UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

COMPUTER SCIENCE HONOURS

FINAL PAPER

2017

Title: **Smart Locker Systems in an Internet of Things environment**

Author: **Marion Mugabirwe Rugunda**

Project Abbreviation: **LOCK-IT**

Supervisor(s): **Gary Stewart**
Sam Chetty
Craig Balfour

Category	**Min**	**Max**	**Chosen**
Requirement Analysis and Design	0	20	
Theoretical Analysis	0	25	
Experiment Design and Execution	0	20	
System Development and Implementation	0	15	
Results, Findings and Conclusion	10	20	
Aim Formulation and Background Work	10	15	
Quality of Paper Writing and Presentation	10		
Quality of Deliverables	10		
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	
Total marks		**80**	

Abstract

Smartphones have become one of the most important devices today. With advancements in technology their capabilities and functionalities have increased. This has resulted in a lot of tasks in homes today being automated which is known as home automation. The most popular smart home system currently are smart door locks. Smartphones have a lot more functionality which creates opportunity for developing a smart locker that is controlled by a mobile and web application.

A mobile application and website were developed for users to manage and control the smart locker system. The system developed will be used for the lockers in the Computer Science honours laboratory. The system was developed over a semester in 3 phases. The system was evaluated using both quantitative and qualitative methods. Users found the interface to be simple, logical, user-friendly and responsive.

1 Introduction

1.1 Home Automation

Home automation or Smart Homes refers to the introduction of technology within the home environment to provide convenience, comfort, security and energy efficiency to its occupants. [1] Home automated systems are becoming increasingly popular due to increased affordability of phones as well as advancements in technology. With these factors, a lot of work is going into automating our daily activities by incorporating of the mobile technologies into the automated systems. [1] Some examples of automated systems are: turning off lights, turning on and off of air conditioning systems or opening a door with a mobile phone.

1.2 Internet of Things

Internet of Things refers to connection and intelligent linking of devices like smart phones, TVs and sensors to the internet and enabling them to communicate with humans and with themselves. [2] According to Hilton [2], the number of devices connected to the internet will increase from 100.4 million in 2011 to 2.1 billion in 2021. It is expected that this will result in a dynamic network of Internet of Things. The development of Internet of Things will revolutionise many sectors like health, transportation, energy and financial services. This has created a wide development space for home automated systems.

1.3 Smart Locker System

Smart locker systems are the most common application of home automated systems and Internet of Things. They are used in some homes and hotels. The locker systems are controlled by a mobile phone. Traditionally the method for accessing doors and lockers is the use of keys to unlock locker. This key can be lost or stolen. Lockers with physical keys can easily be broken into by burglars.

Smart lockers that are controlled by mobile devices are effortless and modern. Since the only access is through an application they are less susceptible to break-ins as it gives the users the ability to track and monitor their locker through their phones. Unlike traditional lockers, smart lockers are easier to allocate to people for limited amounts of time without worrying about people not releasing the lockers for a long period after their allocation time expires.

The main objective of the smart locker system is to have the same functionality as a traditional one but is implemented with modern technologies that are internet connected. It gives users more features and functionality that would be impossible using the traditional lock system.

A smart locker system was implemented that allows a user to control a locker using a mobile and web-based application. The system has many features thus ensuring versatility.

1.3.1 Motivation

In the past, the physical key system was the most feasible one but with the advancements in technology, availability and high distribution of affordable smartphones has made it so that better locking mechanisms are developed.

The traditional keys have various disadvantages. There is little to no management of the traditional locking mechanisms. The monitoring of these systems that exists is highly insecure for example in some places the manager will have copies of all the keys of the locks. This can lead to unauthorized access to the lockers. Since there is no way to track locker system the user has no way of knowing the status of their locker such as when it is unlocked by an authorized person.

The lockers currently in the Computer Science honours laboratory have no locks. This created an opportunity to implement a smart, integrated locker system. The goal of the project was to create a system that allows users to access a locker that is connected to the internet that is controlled by an application on their phones. Some of the features include reserving a locker and opening it during that time.

1.3.2 The Framework

The project is comprised of three distinct sections:

1. The user interfaces for both user website and mobile application as well as one for the administration of the system. It communicates with the web server by sending queries for booking and opening lockers.

2. The webserver contains a database for storing user information relating to bookings and lockers and APIs for communication between front-end and locking mechanism.

3. The locking mechanism that is implemented using an embedded device that is programmed to complement its design.

End Users

The end users of this system are the Computer Science Honours students. The system will hopefully eventually be extended to other parts of the university that require locking mechanisms. The administration website will be used by whoever will be responsible for managing the system.

User Interfaces

There are three user interfaces that the users will interact with. These are the website, mobile application and touchscreen. This is the only part of the system the users will have access to. They will be able to control and manage their lockers through these three interfaces.

This user interface is simple and responsive. It should be easy for the users to navigate efficiently. The mobile application is a cross platform one thus allowing users with different phones with different operating systems to use it.

The Webserver

The web server is the heart of our system. It is the back end for the user interfaces. It is responsible for the communication between the user interfaces and the locking mechanisms through APIs. It is hosted on a virtual machine running Linux. It contains a MySQL database that is used to store user information for bookings and lockers. Some of the web servers' main tasks are:

- Authenticating the user using UCT's LDAP system.
- Receives and process requests from the front end and locking mechanisms.
- Store data from both locking mechanism and front end.

Locking Mechanism

The circuitry and wiring of the locking mechanism was designed and installed with help from the electrical engineering department. A raspberry pi (microcomputer) was programmed to complement the design. The device and circuitry will be hidden away from users at the back of the lockers.

The Raspberry Pi is the brain of the hardware components. Python code was written to establish the communication between the webserver and the raspberry pi. The use of wireless technology reduces the amount of physical wiring that would be needed.

1.4 The End-User Interface: The Smart Locker System Mobile Application and Website

1.4.1 Problem Statement

Traditional locking systems have very limited functionality. The most one can do is lock and open the locker when they need to. This is a problem as if they are not able to monitor their lockers from distant locations. It is also an issue for places with few lockers that have many people that would like to use them. It is difficult to keep track of what lockers are free as the lockers are not monitored.

The users for the application and the website for now will be the computer science honours students. It is important to create user interfaces that are simple, user-friendly and efficient for the required functions.

1.4.2 Motivation

The issues discussed in the problem statement above are the main motivation for the smart locker mobile and website application.

The high distribution and affordability of smartphones has created opportunity for research and development. A lot more devices are connected to the internet and controlled by smartphones. This has led to a lot more tasks in homes being automated.

The lockers in the honours laboratory currently have no locks which created a great opportunity to implement a smart locker system. Smart locks are a good replacement for the traditional deadbolts locks which require a key for locking and unlocking. This is an important project especially for the Honours students as they will be able to secure their belongings in the locker.

1.4.3 Research Questions

The main research question for our project consists of investigating if smart locker systems are the best locking mechanisms for improving security of locks. It will address this by looking in detail at the following questions.

1. Is it possible to control an array of locks using an embedded system that allows the user to remotely open their locks from a mobile phone, website or using an RFID?
2. Is it possible to use a webserver to integrate a cross platform applications with an embedded system?
3. Is it possible to integrate a cross platform application with embedded devices to create a secure smart locker system?

The research question for this paper is number 3. The research question for the user interfaces are sub-divided into two:

Usability

Is it possible to create user interfaces that are intuitive and user-friendly for the controlling of the locker system?

Performance

Are the user interfaces responsive and reasonably performing to improve the end-users experience?

1.4.4 The Framework

The smart locker system has two user interfaces. The mobile application that is a cross platform one and the website. These both have the same functionality. They both provide the users with access to the locker system by allowing them to check the availability of lockers, reserve one and open it if they need it. It has a similar functionality as some smart lockers that have been developed.

1.4.5 Ethical, Professional and Legal Issues

The project design was a user centred design. This means it required users to be involved heavily throughout the design process as well as the evaluation of the system. This required ethical clearance from UCT as well as the faculty of Science.

The faculty of Science Research Ethics committee granted ethical clearance for the user testing of the system. The ethical clearance was granted by Professor Timm Hoffman, who is the chairman of the committee.

The questionnaires given to students during the study required consent from them and they were told what the data we gathered would be used for. All the data collected after evaluation was anonymous. All the students had to read and sign the Informed Voluntary Consent before participating in the study.

2.Background

There has been a lot of work done in this field using Internet of Things concepts. This section gives some insight into the concepts and topics that are relevant to the research. The background of these as well as projects with similar research studies are discussed.

2.1 Internet of Things and Smart Environments

The wide distribution of phones around the world has increased the use of internet. A lot more devices can now be connected to the internet such as laptops, tablets and some watches. This led to the concepts of Internet of Things and Home Automation.

The Cluster of European research on Internet of Things defines the “Things” as being the participants in social processes where they are enabled to communicate among themselves and the environment through data exchange while reacting to real world events and running processes that trigger reactions. [3] Forrester defines smart environments as those using information and technology to make the services of education, administration or healthcare interactive, aware and efficient.

Internet of things has been found to be one of the emerging technologies in IT. [1] People use the internet every day for regular activities such as browsing the web, sending and receiving emails amongst other things. With this platform, more people became interested in using the Internet to let machines and smart objects communicate and coordinate. [3]

2.2 Current Locker Systems

2.2.1 August Smart Lock Home Kit

This smart locker developed in 2011 allows users keep tabs on their locks in their houses. It does this using a mobile application. The system uses modern technologies including voice activation, geofencing, and an If This Then That (IFTTT) channel that allows it to trigger other devices. [4]

This locker supports Apple's Home Kit platform and can be controlled using Siri voice commands. Users can make lock and unlock commands as well as make queries about the status of the lock using Siri. The system also supports manual locking and unlocking of the door in case of power failure or if there is an issue with the users’ phone.

2.2.2 Lockitron Bolt

This lock is quite like the one we implemented. This lock enables one to lock and unlock using Android or iOS smartphone. It allows users to monitor their lock by sending push notifications when door is locked and unlocked. It also gives the user administrative rights by allowing them to add and remove users. [5]

2.3 Web vs. Native Application

Most applications nowadays have both a native and web application for example Facebook and Twitter. Both the user interfaces have the same features and functionality. The major difference between the two is that the native application is only compatible with the mobile operating system it was specifically developed for and must be installed on the mobile device while a web application can be accessed on any web browser on the host platform and requires no installation. Developer must decide whether they want to make a native or web application. Some of the following factors are considered important when making this decision.

2.3.1 Languages

The advent of HTML5 and interrelated technologies such as CSS3 and JavaScript APIs has made these common web tools more powerful and capable to produce web apps that rival native apps in terms of functionality, design, interaction, and use of multimedia. [6] Native applications are developed with the language specified by the operating system.

2.3.2 Hardware Compatibility

The fact that native apps are developed for a particular operating system makes them more compatible with most of the phones' hardware. They can easily access the camera, GPS services and others. This is not the case for web apps. For any application that will require hardware a camera is better suited.

Hybrid apps have become popular lately. A hybrid web app is an application that is neither truly a mobile web app nor a native app. It is also developed with HTML5, JavaScript APIs, and CSS however it runs inside a 3rd party native app container. These apps allow the development in web languages while allowing access to native device APIs and hardware. [6] One of the common frameworks currently is PhoneGap.

3.System Design

3.1 Overview

The smart locker system developed requires an internet connection. The users must have either a Wi-Fi connection or mobile data to be able to use the website or the mobile application as well as for accessing their locker.

3.1.1 The User Mobile Application and Website

These two user interfaces are necessary for the user to control their locker in real time. They must be simple, user-friendly and intuitive to keep the user comfortable as they use them.

Features that are controlled through the end-user interface:

1. Logging in and out of application or website.
2. View status of all lockers (availability).
3. Making and cancelling of locker reservations.
4. Opening of locker

3.1.2 The Administration Website

This website is meant for whoever will be responsible for the administration of the entire locker system. Their roles are defined by the website.

Features that are controlled through the administration locker:

1. Adding and deleting users to database.
2. Viewing all students and their reservations.
3. Adding and removing locker.
4. Opening any locker.

3.2 Technology and Tools

3.2.1 The End-User Interface

The end-user interfaces comprised of two websites and a mobile application. We decided to develop a hybrid application. This is an application that is neither a mobile web app nor a native one. It is developed using JavaScript, CSS and HTML techniques. The Apache Cordova open-source mobile development framework was used for the implementation of the cross-platform application.

HTML5, CSS, and JavaScript

HTML5 is the latest standard from the W3C which is the official, non-profit organization that develops and maintains web standards. [6] HTML5 is both the official recommendation from the W3C as well as a more informal term used to group the actual HTML5 standard together with the new JavaScript APIs, and CSS3. These modernize the capabilities of native languages as well as providing all the required functionality to deliver contemporary web applications to a variety of devices. [6]

Some of the important advantages of using HTML5 and CSS3 are [7]:

- The readable codes are semantically accurate which makes it easier for developers to work with.
- No need for third party plug-ins for audio and video.
- Supports cross-browser compatibility.
- Mobile optimisation and responsiveness
- Local Storage can be used to cache JavaScript and CSS files
- allows the persistent caching of files and data that survives browser sessions and power cycles.

Generic mobile web applications (mobile websites)

This is another term for mobile versions of websites. Some of its features are:

Responsive web design

The number and type of devices that can access the internet now are increasing which is making it difficult to for website designers to determine user content. Traditionally using a minimum target resolution was considered best practice. [6] This limits the user experience of some users and may in the long run need to be redesigned to accommodate the capabilities of new devices. Responsive web design reduces these costs while also enhancing user experience.

Jobe [6] describes responsive web design as a concept where cascading style sheets (CSS) are used to determine a device's resolution and adjusting the visuals of the website accordingly. This renders device specific applications unnecessary as the content can be manipulated for various screen sizes allowing designers to create context-sensitive adaptable websites. We used these techniques when designing the end-user interfaces.

Apache Cordova

Since a hybrid application was being developed a framework that extended beyond a single platform without re-design for a different platform was necessary. This was found in Apache

Cordova. This is an open-source mobile framework using the standard web technologies.

This framework provides two development paths which are: Cross-Platform (CLI) and Platform-centred workflow. There is no platform specific development on the CLI workflow, it supports many mobile operating systems. The platform-centred one is based on building an application for a single platform. [8] The CLI workflow was chosen for the project.

3.2.2 Communication Technologies employed

Users need to have absolute control of the system. This is through the communication between the client (user interfaces) and the server.

This system depends on an internet connection. The client needs to have either a Wi-Fi connection or a device with mobile data capabilities. The user must choose a suitable service provider for data. The internet is the best solution for communication between clients and server as there is a lot of activity and data being transferred between the two.

3.2.2.1 Communication from the Server to the Client

Pull technology was used for communication from the server to client. Pull technology refers to a network communication style where the first request comes from the client and is then followed by the server response. [9] It is used for HTTP page requests from websites. This data requested from the server is stored in JSON format.

JSON stands for JavaScript Object Notation. It is a syntax for data exchange and storage. The server uses Flask framework to convert the data to the appropriate JSON format so that it can be understood and interpreted by the client.

The user must be authenticated before accessing the server. They must use their UCT credentials (username and password) so that they can access data on server.

3.2.2.2 Communication from the Client to the Server

JSON format is also used for communication from client to server. The client needs to be authenticated by the server before they can begin sending data to the server. The client can then begin sending requests to the server.

Ajax calls were used for both communication from the client to server and vice-versa. Ajax is a web technology that provides rich interaction features such as smoother responses and continuous real-time support. It provides dynamic user experience for web applications. [10] Only the section of a web page relevant for a call is loaded at a time instead of entire page.

3.3 System Features and Functionality

The user interfaces went through many phases before the final ones. These will be discussed in detail later.

3.3.1 User Authentication

For the user to access the Smart Locker System they must be authenticated by the web server. The students must enter their UCT credentials into the application. These are sent to the UCT LDAP API that will check whether the students can access the system. If they are successful will they gain full access to the locker system.

3.3.2 Status

Locker Availability

The users can view which lockers are available and those that are not available. The lockers are displayed to the users with the same design they have in the Honours lab.

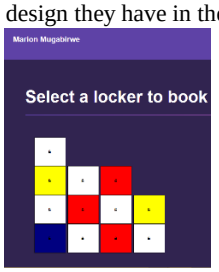

Figure 1: Screenshot of page showing locker status

Time

The users are notified of how long they have their locker and when their reservation time is almost up.

Booking

Users can book any locker that hasn't been reserved by anyone else for given duration.

3.3.4 Profile

Lock/Unlock Locker

Users can unlock their locker as long their reservation is still valid.

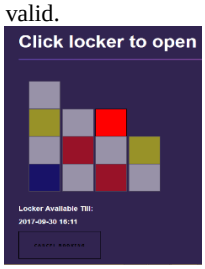

Figure 2: Screenshot showing profile page

Manage Booking

Users can cancel or extend their bookings for a locker.

4. System Implementation

The application was implemented in 3 phases.

1. Feasibility/Prototype Phase: paper prototype was used to gather requirements. Main features of user interfaces developed.
2. First Implementation: all features and necessary were implemented. Integration with back-end was done during this phase.
3. Final System Implementation: the final system was implemented based on feedback from users after the first implementation.

4.2 Iteration 1: Feasibility Prototype

4.2.1 Requirements Gathering

A paper prototype was developed in this phase to get an understanding of what the users would want from the system. 15 participants who are currently doing the computer science honours degree were involved in this phase. The users were asked to perform a variety of tasks and give their suggestions and we also observed them as they did them. Our application has very few features, we noticed that user feedback was starting to be similar after 3 iterations. Some of the common feedback included:

- The colours on lockers confused users.
- Users wanted to be able to undo an erroneous open.
- Uses suggested a cancellation feature.
- Users suggested icons on lockers.

Their suggestions were used to develop the feasibility prototype.

4.2.2 Implementation

Displaying of the list of available lockers to the users

It was noted from the prototype evaluation that the users wanted the lockers on the application to appear exactly as they do in the Honours lab. The lockers in the lab have a variety of colours and shapes. It was important to find a framework that supported this.

The bootstrap framework was used to do this. Bootstrap is one of the most popular front-end component libraries that is used for building responsive website and mobile applications. [11]

Look of websites

Since my section of the project was mainly the front-end it was important to develop a user interface that was simple, responsive and looked good. A customisable template that provided a starting point was used. The template is built on intelligent HTML5 and CSS3.

4.2.3 Evaluation

The feasibility prototype was evaluated by our supervisors Gary Stewart, Sam Chetty, Craig Balfour and the second reader of our project. The supervisors noted the work done and the progress made.

4.3 Iteration 2: First Implementation Prototype

4.3.1 Requirements Gathering

Since most of the requirements had already been collected in the first phase, the requirements gathering was for the team members. Since we had split the project into front-end, back-end and web server, both the front-end and locking mechanism required data from the back-end.

We had several meetings with the team alone and sometimes with the supervisors to establish the data that each section required. For the front-end the following requirements were established:

1. Status of all lockers
2. Status of user

The core features needed for the student system were established for this phase.

4.3.2 Implementation

Both the student and administration interfaces were implemented in this phase. Integration with back-end also happened here.

4.3.2.1 Student user functionality

Authentication of the users

A login page was developed to allow the users to access the system. This required the back-end functionality for authentication. It was done through an HTTP request to server which sent these credentials to UCT LDAP API as mentioned earlier. Both the admin and students need authentication.

Status

Locker Status

Users should be able to see the status of all lockers. A request is sent to the server to return the free lockers. These are displayed on the users' home page if they have not reserved one already. Users can then select one of the available lockers if they want one.

User Status

A user that has reserved a locker is able to view their status. This means they can see how long before their reservation becomes unavailable as well as the locker they reserved.

Opening user locker

Users can open their locker by clicking on the locker they reserved. This prompts a dialog box that will ask the user if they are sure they want to open their locker. This was added to prevent users from opening their locker by mistake.

Manage their booking

Users can cancel their booking if they do not need it anymore.

4.3.2.2 Admin user functionality

Add or delete student

An admin can add or delete students to the system. These can only be Computer Science students for now.

Add or delete locker

An admin can add or delete locker.

View all locker status

An admin can see every students' status. They can see the students' reservations.

4.3.3 Evaluation

This first implementation was evaluated through user testing. Users were asked to perform a series of tasks on both app and website. Users were asked to "think out loud" and make suggestions about the system. They were observed as they used the system. After the tasks, they had to complete the QUIS questionnaire which will be explained in the next section.
6 students evaluated this system. These were all Computer Science honours students who had done the Human Computer Interaction (HCI) module. They were chosen since they had learnt the HCI principles and would be in a better position to evaluate and give feedback on the system.

From this testing, some problems with the system became visible. Some of the problems found or recommended changes include:

Login

- Make the login instructions clearer.
- Enter button should be activated as it feels more natural
- The text showing invalid details should be in red.
- Users did not like where the login section was and said it would be better if it was always the first thing they saw when they accessed the website.

Status

- Users said the buttons needed to be more responsive as it was difficult for them to see that they were buttons.
- Colours of buttons confused some users
- Customise the disabled buttons that show locker is unavailable

General feedback

- Show that the reserving time is in hours
- Give a longer leeway for booking
- Confirmation for actions performed
- Indication for users to enter a number on booking page
- Make the instructions for different functionality better
- Use better dialog boxes that fit the webpage
- Brighten fonts compared to website

4.4 Iteration 3: Final System Implementation

4.4.1 Requirements Gathering

From the user testing it was clear a lot of changes needed to be made. Most of the changes were to do with the look of the system. The requirements for this phase came from the users as stated above in previous phase. Some of the features were implemented and those that were not are included in the future work section.

4.4.2 Implementation

Most of the features suggested by the users could be implemented easily. The most significant changes added will be discussed here.

Toast Messages

These are a fast way to tell users something. They are typically short pop-up messages that will appear on the screen for a few seconds and then disappear. [12]
Originally, I was using the default dialog boxes for the browsers. The materialize framework provides this functionality. It provides a way to send unobtrusive alerts to the user. These messages are well placed and sized. They were used to show users a confirmation of their booking, cancellation confirmation amongst other things.

Icons

The lock and unlock icons were added to the buttons representing the lockers. This was to make the status of a locker clearer to users. The available lockers had the unlock icon while those that are unavailable have the lock icon placed on them.
These icons were added from the font awesome library. Font Awesome is a library of icons.

Responsiveness of buttons

Users stated that the buttons needed to be more responsive to show when they are hovered over. This would make it easy to see which ones are functional especially for showing status of lockers. Hovering effects were added to the buttons using bootstrap framework.

Fonts

The size and type of fonts were changed to make instructions clearer. A monospaced font was used. The placement of instructions was also changed as users did not seem to see them before.

Tips and Tutorial

A tips and tutorial section was added to the start of the website and application. This was because users complained about not understanding how to get started with the system. This section also included contacts of the admin staff that users could contact in case of any issues.

4.4.3 Evaluation

The final evaluation was a user experience testing that will be discussed evaluation section following.

5 Evaluation

Software testing is important as it used to test effectiveness of product and improves the quality of the end-product. [13] It is important to test to find any oversights or issues that developers may not be aware of.
There are two types of testing we did that will be discussed in this section.

5.1 Design

Before starting the testing process, it was important to have a plan in place for how this would be done. This was so we could know the type of testing we would be doing as well as the users who would be involved in this process. Originally, we had planned to do testing during development however there was not enough

time to do this. We settled for functional testing and user acceptance testing.

5.1.1 Functional Testing

Functional testing is a testing technique where all features and functionality of software is tested ensuring all the scenarios are covered including all failure paths and boundary cases. [14] This prevents any overseen bugs from reaching the user. The testers need to understand the expected performance of the system.

5.1.2 User Acceptance Testing

This is the most important part of the software testing as the project is a software engineering one and depends a lot user feedback.

Usability is defined as the ease of use and learnability of a human-made object. In software usability is the degree to which software developed achieves quantified objectives with the effectiveness and satisfaction that is required for a context. [14] The user interface is studied under controlled conditions.

For this testing users were given a set of tasks that needed to be completed. To access their interaction with the system we observed them and asked them to "think aloud" as they used the system. After completing the tasks, the users were given a QUIS7 questionnaire to fill. QUIS7 questionnaire allows users to provide a subjective assessment of recently completed tasks thus providing a standardised measure of the quality of the system as well as overall usability. [15]

5.1.3 Data

The users must be already added to the server. This way they can use their UCT credentials to access system. The admin website was used to add the users who would be accessing the system.

5.1.3 Users

Three types of testers were required for evaluating the final system.

1. Functional testers: the members of the team were responsible for this testing. Since the project was divided into front-end and back-end it was important to work together to make the integration simpler and ensure that each section was performing its required functionality.
2. Experts: these are honours students who have completed the HCI honours module.
3. Target users: this was comprised of any students in the computer science honours class. This is because the system implemented was for the honours lockers which made them the target users.

5.1.4 Testing Process

User experience testing was done over a one week period at the project end while functional testing was done after the first iteration. Functional testing was done first to improve the experience of the target users by finding any overseen bugs. The expert users then evaluated the system and gave their feedback which resulted in changes which made for a better system for the final testers, the target users.

5.2 Results

Each group provided different feedback that would be added to the system. This way the next testers would not make the same recommendations.

5.2.1 Functional Testing

Only a few errors were found by the other team members as they interacted with the front-end. These were likely overseen during development but were noticed immediately.
One bug found was that users could book more than one locker at a time. This is not the expected functionality as users must only be able to book one. This happened to be both a back and front-end problem. To fix on the front-end users were prevented from going back to home page where they pick a locker. If they do go back they would be notified through a toast message that they have a locker already and cannot pick another unless they cancel their previous reservation.

The text showing that the user had entered the wrong credentials happened to be repeated as many times on the screen as the user entered the wrong details. This is because the string showing this text was being appended rather than being cleared and placed back.

5.2.2 Expert Testing

This was the most beneficial testing of all the three as it was evaluating the look and feel of the system using HCI experts. They identified the following issues in their evaluation:

Space Optimisation
The expert users noted that a lot of space on the website was not being optimised. They suggested moving the information from the bottom of screen to the unutilised space. This would prevent users from having to scroll down but rather just look to side.
On the booking page one expert suggested placing the entire locker screen there in case users needed to remember which locker they had booked as well as utilise the space on that page.

Headers
The names of the user were placed as header on each page and the instruction for whatever action needed to be taken was placed as second header. This was confusing for some users and they suggested only placing the instruction as main header.

Feature explanation
Some of the users found the lockers displayed as buttons confusing. They stated that the colours were confusing and

thought there was a meaning to them. It was explained to the users that the colours were like the ones on the lockers in the Honours lab. This suggestion was taken into consideration but it was decided to keep the colours so users could know which locker they were selecting. A tutorial and tip page was added to explain this to users. Icons were also added to the buttons displaying lock or unlock icon depending on the status of the lockers.

Placeholders

The experts suggested adding placeholders in sections were users needed to input some data. These would guide the users on what data was needed for a text box. These would disappear as the user entered text. These would be an addition to the labels next to each textbox thus users would not need to remember the input needed.

Website/Application Consistency

The experts noted that some features were inconsistent with the overall layout of the user interfaces. The most notable one was the dialog boxes. Originally, the default dialog boxes for browsers were being used. They did not look anything like the website thus needed to be changed. The materialise framework was used to create better dialog boxes matching the look of user interfaces.

5.2.3 User Experience Testing

After the recommended changes from the expert users were implemented to the system it was finally ready for user experience testing.

Most of the usability problems are normally detected by six test subjects and testing with anymore will not reveal anything new. [16] This was clear in our study after a few tests users were simply giving the same feedback. Testing was done with 6 Computer Science Honours students. Most of whom are familiar with web and application development which made them ideal for the testing.

The testing was done individually after gaining consent from the users. They were given a set of tasks to complete. They were requested to complete these tasks on their own and only seek our help when confused.

5.2.3.1 Findings

Users typically completed each the entire set of tasks in less than an hour. Each task and findings from task are summarised below:

Logging In

Users did this easily but had concerns on whether system was storing their passwords and if system was secure.

Booking a Locker

The addition of the lock and unlock icons were helpful to the users to buttons showing the buttons. They could select one easily.

Opening Locker

Users found this page confusing and suggested putting all lockers there so they know which lock they had booked.

Cancelling Reservation

Users suggested adding tooltip to explain this feature. This was added.

5.2.3.2 Feedback

After completing these tasks users were required to fill in the QUIS7 questionnaire.

QUIS7

This questionnaire typically consists of 10 sections for specific system aspects. Only four were relevant to the project and these are:

1. Overall User Reactions: how the user felt about the entire system.
2. Screen: evaluate the interface features and whether the sequence of screen was clear and logical.
3. Learning: determining the simplicity of the system. This is to determine whether getting started and eventually using advanced features was easy for users given how short a time they had to interact with system.
4. System Capability: this was for evaluating the system's ability to execute an action.

The QUIS7 questionnaire had a Likert scale from 1-9 for these sections. The results for the 6 users are illustrated in the graph below.

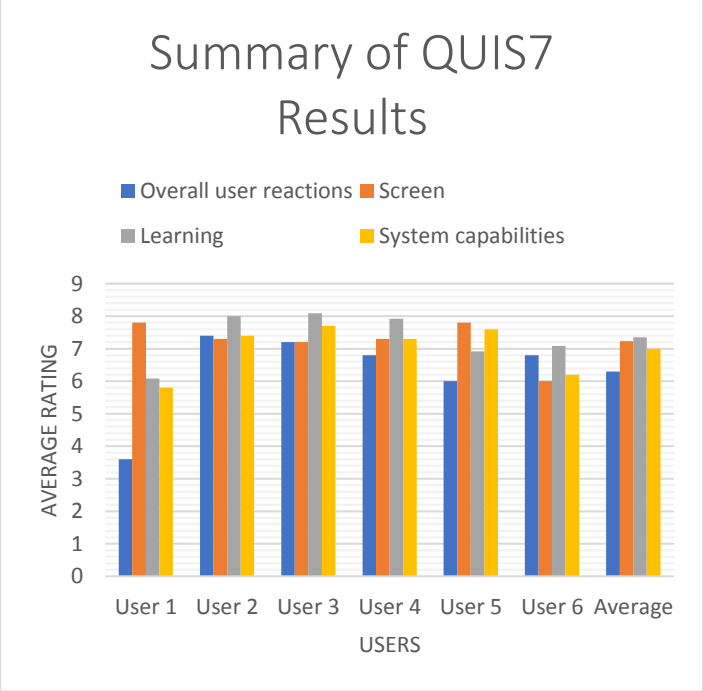

Figure 3: Graph showing a summary of QUIS7 results from 6 users

The results showed that overall user reactions were not as expected as the average was 6.3/9 meaning there is quite room for improvement. The individual ratings varied from user to user.

The screen reaction results with an average of 7.23/9 showed that the users found the features clear and the sequence of screens quite logical.

The learning results had the most discrepancies between the users as one user averaged 3/9 while another averaged 8/9. This discrepancy was caused by the lack of a help feature with tips and tutorials for using the system. This feature was added as a result.

Lastly the system capabilities results were similar for all the 6 users with an average of 7/9. This showed that the users found the system to be responsive, fast enough and reliable as they were hardly any system failures.

The general feedback is that users found the system to be reliable and found the interface simple and user-friendly. There is need for improvements in some sections which are covered in the future work.

6. Future Work

Some of the ideas and features that were mentioned in the project proposal were not implemented due to time and budget constraints. Some of the features and functionality were also beyond the project scope or hindered by the mentioned constraints. The few that we would be implemented in future iterations are:

6.1 Push Notifications

This is a message that pops up on a mobile phone. They are sent even when user is not in the app or even not the device. This is like the working of SMS text messages. They available on most mobile platform. [17]

The notifications would be used to alert the user when their booking is almost done or if they have exceeded their time limit. This way the users would not be penalised.

6.2 Continuous Data Update

In the future data would be updated continuously through threading. The user would not need to refresh page just to see updated status of lockers.

6.3 Token Based Authentication

This is a security technique allowing users to access server using a security token provided by the server. We had initially tried to do this but had a lot of issues implementing it and time did not allow us to correct them.

The advantage of this besides security is user would not have to authenticate themselves with username and password each time but rather just one layer of authentication as token is used for all other resources.

7. Conclusions

The number of smartphones is increasing due to various factors such as affordability. The types of devices that can be connected to the internet has also increased. These devices range from tablets to watches. These have many features and functionality. This has created a platform for developers to develop some incredible systems and concepts. One such concept is the Internet of Things. This concept is used in the implementation of home automated systems, the most popular being smart locker systems.

The smart locker system that was implemented is made of 3 parts: the front-end, web server and the locking mechanism. The web server is the heart of the entire system. The front-end and locking mechanism both communicate through the web server. It is responsible for most of the functionality of the system. The implementation was carried out in 3 phases each consisting of requirements gathering, implementation and evaluation of the system.

The three phases were the prototype that was developed and evaluated with and by the users, software feasibility prototype evaluated by the supervisor Gary Stewart and the second reader. The final system was evaluated by the users.

The research question about integration of webserver and user interfaces was only answered in the final implementation phases. The integration of hardware and software took a bit of time which is why it was only accomplished later in the project.

The testing consisted of 3 types: functional testing by the team members, expert testing by HCI experts and user experience testing. They were carried out in that order with each testing optimising the system for the next testers. Testers were all computer science UCT students and who had all consented to participating in the study. Ethical clearance was granted from the faculty for this study.

The system was implemented over a three-month period. This placed a time constraint on development thus certain features and functionality could not be implemented. The system implemented though it's a prototype it can be used as a starting point for further implementation of smart lockers around UCT.

The research questions were answered and showed that it is possible to use a webserver to integrate a cross platform applications with an embedded system that gives control to users using either a website or application whose user interfaces are intuitive, responsive and user-friendly.

8. Bibliography

[1] Rajeev Piyare, 2013, Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone, *International Journal of Internet of Things*, 2(1), pp.511

[2] Baraka, K., Ghobril, M., Malek, S., Kanj, R. and Kayssi, A., 2013, June. Low cost arduino/android-based energy-efficient home automation system with smart task scheduling. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on* (pp. 296-301). IEEE.

[3] Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), pp.1645-1660.

[4] August Smart Lock. *August.com* Available at: http://august.com/products/august-smart-lock-bluetooth/ [Accessed September 17, 2017]

[5] Bolt. Available at: https://lockitron.com/ [Accessed September 17, 2017]

[6] Native Apps vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies*, 7(4).

[7] Taylor B, Advantages of HTML5 and CSS3. Available at: http://www.business2community.com/tech-gadgets/advantages-of-html5-and-css3-0322078#2Ik64SX1qehtkeAx.97 [Accessed September 17, 2017]

[8] Cordova. Available at: https://cordova.apache.org/docs/en/latest/guide/overview/ [Accessed September 20, 2017]

[9] Pull vs Push Technology. Available at: https://simplicable.com/new/pull-vs-push-technology [Accessed September 20, 2017]

[10] Kim, D.J., Yue, K.B., Hall, S.P. and Gates, T., 2009. Global diffusion of the internet XV: Web 2.0 technologies, principles, and applications: A conceptual framework from technology push and demand pull perspective. *Communications of the Association for Information Systems*, 24(1), p.38.

[11] Bootstrap. Available at: https://getbootstrap.com/ [Accessed September 21, 2017]

[12] Toasts. Available at: https://developer.android.com/guide/topics/ui/notifiers/toasts.html [Accessed at September 21, 2017]

[13] West, B.T., Welch, K.B. and Galecki, A.T., 2014. *Linear mixed models: a practical guide using statistical software*. CRC Press.

[14] Walton, G.H., Poore, J.H. and Trammell, C.J., 1995. Statistical testing of software based on a usage model. Software: Practice and Experience, 25(1), pp.97-108.

[15] Sauro, J. and Kindlund, E., 2005, April. A method to standardize usability metrics into a single score. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 401-409). ACM.

[16] Nielsen, J., 1994. Usability engineering. Elsevier.

[17] Push Notifications Explained. Available at: https://www.urbanairship.com/push-notifications-explained [Accessed at September 27, 2017]