leader/observer. This could potentially be a critical problem, but it is hard to say as the leader/observer is quick to help the user on the right path. To solve this problem we would need to be more consistent with the data outputted in the search fields.

Problem 28: The user deletes an item through the search specific queries, and gets a new page where the menu is missing, and there is several error messages. This problem is categorized as a serious problem since the user would not be able to get away from the screen, without using the back button. It could be discussed if it is a serious problem if the user is a novice computer/internet user. The solution would be to change the html code, to load the correct CSS-code, so the menu is loaded correctly.

Feedback

Problem 3: The assignment that the user is given tells him to attach a PDF-file to an item. But since there was no file prepared, he does not choose anything and upon pressing the submit button he is presented with a white screen and an error message. This is categorized as a serious problem since it potentially could keep the user from doing his job. To prevent this problem there should be either validation of data or an error screen that lets one go back and do the job right.

Problem 13: The user searches for a serial number, but the query does not display the serial number and the user has to double check the search to be sure it is the right item he has found. This is categorized as a serious problem, since the user could use an unnecessary amount of time, checking if the output is correct. The problem could be solved by displaying more correct data. For example the same data as in the search fields.

Problem 27: The user deletes an item from the search list, but there is no pop up warning. This is categorized as serious, since the user could accidentally delete items from the database without noticing. To solve this problem one would need some kind of feedback to prevent the user from deleting, without noticing. This could for example be a pop up box that warns the user he is about to delete the specific item.

Problem 30: After the user pushes the "Add person"-button he is sent to a screen full of errors. This is categorized the same as problem 28 and the solution is the same.

Problem 36: The user tells the test leader that he expected some feedback after he has lent out equipment. This is a serious problem since the user used time on checking if he had lent out the equipment. This would be solved by adding feedback to the function as the user points out.

Problem 38: After the user returns the equipment he would have like some feedback. Is categorized as cosmetic and could be solved the same way as problem 36.

Language

Problem 31 The user have great problems finding the menu point where he can lend out equipment to a person. This is categorized as a serious problem, since it takes a fair amount of time before the user finds the function. The solution could be to add better descriptions to the menu points, and maybe adding an html "title" to the menu point that gives the user a better idea of what can be done, if the user clicks the link.

Problem 32: The users' first impulse is to go the "Enter new data"-screen to lend out equipment. This is categorized as a serious problem since this means that the user does not understand the menu points of the system. This could be solved with better description of the menu points and perhaps with a little more training in the system.

Problem 33: The user goes to the "lend out equipment"-screen but he does not really think it is the right place to go to lend out equipment. This is categorized as a serious problem since this means that the user does not understand the systems texts (labels) and this means that the user could use substantial amounts of time, to find the right function. This could be solved by naming the menu points with more saying titles.

Problem 34: The user notices the header of the table called "lend to user" in the "Lend out equipment"-screen, but he does not expect this is the way to lend out equipment. This is categorized as serious problem, which shows that there is a lack of information on how to execute different functions. The problem could be solved by more text that explains how to lend out the equipment. Maybe a page that helps to explain how to use the different functions.

Learnability

Problem 5: The user enters "lager" (stock) in the comments field while entering the info about an item. The comments field is supposed to be used as a field where one writes the technical specifications about the item. This is categorized as a cosmetic problem. To prevent this one could change the name comments to something more saying, like technical specifications etc..

Problem 6: As he enters a new piece of hardware, he does not add an invoice to it. This is categorized as critical, because there is no other way to add an item to an invoice without doing it from the start. To prevent this, one would either not be able to enter an item without adding an invoice, or there should be a separate functions to be able to add invoices to items. The first idea is not really favorable since this does not give the freedom to add things whenever one wants.

Problem 9: The user wants to go back and add an invoice to an item, but he does not know how to do this. This is categorized critical since this is not possible and the user will never be able to fulfill his task. To prevent this, one could make a function that makes it possible to link an item to an invoice. Another way is to alter the "Edit"-function so it actually links the item together with an invoice.

Problem 11: The user tries to bind an item to an invoice using the "Edit" function. This is however not possible. This is categorized as a critical problem, the same as problem 9. To solve this, one could use the same approach as problem 9.

6.1 Introduction

This chapter will discuss and conclude the results of the project, and subsequently put them into perspective.

6.2 Discussion

The discussion in this section will be based upon the different parts of the report in relation to the project delimitation, in section 1.2.

Development method

The idea behind using the agile methods that we have chosen in the project, was to be able to respond fast to changes in the requirements. While this was indeed a noble intention, it was not used as much as we would have liked. There were very few changes in the requirements and that raises the question, did we have to go agile?

When considering the circumstances at the starting point of a project, one can never determine if it is subject to change. Therefore we believe that our choice was the right one. When considering how many changes the project actually had, a traditional method would have been able to handle the project without any complications as well. But on the other hand, the lack of regular contact with the customer may have contributed to absence of requirement changes.

It could be discussed whether or not a choice of using agile methods were the right one, especially since the development experience of the group is based on traditional techniques. One problem with XP and Scrum is, that it requires a fair amount of experience to be able to start developing without a huge amount of documentation. The choice of Scrum was based on an understanding that it used more documentation than XP suggested. When that is said, a traditional method could once again be considered a

possible alternative, because it is one of our main competences. In spite of the intended development methods, the actual process was a mix of agile developing (XP and Scrum), where code was made as simple as possible, with a fairly extensive traditional analysis taking place afterwards.

Scrum as a whole worked well. A difficulty with our use of Scrum was that we did not use the stand-up-meetings. The point of these short meetings was to brief all other group members in the project status. We did have meetings and though they were not performed daily (Each Monday), they served as a briefing for the other group members. The question is if a daily meeting would have helped with the lack of overview, that the project sometimes faced. A fact is that the overview of the report was primarily in the hands of the project manager. This meant that some of the group members at times did not know where to put their efforts. The stand up meetings could very well have prevented this.

Another thing that we did not make use of, was the risk analysis. The risk analysis was made to identify and cope with any problems, that the project might face. We used a fair amount of time making the risk analysis, and making up handling scenarios for possible situations that could occur. Even though we used time to complete the document, it was not used when project problems occurred. Interestingly enough, when a problem araised, one that the risk analysis should have been able to cope with, it was not consulted. This lead to unnecessary panic in the group. One of the biggest problems to comprehend was facing the fact of not having any framework. While the risk analysis purpose is to identify problems, some problems may slip its scope. The fact is that the framework problem was not a problem included in the analysis, and therefore it could not have helped us out of this particular situation. While it was not used as intended, the group recognize the potential of this tool to foresee possible problems and thereby avoid them.

Initially we intended to keep the contact with the customer on a regular basis. The idea was to involve the customer as much as possible in the analysis phase and afterwards after each sprint. Scrum describes that in a sprint review meeting, a contact person should be available to the development team at all time. We quickly discovered that this what not a possibility due to the fact the contact person was the same as the customer and he had a job that needed his attention. The solution was to make one of the group members a representative for the customer. This should have helped us answer some of the questions that we would have asked the contact. This did not work as intended, instead of using the representative in the group we based the analysis on the first interview with the customer and thereby we might have prolonged the analysis period unnecessary. If we had used the representative as intended we might have been able to conduct a proper customer involvement, but the question is if he would have had enough knowledge to be able to give us the proper information.

Our customer expressed a wish for documentation, with the intention of applying further development, by third party. However we have not discussed with our customer the nature and extent of the documentation. We have by instinct used traditional methods for documenting, and thereby one could discuss whether our development method

is a mix of both agile and traditional methods. The result of this is a combination of traditional analysis and more or less agile development, that is sprint oriented.

The design phase is characterized by the lack of defined programming guidelines. The fact that all of the files are in one folder is the reality of the project. Dividing the files into several folders would have been the least that could have been done to create a better overview. Another thing is the naming of the various files, again it would have been good with some naming guide lines. XP defines a code standard but we did not use it because of lacking HTML, PHP and Java Script experience.

Analysis

Another aspect the of the project was the slow start of the analysis work. A problem that the group had faced before. The slow start could be based on the necessity of learning basic techniques for designing, and implementing some of the things the project involved. This however, could be a part of the learning process that a university project faces. That is, a slow start, because of the need to learn new techniques. One would expect that in a working environment, one would have these skills, and therefore can start analysis and development at a much earlier stage. If we had started the analysis work at an earlier point, we would probably have had the time for more sprints.

The group have knowledge about developing software for inexperienced users, but it is the first time an experienced user is involved. This means that in some ways the user knows more about the problem area than we do. An inexperienced user knows what he wants to do but not how it should be done. An experienced user knows exactly what he wants and has an idea of what is possible to make. This means that we have a hard time guiding the customer to get a more appropriate system than the one he thinks he needs as we would normally be able to do with less experienced users.

In the analysis phase the database design has been subject to many changes. The course “Databases” made it clear to us that a good database design would give a lot less problems when programming the system. A problem with this is that we maybe had to much focus on developing a good database design and thereby neglected other important things like the fact that we are missing a framework in our system. The question is if we would have realized that we needed a framework if we had used less time analysing and developing the database. That being said the thorough design of our database has meant that there has not been any great changes in the code. If we were to change the database structure late in the development period the amount of code that would have to be changed would also be significant.

The database structure depended of the analysis made in the start of the project. The analysis stood pretty much still in the beginning of the project. This meant that it took quite a long time to get the database structure right. A consequence of this was that the development of the system was postponed. Our course in database design stressed that a good database structure was one of the first building blocks in developing our system. Therefore it had to be accurate as possible to prevent too many changes in the code later on. As a result of this the database was ongoing a lot of changes in the

analysis phase of the project period, which could be a consequence of lack of contact with the customer.

One aspect of the analysis, and developing of the system were the object oriented paradigm. The whole construction of our system, is based on what people may call “spaghetti code”. Therefore it made no sense for us to make a class diagram. The EER diagram describes the structure of the database and it includes some elements from the object oriented paradigm e.g. inheritance. The class diagram describes the structure of a system. The class diagram would have made sense to use if we had a framework, but since we do not have that feature implemented yet, it did not make any sense to create one.

The design of the user interface has mostly been based on our own design ideas. The customer had some influence but in general he was not concerned with the design as long as it worked. The usability test shows us that we missed some general defined design principles. An example is that the user interface is not consistent in for example the placement of table data displayed on the web page.

A question is if the customer really wants security in the system. The choice of securing the system was an important aspect of the project is that it is not only our system that provides security but also the web server. This does not lie in the scope of this project though. If we were to listen to the customer there would probably not have been as much security as the system holds currently. But the group realised that when dealing with sensitive data some level of security is needed.

The code

First of all, lets state that the level of programming skills in the group were relatively vague. Especially in PHP, which no group members had ever programmed before. This have of course raised a major issue, if skills needed to successfully complete the project were beyond those of the group, how could it ever end succesfully? However, the project faced another challenge in the form of user-requirements. The user wanted a web-driven database system, and since MySQL were the primarily focus of the DB-course, MySQL and PHP seemed like the obvious choice.

The group were also novice when it came to the HTML programming languages, though more experienced than with the PHP. So apart from learning the fundamentals of databases, their implementation and so on, two programming languages needed to be learned. How could we deal with this scenario? As part of the Software Engineering course, risk analysis were implemented in the hope of developing tools to comprehend scenarios as this. But even though the group agreed that the programming level of the group were a major concern, there simply was no possible way of handling this. So a great deal of time was spent on learning these programming languages and getting a basic understanding of how they worked individually and possibly most important how they could support one another. But how to deal with such a scenario in the future? Well, this was concluded not to be a concern. Simply relying on being able to learn two programming languages on a sufficient scale for a project simply is not a possibility.

The chosen style of programming must be within the competence of the project group.

The system that was developed was based on, what people would call “spaghetti code”. This type of coding style has no particular framework. The inexperienced use of XP might have had an influence on this. The mantra of coding the simplest possible solution, to every little problem makes it difficult to maintain an overview of the system as a whole. This means that a lot of duplicate code has occurred, in spite of what previous experience has taught us about this.

This actually also explains how the system was created, simply by trying and testing one feature, getting it to work. When this was done, the next feature underwent same kind of implementation. It was simply a matter of lacking programming skills. This raised a paradox in which the group was aware of the object oriented programming environment, and how a object oriented programming style would result in a much more flexible and effective program. But programming in such a way with PHP was beyond the capabilities of the group. However, the programming skills in the group were high enough to realize this problem, but too low to solve it appropriately. This also explains the rather frustrating process which the group have been through, trying to make functional code in an unstructured environment.

From our perspective, the chosen programming language might not have been the right choice. The point is that the technical skills of the group members were oriented towards .net applications. The question is if it would have been wiser to decline the project and to chose something that was more in our line of experience, which is C# applications.

One of the requests from the customer, was to transfer the data from the old database into the new one. This request has not been accomplished. One of the main reasons for this, was that we could not find any method to do this in an easy way that matched our time frame. This problem is that we have no knowledge about database files in the Perl language. The problem could probably be solved by scripting the old database file but we do not currently have the time or expertise to do this.

Testing and missing features

The load time performance test yielded good results. The fact that the web page used a little over 12 seconds to load 600 items, which is the current number of items in the old database, indicates the need of further optimization of the web page. This could be done by dividing the number of displayed items into multiple pages with some sort of page splitting script. Another aspect is whether or not it would take 12 seconds for the items to load on one of the institutes servers, which is most likely to perform better than the one we used.

Another testing type that was considered was unit testing. One of the reasons for not conducting unit tests, is the study regulation stating the need to include performance testing in the project Regulation [2010]. With a limited amount of time, it was decided to focus on a load test, as this seemed reasonable to illustrate the performance of our database oriented system. The lack of classes does not favor the use of unit testing either, as unit testing of our code would not apply to the system as a whole.

The fact that we did not use a framework was realized too late in the project. This meant that we were not able to implement it and therefore it became one of our unimplemented features. The use of a framework, for example MVC would have helped with consistency in the code, also the use of a lot of duplicate code could have been prevented. One of the main ideas with the system was that it could be accessed from several types of browsers. The fact that the system is not compatible with Internet Explorer is a concern with the current design due to it being a fairly common browser. Since the customer does not use Internet Explorer this is not seen as a major problem at this moment but as soon as the system has to include more users like students and employees the system should be compatible with this browser.

Mostly we have been satisfied with the design of the user interface. But the usability test had also shown that in various areas the design lags consistency, feedback, understandable language and learnability. The consistency problems are due to late changes in the CSS code. The problem occurs where the style sheet is not added to the HTML code. The feedback problem is an overall problem. In the first HTML code we had added the PHP script to a new HTML page + feedback text, but after we copied the PHP to its own scripts the feedback text was deleted and afterwards forgotten. One could discuss that text feedback is not always needed especially if one can see that the website reloads. The understandable language problem could be discussed whether or not it is a big problem. One could say that this would stop being a problem after some training with the system. But on the other hand if the system is expanded many new users would have the same problem. The learnability problems could also be solved with some training in the use of the system. On the other hand one should always strive for making a system that makes it as easy as possible for the user to do the work at hand.

6.3 Conclusion

6.3.1 Introduction

After an extensive discussion of the project and the problems which it has encountered, the relation of the two is now in focus. In order to conclude this project we will answer the problem definition, which made the foundation for the project.

It is stated:

How do we design a new system for the administrator at the institute of mathematics, that fulfills the users requirements?

This problem definition aimed extremely wide, therefore a thorough delimitation were conducted. This, among other things, defined our primary user as the system administrator. This also empathizes the importance of choosing the right project management methods by considering the customers' requirements and our timeframe at all time

6.3.2 Project management - method choice

When looking at the development method for the system, it becomes clear that we have not taken full advantage of the agile principles. Instead we have used a combination of the agile and traditional paradigm. Our lacking experience of working with agile methods is to blame in this case. Our experience in working with traditional methods also could be held responsible for the development. An experience in traditional methods that is caused by the fact, that it have been used at the previous project.

When looking at our use of Scrum we have utilized many of the tools that Scrum provides. It has helped us structure and improve the process, even though we have not taken fully used all of the Scrum tools. One feature, in particular, that we have not taken advantage of was Stand-Up meetings, but this was considered less important. This was due to the fact that we had close face to face communication during the entire developing process.

Our contact with the customer has been insufficient in comparison with the choice of agile methods. But the influence on the system has been minimal since the requirements have not changed much. However, the lacking contact, might cause that we would not have noticed if any changes arose. Had the problem domain been larger this would potentially be a major concern, but the usability testing of our system only showed few errors caused by lack of customer involvement.

The programming skills of the group have been vague from the start, and we did overestimate our own programming experience. This combined with a programming

method such as XP, called for an extremely bad match. The unstructured style of programming in XP really puts faith in the developers' abilities. We simply did not have the means to accomplish this sort of method.

When considering the chosen methods in a broader perspective, a traditional method would have made more sense in the particular project, since the system have not been subject for major changes. But taken to account that there was no way of knowing this in the beginning of the project, we would argue that the choice was the correct. When putting into perspective that it is the first time we used agile methods, we consider it to be a success. Mainly because we at the end had a working prototype of the system for the customer.

6.3.3 Tools

The documentation process has been somewhat concise. Documentation was only written to the extent that it was considered necessary. This suites the agile principles very well, but is perhaps not preferable for our customer. This became a concern since the system administrator wanted a modular system, which was well documented, in order for a third part to further develop the program.

But in the end we still found that the system was as well documented as possible, due to the coding style which were not object-oriented. Another concern is the definition of the term documentation, whereas the group sees it as both the analysis and design section, the customer might only want a class diagram and a description of which scripts does what. The documentation discussion between us and the customer have not been done in this project, but would be essential before releasing the program. Based on the customers requirements some documentation is necessary.

We learned about the possibility of using a framework, such as MVC. This was unfortunately too late in the process, so there were no time to implement it. This is regrettable, since we would have liked our code to have been more structured, but unfortunately it is a fact. A solid framework is therefore an essential part of the structure and the functionality of a system.

The usability test shows that there is a lot of problems in especially in the four areas consistency, feedback, language and learnability. But with because we made the system we can also see that many of them are fairly easy to correct. Some of the problems exists because that the team has not been keen enough to correct them when changes to the CSS code occurred. These would not have happened if the project had used a framework or at least not to the same extent. But these problems also show that the system is not yet finished.

6.3.4 Database system

Since one of the main themes of this semester is “database systems”, much effort has been put into designing the database fundamentals. Here of course the EER diagram in particular. Even though this has been subject to many changes, it has been created according to our expectations. The use of the SQL language has been rewarding, since it is relatively easy to learn.

The overall database design has been a trial-and-error approach. Which have been found to be a very successful solution. Not only did we get a good understanding of how to structure a database, we also learned to combine this with PHP. A main concern was that we would, encounter scenarios where major changes were needed in the database. If these occur late in the development it would have significant impact on our code and potential set us critically back. A fact is that a well-defined database structure is fundamental.

6.3.5 Summary

To design a new system for the administrator we had to start from our customer expectations for the system. Basically it should be able to contain the same information as the old, but with some new added features. These features were not critical, but they would be appreciated. However, did we reach this goal? We made a functional system, which the customer can use and live up to most of his expectations. This has however not been done in an elegant way. The code is unstructured and is in serious need of a framework. When this is said, the coding style only affect the modularity of system, so in order to expand it, serious work is needed. The reasons for the bad code have now been identified, and only leave us wondering: “should we have taken on this project?”

In any other case than a school project, no. But since this is a learning experience, this project has seriously pushed the boundaries of our capabilities, and showed the pros and cons of using the agile methods.

6.4 Putting into perspective

Like so many other projects, the time factor has played an essential role, as it has been a gateway for this project. As in so many other cases, the project management can actually be held responsible, since the primary role of project management is to structure the process. But how did it get so far? And why? As this report discussed and concluded the lack of skills where one of the main reasons for the outcome of project. But what if this project went on for another month? Taking in to consideration the kind of knowledge the group have obtain during the project. To put things into perspective we have considered what we would implement if we had two-four more sprints to work on the system.

One of the main things that needs to be implemented is a page split script. This is not described on the product backlog because it is not critical for the system to run, but it would enhance the performance of the system drastically. This feature would also mean that the customer would use the system more efficiently. This is because there is around 600 items in the institutes current database and the performance check of the system (See section 5.2) found that, it would take approximatly 12 seconds to load all the items, which is far longer than most people wants to wait for a page load. Therefore the page splitting script has a very high priority in future implementation.

If we look at the remaining user stories in our product backlog (See figure F.3) we can see that implementation of more users is in focus. We need a way to send out reminders to the students and employees when their return date is coming up. This would be a great way to make sure the lent equipment is returned on time. If the feature should involve sending an E-mail, SMS or both is yet to be discussed with our customer. After this is implemented, the next step would be to involve the students and employees in the process, so they would be able to book equipment from the system. First of all, we would need to list all of the available equipment in the system. That being all the equipment in stock, and not already booked. When this is made, it should be possible for the students and employees to book the equipment they need, and afterwards the administrator should be able to approve the booking. (See figure 3.4) Our 16th user story describes that we need a list of booked equipment for the administrator. This would probably be where he could approve the bookings. The students and employees should be able to keep an overview over their rented and booked equipment as well as canceling bookings.

Another implementation that is not critical, but important, is the implementation of a framework such as MVC. This would be an important feature, when considering future development and maintenance.

The system as it is has a lot of usability problems that needs to be addressed, especially the more system critical where one ends up on a blank page or a page filled with error messages, these are however easily fixed. But there are also some problems with some of the functionality, things that was not revealed by the test, but which the team has discovered since. One could probably make these into a bug report which could be sent with the system so the administrator would have an easier time getting it resolved by third party if needed.

BIBLIOGRAPHY

CamStudio [2010]. <http://camstudio.org>.

Chen, P. [1976], The entity-relationship model - toward a unified view of data acm transactions on database systems. 1: 9–36.

cnet.com [2010]. http://news.cnet.com/8301-1001_3-10030888-92.html.

Firefox [2010]. http://en.wikipedia.org/wiki/Mozilla_Firefox.

Larman, C. [2004], *Agile and Iterative Development*, Pearson Education, Inc.

Mathiassen, L., Munk-Madsen, A., Nielsen, P. A. and Stage, J. [2001], *Objekt Orienteret Analysis & Design*, 3. udgave edn, Marko.

MySQL [2010]. <http://www.mysql.com>.

NCH [2010], ‘Debut video capture software v.1.49 nch software’.

NetBeans [2010]. <http://netbeans.org>.

phpMyAdmin [2010]. <http://www.phpmyadmin.net>.

PHP.net [2010]. <http://php.net/manual/en/security.database.sql-injection.php>.

Regulation, B. S. [2010]. http://www.fsn.aau.dk/fileadmin/fsn/dokumenter/7_studieordninger/f09/baitsept2008_sept2009.pdf.

university, A. [2010], ‘Soe slides’, Online. <http://www.cs.aau.dk/~jeremy/SOE%202010/resources/SOE15coursesummary.pdf>.

Wikibooks [2010]. http://en.wikibooks.org/wiki/PHP_Programming/SQL_Injection.

Wikipedia [2010a]. http://en.wikipedia.org/wiki/Enhanced_Entity-Relationship_Model.

Wikipedia [2010b]. http://en.wikipedia.org/wiki/Transport_Layer_Security.

Wikipedia [2010c], 'Risk analysis (business)', Online. [http://en.wikipedia.org/wiki/Risk_analysis_\(business\)](http://en.wikipedia.org/wiki/Risk_analysis_(business)).

APPENDIX A

USER STORIES

The user stories we made are based on an interview with our customer. During this interview he explained what he wished this system to do, and how he expected it to be used. Some of these functions he mentions we had to interpret a bit, but none of the functions that we mention is ideas that we made up, they are all based on the customers comments and opinions. From time to time we had some questions for him and this gave him some new ideas for improvements along the way. As part of Scrum this was not a problem and new features were added to the product backlog.

The user stories was made in three phases. The first phase where we started the brain storming we felt it would be too difficult to brain storm with 6 people at a time. So we divided our group into two smaller groups. At this point each of these two groups would brain storm and then we would compare the work afterwards. In order to compare the brain storms we had to agree upon a template for the sentences so the user stories were easier to compare. An example of this template is:

“As an [User Type] I should be able to [Function] in order to [Business Value].”

One of the groups wrote the User Stories down on small pieces of paper while the other group did it on a laptop and then printed it out, and cut out each sentence. Though we only needed to come up with user stories for the administrator we felt it would make sense to identify all possible people that could end up using the system.

At this point we had 3 user types identified, an admin, an employee and a student. Each user type had different needs, and therefore we had to make user stories for each one. This resulted in a myriad of user stories. At this point we proceeded to the second phase. Now we had to allocate a group to review all the user stories. This group would be responsible for filtering all the stories, which ensured that all the user stories referred to something the customer had said and that none of them overlapped.

First off all the user stories was compared, many of them sounded a lot alike. Therefore we chose the one sentence that either sounded the best or included most of the functions that the customer was seeking. After the filtering the group agreed that we should concentrate about the administrator stories, so they had the first priority. The reason for this is that we knew that this system would not make any sense without an administrator, secondly the lack of time were a factor and since the we were developing for an administrator it would make sense to make the functionalities that he wants first. According to the customer this was his main concern as well, the functions that he was going to use were to be implemented first, all other functionalities were just “good to have“. When all the functions of the administrator were implemented we could think about the Employees and lastly we would do the Student functions.

One person from the group read aloud the story, and we would check once more if there were any duplicates in each category. After this part we did not do anything about Employee and Student stories. All the Admin stories would at this point be prioritized from 1-5, where 1 was top priority and 5 was the lowest priority. The following user stories were identified.

1. As an admin I should be able to log in to the system in order to use it.
2. As an admin I should be able to enter new EQ in order to register new EQ in the database.
3. As an admin I should be able to delete EQ in order to keep the database updated.
4. As an admin I should be able to edit EQ data in order to keep the database up to date.
5. As an admin I should be able to easily enter identical EQ (eg. A bunch of new monitors) in order to avoid entering the same data multiple times.
6. As an admin I should be able to search my EQ in order to easy and logically find all EQ (Full access).
7. As an admin I should be able to print a list of all EQ in order for “SKAT“ can take a look in our assets and see that our accounting is correct.
8. As an admin I should be able to find specific and detailed information about the EQ in order to find the Guarantee, defects, OS etc.
9. As an admin I should be able to keep track of all EQ in order to keep status whenever needed rented as well as in stock!
10. As an admin I should be able to associate the EQ to a location or a loaner in order to know if the rented item is at a location at the university or its rented to a private location.

11. As an admin I should be able to edit "Repair Status" on EQ in order to keep the database updated.
12. As an admin I should be able to accept returned EQ in the system in order to administrate the EQ.
13. As an admin I should be able to send out reminders to people who have rented some EQ in order to get the EQ back in time.
14. As an employee or student I should be able to find information about the EQ I am able to rent in a easy way in order to select the correct EQ for my needs.
15. As an employee or student I should be able to book EQ in order to be able to rent EQ I need.
16. As an admin I should be able to get a list of booked EQ in order to see status of EQ and who has booked it.
17. As an employee or student I should be able to keep track of my rented EQ in order to know when to return my rented EQ, and to keep track of what i have rented.
18. As an employee or student I should be able to cancel booked EQ in order to update the list of EQ I need.

APPENDIX B

PLANNING GAME

These are the results of the planning game in the project group. Each user story number corresponds to the user stories listed in appendix A, with each group members estimation of ideal developing hours. In the first round each person estimated how long the user story in question would take to develop. Before the second round, the persons with the highest and lowest estimates, had to argue why they assessed the user story as they did, in the first round. Then the second round began and people would once again estimate the user stories, but this time with the new arguments stated in mind. Then the average assessment was calculated for each of the user stories. This average assessment was then the estimated development hours used in the product backlog.

1. round

User story	Michael	Christoffer	Daniel	David	Simon	Max	Average
1	8	7	10	4	10	8	7,8
2	4	17	5	5	5	8	7,3
3	5	3	8	5	2	2	4,2
4	6	6	3	10	8	3	6,0
5	15	5	100	5	10	8	23,8
6	3	8	40	30	10	8	16,5
7	3	2	8	5	20	5	7,2
8	3	4	12	7	15	8	8,2
9	1	2	8	2	1	1	2,5
10	5	7	7	8	5	2	5,7
11	1	1	5	2	2	2	2,2
12	2	1	3	5	2	2	2,5
13	100	7	5	20	30	20	30,3
14	5	7	5	5	8	7	6.2
15	30	25	26	15	18	20	22.3
16	10	12	8	2	5	7	7.3
17	12	8	5	2	3	2	5.3
18	2	8	9	10	2	7	6.3

2. round

User story	Michael	Christoffer	Daniel	David	Simon	Max	Average
1	8	10	10	10	15	20	12,2
2	17	17	17	20	20	17	18,0
3	5	5	5	5	5	5	5,0
4	10	10	10	10	8	5	8,8
5	25	19	40	25	30	20	26,5
6	10	23	25	25	25	25	22,2
7	1	5	5	5	5	5	4,3
8	5	6	8	7	8	8	7,0
9	2	3	2	3	2	2	2,3
10	6	8	10	8	10	8	8,3
11	2	2	2	2	2	3	2,2
12	5	5	5	5	5	5	5,0
13	8	10	8	20	15	20	13,5
14	4	2	2	2	2	2	2.3
15	20	10	13	15	14	15	14.5
16	7	8	6	2	5	4	5.3
17	11	5	5	2	3	2	4.7
18	5	7	5	5	5	5	5.3

APPENDIX C

DATATYPES IN DATABASE

This chapter describes each attribute, with name, type, constraint, key and comments, in its respective table.

Attribute name	Type/ constraint	Key	Comments
AAU_Reg_No	Int(5)	Foreign key from Item	
HW_ID	VarChar(10)/ Not NULL		Holds the information about what type of hardware it is e.g. a screen, computer etc.
Serial_No	VarChar(30)/ Not NULL		Serial numbers is often listed with a combination of letters and numbers therefore we use varchar.
Model_No	VarChar(30)/ Not NULL		Same as Serial_No
Comments	VarChar(255)/ Not NULL		The Comments field is holding all other information about a piece of hardware, therefore it must be long enough to hold the information.

Figure C.1: *Hardware*

Attribute name	Type/ constraint	Key	Comments
Internal_Index_No	Int/Auto Increment	Primary key	The internal index number is used instead of the invoice number found on an invoice. This is because two invoice numbers potentially can be the same. This way we are insured that it is unique.
Invoice_No	VarChar(15)/ Not NULL		This is the listed number you can find on an invoice
Supplier	VarChar(30)/ Not NULL		The supplier of the item
Date	Date/ Not NULL		The day the invoice is recieved
Ref_Scanned_PDF	BLOB		Holds a reference to a scanned PDF, it is set as a path on a computer

Figure C.2: *Invoice*

Attribute name	Type/ constraint	Key	Comments
Association	VarChar(20)/ Not NULL	combined key	The association of a loaner, e.g. student or employee
Username	VarChar(10)/ Not NULL	combined key	username of a loaner, this is his/her AAU user name
First_Name	VarChar(30)/ Not NULL		First name of the loaner
Last_Name	VarChar(30)/ Not NULL		Last Name of loaner
Email	VarChar(30)/ Not NULL		Email of the loaner
Phone	Integer		Phone number of the loaner

Figure C.3: *Person*

Attribute name	Type/ constraint	Key	Comments
AAU_Reg_No	Int(5)/ Not NULL	Primary key	The AAU registration number. This number consists of 5 numbers and is unique for all equipment on AAU
Type_ID	VarChar(10)/ Not NULL		Signals what type of equipment it is e.g. hardware, software.
App_Index_No	Int(6)/ Not NULL		Apperatus index number, a number which all equipment have, not unique though
Status	VarChar(10)/ Not NULL		Status of the item, where it is, e.g. lend out, home, repair.
Date_of_Purchase	DATE/ Not NULL		Purchase date of an item.

Figure C.4: *Item*

Attribute name	Type/ constraint	Key	Comments
Institute	VarChar(10)/ Not NULL	combined key with room no	The AAU registration number. This number consists of 5 numbers and is unique for all equipment on AAU
Room_No	VarChar(10)/ Not NULL	combined key with institute	Room number of a location.
Address	VarChar(40)/ Not NULL		Adress of a location.
Description	VarChar(40)/ Not NULL		Discription of a location.

Figure C.5: *Location*

Attribute name	Type/ constraint	Key	Comments
Internal_Index_No	Int	Foreign key from Invoice	
AAU_Reg_No	Int(5)	Foreign key from Item	

Figure C.6: *Belongs_to*

Attribute name	Type/ constraint	Key	Comments
AAU_Reg_No	Int(5)/ Not NULL	Foreign key from item	The AAU registration number. This number consists of 5 numbers and is unique for all equipment on AAU
Rent_Date	DATE/ Not NULL		The date the equipment was rented
SDOR	DATE/ Not NULL		Scheduled day of return
Username	VarChar(10)/ Not NULL	Foreign combined() key from person	A username given to the user by they system administrator
Association	VarChar(20)/ Not NULL	Foreign combined() key from person	Email of the loaner

Figure C.7: *Rented_to*

Attribute name	Type/ constraint	Key	Comments
AAU_Reg_No	Int(5)	Foreign key from Item	
Institute	VarChar(10)	Foreign key from Location	
Room_No	VarChar(10)	Foreign key from Location	

Figure C.8: *Has_A*

APPENDIX D

USABILITY TEST

D.1 The test

The following pages contain the test, that were designed and applied to the system administrator of the Institute of Mathematics. Section E then lists the problems that were found.

Usability testing

Introduction

Thank you in advance because you will spend time on this test of the system. It is a great help, because we can discover errors and shortcomings that we ourselves have not discovered.

In advance we will remind you that this is a test of the program, and not you.

The system was developed for use by the Institute of Mathematical Science. It manages equipment at the Institute, and among other things allows you to rent out equipment.

The test you are about to do, consist of a series of tasks which reflect the daily use of the system. These tasks must be done in sequence starting at task number one.

The tasks should preferably be solved in the following way:

Start by reading the task out loud, and then explain us how you understand it, and how you intend to solve it.

Now start solving the task. An important point in solving the task is that you "think out loud" about the things you do in the system, and what your plans for that task is. In addition, it is important that you comment on things which surprise you, things you do not like, and things you find illogical.

You are welcome to ask for help during the test, but you will most likely not be helped to a solution of the task. Possibly you will receive small clues that will help you move forward.

Task 1

You have just gained access to the system and can now log in for the first time.

Your username is: **FinnBanke**

Your password is: **google**

Log in to the system.

Task 2

After showing up at the office a gray Tuesday morning, you observe that you have received a package. In the package there are two new screens and a laptop that you have ordered for the institute.

Before you enter the equipment in the system you scan the invoice into a PDF file on your computer.

Once you have your PDF file, you can enter an invoice in the system and the PDF file can be uploaded.

(It is not necessary to have a PDF file to create a invoice, but to keep data consistent it is recommended)

Task 3

Since you now have created a invoice, the received equipment can be entered into the system, and binded to the created invoice.

The two screens have been awarded the AAU Reg No **991** and **992**. They have the following specifications:

Model	LE19B455C4WXXE
Size	19"
Type	LCD
Aspect Ration	16:9
DVB-T tuner	Yes
DVB-C tuner	Yes
MPEG4	Yes
Resolution	1366 x 768
Full HD	No
HD Ready	Yes
Power usage	25 W
Connection Type	S-videocable, Component (Y/Pb/Pr), Scart, HDMI

Task 4

You remember that you weeks earlier entered a laptop similar to the new one you received in the package.

To avoid having to enter all data again, you can load data using the AAU Reg number. But you can not remember this number. You know however that the laptop has Serial no: **1789**

Task 5

An employee comes to you in great despair! His home office laptop has stopped working. After you have examined the laptop and tried various attempts to breathe life

into it, you choose to send it to repair. However, you can not remember where it was purchased, and therefore need the invoice.

The laptop has the AAU Reg Number: **32**

Task 6

You have discovered that the description of a piece of equipment is incorrect. This must be corrected. Right now the equipment have the following specifications:

Intel Core i3, 2.13 GHz, 4096 MB DDR3, 320GB HDD, 17.3 inch, NVIDIA GeForce 512MB G105M.

Instead it should be:

Flash-based, 512 MB, USB.

The equipment in question has AAU Reg Number: **24**

Task 7

It is time for spring cleaning , which means that some old pieces of equipment that is hopelessly outdated needs to be trashed or given to charity!

After having taken note of the AAU Reg numbers in question they must be erased from the system.

AAU reg. No's: **51** and **52**

Task 8

It's Friday afternoon and you're soon off for the weekend! Suddenly a student runs through your door breathless!

He says that he is critically lacking a screen for his group room, because his own is broken, and his group must submit their project on Monday!

The problem is that the student has not been created in the system.

The young man has following personal information:

Fernando Ramirez

Phone: 52 45 75 23

E-mail: FernRam@mexico.mx

Wanted username: FernRam

Task 9

After creating Mr. Ramirez in the system, it is time to find a screen for him, and then lend it to him.

Task 10

It has turned monday, and Fernando returns the screen after having handed over his project. It now needs to be returned into the system.

APPENDIX E

USABILITY PROBLEMS

This section will identify the problems found during the usability test and categorize these problems using the cosmetic, serious and critical notations.

Problem no.	Time	Problem description	Category
1	01:10	Uses both big and small characters but the system is not case sensitive	cosmetic
2	03:18	The user cannot easily figure out which menu point to choose when he needs to enter a new invoice. Chooses "Enter new data" which is the most likely menu point	cosmetic
3	04:48	The task tells the user to add a PDF file to the invoice, though there is no file he can choose he does not choose one. Then upon pressing the "Enter Invoice" button he gets a blank screen with text that explains to him it was not possible to do the task. There is no way to exit the screen without pressing the back button in the browser	serious
4	05:58	After entering another invoice with a file he is still send to a blank screen that tells him it has been entered, though there is no back button and he has to use the browsers back button	serious to critical
5	09:20	The user uses the comments field as a status field thus writing "lager" (stock) in it	cosmetic
6	09:33	Does not add an invoice to the hardware he is entering	critical

7	10:10	Wonders if, when he enters more (detailed) information to one item, that it also will be copied when using the load data button in the "Enter hardware"-screen or if it is only the data he can see	serious
8	12:37	He thinks that if he adds information about invoice he will be able to copy this data when he loads data from the "enter hardware"-screen	serious
9	14:14	The user wants to go back and add an invoice to the item he previously entered and he does not know how to do this. This is however NOT possible	critical
10	14:54	The user knows that he can change the invoice number from the edit screen but could not remember the aaU registration number	cosmetic
11	15:34	He tries to add an invoice to a piece of hardware using the edit function, this is NOT possible, but he thinks that he is done with the task	critical
12	15:48	After he pushes the save changes button he gets sent to an "Equipment updated"-screen. This breaks consistency	cosmetic
13	17:40	The user searches for a serial number, but when the query returns the search information the serial number is not displayed	serious
14	17:50	The data from the search lags an overall structure through out the webpage	cosmetic to serious
15	18:21	Finds the serial number he is looking for using the "Edit"-button, though he would have liked a "show details"-button instead	cosmetic to serious
16	18:30	After the user finds the information on the "Edit"-screen he uses the "Back"-button this does however not return him to his search query but rather the "View all equipment"-screen	cosmetic
17	18:40	The user hesitates to go to the "enter new data" in the menu and "enter new equipment" in the contents window. He is deciding if it is the same thing or not	cosmetic

18	19:07	The user needs to duplicate an item using the load function in the "Enter new hardware"-screen. First he needs to find the aaU registration number by searching the items serial number. He thinks he finds it, but instead he finds the apparatus index number which he then enters into the load function. This results in faulty data being loaded into the form	serious
19	19:57	Thinks he can get a details page by clicking the table	cosmetic
20	22:26	The user searches for an aaU registration number 32 to find the supplier. When the search appears he chooses to use the "edit"-button to be able to see the details and find the invoice number so he can search that to find the supplier. What he does not notice is that the supplier is already written on the "Edit"-screen	serious
21	23:20	Uses the invoice number to find the supplier. But he does not first notice that the query gives him the answer	cosmetic
22	23:52	When the user searches for the invoice he does not have the ability to see/download the file from the search result. Though he notes that the supplier is Wupti	serious
23	24:22	Missing titles while searching for "specific Invoice"	cosmetic
24	25:45	The comments field is very confusing because of the small size	cosmetic
25	26:05	same as problem 12	cosmetic
26	28:05	Cannot remember if it is possible to delete using the search function	cosmetic
27	29:15	While deleting from the search specific item results there is no pop up warning	serious
28	29:17	The menu disappears after he pushes the delete button and there is several error messages	serious
29	32:40	Takes a bit for the user to realize that the phone number needs to be written as one without spaces	cosmetic
30	33:31	After pushing the "Add person"-button the user is sent to a new screen full of error messages	Critical
31	35:18	Has great difficulties with finding a menu point where he can lend out equipment to a person	serious

32	35:48	Tries to lend out equipment through the "Enter new data"-screen	serious
33	36:03	Enters the "lend out equipment"-screen, but it does not make sense to him that this is the right screen	serious
34	36:08	Even though the user notices the header with the writing "lend to user" he does not think that this is the way to do lend out equipment	serious
35	36:46	The user expects a list of equipment that is lend out when he enters the "lend out equipment"-screen	serious
36	38:01	The user notices that there is no feedback after he has lend out equipment	serious
37	38:35	The user still has another perception of the "lend out equipment"-screen. He also comments that he has another understanding of the text in the menu point "lend out equipment"	serious
38	40:01	After the user has returned equipment he says that he would have liked some kind of feedback	cosmetic

E.1 Remaining problems discussion

This section will as section 5.3.3 discuss the rest of the problems missing from this section.

Problem 1: The user uses both big and small letters in his username, but the system is not case sensitive. This would not really be a problem for the user to be able to log into the system it could however be a problem for the security of the system since it would be easier to guess a username. This is only categorized as cosmetic since the user only stops few seconds before he logs in. The security issue could be prevented by changing the system so that it is case sensitive.

Problem 2: The user is not absolutely sure which menu point to choose when he needs to enter a new invoice into the system. After a few seconds he chooses what he thinks is most likely to let him enter a new invoice, which is the "Enter new data" menu point. This is categorized as cosmetic since the user only uses little time to pick the right menu point: To prevent this problem one could choose a better text for the menu point and maybe adding a "title", which is a small pop up box that appears beneath the cursor which can hold different information in this case a help text.

Problem 7: The user thinks that he can enter detailed information about both item and invoice and when he copies the data using the "Load data"-button this information will be copied as well. The problem is categorized serious since it could prevent the user from entering correct data it could also mean that a lot of time has been used without getting the expected result. One is probably able to prevent this by giving the user instructions about how the system works, though one could discuss that the system is

not build properly if the user thinks that this would be intuitively.

Problem 8: The problem is essentially the same as problem 7 and the solution would be the same.

Problem 10: The user cant remember the aau registration number he wants to edit so he has to find it in "view all equipment". This problem is categorized cosmetic since the user knows where to find it and does not use long to find it. This is hard to prevent and can be discussed if this is really necessary to fix one could make a function that writes the last 5-10 aau registration numbers in a list but seems a bit overkill.

Problem 12: The user gets sent to an "Equipment updated"-page which brakes consistency. This is categorized as a cosmetic problem which can be fixed by returning to the screen one came from and adding the "equipment updated" to for example the top of the screen. This however touches a deeper problem in the system which is that it lags feedback on the things that the user does.

Problem 16: After the user uses the back button in the "edit"-screen he gets returned to "view all equipment", he should have been returned to his search query. This has been categorized as a cosmetic problem since the user does not notice the faulty back button. This could be solved by using a "history(-1)" reference to the button instead of the hyper link to "view all equipment".

Problem 17: The user is deciding if the menu point "Enter new data" and the "Enter new equipment"-button on the "view all equipment"-page is the same. This is categorized to be a cosmetic problem that only slows the user a short amount of time. The solution to the problem would be to be more consistent with the names of the buttons and the menu points.

Problem 19: The user thinks he can get a detailed view of an item by clicking it in the table. This is categorized as a cosmetic problem since the user does not use much time on it. The solution would be to add a "details"-button to the search queries.

Problem 20: The user searches for aau registration number 32 and looks in the "Edit"-screen find the invoice number to be able to search for the invoice and find the supplier. This is unnecessary though since the supplier is stated on the "Edit"-screen. This is categorized as a serious problem since it makes the user do unnecessary work. This problem is also hard to solve, but more training in the system would probably ease the use of it.

Problem 21: The user does not notice that his search for the invoice gives him the supplier as well. He uses the "edit"-function to find the supplier. This is categorized a cosmetic because the user finds the information fairly quickly after all. This could also be due to the fact that the user is not experienced with the system, another reason could be due to that the information from the search queries are not consistent.

Problem 22: The user does not have the ability to download/see the PDF directly from the search query. This is categorized as serious since this could take some time before the user figures out that he needs to go to the details page to be able download the PDF. This is categorized as a serious problem this is due to the fact that one cannot go from the search query to the details page. One needs to go to "view all equipment" to be able to get hold of the PDF. This could be fixed by adding the details page to the

search query, maybe even adding a hyper link to the query search so you can download it directly from the search.

Problem 23: After searching for an invoice the user gets a table that is missing titles so it is hard to tell which information is which. The problem is categorized as cosmetic. To solve this one would have to add titles to the search data.

Problem 24: The user needs to edit the comments field of a product but this is very confusing since the field is not very big. This is categorized as a cosmetic problem. This can be solved by changing the comments field into an area (which is easier to assess).

Problem 25: Is the same as problem 12.

Problem 26: The user cannot remember if he can delete through the search function. This is a cosmetic problem. This is probably to to lack of experience with the system, also the user could remember that he could delete from the "view all equipment"-screen.

Problem 29: The user needs to enter a new person into the database. But as he enters the phone number he uses spaces between every second digit, the text field is limited to 8 digits. The users takes some time to find out that the 8 digits need to be written after each other with no spaces. The problem is categorized as a cosmetic since it does not take the user long to figure out that the phone number needs to be written in a certain way. The problem could be avoided with a help text besides the text field that shows how the phone number is written.

Problem 35: The user expects a list of items that are lend out when he enters "lend out equipment". This is categorized as a serious problem since this means that the texts on menu points etcetera are not self explanatory and therefore could mean the user uses an unnecessary amount of time on figuring out how to use the different functions. Could be solved with better explanation for these texts.

Problem 37: The user has another understanding of the "lend out equipment text". Would be solved by correct problems 33-35.

APPENDIX F

PRODUCT BACKLOG

This section will show the features/functions that were in the product backlog. For each sprint, we made a sprint backlog containing user stories from the product backlog. But some of the functions mentioned in the product backlog, were not implemented. These functions could be implemented by having some more sprints, but this was not possible to due the time frame for this report. We managed to complete 4 sprints within our time frame. The sprints started Tuesday morning with a group meeting. At this meeting we would review the sprint that we just had, and what should be done in the following sprint. This is a guideline for how to understand the figures in the following part. In the first column, is the number of the function this indicated the importance of this function, the lower number, the more important it is. The second column is the assignment described. The next column is the initials of the person responsible for this function. The status column was updated when someone took this task. We had 3 different statuses that could occur: Done, In progress and Not implemented. The last column describes the work hours left for this function to be complete. The first number was the estimate we did as a group during the planning game. The following numbers are the new estimates each from Tuesday to Monday.

	No.	Task	Person in Charge	Status	Remaining work hours					
					Est.	Tue	Wed	Thu	Fri	Mon
Sprint Backlog 1#	1	As an user i should be able to log into the system in order to use it.	DN + CHP + MKN	Done	12	12	2	2	1	0
	9	As an admin i should be able to keep track of all EQ in order to keep status when ever needed – rented as well as in stock!	DRF	Done	2	2	2	0	0	0
	#	Design of webinterface	DN + DRF	Done	12	12	8	8	0	0
	#	Install of Development programs	ALL	Done	2	2	0	0	0	0
	#	Create tables in DB	MS	Done	2	2	1	2	0	0
	Total				30	30	13	12	1	0
Sprint Backlog 2#	2	As an admin i should be able to enter new EQ in order to register new EQ in the database.	MKN	Done	18	18	12	8	7	0
	3	As an admin i should be able to delete EQ in order to keep the database updated.	SMS	Done	5	5	5	5	2	0
	#	Dynamic Menu	DRF	Done	5	5	0	0	0	0
	7	As an admin i should be able to print a list of all EQ in order for "SKAT" can take a look in our assets and see that our accounting is correct.	DRF	Done	4	4	1	1	0	0
	6	As an admin i should be able to search the database in order to easy and logically find all EQ, persons etc.	MKN + DN	Done	22	22	22	22	10	0
	Total				54	54	40	36	19	0

Figure F.1

Sprint Backlog 3#	4	As an admin i should be able to edit EQ data in order to keep the database up to date.		Done	9	9	5	2	1	0	
	5	As an admin i should be able to easily enter identical EQ (eg. A bunch of new monitors) in order to avoid entering the same data multiple times.		Done	27	27	20	20	2	0	
	8	As an admin i should be able to find specific and detailed information about the EQ in order to find the Guarantee, defects, OS etc.		Done	7	7	5	3	0	0	
	Total					43	43	30	25	3	0
Sprint Backlog 4#	10	As an admin i should be able to associate the EQ to a location or a loaner in order to know if the rented item is at a location at the university or its rented to a private location.	MKN	Done	8	5	4	4	4	0	
	12	As an admin i should be able to accept returned EQ in the system in order to administrate the EQ.	DRF	Done	5	5	0	0	0	0	
	11	As an admin i should be able to edit "Repair Status" on EQ in order to keep the database updated.	DRF	Done	2	2	2	0	0	0	
	Total					15	12	6	4	4	0

Figure F.2

Unimplemented Features	13	As an admin i should be able to send out reminders to people who have rented some EQ in order to get the EQ back in time.	Not implemented	14
	14	As an employee or student I should be able to find information about the EQ I am able to rent in a easy way in order to select the correct EQ for my needs.	Not implemented	2
	15	As an employee or student I should be able to book EQ in order to be able to rent EQ i need.	Not implemented	15
	16	As an admin i should be able to get a list of booked EQ in order to see status of EQ and who has booked it.	Not implemented	5
	17	As an employee or student I should be able to keep track of my rented EQ in order to know when to return my rented EQ, and to keep track of what i have rented.	Not implemented	5
	18	As an employee or student I should be able to cancel booked EQ in order to update the list of EQ i need.	Not implemented	5
	Total			46

Figure F.3

APPENDIX G

CD

The enclosed CD-rom contains the system files, including the html and php files along with the mysql database files. A README file, that describes how to setup the system at another location, is appended. An example of how the system should be build according to MVC is also included. The CD also contains the initial interview with the customer as an audio file and a video of the usability test (Notice: They were carried out in Danish). The customer also signed an consent form in order for us to use the material in the report.

After the project completion, until the examination, the website will be available on the internal local area network of the Institute of Computer Science, at <https://172.25.24.201>. In order to reach it, it is necessary to use the same subnet, which means that it will not be available from the wireless network, or Virtual LANs different from the one the website uses. As an alternative, it is possible to use “rdesktop 172.25.24.201” from the application servers of the institute, eg. bart.appl.cs.aau.dk, to get a remote desktop connection to the server, in order to use the system.

The remote desktop login is:

- User: b403a
- Password: google42