



LAUREA
AMMATTIKORKEAKOULU

Uuden edellä

Network Lab Information System

Festus, Anyanwu Stanley

2015



Laurea University of Applied Sciences
Leppävaara

Network Lab Information System

Stanley Anyanwu Festus
Degree Programme in BIT
Bachelor's Thesis
June, 2015

Stanley Anyanwu Festus

Network Lab Information system

Year 2015 Pages 25

The Network lab as the case company is operating within Laurea University of Applied Sciences have been facing challenges in daily operations and seeking to improve their standards and ease their operations. The Network Lab lend equipment (cameras, server machines) to students and other project groups conducting researches within Laurea Leppävaara unit. The aim of this development project is to propose a solution that would help Network Lab as the case company to carry out their daily operations with a computer system. One-to-One interview was used as the research method to better understand the current system of operation and the problems the stakeholder has been facing. A database driven system was designed and proposed that will store customer data and keep records of acquisitions made during their operations.

Keywords

UI, User Interface, RWD, Responsive Web Design, FOSS, Free and Open Source Software, Database, WAMP, Computer System, AngularJS, Software Development Life Circle, SDLC, PHP, HTML5, MySQL, Use-case diagram, Information Systems

Table of contents

1 Introduction.....	7
2 Objectives and Aims	7
2.1 Aims	7
2.2 Objectives.....	8
3 Knowledge base	8
4 Design and Implementation	9
4.1 System’s requirements	9
4.1.1 Functional requirements.....	10
4.1.2 Non-functional requirements	10
4.2 Design	11
4.2.1 Responsive Web Design.....	13
4.3 User Interface	17
4.3.1 Project directories.....	19
4.3.2 Main Page	20
4.3.3 Registration Page	21
4.3.4 Log in page.....	23
4.3.5 Admin page	25
4.4 Database design.....	26
5 Suggestions for improvement of functionality and interactivity.....	28
5.1 More functionalities with possible technologies.....	28
5.1.2 Show/hide with AngularJS.....	28
5.1.3 Message board.....	29
5.1.4 Extension of admin privileges.....	29
6 Conclusion	29
References.....	31
Figures.....	31

1 Introduction

The Network Lab is a laboratory operating in Laurea University of Applied Science, Leppävaara unit. The Lab controls over 50 devices ranging from server machines to cameras, routers and switches with which it runs and carries out lectures, projects within the lab, and also lends some of its devices and equipment to individuals or group as well as students that needs to use these equipment within the school premises/projects and group conducting re-search in the school.

Situations arise where the Lab have to keep track of these equipment/devices and doing so without a system that could electronically record the lending and returns process is proving to be too challenging for the Lab conductors. For this reason a proposal is being made to develop a database system with a server and a user interface/mobile application with which borrow-ers can check for availability of equipment they wish to borrow and the service providers can keep track of equipment being it lends out.

At the moment, records of borrowing are being kept on written papers and also memorized. Inflow of freshly received equipment needs to be recorded in other to know what new equip-ment is being added to the collections, the state of older ones and what should be disposed of and what not.

This thesis would be aiming to solve the problem of tracking the lab equipment by designing and developing a database system that would record all borrowings and lending i.e. providing users with availability of devices and a user interface from which users can check for availa-bility and make bookings as well. The user interface will be designed and developed to be user-friendly so that visitors can navigate their way around and find information with ease and good download speed.

2 Objectives and Aims

2.1 Aims

One of the main aims at this thesis project is to deliver a solution to meet the needs of the customer; to provide a system that will aid the operations of the Network Lab. The develop-ment and implementation aims to produce a solution that will support all devices and browser platforms regardless of the screen size and create a generic site need to give visitors great experience.

Another focus is to get hands-on practice in ICT field. This means that during my entire study period the school curriculum had touched several ICT areas that are vast ranging from Networking to Information Security, and programming languages like Java. There is an underlying personal aim to develop myself in a particular field of IT and also have a portfolio upon graduation and employment benefit. For this reason, I have also taken up this challenge to develop this system and apply possible frameworks and programming languages available in IT industry mainly focussing on front-end programming/scripting. We would see about this in the following chapters.

2.2 Objectives

During the preparation of this thesis and project proposal, the objective of this project was to develop and present a working solution i.e. a solution that the stakeholder can use to follow up on borrowers using a database to store data of users of their services - that will aid operations the Network Lab to keep track of equipment acquisitions, using a sustainable open-source software or other Web technologies to design and develop a user-friendly system to meet customer's needs.

In the final implementation, open-source software wasn't considered as an appropriate tool and was finally dropped for latest HTML technology (HTML5) will be use to implement this project as well as open-source server-side programming language (PHP) and MySQL to design a database system that can be supported on all mobile devices and desktop Web browsers (Safari, Firefox, Opera, IE and Chrome). This was due the reasons mentioned above - to apply hands-on practice for personal development.

3 Knowledge base

Network Lab being the customer/stakeholder have presented a business/operational need for a system that will ease their daily operations in form of an inventory system that would register acquisitions made by users. As specified above in the system requirement section, the customer has expressed concerns about having a system that can be accessed from all platforms regardless of the device screen size.

In an internet article, Frost demonstrated the use CSS @media rules to implement a Mobile-First Responsive Design and elaborated on the need for adaptivity by using fluid grids to create good layouts for different screen sizes Frost, B (2012).

Internet research was carried out within this problem domain during the preparation of the system's proposal to find out all possible implementation methods either by using existing open-source software solutions that could be customized to fit the needs of this project. The other purpose of accepting a system's development project as base for bachelors thesis was also to apply and develop the programming technics that have been acquired during my study period as mentioned above and that has prompted me to implementation of this project with pure front-end programming and scripting languages and also implemented back-end languages where applicable as against for example Twitter Bootstrap framework and that helped also me to understand the underlying technology in these kind of frameworks.

Further down in the implementation chapters, we would see screenshots of the system in different modes. The look and feel of the system in handheld device, medium size device and desktop size device.

4 Design and Implementation

This thesis is a project-based thesis. As a project-based thesis, a constructive approach. The project implementation process followed software development life circle starting with user requirement gathering, design, and construction and testing. Agile development method was applied because it best suits my technical level. It was also very suitable since there was only one technician and one stakeholder to hold meetings with.

4.1 System's requirements

Problem domain

As explained in the introductory section, the client has been facing some problems with their current processing system. The main challenge is that the process of storing equipment that is available to the public for lending purposes and there specifications, storing borrowers/users' data has been very rigorous. The loan process have been carried manually. The processes involved are checking the inventory for availability of equipment, recording borrowers' information on a piece of paper having been the case that is eventually transferred into a hard cover folder. In many cases, transferring borrower's information on the main folder never happens the due to other commitments to the part of the handlers of the network lab.

Another issue the operation is also facing is that this process is managed by several personnel in the lab, that is to say that one party could issue a loan and misplace or forget to transfer the information on the main folder and no other personnel would be aware making it easy to

lose equipment. For this reason, the need for a system to help efficiently in the process carried out by handlers.

The first stage in designing a solution to this problem is to determine the requirements of the new system. An interview was conducted as a research method for requirement gathering and analysis with the stakeholder to determine the functionalities of the new system.

✓ Name of interviewee	Salminen, R.
✓ Interviewee's job title	Senior Lecturer/Lab Caretaker.
✓ Company	Laurea Network Lab.
✓ Type of communication	Interview with Lab Caretaker.
✓ Date of interview	4 September 2014.
✓ Location	Leppävaara, Espoo.

Below is the list of functional and non-functional requirements for the new system generated by that study. The list below is subject to improvement during the project life cycle until the final delivery date.

4.1.1 Functional requirements

- Users can search inventory for availability of equipment
- Database search with different keywords or by category
- Sign up
- Log in possibility to book equipment with required information

4.1.2 Non-functional requirements

Security requirements

- Only system's administrator can insert and delete devices from database
- Only system's admin can see registered users and edit their accounts but not password
- Only registered users can rent equipment

Operational requirements

- System should support any web browser users uses to access service
- Anyone can search the for equipment

Use case diagram

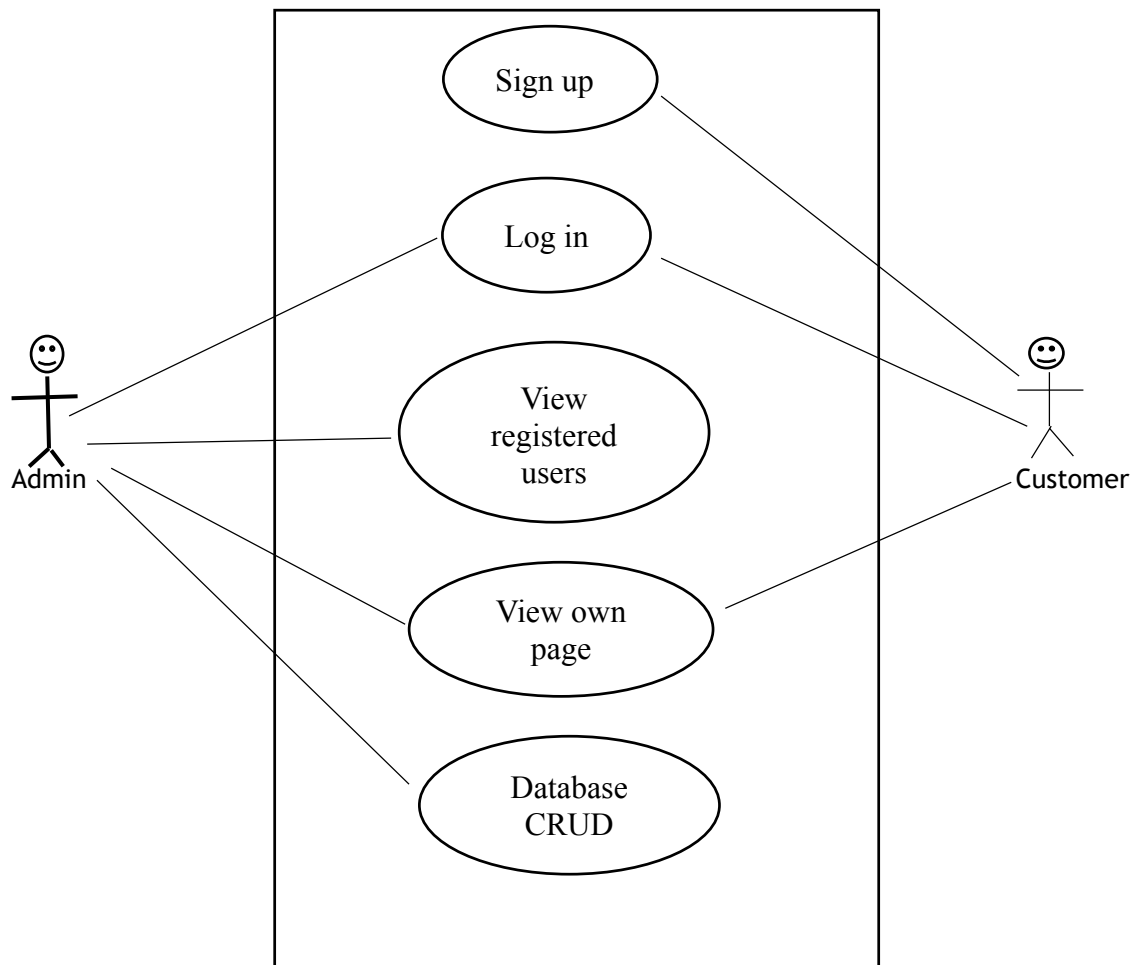


Figure 1 Use-case diagram of the proposed system

4.2 Design

In the design stage of the project's system design life cycle, it started with sketching out different user interfaces for every page starting from the main page (index.html) to search page and log in and registration page. It was a challenge to come up with application and practical user interfaces that have to be made to blend with the functionalities of the systems so it doesn't look too empty because is the first of this kind of system, so the possible look and feel of the registration page and log in forms have to designed cautiously. There were a few sketches before the final decision to go with one. Below is a figure of some other the sketches drawn.

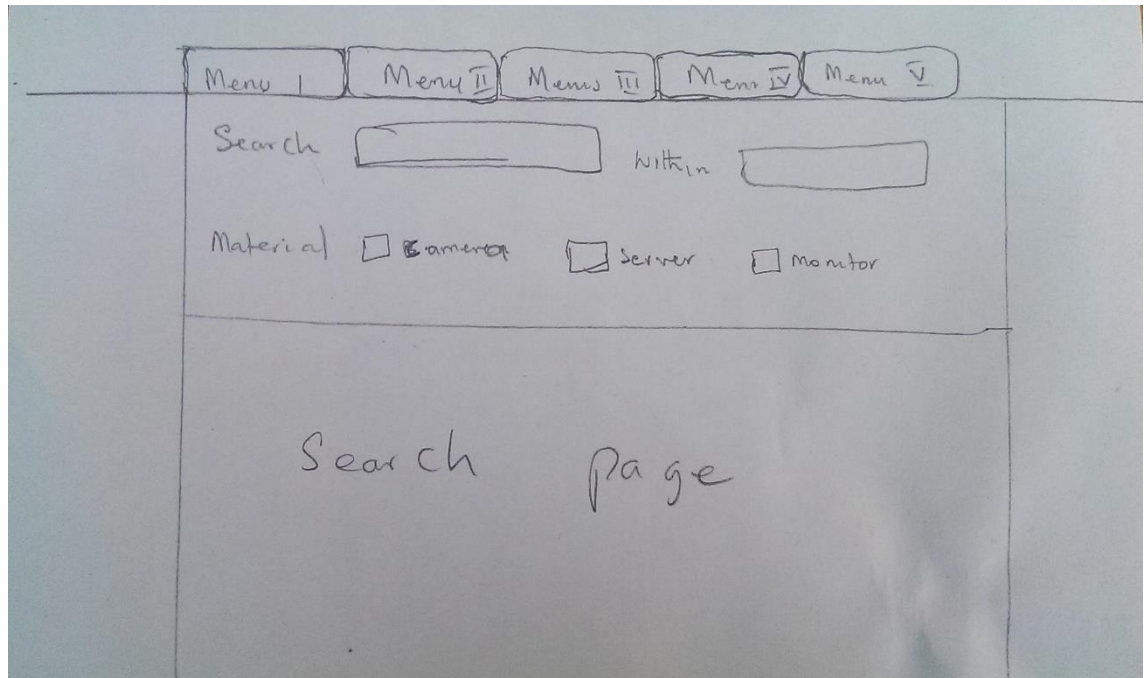


Figure 2 System's search page

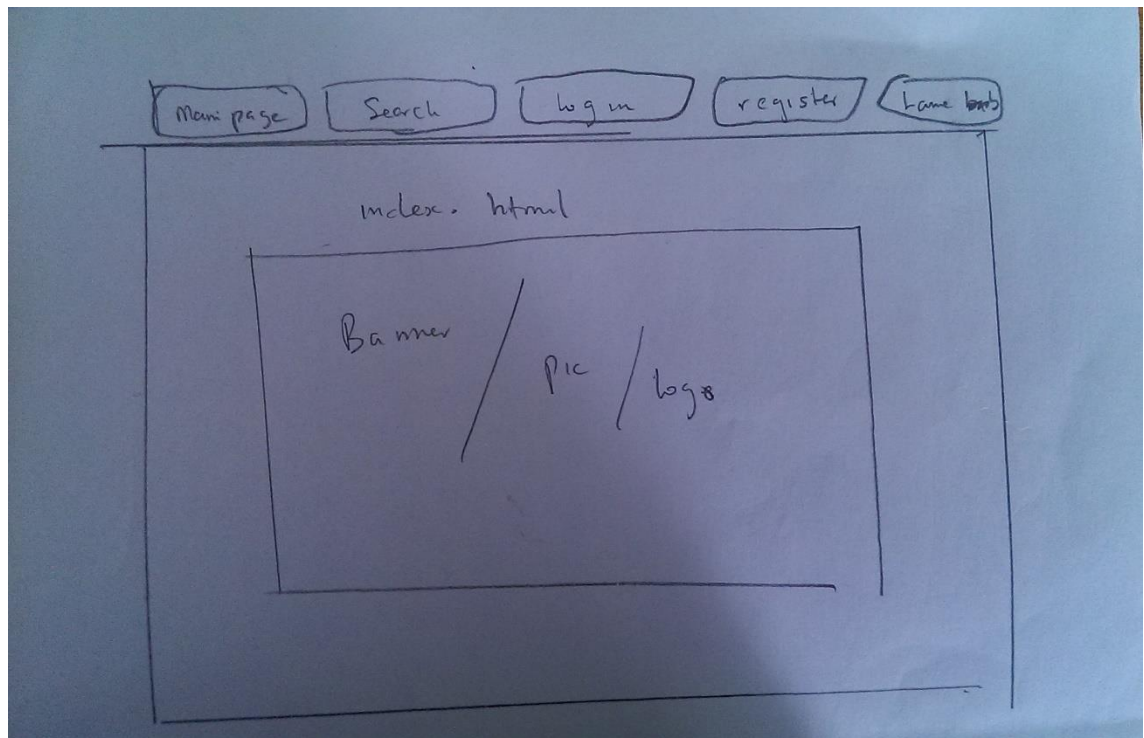


Figure 3 System's main page

The figures above represents an initial sketch for the search page with the navigation menu at the top of the page and the main page navigation page. In the final implementation, the navigation menu was brought to the left hand side of the page and a banner of Laurea

University of Applied Science now sits on as the header of the system. We would go through more on this pages in the later chapters.

4.2.1 Responsive Web Design

The system's design and implementation have followed the RWD guidelines described above to build for device with small screens and progressively reacting to larger screens. Using CSS3 media types, devices with different screen sizes have been designed access the web application and interact with the system but different UI displays to fit the scale. With progressive enhancement, the different features are that as the device screen gets bigger, the better presentation. According to Wikipedia (Progressive enhancement 2015) progressive enhancement gives access to devices to a website's basic content regardless of software version and hardware as it is in this case screen size using web technologies. We will take a look at the different views the system provides to different devices below implemented with CSS @media web technology:

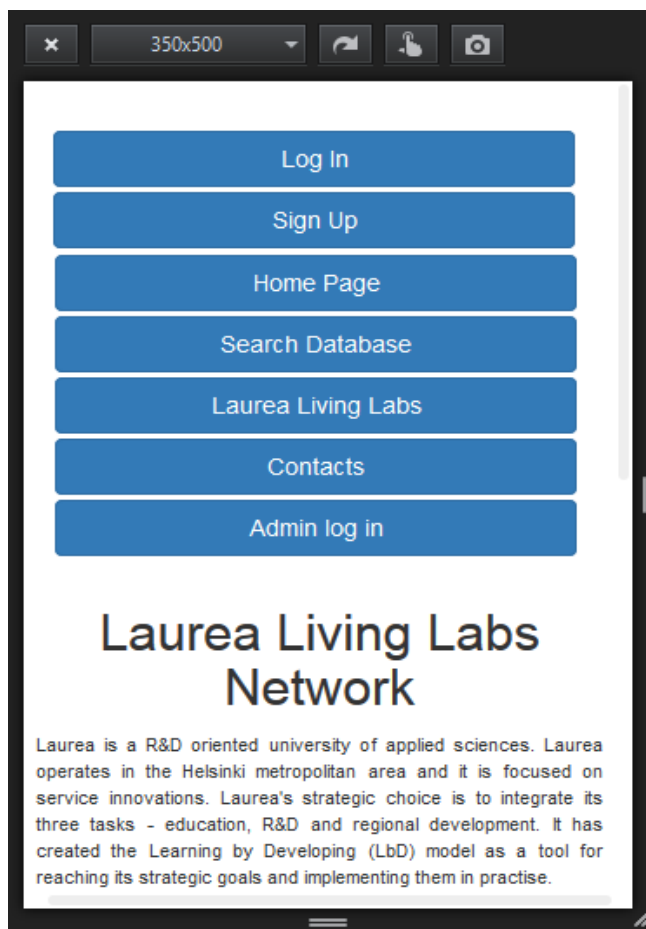


Figure 4. A mobile display of the system

The figure above is a screenshot of the proposed system in handheld device. The screenshot

was taken with *Mozilla Firefox* responsive web design developer tools. And below is a snippet implemented to achieve the above result.

```
@media (max-width: 500px) {  
  #header {display: none}  
  #info-col {display: none ;}  
  #footer {display: none ;}  
  #container {width: 300px;  
  padding: 0px;  
  margin: 0px;}  
  #reg-navigation {position: absolute;  
  margin-left: 7px;}  
  #reg-navigation li a, #nav li a {width: 300px}  
  #nav {position: absolute;  
  margin-top: -45px;  
  margin-left: 3px; }
```

To support handheld devices in the concept of progressive enhancement, only the website basic content have been designed to be accessible to small devices as mentioned above. The navigation links uses extended @media rule to create a fluid grid display that shrink to fill the entire width of handheld devices but will spread out to fill the width of any screen as it grows bigger.

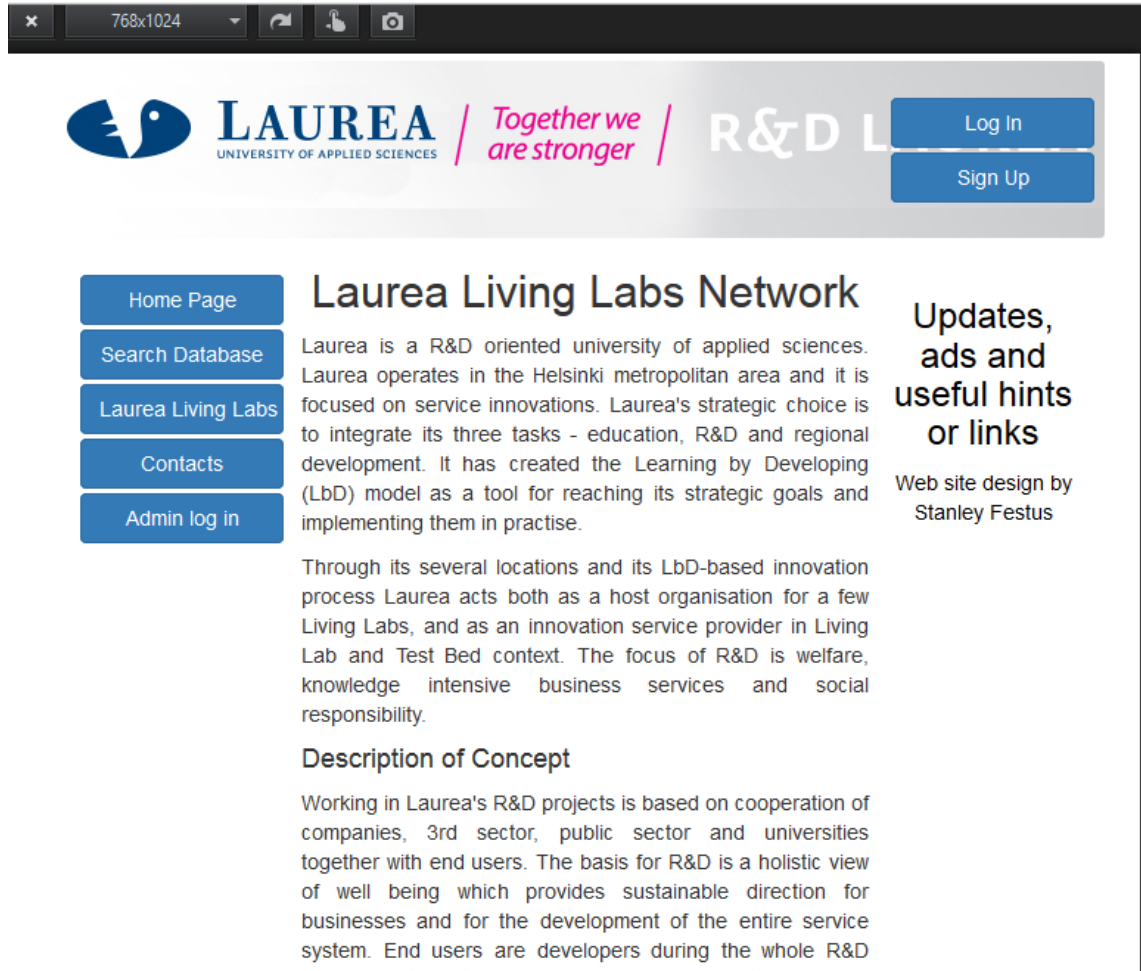


Figure 5 Display for tablets

The above screenshot displays the website with a bigger screen (tablet) that could as well be a 10 'inch' handheld device but was taken with *Mozilla Firefox* developer tools.

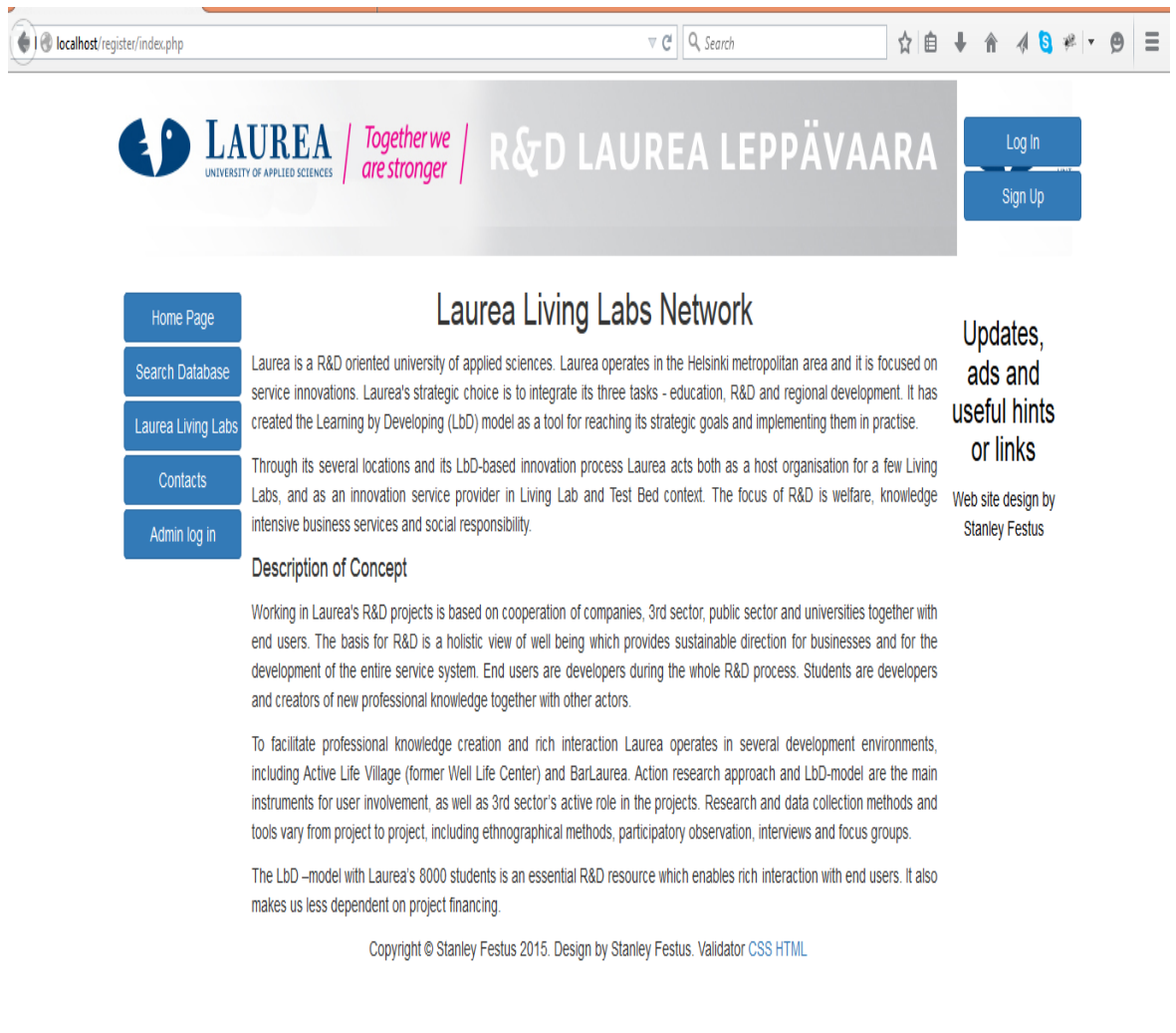


Figure 6 shows the website in desktop screen

The above screenshots have been used to demonstrate mobile-first paradigm and the power of progressive enhancement. All three figures displays different views as the device screen size progresses. Figure 1 shows what the user would get where device width is 350 pixels and height is 500 pixels. There are some features unavailable to smaller screens like the banner header, the advert section to the right website that can be used to incorporate services from other sites RSS feeds, weather and so on - we will come back to this later in the improvement chapter and even the footer.

Only basic features are available when accessing the site with a handheld device. It is now obvious that with a smaller sized screen, one gets to interact with the system and access all related links available but if we take a look at the figure 2, it clearly gives more in the presentation like banner in the header.

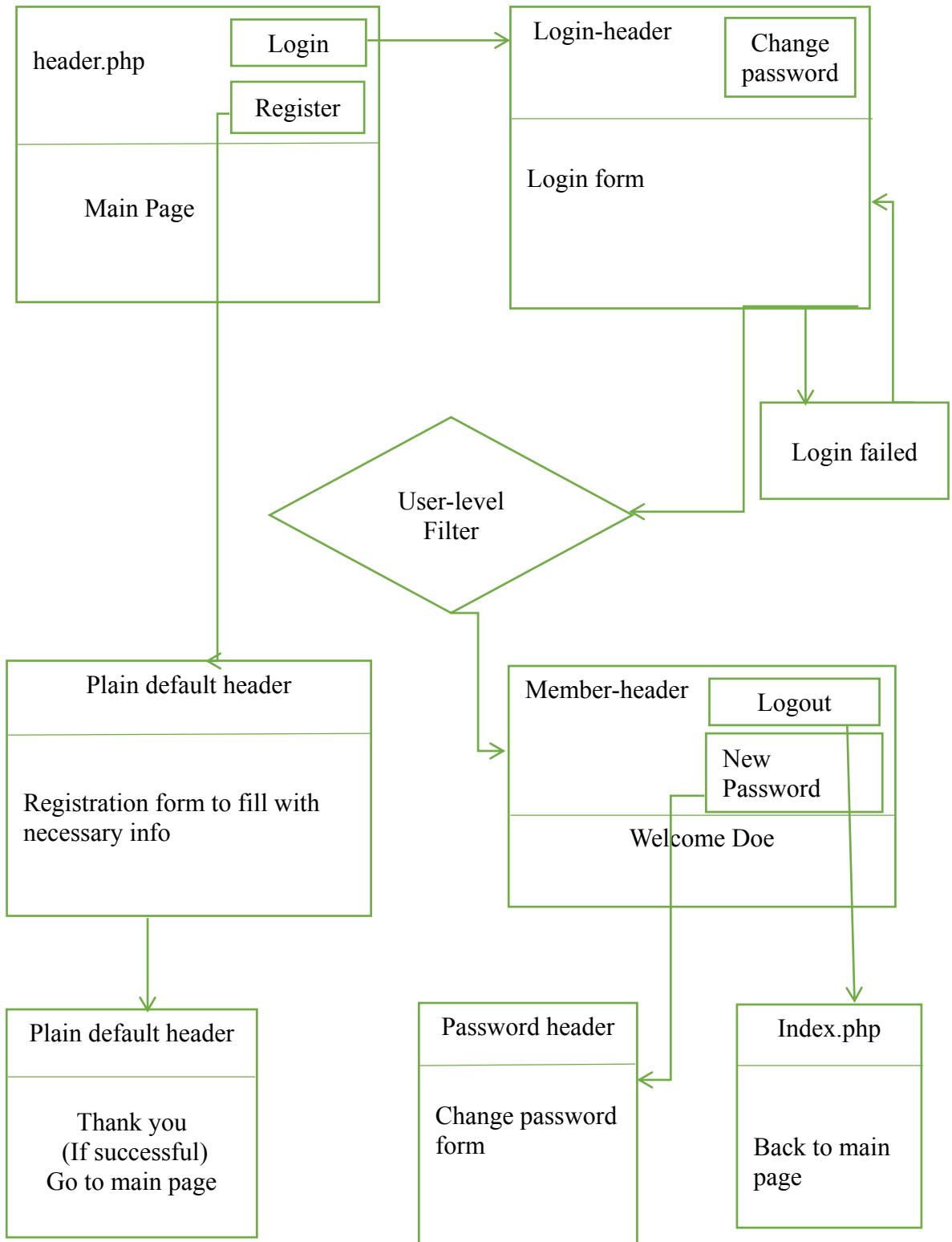


Figure 7 the system's flowchart

4.3 User Interface

An integral part of this documentation is the interview conducted by the business analyst as a research method to gather user requirements. The deliverables of the interview were functionalities that would suit the needs of the client as agreed by the analyst and stakeholder. All of these taken into account, the system's architecture have implemented all functional and non-functional requirements and have separated user interface into two sections i.e. ordinary users and admin user for security reasons.

The reasons are that users' privacy will be guaranteed i.e. every user can only access his/her activity log i.e. filtering and an admin user should have an extensive right to the system as well. There is a flow chart through and around the system's architecture. This flow chart is intending to help any developer who would be improving on this system.

The system's flow chart (see figure 6 above) demonstrates the basic forms of navigating through the system. When a visitor types in the URL of the system, he or she would be directed to the main page (index.php) with several other navigation menu. From here the visitor can navigate around to anywhere in the entire system except for some pages with special rights. The vital links in the main pages are the registration page, log in and search database pages. From the main page, users can search database for available services and if found are now expected to log into the system with their account credentials to be able to use the service.

If a user does not have a registered account, he or she is then expected to register into the system because only registered users can access services provided. If the visitor/user logs in, he or she would be directed to own page as shown by the diamond shaped activity labelled *user-filter* in the flow chart. The entire flow chart gives a quick view of the order of activities provided. Although not included in the flow chart but in other to clear any confusion, the diamond's other direction when anyone signs into the system is to redirect according to the credentials to the admin page. It is intended that there would be one admin at a time.

The use of search page is intended for any user to access and search the database for services i.e. equipment available for loan. If found, then the user can then have move to borrowing the equipment if and only he or she is a registered user. Otherwise, he or she would have to be registered and the registration is visible on every page. During registration, the inputs expected are the users name, surname email and password where email would be used as ID and the password as password. When connecting to the database, the email and password will be used to fetch other user data in the system.

4.3.1 Project directories

The project team came to a conclusion to apply PHP server-side scripting language and MySQL as the RDBMS programming language web technologies for project implementation. During the project planning phase, there were a few options listed out as possible web technologies amongst are JavaScript as front-end scripting language -JavaScript script frameworks like AngularJS - NodeJS for server-side scripting. The reasons for choosing PHP and MySQL strong depended on the availability of FOSS in the market and the project's programmers' basic knowledge of PHP instead of starting completely with for example.

The entire project resides in a file directory named registry with other subdirectory like *dist*. directory which contains CSS stylesheet folder, script folder and font folder. The rest of the files that reside in the main directory are basic php files that holds the main structure of the system. In essence, the system was designed to fit basic functionality for now as the first of its kind but there will suggestions on how to improve the system taking into considerations stakeholder demands and expectations that were gathered during requirement gathering stage.

These said functionality have not been implemented for now because they are not part of the scope of the first part of this project but there are plans on ground for them to be included in future to create better user experience. Below is a figure that shows the basic family tree of the project files:

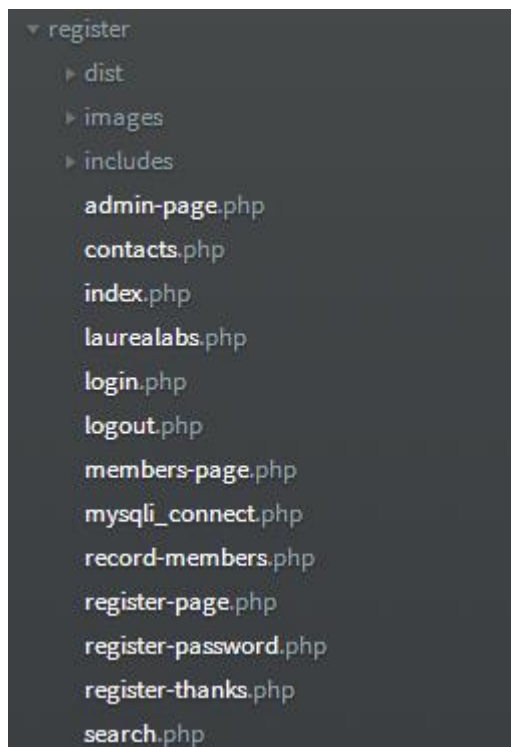


Figure 8 A screenshot of the system's directories

The *dist.* folder contains a downloaded version of Twitter Bootstrap codes but customized to the needs of this project. The version downloaded is Bootstrap v3.3.1 licensed under MIT GitHub. The project applied only Bootstrap CSS minified version (*bootstrap.min.css*) and can also be modified to fit later needs of this project. All the different Bootstrap files (*bootstrap.min.css*, *bootstrap.min.js*, and font files) have been included into the project even though the system doesn't contain any front-end scripting because some of the files have (Bootstrap Documentation 2015) plugins that are or can be dependent on each other. It is possible to apply them later when developing the system and adding more functionalities. We would see more that in the suggestions for improvement chapter.

The other folders (includes folder, image folder) have been declared with self-explanatory names that explains what they do within the system. The *image* folder contains all images used in the system (the only *image* in the folder is the system's banner) whilst *includes* folder contains all separate *php* files attached to the system with `php include()` function. We would now go through the other files one after the other taking us through the snippets that make up the entire system. First, we would start with *index.php* page that holds all other pages within the project.

4.3.2 Main Page

The main page holds navigation links to every page giving visitors the much needed access. The most important pages to navigate to are the log in page, the registration page and search database page. Also when visitors log out of the system, they will be redirected to the main page. With the snippets, the important issues to take note of is the CSS link file and the include files wrapped in PHP tags. There are two CSS files namely *includes.css* and the bootstrap CSS file. Within Bootstrap, the components that were applied are the Glyphicons component, Typography, Form and Button. In the main body of the main page, the header, navigation menu, and footer all in separate files and served to website using `PHP include()` function so updating would be easy.

Since all the pages have a lot of similarities with the headers and menu buttons, the project team concluded that it is easier to have separate files like this but then with some pages, redundant links were removed. At this point, it is important to mention that this practice have also been applied in other areas e.g. the snippet to connect to the database is in its own file and called into where ever it is needed using `PHP require()` function.

```

register/index.php - Brackets
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <title>Home page</title>
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8 <meta name="author" content="Stanley Festus">
9 <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
10 <link rel="stylesheet" type="text/css" href="dist/css/includes.css">
11 <link rel="stylesheet" type="text/css" href="dist/css/bootstrap.min.css">
12 </head>
13 |
14 <body>
15 <div id="container" class="container">
16 <?php include('includes/nav.php');?>
17 <?php include('includes/header.php');?>
18 <?php include('includes/info-col.php');?>
19
20 <div id="content"><!--Start of page content-->
21 <section>
22 <h2>Laurea Living Labs Network</h2>
23 <p>Laurea is a Finnish oriented university of applied sciences. Laurea operates in the Helsinki metropolitan

```

Figure 8 Screenshot of the index.php page

4.3.3 Registration Page

Below is the code used to create registration form. We urge the reader to pay attention to the form field string length and size.

```

<form action="register-page.php" method="post">
<p><label for="fname" class='labeling'>First Name</label>
<input type="text" id="fname" name="fname" size="40" maxlength="40" value="<?php if(isset($_POST['fname'])) echo $_POST['fname'];?>" />

<p><label for="lname" class="labeling">Last Name</label>
<input type="text" name="lname" id="lname" value="<?php if(isset($_POST['lname'])) echo $_POST['lname'];?>" maxlength="40" size="40" />

<p><label for="email" class="labeling">Email</label>
<input type="email" id="email" maxlength="60" size="40" name="email" value="<?php if(isset($_POST['email'])) echo $_POST['email'];?>" />

<p><label for="pword1" class="labeling">Password</label>
<input id="pword1" name="pword1" type="password" size="12" maxlength="12" value="<?php if(isset($_POST['pword1'])) echo $_POST['pword1'];?>">&nbsp; Between 8 and 12 characters
</p>

<p><label for="pword2" class="labeling">Confirm Password</label>

```

```
<input id="pword2" name="pword2" type="password" size="12" maxlength="12" value="<?php if
(isset($_POST['pword2'])) echo $_POST['pword2'];?>"> </p>
```

```
<p><input id="submit" type="submit" name="submit" value="Register"> </p>
</form>
```

The sizes and length of the form input field are made to fit the length of the database table's column attribute string values. The form application fields are limited to the necessary information required to register to this system i.e. name, surname, email (as log in ID) and password. Keep in mind that this system is the first of its kind so it is very possible that in future more functionalities can be added.

The next is the piece of code that processes the information input from the user, inserting the info into MySQL table while interacting with the user if there any errors. We would look at some necessary parts of the snippets.

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
require ('mysqli_connect.php');
```

The snippet above specifies the request method the form input is sent to the server. If the method is 'POST', there and then a connection to the database is opened.

```
$error = array();
if(empty($_POST['fname'])){
$errors[]="You forgot to enter your name";}
else {
$name = mysqli_real_escape_string($conn, trim($_POST['fname']));
}
```

The input values received are being wrapped and inserted to the database using `mysqli_real_escape_string()` function. What the function does is that processes any special character within the string. For example member's name is Matti-Veli or Mac' Donald, the functions makes sure that the string is processed (West 2013, 55).

```
if(empty($_POST['lname'])){
$errors[] = "You forgot to enter your surname";}
else {
$name = mysqli_real_escape_string($conn, trim($_POST['lname']));
```

```

}
if(empty($_POST['email'])){
$errors[] = "You did not enter your email address";}
else {$email = mysqli_real_escape_string($conn, trim($_POST['email']));}
}
if(!empty($_POST['pwd1'])){
if($_POST['pwd1'] != $_POST['pwd2'])){
$errors[] = "Your passwords don't match";}
else {$psword = mysqli_real_escape_string($conn, trim($_POST['pwd1']));}
}
else {$errors[] = "You forgot to enter a password";}

if(empty($errors)){
$query = "INSERT INTO customers (user_id,user_level, fname, lname, email, psword, registra-
tion_date) VALUES (' ', ' ', '$fname', '$lname', '$email', SHA1('$psword'), NOW())";

```

Another important part of this programme is the security issue and that was taken into consideration by MySQL SHAI() function. SHA1('\$psword') is a function for protecting a password. "The acronym SHA stands for Secure Hash Algorithm, which is a security measure created by the US National Security Agency" (West 2013, page 42). It instructs MySQL to encode the password by converting it to a string 40 characters in length (no matter what length of password the user enters into the form).

4.3.4 Log in page

As mentioned above, a few individual pages share the same *headers* file but with some others like the *log in* page, there is some customization to make sure that only buttons necessary are visible. Another important thing to note is that both the admin log in and the general user log in share the same log in page.

The log in page design took a different turn because privacy and security issues that were taken into consideration. In the use-case diagram above (see use-case diagram above) at the planning stage of this project it was clear that there will be different membership types. The database holds one table at the moment is a table that is meant to hold registered members/users. As planned initial that there will be an administrator who will be administering this site (accessing registered members and altering the database) this admin member would have certain privileges and rights that every member cannot have.

The admin and ordinary users have been separated by creating user levels. The snippets in the log file shows how session during log in a session is created using user credentials and then checks the *user-level* column and if it is equal to '1' then the user is redirected to the admin page (admin-page.php, member-page.php, see figure below) otherwise the user is re-directed to the member page accordingly.

Showing rows 0 - 8 (9 total, Query took 0.0007 sec)

```
SELECT * FROM `customers`
```

Number of rows: 25

Sort by key: None

+ Options

	user_id	user_level	fname	lname	email	psword	registration_date
<input type="checkbox"/> Edit Copy Delete	1	0	Maria	Riika	riikamaria@gmail.com	e2d918adde1e76443def13c8194e664a6e811493	2015-02-05 13:23:43
<input type="checkbox"/> Edit Copy Delete	2	0	Chineso	Festus	chineso@festus.com	e2d918adde1e76443def13c8194e664a6e811493	2015-02-09 11:30:08
<input type="checkbox"/> Edit Copy Delete	3	1	Admin	Administration	admin@example.com	d033e22ae348aeb5660fc2140aec35850c4da997	2015-02-09 11:45:43
<input type="checkbox"/> Edit Copy Delete	4	0	Uchenna	Festus	uche@festus.com	ca8a0d0991a62659a91ec3172b614dd4c142e7a9	2015-02-09 11:46:07
<input type="checkbox"/> Edit Copy Delete	5	0	Kelewe	Anyanwu	kelewe@hotmail.com	d16b6120cea33c3811a36224df6e1a0dfeaa74c6	2015-02-09 11:46:32
<input type="checkbox"/> Edit Copy Delete	6	0	Vamma	Aivo	akosuot@gmail.com	e562ee5a6baf946f41284318ed349ef114b37188	2015-02-09 18:06:23
<input type="checkbox"/> Edit Copy Delete	7	0	austin	jibrin	b-man13@hotmail.com	4d1168a538fbce95ccefa93a6316d2bddf388113	2015-02-26 23:22:48
<input type="checkbox"/> Edit Copy Delete	8	0	gabe	gabe	gabe@something.com	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	2015-05-01 16:26:14
<input type="checkbox"/> Edit Copy Delete	9	0	Antonius	Camara	camara@example.com	d16b6120cea33c3811a36224df6e1a0dfeaa74c6	2015-05-12 14:31:06

Check All With selected: Change Delete Export

Number of rows: 25

Query results operations

Figure 9 Screenshot of the database table illustrating the user level column


```

22     else{$email = FALSE;
23     echo '<p class="error">You forgot to enter your mail</p>';}
24
25     if (!empty($_POST['pword'])){
26         $p = mysqli_real_escape_string($conn, $_POST['pword']);
27         else{$p = FALSE;
28         echo '<p class="error">You did not enter your password</p>';}
29
30     if($email && $p){
31         //retrieve the user_id, first name and user_level
32         $q = "SELECT user_id, user_level, fname FROM customers WHERE (email='$email' AND psword=SHA1('$p'))";
33         $result = mysqli_query($conn, $q);
34         // count number of rows
35         if (@mysqli_num_rows($result)==1){
36             session_start();
37             $_SESSION = mysqli_fetch_array($result, MYSQLI_ASSOC);
38             $_SESSION['user_level'] = (int)$_SESSION['user_level'];
39             $url = ($_SESSION['user_level'] === 1)? 'admin-page.php' : 'members-page.php';
40             header('Location: '.$url);//make the browser load members or admin page
41             exit();
42             mysqli_free_result($result);
43             mysqli_close($conn);}

```

Figure 10 Screenshot of the snippet of the log in page

4.3.5 Admin page

At the moment we presume the logic behind the admin page design is clear concerning privileges and rights. The admin enjoys certain privileges ordinary members do not. Only the admin can see registered members, alter the database data directly for example to insert available equipment except *change password* functionality which every registered member have access to. When the admin page is logged on, it would look like this:

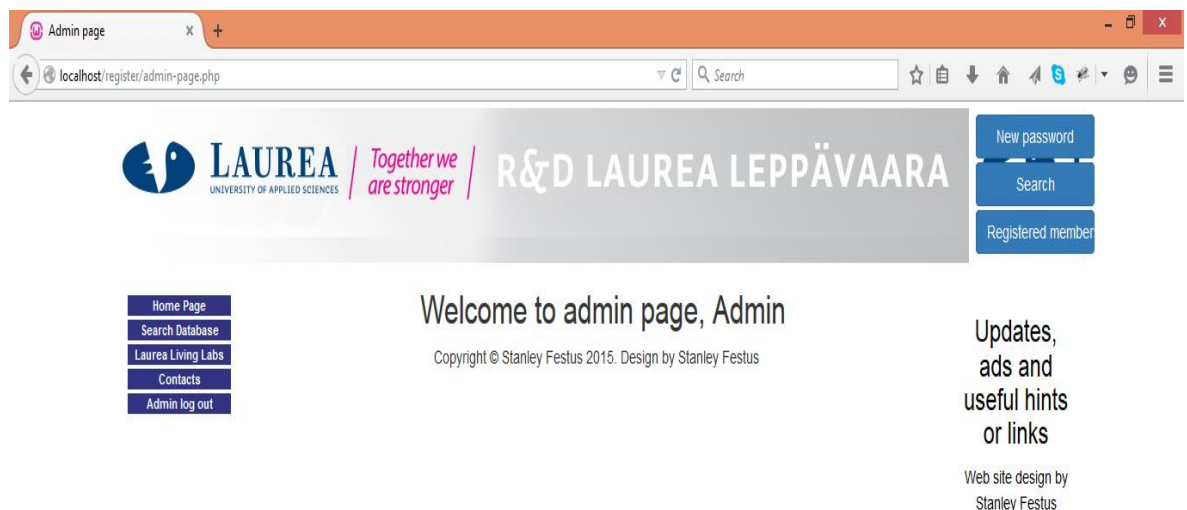


Figure 11 screenshot of the admin page

What we are seeing is an admin page with access to registered members, search button for equipment available and free to be loaned, log out button and password change possibility.

We would see shortly the difference from an ordinary member page. There are plans to add some more functionality to the admin page but it can be found in the improvement chapter.

4.3.6 Member page

The members' page with reference to use-case diagram above has only the basic functionalities designed to fit the needs of members with security concerns in mind. When a member log in, the system would start a session and then search the database table's user level column to see if the value is 0 or 1 and redirects the user to the appropriate page (admin page or member page). The buttons provided are buttons to view own loans, change password and log out button.

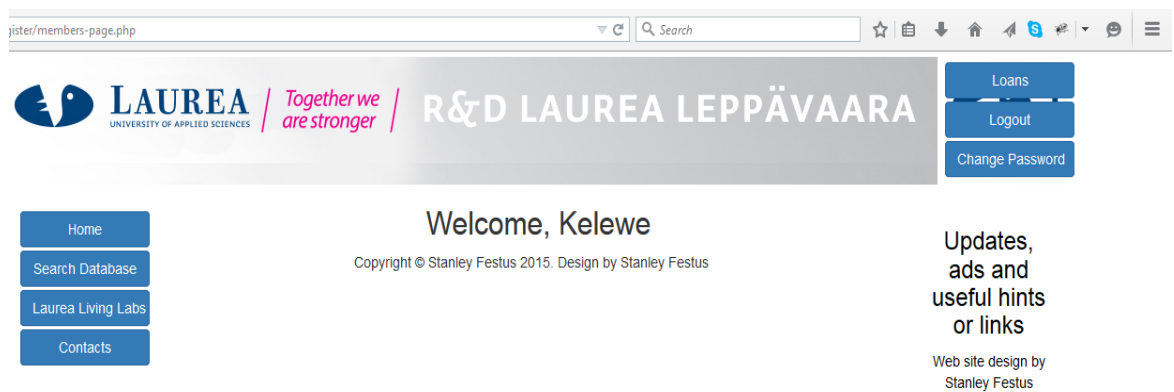


Figure 12 A screenshot of the member-page

The above screenshot illustrates the basic functionality a member has access to but can be improved by adding more functionalities like a possibility for the member to delete his/her own account if he/she deems it necessary or does not have need of this service anymore. More on this in the improvement chapter.

4.4 Database design

The database design was first introduced by doing a sketch with columns and rows that fits the functionalities of this system based on the use-case diagram above so that data can be stored and retrieved and also user-rights. According to the system's requirement, only registered members are allowed to make use of the services provided with this system. To satisfy the requirement, the user would register providing a name, surname, email (user log in name) and password. One database was created for this project named 'registry' and it consists of one database table named 'customers'.

The names have been chosen that reflect and explain explicitly the use of this database and table but we will still take some time to explain the rows that play vital roles.

The customer table below screenshot illustrates the 7 rows that have been created:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 user_id	int(8)		UNSIGNED	No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	2 user_level	tinyint(1)		UNSIGNED	No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	3 fname	varchar(40)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	4 lname	varchar(40)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	5 email	varchar(60)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	6 psword	char(40)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	7 registration_date	datetime			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Figure 13 Screenshot of the database structure

A separate file has been created for connection to the database. The code snippet contains the database name (*registry*), user (*Riku*), password (*Salmi*) and the name (*localhost*).

The above screenshot represents the table rows and their columns; the name titles given are:

- ✓ User_id
- ✓ User_level
- ✓ Fname
- ✓ Lname
- ✓ Email
- ✓ Psword
- ✓ Registration_date

The user ID is the primary key and as type 'int' with the length of 6 and the box A_I ticked to indicate automatic increment. User level has a length of 1 and it is used to literally to create

user levels between ordinary user and administrator. The name (fname) and surname (lname) are also important but do not have much significance after a user is registered. The email acts as a user's log in ID with the length of 50 and type VARCHAR. The password's cell has been encrypted PHP SHA1() function as mentioned above and the function instructs MySQL to convert the user's password into a 40 characters long string to secure every password. The seventh row (registration_date) stores the date of a particular user's registration. This happens with this row being programmed with PHP NOW() that gives command to MySQL to convert the date.

5 Suggestions for improvement of functionality and interactivity

5.1 More functionalities with possible technologies

This section deals with the possible improvement of this system. Adding functionalities to improve interactivity and user experience. At the planning stage of this project, the project team weighed several implementation technologies (programming languages) and frameworks and how any would best suit this kind of project from the many available Chesters, J. (2015)

- ✓ jQuery - an open source cross-platform library, jQuery is designed to simplify the client-side scripting of HTML. jQuery is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications.
- ✓ AngularJS - an open-source web application framework maintained by Google and a community of individual developers and corporations that aims to simplify both development and testing of single-page applications. Angular uses an MVC framework, along with components commonly used in rich internet applications.
- ✓ JSON - JSON is a format for storing and transporting data and is often used when data is sent from a server to a web page.

5.1.2 Show/hide with AngularJS

One of the proposed functionality that would improve interactivity is to use Angular ngHide and ngShow directives to hide or show elements that are holding any particular equipment. If for example a camera in the database is on lease, the element holding this data in the UI () semantically to add ng-hide directive as its attribute to hide and ng-show di-

rective to show the camera when it is again available for lease. The logic is that when a visitor enters possible search words into the search form field, a list of related items will be displayed but only available items are displayed showing only items that are not loaned out. A complete tutorial on this can be found in AngularJS documentation (AngularJS Documentation 2010-2015).

5.1.3 Message board

Message board is a type functionality that can also be known 'sending alert messages'. By message board, the project team is proposing that this system is developed further such that there will an automatic alert to customers in possession of any equipment regarding the end of loan time approaching. PHP mail() function can be used for this implementation purpose.

5.1.4 Extension of admin privileges

The system presently features buttons that allow the admin to view the registered members and their information. To further develop this system, it would be a good to ease the work of the admin especially in cases where the technical know-how of the admin is not known, and add more functionality with which the admin can delete members (in case they deceased, change of address) so instead of altering the database manually the admin can do so through a UI (user interface).

6 Conclusion

A database-driven website have been created with one table as demo to hold registered customer/user data as main deliverable to the stakeholder to ease its daily operations. The system comes with different layouts to support different screen sizes ranging from mobile phone devices to tablets and desktops applying mobile-first paradigm.

This thesis project work is not without limitations. The project omitted some functionalities that were afore planned to be part of the implementation initial plan but have been all added in chapter 5 as part of suggestions for improvement. The project team was a one-man team that worked on the front-end and backend and as the business analyst in this project. A three man team would have had a more effective work done instead of a one-man team with one carrying out the task of business analyst, the other concentrating on the front-end and the third on the back-end tasks.

The project team learnt during this implementation that supporting devices with different screen-sizes can also take a different approach and the approach is known as 'browser sniffing' instead of CSS media approach applied in this project (Patel 2014).

References

1. AngularJS Documentation 2010-2015. Accessed 2015 <https://docs.angularjs.org/api/ng/directive/ngHide>
2. Bootstrap Documentation 2015. Accessed 2014 <http://getbootstrap.com/components/>
3. Chesters, J. (2015) JavaScript Frameworks in the Real World. Info Queue, 2 March. Accessed 24th May 2015. <http://www.infoq.com/research/javascript-frameworks-2015>
4. Frost, B. 2012 Creating a Mobile-First Responsive Design. 16th April. Accessed 20th September 2014 <http://www.html5rocks.com/en/mobile/responsivedesign>
5. Patel, S.K. (ed.) 2014. Responsive Web Design with AngularJS. Birmingham - Mumbai: Packt
6. West, A. (ed.) 2013. Practical PHP and MySQL Web Site Databases: A Simplified Approach. New York: Apress

Figures

Figure 1. Use-case diagram of the proposed system, Figure 2. A Sketch System's search page, Figure 3. System's main page, Figure 4. A mobile display of the system, Figure 5. Display for tablets, Figure 6 shows the website in desktop screen, Figure 7 the system's flowchart, Figure 8 A screenshot of the system's directories, Figure 9 Screenshot of the database table illustrating the user level column, Figure 10 Screenshot of the snippet of the log in page, Figure 11 screenshot of the admin page, Figure 12 A screenshot of the member-page, Figure 13 Screenshot of the database structure