

Bachelor's thesis
Information Technology
Internet Technology
2015

Juuso Kujanpää

DEVELOPING A BOOKING SYSTEM

–Case: TUAS Cisco laboratory



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology

2015 | 40 pages

Patric Granholm, Vesa Slotte

Juuso Kujanpää

DEVELOPING A BOOKING SYSTEM

The purpose of this thesis project was to develop a new browser-based online booking system for the Turku University of Applied Sciences Cisco Laboratory. The new system was needed because the current system was outdated and system functionalities and user interface requirements needed to be updated. The development for the new system was started with a new list of requirements from the laboratory staff and taking into consideration the basic idea from the current system. The main goals were to improve the usability and the security of the system. PHP5 and MySQL were used as development tools for the server side development and Twitter Bootstrap front-end framework including HTML, JavaScript and CSS for the browser side. This thesis work concentrates on how to develop a booking system for a specific situation and how to achieve the best results on front and back-end.

KEYWORDS:

Responsive, web development, Twitter Bootstrap, Database

CONTENTS

LIST OF ABBREVIATIONS (OR) SYMBOLS	5
1 INTRODUCTION	6
2 TUAS CISCO LABORATORY	7
3 BOOKING SYSTEMS	8
4 THE CURRENT BOOKING SYSTEM	10
4.1 Booking in the current system	11
4.2 Problems in booking	13
5 THE NEW BOOKING SYSTEM	15
5.1 Why develop a new booking system	15
5.2 Design principles	15
6 DEVELOPMENT TOOLS	16
6.1 PHP and MySQL	16
6.2 Bootstrap	16
6.3 Debian and Apache	17
6.4 Platform for the application development	18
7 DEVELOPMENT PHASES	19
7.1 Collecting requirements	19
7.2 Database	20
7.3 Making a booking	20
7.4 Booking form page	23
7.5 Form handling and security	24
7.6 Verification email	25
7.7 Handling resources	25
7.8 Admin side	26
7.9 Handling holidays and sick leaves	29
7.10 Using cron for swipe and reminding	30
7.11 Attendance page	31
7.12 User accounts and login	31

7.13 Password protection	32
8 CONCLUSION	34
REFERENCES	35

PICTURES

Picture 1. Front-page of the current booking system.	10
Picture 2. Bookings-page of the current booking system.	11
Picture 3. Dropdown menu to choose the day in the current booking system.	11
Picture 4. Table showing the bookings and available tables in the current booking system.	12
Picture 5. Booking form and captcha in the current booking system.	13
Picture 6. Picture of the database structure taken from phpMyAdmin designer.	20
Picture 7. New system front-page view on mobile and desktop.	21
Picture 8. User view on the left admin on the right.	22
Picture 9. Booking form page in the new system.	23
Picture 10. Sweetcaptcha to block automated bookings and spam.	24
Picture 11. Rules text editor when logged in.	27
Picture 12. The exceptions page has the same structure as the other resources pages.	28
Picture 13. All bookings are shown on the bookings page.	29
Picture 14. Structure of hashed password (Source: http://php.net/manual/en/images/2a34c7f2e658f6ae74f3869f2aa5886f-crypt-text-rendered.svg)	33
Picture 15. Matching the password with hash_equals function.	33

LIST OF ABBREVIATIONS (OR) SYMBOLS

PHP	Programmable Hypertext Preprocessor. Server side scripting language.
JavaScript	Browser side scripting language.
MySQL	Open source database system.
phpMyAdmin	Open source database management interface for MySQL database.
Bootstrap	Open source front-end framework including CSS styles and JavaScript library.
Responsive	In internet terms a service that scales on every screen size.
CMS	Content Management System. System for fast creation of websites and content management.
IDE	Integrated Development Environment. Application for software development.
ERP	Enterprise Resource Planning.

1 INTRODUCTION

Since the Internet's rapid growth in the past few years more and more daily tasks and operations are done online using computers and mobile devices (Internet World Stats, 2015). One of these tasks is making a reservation or booking for different services.

Many types of booking systems exist offered by various software developers and companies and there are endless situations where they are needed. There are booking systems for restaurants, hospitals, barber shops and schools examinations etc., booking systems are everywhere. Developing a good booking system that would work on every single situation, device, and business can be very complicated because every situation is somehow different.

Some companies offer their booking system as one for all and it might be integrated to the company's existing ERP system. Often, when one system is applied in every situation, the result is more or less a compromise. If the booking system is very flexible, it is often unnecessarily complicated for the end user and for the administration. This means unnecessary clicks or touches and administration side with features that are not actually needed. Using a booking system should be logical and easy to satisfy all the users on both sides of the system.

This thesis project concentrates on how to develop a new booking system for the Turku University of Applied Sciences (TUAS) Cisco laboratory. It considers what issues the current system had and what issues are faced when developing a modern booking system and then it explains step-by-step how this booking system was developed.

2 TUAS CISCO LABORATORY

The TUAS Cisco laboratory is a laboratory environment where students of TUAS complete Cisco courses' laboratory exercises and skills exams offered in the Cisco Networking Academy (<https://www.netacad.com/>). In the laboratory exercises students use Cisco network devices, such as routers and switches, to practice their networking skills. The devices needed in the laboratory exercises vary by course and type of exercise.

Students work mainly in pairs but it is also common to work individually or in groups of three. Each group or individual reserves one table from the laboratory. Every table has three desktop computers and other devices needed are picked up from the storage room.

Booking for the laboratory table is done by using the online booking system, one to eight days before going to the laboratory. Students book the table, always for one full session minimum. Sessions usually last approximately 4 hours and are held one in the morning and one in the afternoon. The laboratory can be open for zero to five days in a week depending on the school's budget and holidays. Students are able to cancel their booking, by using the booking system, until a specified time in the evening of the day prior to the booking. After the cancellation window closes, the students have to inform the laboratory staff via email.

In the laboratory students are guided by professors or laboratory staff qualified for certain courses only. This means laboratory is only available for the courses which the staff in the laboratory are qualified to supervise. To keep a record of exercises completed, each student has a laboratory book where a qualified staff person signs after each exercise is completed.

3 BOOKING SYSTEMS

Booking systems are used in more and more places all the time. Even when buying something online the item is booked for the prospective buyer but if not generalizing too much, booking systems could be limited to systems where some resources are booked for some time range.

Booking systems can be quite simple or on the other hand, very complex when looking at their functionalities. Most of the booking systems functions do not become visible to the user or the administrator. These functions could be, for example, automated reminders to users and calculations to avoid overlapping bookings. The more resources and other changing factors a system has, the more processing needs to be done.

A booking system that holds table bookings for a restaurant can be made very simple. The customer chooses a table of as many seats she/he wants at a specific time and date. There are two factors to take in consideration, the size of the table and time. Each restaurant has its own average time the customers stay in and for how long the table has to be reserved. The most simple possible database table for holding the bookings for restaurant reservation has columns for table size, time/date, and the booker name.

A good example of a complex booking system could be for a beauty salon which offers various types of services with different duration and then different employees capable for different services. Unlike in the restaurant example, here are multiple different durations and the system has to take these in consideration. There are two solutions for handling the time issue. The first solution would be which part of the day or a week is reserved only for certain service or services of the same duration. In this way the system does not have to make calculations of which service would fit in the certain time slot and which not and the employees' work time can be used very effectively when there will be no extra free time between customers. The problem in here is that what if the customer wants a different service, for example her/his nails done, instead of a haircut that system is

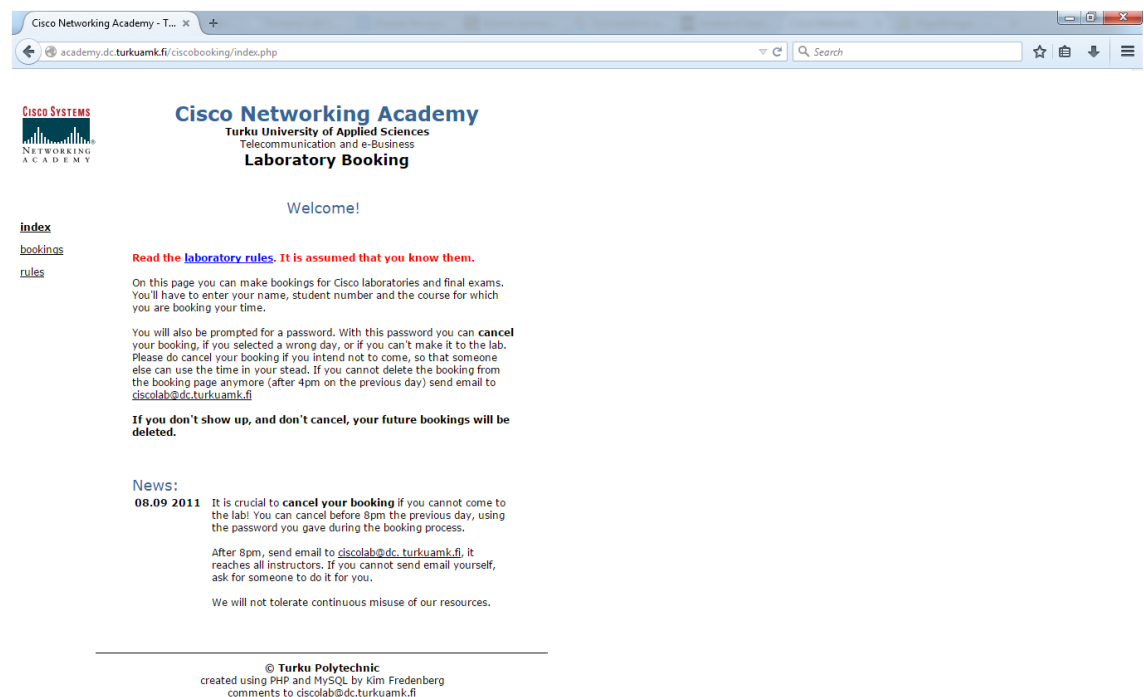
offering. This will make the salon empty just because the system was not dynamic enough. A dynamic system which calculates which services fit in the time slot will be more flexible for the customer, but might end up with employees having unwanted free time when they just wait for the next customer.

The decision between different options has to be made carefully depending on the situation. A good booking system saves valuable time for the company and therefore resources are left for more important tasks. A badly developed booking system, on the other hand, can just take a lot of work from employees to maintain and might even scare the customers away. For customers, a good booking system is accessible with every device and it is fast and especially easy to use.

4 THE CURRENT BOOKING SYSTEM

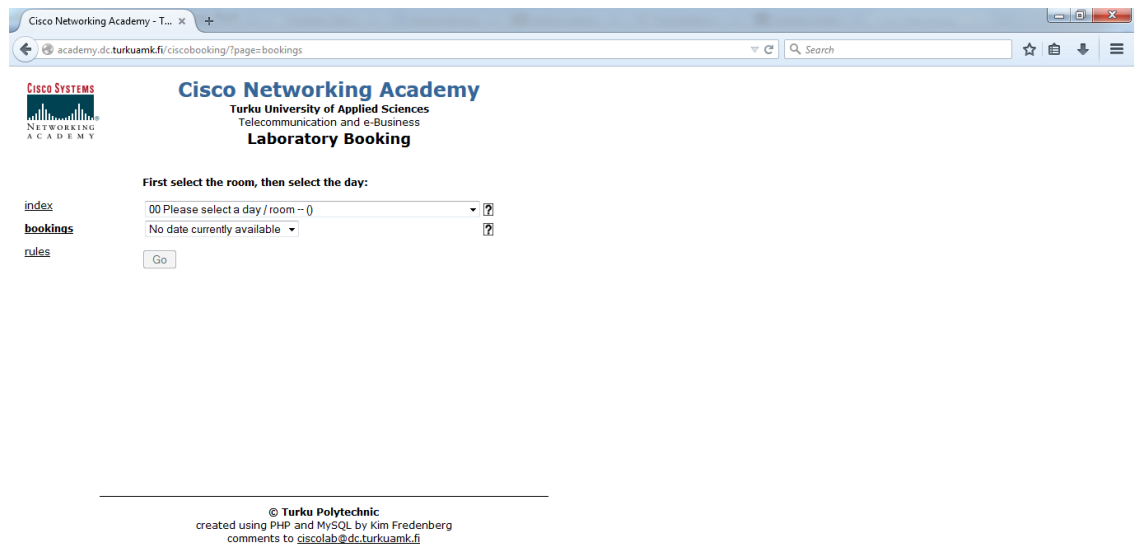
To modernize the booking making and keep better records, the first booking system for the TUAS Cisco laboratory was developed more than 10 years ago. Since it had been developed a long time before smart phone use started to increase, it was designed for desktop computer use only and had no responsive features.

On front page of the system, the user can see news feed and general instructions (Picture 1). The front page also has links to rules and bookings page. All the content on the page is always aligned to left.



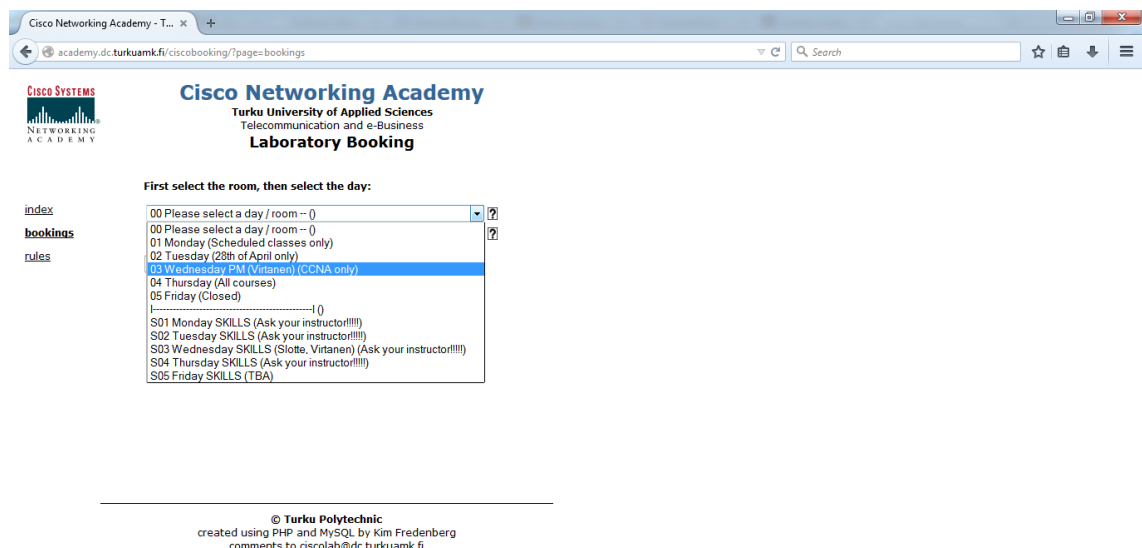
Picture 1. Front-page of the current booking system.

4.1 Booking in the current system



Picture 2 Bookings-page of the current booking system.

Booking starts on the “bookings” page (Picture 2). From “bookings”, the user is asked to choose a room and a date from two separate dropdown menus (Picture 3). The “room” is actually a day of the week. Then the user submits the form by clicking on the “Go” button.



Picture 3. Dropdown menu to choose the day in the current booking system.

On the next page, the user needs to choose a table (Picture 4). The user is able to choose multiple tables but there is actually never a need for this.

The screenshot shows a web browser window with the URL `academy.dc.turkuamk.fi/ciscobooking/index.php?page=bookings`. The page header includes the Cisco Systems logo and the text "Cisco Networking Academy, Turku University of Applied Sciences, Telecommunication and e-Business, Laboratory Booking". The main content area displays the date "Tuesday the 28 of Apr, 2015" and the room "Room: 02 Tuesday 28th of April only".

On the left side, there are links for "index", "bookings", and "rules". The main table is structured as follows:

Time: 8.15-11.30				
Tbl	Name	Sem	Lab	Book
1				<input type="checkbox"/>
2				<input type="checkbox"/>
3				<input type="checkbox"/>
4				<input type="checkbox"/>
5				<input type="checkbox"/>
6				<input type="checkbox"/>
7				<input type="checkbox"/>
8				<input type="checkbox"/>
9				<input type="checkbox"/>
10				<input type="checkbox"/>
11				<input type="checkbox"/>
12	Skills	-- CANCELL...	Ossi	delete
13	Skills	-- CANCELL...	Ossi	delete
14	Skills	-- CANCELL...	Ossi	delete
				book
Time: 12.15-15.30				
Tbl	Name	Sem	Lab	Book
1				<input type="checkbox"/>
2				<input type="checkbox"/>
3				<input type="checkbox"/>
4				<input type="checkbox"/>
5				<input type="checkbox"/>

Picture 4. Table showing the bookings and available tables in the current booking system.

After choosing the table, the user again needs to click on a button now called "book". Then finally comes the booking form (Picture 5). The form asks for names of the participants (2-3), name of the course, which labs the users are going to do, password for cancelling the booking, repeating the password and last the "CAPTCHA" to avoid bookings done by automated scripts or bots.

Cisco Networking Academy
Turku University of Applied Sciences
Telecommunication and e-Business
Laboratory Booking

Date: 28.04 2015
Time: 8.15-11.30
Room: 02 Tuesday

Names	Std. nums.
1. <input type="text"/>	<input type="text"/>
2. <input type="text"/>	<input type="text"/>
3. <input type="text"/>	<input type="text"/>

Select lab:
CCNA - 1 3 - CCNA 1 labs

Enter password:

Password again:

Enter the text from the CAPTCHA image into the textfield:
NTHMVH

Submit the booking:

* required fields

Picture 5. Booking form and captcha in the current booking system.

4.2 Problems in booking

In aspects of usability and simplicity, there are multiple issues in the way the booking is done in the current system. The issue on the first page when the user enter the system is that the user should read the instructions every time when booking. When users usually book multiple times per each month, this can be considered to be totally unnecessary.

On the booking page, the users choose the day of the week and date separately, and if they want to do skills exam, they have to choose another line from the dropdown that has “skills” on it. This seems unnecessarily complicated.

Choosing the table has really no meaning because there are not actually different types of tables in the system or in the laboratory because the tables are identical and they are not numbered.

On the booking form page, the user needs to fill in the names of two users while there is no rule that says that the user cannot come alone. This means the second field is often filled with some nonsense just to get the form approved. Another field that is not actually in use is the field for which labs of the course the user is

going to complete. This would provide information if the system would reserve the devices for the user but the current system does not and also the devices used in different labs of the same course do not differ so much that it would make any difference when booking.

5 THE NEW BOOKING SYSTEM

5.1 Why develop a new booking system

The idea for the new booking system was brought in mainly because of the security issues that the current system was having. The most important issue was that users were able to make bookings outside the allowed timeframe. In addition, the idea behind the current system was considered outdated and needed more resources. Teachers also wanted more features automated. From the users' point of view, the user interface needed to be updated to match the standards of modern web design.

5.2 Design principles

The development for the new booking system needed to start from the beginning but still keeping in mind the basic functionalities of other booking systems and keep in mind that the problems of the current system will not be repeated. The key characteristics for the new system were considered as security, usability and automation. To accomplish a modern design which satisfies the requirements and functionalities what is expected at the moment, mobile-first and minimalist design principles were considered. In short, minimalist web design means cutting all the nonessentials elements and details to minimum (Knight, 2009).

6 DEVELOPMENT TOOLS

The tools that are used on a web development projects have a great affect to the projects development duration, degree of difficulty, debugging and, future development. For the future development recognizability of the development tools is very important. Choosing generally well known tools helps to keep the systems learning curve as low as possible for the future developers.

When choosing tools to be used to develop the system the following things were considered:

- Size of the project
- Type of the project
- Recognizability of the tools
- Knowledge of the author

6.1 PHP and MySQL

The current system was developed using PHP and MySQL and since the author was familiar with them PHP and MySQL were also chosen for the new system. PHP and MySQL are also very popular worldwide and many known sites or CMS systems like Facebook, WordPress and Drupal use them (w3schools.com, 2014). No frameworks were used because using some specified back-end PHP framework such as ZEND or Yii would make it much more difficult for new administrator to learn the system, except of course if they happened to know the framework already.

6.2 Bootstrap

For front-end development the Bootstrap framework (Twitter Bootstrap) was used. Bootstrap is an open source front-end framework and its code is licensed

under the MIT license. It consists of predefined CSS-styles for buttons, tables, forms etc. and some JavaScript extensions. Since version 2.0 it is based on grid structure and has responsive design features. Version 3.0 introduced the flat design which is the current standard on all the major websites such as “google.com” and “facebook.com”. Bootstrap is the most popular front-end framework with good documentation so it is very good for the future maintaining and development of the system (Gerchev, 2014). (getbootstrap.com, 2015)

Using Bootstrap framework makes the front-end development much faster and easier since its responsive features and premade styles made for the developer. Since there were no special requirements for the design in terms of colors and shapes, Bootstrap gives a clean basic design that pleases the eye of the user.

For structuring the page. Bootstrap uses the so called “grid” system that divides the screen into 12 vertical grids. The amount of grids and many other parts of bootstrap are customizable. Customizing is easy to do on their website before downloading the package and it is easy to override the bootstrap rules for CSS classes by adding your own.

6.3 Debian and Apache

As a development server, Debian with an Apache web server was used. Debian is a Unix-based computer operation system used widely in server computers and for hosting web-pages. It is also one of many Linux distributions (Debian, 2015). Apache is a HTTP server application to run dynamic websites such as the ones made with PHP. Apache also includes a MySQL database and a phpMyAdmin browser-based application for database management.

This set up was chosen because the final production server was most likely to be the same.

6.4 Platform for the application development

The development platform was NetBeans which is a free development environment for Java, PHP, HTML5, and C/C++. Netbeans runs on Windows, Mac OS X, Linux, and Solaris. (Netbeans, 2015) It has a pre-installed module for FTP connection to modify code on a remote server and it also has a version control built in which was helpful in debugging.

7 DEVELOPMENT PHASES

The development of the system was quite straight forward and followed the “development phases” path (below). Since this was the authors first web application in this size improvements to security and functionalities were made along the development. After every major step taken in the development the functionality was tested and re-evaluated.

1. Collecting requirements
2. Choosing development tools
3. Designing and building the database
4. Designing user pages
5. Designing admin pages
6. Creating forms and form handling
7. Adding sign in functionality
8. Adding automated functionalities
9. Improvements to usability and security

7.1 Collecting requirements

The system development started by collecting all the requirements for the system keeping in mind that some features might be changing during the development. Requirements were mainly given by staff and but discussed together if they are possible, reasonable or if they fit within the time-frame of this thesis.

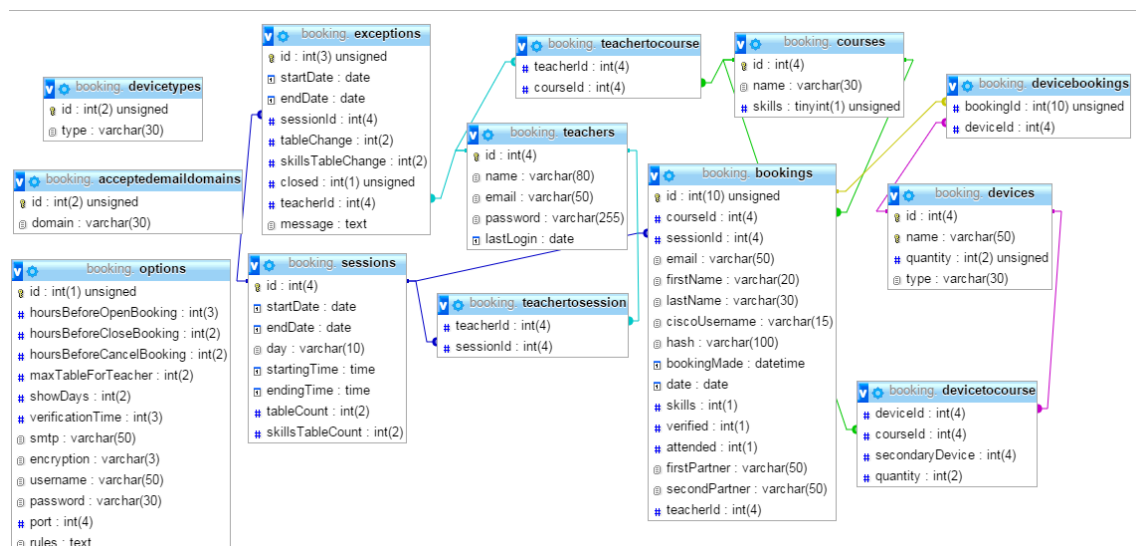
List of requirements:

- User pages will be optimized for desktop, tablet and smartphone
- The system has to reserve a table, devices and a teacher when booking is made.
- The system has to automatically send emails to users as a reminder and in a case of cancellation.

- The system includes a feature to follow up if users have attended the laboratory session they booked for.
- Exceptions such as laboratory cancellations can be made for Laboratory sessions.

7.2 Database

The database consists of 13 different tables of which 3 are separate tables with no relation to any other tables (Picture 6). Six of the tables have relations with each other directly or through a junction table.



Picture 6. Picture of the database structure taken from phpMyAdmin designer.

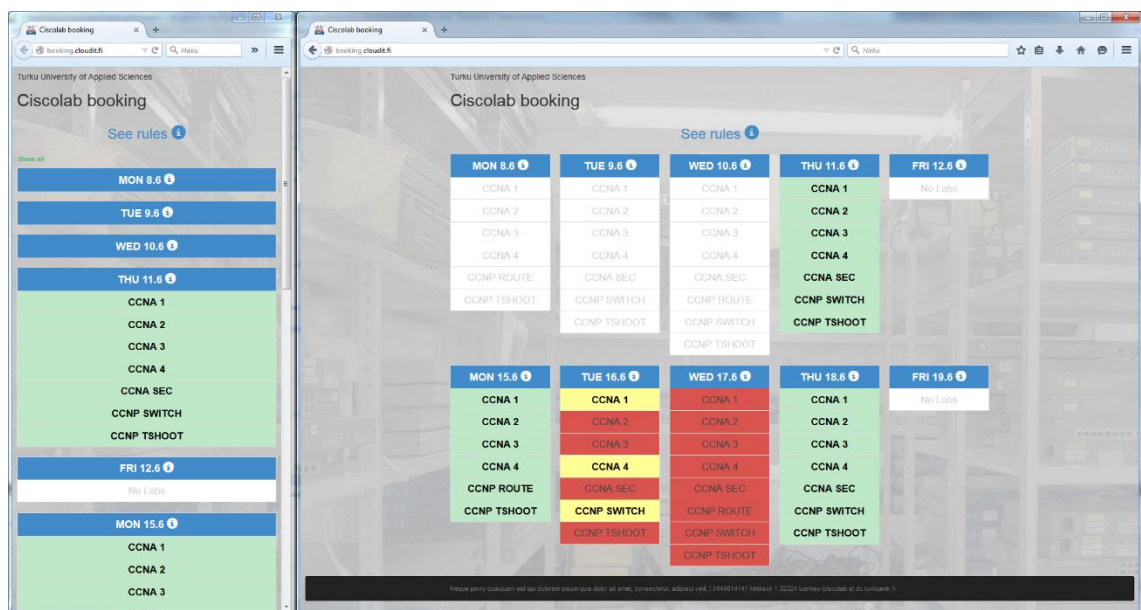
7.3 Making a booking

This part was considered very important because it is the most used functionality of the whole system. Bookings are made many times a day on average and a person who is coming to laboratory every time it is open, is using this system almost every day. The user pages are designed with the mobile-first design method meaning first design the mobile view and then the desktop view (Sexton, 2015). It was also assumed that in the future most of the bookings will be made by using mobile devices and mainly with smart phones.

The main idea was to keep the user pages and booking process as simple as possible. To achieve this, the number of mouse clicks and/or touches was tried to keep to a minimum. The final structure for making a booking was to divide the process in four steps.

1. Choose date
2. Choose course
3. Choose session
4. Fill out the form

These steps are divided on two pages. On the first page, the user chooses the date wanted and then the course wanted. Then, on the second page, the user chooses a session or multiple sessions, fills out the form, and submits it.

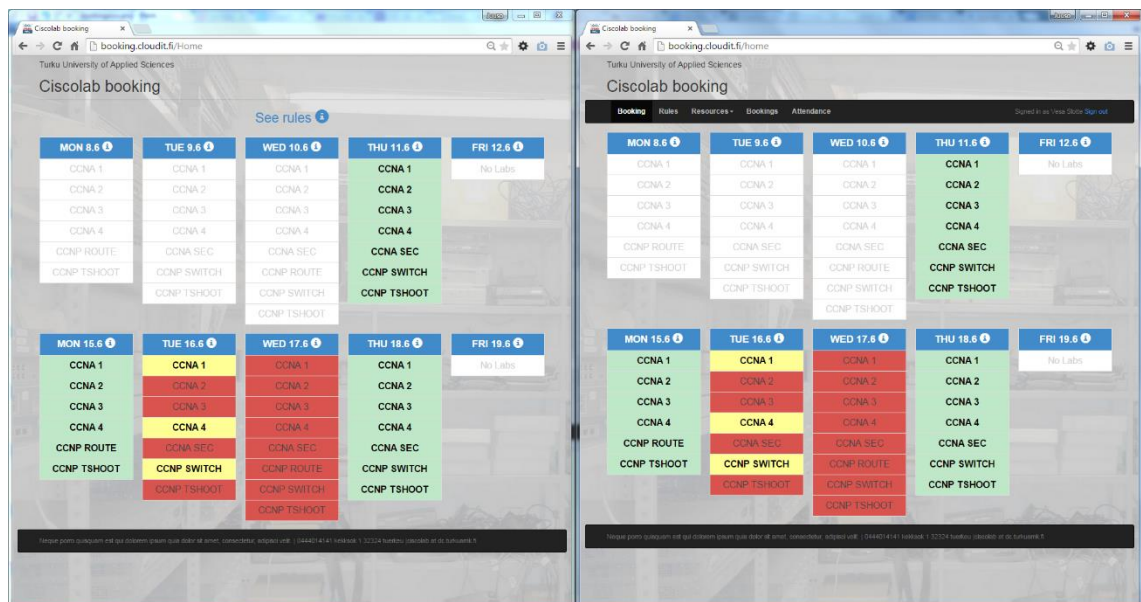


Picture 7. New system front-page view on mobile and desktop.

The first page had to show enough information for the user so that the user is able to go to the second page directly from here. Information had to be shown directly without any additional clicks by the user. This was done by showing the data on 1*x tables where the date was as the column header and the course as a table row item. Each day was shown on a separate table to achieve the desired

scalability to different screen sizes. On a smart phone the user sees everything in a vertical list since the tables set on top of each other (Picture 7). The desktop view offers a five-day week schedule. To get five tables fit correctly on each row, Bootstraps original 12 grid structure was changed to 15, each table getting the width of three grids. This is the only modification carried out to the bootstrap settings.

By using different colors the system informs the user if there are places left (green) or not (red), if the laboratory is closed (white “No labs”) or unavailable (white) and if there are only tables left for skills exam (yellow).



Picture 8. User view on the left admin on the right.

In the beginning bootstraps navigation bar was used for navigating between the three pages “front”, “form” and “rules” but later on the bar was hidden from the user side in the part of minimalizing the design (Picture 8). Plain link to the rules page was added to the front-page and “Back”-links to “rules” and “form”-pages.

7.4 Booking form page

Turku University of Applied Sciences
CiscoLab booking

[← Back](#)

Booking for: CCNA 1

Tuesday 26.06.2015

Lab 08:15 - 11:45
 Skills exam 08:15 - 11:45
 Lab 12:15 - 15:45
 Skills exam 12:15 - 15:45

Firstname:

Lastname:

Cisco username:

Email:

We'll send you an email to verify your booking!

Lab partners

First partner(optional):

Second partner(optional):

Verify your mail existence
Drag mister worm to his apple.

Powered by userCaptha

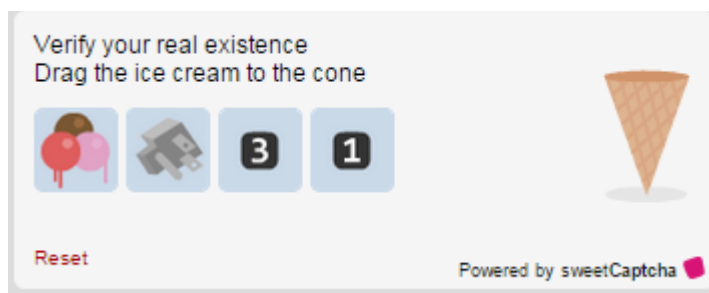
Bookings made:
No bookings made yet!

Negue porro quisquam est qui do lorem ipsum qua dolor sit amet, consectetur, adipisci velit. | 0444014141 kekko@tutkimus.fi | 0444014141 kekko@tutkimus.fi

Picture 9. Booking form page in the new system.

After clicking on the desired course on the front page, the user goes to the booking form page where the session information is shown to the user (Picture 9). The user chooses the session or multiple sessions from the list of checkbox options. If the user wants to book the laboratory for the whole day, s/he can choose both, morning and afternoon at once, and therefore fill out the booking form only once unlike in the current system where user has to start the booking process from the beginning. If the session happens to be full, the checkbox field will be disabled and the user is not able to choose it.

If there is a skills exam available for the session, the course system restricts the user from choosing a normal lab and skills exam for the same session. Furthermore, only one skills exam for a day is allowed.



Picture 10. Sweetcaptcha to block automated bookings and spam.

To block automated bookings and spam captcha is added in the end of the form (Picture 10). Sweetcaptcha was chosen because it is based on image identification and not in the complicated unreadable texts. Sweetcaptcha also supports touchscreen input. Blocking automated bookings is something that has to be observed and probably updated once in a while during the system is in use because hackers are all the time developing new ways to automatically validate captcha.

7.5 Form handling and security

When a user submits the booking form Bootstraps JavaScript validates if email format is correct and notifies, if not before the form is sent to the server. The browser checks if any of the required fields is empty. These functions are done on the user side and are a way to improve the usability of the system. These functions do not improve security because the user is able to modify the html file and JavaScript functions.

To protect the system the form input data has to be validated by PHP. All the input data is at least put through three PHP built-in functions to discard unwanted characters and data. The trim function takes all the additional spaces from the beginning and the end of the string. Then the stripslashes function ensure that no slashes are input and, finally, the last htmlspecialchars function converts html characters such as "<" and ">" from the input to "<" and ">". This protects the system from users injecting HTML or JavaScript code. For inputs where it is known that only normal alphabets are needed we can check with preg_match function for the characters allowed. To ensure that the user has input a valid email

address, PHP has a function `filter_var` that compares the user input against the syntax defined in RFC 822 (More info: <https://www.cs.tut.fi/~jkor-pela/rfc/822addr.html>).

7.6 Verification email

After data validation is completed, the system sends the information to the database and sends the user a verification email. This email gives the user options to verify or cancel the booking. Verification has a time limit set in the system options. If the verification time runs out, the system removes the booking automatically. The user is able to cancel the booking up to a certain time, one day before the booked session. This means that if the booking is made for Monday 24th of January the booking can be cancelled at some time on Sunday 23th of January. This time is also set in the system options.

To verify the booking, the user clicks on the “Verify”-link in the email which takes the user to verification page with a variables set for the ID of the booking and a hash. The hash is created when the booking is added to the database. The system then looks for matching hash and booking ID combination and verifies the booking if one is found. The hash is created to authenticate the verification and especially the cancellation. Without the hash deleting other users booking would be fairly easy and the system could be misused. The Verification email can be considered as an updated way to manage ones bookings compared to the current system’s password protection.

7.7 Handling resources

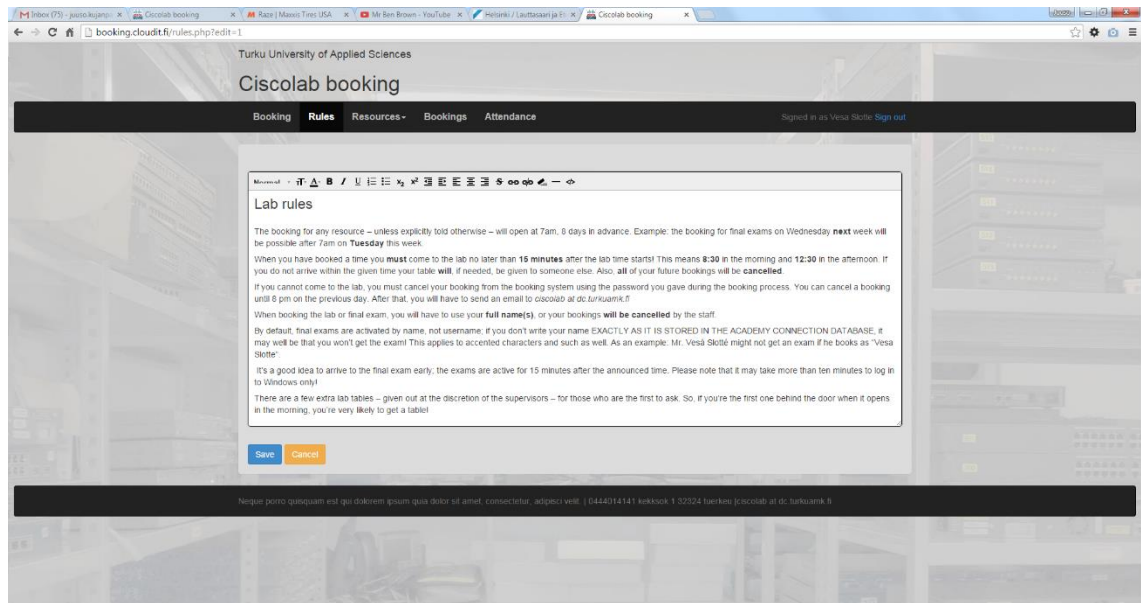
After the booking form is submitted and validated system checks once more that the needed resources are available before feeding all the information to the database.

To check if there is a free table, the system compares the number of tables set for the session in the “sessions” table to the entries in the “bookings” table. The number of tables are different for the skills exam and for laboratory exercises. Device availability is checked by comparing the “devices” table information to the “deviceBookings” table. The information which devices are used on each course is found in the “deviceToCourse” table. This table also has columns for quantity which indicates how many of these devices are needed and for secondary device if it is possible to use some other device for the course. For the teacher resource system checks if the teacher is available and that the teacher does not have too many tables to supervise for that date and session. The limitation for how many tables one teacher can hold at one session is found in the “options” table and it is compared to the amount of entries with this particular teacher in the “bookings” table. If the all the resources are available, the data is set to the database.

7.8 Admin side

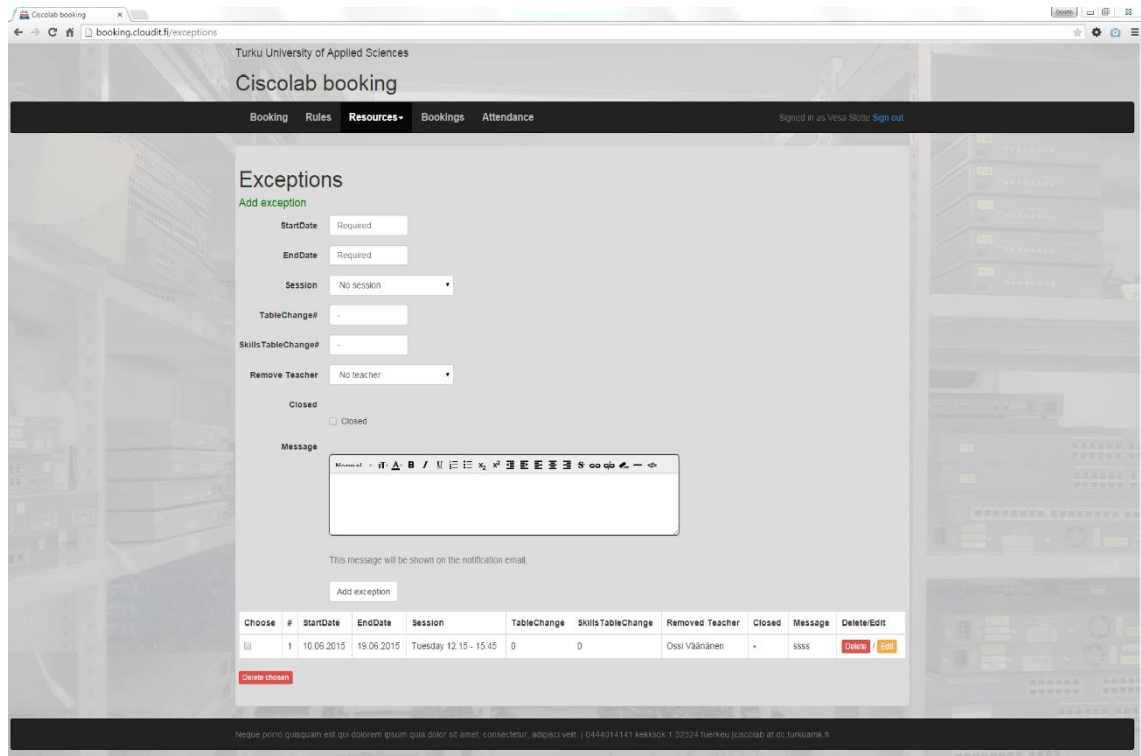
The admin side is protected from users and unauthorized users by holding the login information in a PHP session variables. There are no login-links or buttons on the page because basic users have no reason to login. This reduces the amount of false login attempts. The admin side has multiple pages, therefore for the navigation bar is shown after a successful login.

The navigation bar contains direct links to front-page, rules, bookings and attendance pages and then a dropdown menu for resources. In addition to resources under the resources dropdown also exist options and exceptions because these are all in away part of the settings of the whole system. After the testing period the final division of the navigation bar can be changed if needed.



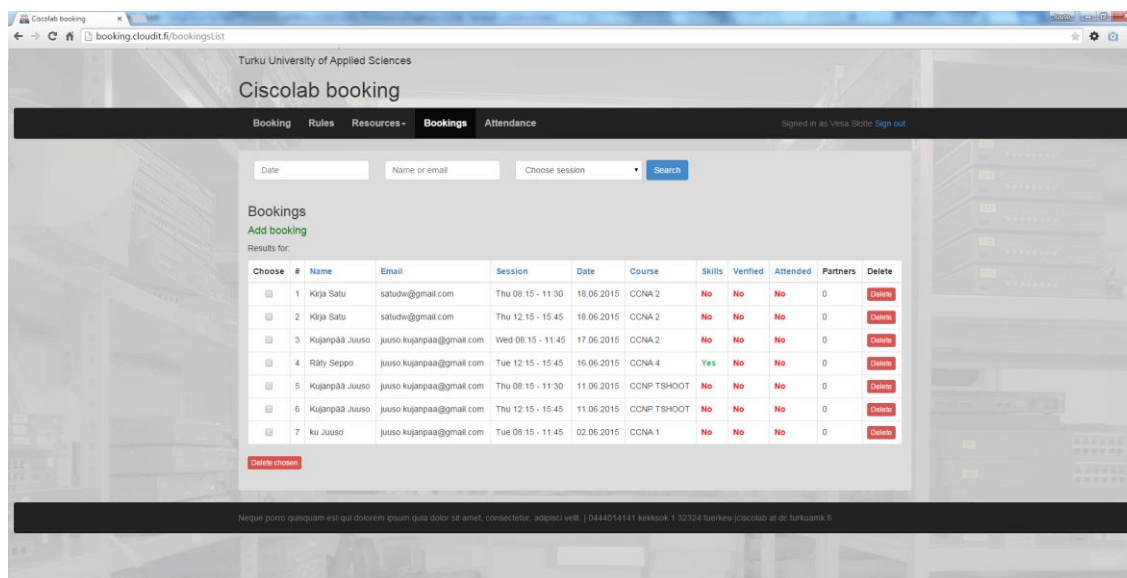
Picture 11. Rules text editor when logged in.

The front-page and the rules page are the same as in user side except on rules page, the admin can see the “Edit” button on the bottom of the page (Picture 11). When the edit button, which is actually a link, is clicked, the page is reloaded with the variable “edit” set to 1 in the address bar. This enables the JavaScript text editor plugin which now shows the text in the editor. After changes are made, the text can be either saved or the process can be cancelled.



Picture 12. The exceptions page has the same structure as the other resources pages.

The pages “Teachers”, “Courses”, “Sessions”, “Devices” and exceptions all have the same structure (Picture 12). First, the form and then following a table listing already added content. The form is first hidden when coming to the page and can be easily shown by clicking the “Add *Resource*name” link when adding resources or exceptions. Editing and deleting resources is possible in the table.



Picture 13. All bookings are shown on the bookings page.

On the bookings page, the laboratory staff are able to browse, delete, and search the bookings in the system (Picture 13). The staff can also add bookings from here without any restrictions, in other words, the system skips the resource checking procedure and not all information is required.

The search can be carried out by date, session and the name or the email of the booker. The search function takes the information from the search form and makes a query to the database. The table sorting also uses a simple query to the database. This is not the best way to do this since the information is already fetched and there would be no need for additional queries but because there are not so many users on the admin side it was thought it was better keep the code simple.

7.9 Handling holidays and sick leaves

Since the system schedule works so that the staff creates the so called sessions for a week and then these sessions are repeated every week within the time session is “alive”, the system needed a way to handle the exceptions that might occur once in a while. These exceptions are holidays, sick leaves, meetings or other

obstacles that make the laboratory space, tables or someone or all from staff personnel unavailable.

To enable exceptions, a page called “exceptions” was created. This page is again a form where the staff is able to add exceptions to the schedule. The staff is able to remove personnel, add or remove tables, or close the entire lab for a specified period of time. For sick leave situations or other last minute exceptions where any of the resources have already been reserved, the system checks if there are bookings made that the exception affects and sends a cancellation notice to the students via email. In exceptions where a staff member is removed, the system first tries to assign another person from the staff to the booking but if it is not possible then sends the notification and cancels the booking.

7.10 Using cron for swipe and reminding

To run automated scripts to remove unverified bookings from the database and send reminders, cron daemon is used. Cron daemon is a service in UNIX-based computer operation systems used for scheduling tasks. Tasks that need to run are set in the crontab file. The crontab file is a simple text file where each task is represented on its own line. Each line tells the service what needs to be run and when. (cognition, 1999)

To swipe unverified bookings from the database, cron is made to run a file called “swipe.php”. This file first checks in the system options what the time limit for verifying the booking is and then subtracts this from the current time using “strtotime” function. After this, a simple query to the database is done to delete all entries where booking is done earlier than the result of the subtraction. This task is run every minute.

To remind users of their booking for the next day, cron is set to run the “sendReminder.php” file at 18.00 every day. This file goes through every booking for the next day and sends an email to the email addresses. From this email, the users are still able to cancel their booking.

7.11 Attendance page

The page to manage attendance of the laboratory sessions does not exist in the current system and was not in the original list of requirements. Until now, this has been done on paper meaning that the staff prints the list of people booked the laboratory and then just marks with a pen if this person or group has arrived. This piece of paper will end up in trash or in a pile of paper that no one will look at afterwards. Now this information will be in the database and all the wanted statistics can be extracted out of it.

Differing from the other admin side pages the attendance page was designed to be used on a tablet device because it was thought to be good to use the existing tablet device for this rather than a laptop or especially a desktop. This page also works as a list of bookings made for the day. The staff can see the name and email of the bookers and then course name, partners, whether the booking is made for skills exam and when the booking was done. The information about when the booking is done is important because if there is some issues with some resources then a fair decision can be made on whose booking will be cancelled.

7.12 User accounts and login

The user accounts in the system are only for administration use. This means that only laboratory staff can have an account. New users can only be added by a logged-in user. The users in the system are called teachers and new users can be added in the Resources->Teachers page.

When adding a new user, the system asks for a name, email, and password. The password is prompt twice to avoid typing errors. In this form user can be assigned for some courses and sessions directly. This can also be done from the Courses and Sessions page.

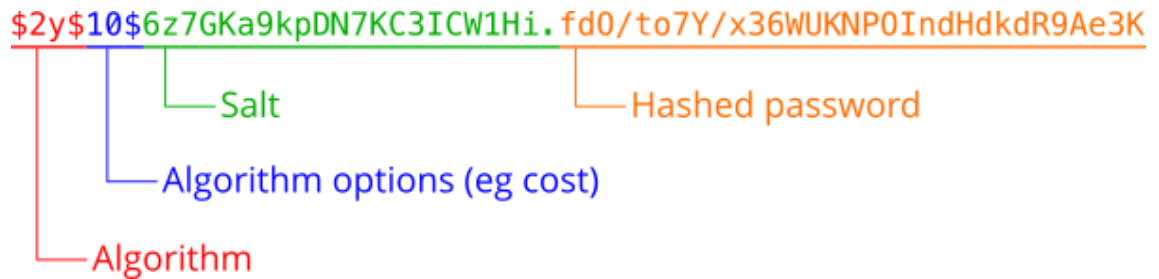
When the form is submitted, the system encrypts the password and other information is set to the database as it is. Since every user has a unique email address, the address is used as the username for login.

Login in is possible by typing /manager after the site URL. The system prompts the user for email and password. After submitting the username and password, the form is passed to login.php file which makes the comparison with the data in the database.

7.13 Password protection

When a new user is added the password for the system has to be made unreadable before it is stored in the database. If the passwords are saved as plain text someone who has access to the database is able to read the passwords. Since the server in use will be a shared server where multiple users will have access, this is a very important part of the systems security. Users also tend to use same passwords in multiple places which means the leaked password cause much harm.

After the system has verified that the given passwords match, a secured version of the password is created. First, for additional security, the system creates a random “salt” which is additional data that is used in hashing to add randomness to the hash result. This especially helps when trying to protect the system against attacks where precomputed lists of already known hashing results are used to expose passwords. These lists are made from common passwords that people use. Then the password is hashed with the PHP built-in function called “crypt()”. The crypt function takes two parameters: a string to be hashed and the salt as a base for the hashing. After the password is hashed it is inserted to database with the other information. (PHP.net)



Picture 14. Structure of hashed password (Source: <http://php.net/manual/en/images/2a34c7f2e658f6ae74f3869f2aa5886f-crypt-text-rendered.svg>)

The final result of the hashing can be separated into four parts (Picture 14). These parts are used when comparing the hash to the password that user gives when login in. The algorithm part tells which algorithm is used for the hashing. The algorithm options tell the “cost” of the hashing. Higher “cost” means more security but it also needs more processing power to create. Last comes the salt and the hashed password.

When logging in, the system fetches the password hash from the database and then the hash is compared to a hash created from the user given password using the hash from database as the salt (Picture 15).

```
$match = hash_equals($hash, crypt($password, $hash));
```

Picture 15. Matching the password with hash_equals function.

For additional security the user names, in this case email addresses could be hashed. This would again protect the users from the leaks that happen through the database. In addition every time when user logs in, the system could create a new hash for the password. (Elbert, 2010) At the moment these additions were considered to be unnecessary because of the limited number of users having the access to the system.

8 CONCLUSION

At the moment of writing this thesis the development process reached a point where all the required functionalities are working. The goals of the project to add security, usability and automation were accomplished.

The next step in developing the system will be testing in its real environment. During testing, security and usability will be enhanced further if necessary and probably some functionalities will be added or even removed. Currently the system is not using all the data that it has the possibility to use in terms of statistics and reports to the staff. This is future development area.

The development process has been challenging and very illuminating about back-end web development. Finding the right solutions and methods to achieve the desired result has been also rewarding. Even though this particular system is developed only for one specific use it has ideas and solutions that can be applied in other booking systems as well.

REFERENCES

cogNiTion (1999). Intro to cron. Available at: <http://www.unixgeeks.org/security/newbie/unix/cron-1.html> [accessed 29.1.2015]

Debian.org (2014). What is Debian?. Available at: <https://www.debian.org/intro/about.en.html> [accessed 28.11.2014]

Elbert (2010). How to store passwords safely with PHP and MySQL. Available at: <https://alias.io/2010/01/store-passwords-safely-with-php-and-mysql/> [accessed 2.3.2015]

Gerchev I. (2014). The 5 most popular frontend frameworks compared. Available at: <http://www.sitepoint.com/5-most-popular-frontend-frameworks-compared/> [accessed 3.3.2015]

Internet World Stats, 2015. Internet growth statistics. Available at: <http://www.internetworldstats.com/emarketing.htm> [accessed 27.12.2014]

Knight, K. (2009). Minimalist web design: When less is more. Available at: <http://www.webdesignerdepot.com/2009/12/minimalist-web-design-when-less-is-more/> [accessed 12.12.2014]

Netbeans (2015). The smarter and faster way to code. Available at: <https://netbeans.org/> [accessed 23.3.2015]

PHP.net (2014) Safe password hashing. Available at: <http://php.net/manual/en/faq.passwords.php> [accessed 27.12.2014]

Sexton P. (2015). Mobile first principles. Available at: <https://varvy.com/mobile/mobile-first.html> [accessed 9.6.2015]

w3schools.com (2014). PHP MySQL database. Available at: http://www.w3schools.com/php/php_mysql_intro.asp [accessed 1.12.2014]