LOCKIT

Raees Eland
ELNRAE001
University of Cape Town
elnrae@myuct.ac.za

Marion Mugabirwe
RGNMAR001
University of Cape Town
rgnmar@myuct.ac.za

Norman Pilusa
PLSNOR001
University of Cape Town
plsnor001@myuct.ac.za

1 Project Description

With the advancements in technology, a lot of tasks in homes are being automated. Home automated systems have the potential to increase the standard of living for people in many sectors such as health, education, security and other aspects of our daily lives. Some of these tasks include controlling temperature in homes, lights and cameras. The most popular smart home system currently are smart door locks. The newly renovated Honours lab lockers currently have no locking mechanism, which is not ideal for security and usability reasons. The current locker systems that have been developed offer only basic security and fail to provide control in real time. They also use outdated electrical and communication technologies that hinder the use and development of smart systems. Using the current developments in Internet of Things and advancements in Home Automation Systems, we are proposing a smart locker system to implement in the Honours lab. 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 lock. It is also important because it gives more control to the users. The project proposes building an integrated system using a Raspberry pi (embedded device) controlled by an Arduino (microcontroller) that will manage the locking mechanism and the user will control the entire system using a mobile and web application. The idea is to build a system that is usable, convenient, efficient and self-managing. Some of the current smart locks can be opened from a smartphone but do not have a dedicated smartphone app and requires a smart home hub to control it remotely. The system will allow the user to lock and unlock the door through an application which allows for more features thus ensuring versatility.

2 Problem Statement

Students require lockers to provide a personal secure storage for their belongings to reduce their load as they commute between classes on campus. The use of traditional physical keys for locks poses problems in the modern world.

2.1 Traditional methods for Access Control

The most common traditional method for accessing doors and lockers is the use of keys. 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.

Besides that, keys have various disadvantages. They require one to carry a key with them always as they must be physically present to unlock locker. This key can be lost or stolen. The user of a lock must manually lock the locker and it also adds an extra step in the unlocking process. Lockers with physical keys can easily be broken into by burglars. It is easy to make a copy of a key thus they do not provide security.

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. Traditional locks would require the locker administrator to physically go to the locker and open it with a master key or cut padlocks.

2.2 Management of locker system

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.

Our system proposes a smart phone integration for monitoring the status of the lockers. The users as well as administrator can monitor the system to prevent any unauthorized access. The current locker systems lack these features that provide information about locker status to the users for security purposes.

2.3 Clients and Users

The current clients for this projects are Computer Science Honours students. This project can be extended beyond this in the future. The project could be extended to lockers around campus. The system can also be extended to other real-life situations. The possible ones are hotels, homes and hospitals. All these places require secure locking systems. Our proposed would give them a system that gives them control of their locks in real time. The administrators of the system should be able to override the system and make any changes required. The users in these places would be doctors, lecturers, nurses and security personnel.

2.4 Main research question

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?

3 Methods and Procedures

3.1 Software Engineering

We will be using the Embedded System lifecycle.

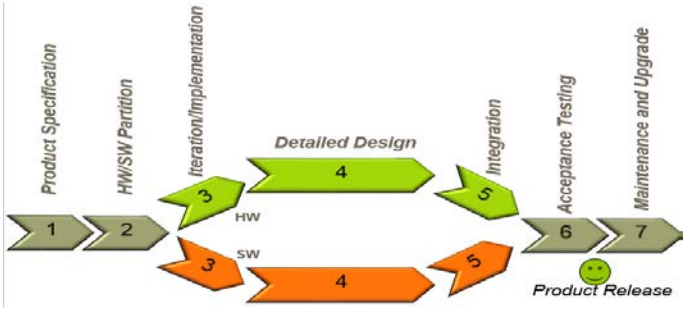


Figure 1: Embedded Lifecycle

Each phase has a predefined start and end. All the phases have deliverables that are used to communicate with the next phase. This methodology allows various tasks to be assigned to specialized teams. One of the team members will be working on interfacing with the hardware while the other two will be looking at the software components. (Website and mobile application)

This methodology has all the outcomes and artefacts of each phase pre-defined. This makes it easy to evaluate the progress of the project. To minimise risks we do not intend to follow this model step by step as requirements may change and we need to be flexible. Weekly meetings with supervisor will help monitor requirements changes.

3.1 User Interfaces

3.1.1 Methods

3.1.2 Testing

3.2 Webserver

3.2.1 Methods

The webserver contains a database for storing user information relating to bookings and lockers and APIs for communication between front-end and locking mechanism. The webserver will be hosted on a virtual machine running Linux. The virtual machine will be created and configured by the systems.

On the virtual machine, apache HTTP server software will be used for the server. Apache is open source and has a lot of supporting documentation and user community.

A MySQL database will be installed to be used as a store for user information. MySQL was designed for web-based application and it offers a high performance and robustness for read and write operations. The schema of the database will be designed first then implemented.

The APIs on the server require a versatile tool, so we will use flask to create APIs. Flask is a web frame work for python. The APIs we develop will be based on use cases of the system. Each API is a module to handle a set of related functions or example there will be an API for handling communication with database, one for communication with locking mechanism and one for processing requests from front-end.

Authorisation of users is handled by the UCT LDAP. Request for authentication from front-end will be redirected to UCT LDAP and the results of LDAP will be used to grant or deny user access.

3.2.2 Testing

Server operation:

Connection to the webserver will be the first thing tested. To test server operation, we will attempt to connect to it from different devices not used during its development using the server's IP address and domain name.

User authentication

To test whether user authentication works, the details of a known UCT honours student will used to access the UCT LDAP and print out results to screen. Another non-existent student will also be requested from the UCT LDAP and the results will be printed to screen and compared for validity.

Database read/write

Information such as booking time, duration etc. will be stored and read from the MySQL database. To test successful reads or write, a write will be followed by a read to test successful write. A read is successful if it outputs an expected result to screen.

API calls/responses

The different APIs will be tested using requests. Each API will keep a log of activities particularly requests it gets and responses it gives. For every request made there is an expected result, the log will be used to check if the API received the request and what it has returned. The result returned will be checked if it corresponds to what is expected.

3.3 Locking Mechanism

3.3.1 Methods

Developing a working and well-functioning locking mechanism with the required circuitry will be done in stages. First a well-designed and detail plan will be developed alongside help from the engineering department. Taking this design, it will then be simulated on a computer using electronic engineering circuitry simulation software. Using the design a prototype will be built using LEDs. The LEDs will be used in place of the solenoid locks. Each LED will correspond to one lock. By controlling the embedded device remotely, the device can simulate the unlocking of a locker by turning on the corresponding LED. The embedded device will also need to be programed to complement the design of the locking mechanism. Once the hardware simulation works, building of the final locking mechanism can take place.

With help from the electric engineering department, the first step is to wire the system and install the locks on the prototype lockers provided. The system will be designed in a way that makes it easy to debug and locate a lock module that is not working. This procedure depends heavily on a detailed and well developed design. From this design, it will then be easy to simulate it using the LEDs and then building the locking mechanism.

3.3.2 Testing

Various tests will be conducted during the development of the lock mechanism. First all locks will be tested to see if they all function correctly. Once the programming of the embedded device is done, it will be connected to an array of LEDs that represent the locks. Commands will sent to the device remotely to test if the device is doing what it needs to do (open the locker specified). While the device is being programmed, the construction of the locking system will take place. Once this is complete stress tests and stability tests will be done to make sure the locks are secured and safe from physical threats. The next part will be to test the system when the device is connected to the locks. The device and circuitry will be hidden away from users at the back of the lockers. User tests can then commence for the locking mechanism. The tests will be conducted from the command line until integration with the user interfaces and webservers happen.

3.4 Integration

Integration of the three components of the system will a critical part of the system. To ensure smooth integration, each component of the system will use log files to record its outputs and the format of the output.

The log files will be shared between team members on a version control repository, bitbucket. Some of the outputs will be inputs for some other component. However, expected inputs will be predefined to ensure independent development. As development progresses, formats of inputs and outputs may change. Regular meetings will allow early tracking of these changes.

The front-end will create a log of requests which will be inputs for the webserver. The webserver will have a log with some outputs that are inputs to the front-end. The other outputs from the webserver will be inputs to the locking mechanism.

The final stages of development iterations will include replacing logs with actual connection of the relevant components.

3.2 Prototype Design

Time will not allow us to develop a functional prototype to emulate the working for our entire system. However, we will create an initial paper prototype. During prototyping we record user suggestions and create a list of requirements to meet. When our design has all the requirements incorporated we will use a Likert scale to record user satisfaction.

Prototyping process

- Develop low fidelity paper prototypes to encourage user inputs.
- Create scenarios to help users visualise the system in action
- Use iterative approach to allow project flexibility during development

The use of low fidelity paper prototypes encourages users to give inputs to the design of the system. Research has shown that users give more input when using paper and sketches than when using a well-designed prototype using software.

Research [4] has shown that scenarios help users understand a system. Scenarios can be used to identify cases which are not easy to spot otherwise. Scenarios also encourage user participation because it is engaging when used during prototyping.

Iterative development ensures continuous deployment. It allows early detection of problems in the system. This is very important because our project will split into three parts which makes likely to have issues especially with compatibility.

3.3 Testing Procedure

Usability studies will be conducted during the design process for both the web and the mobile application. This will help guide the design of the user interfaces by giving insight into the functionality. After developing the applications, another set of

usability tests will be conducted to evaluate the features and get feedback from the user. Any changes needed to be made will be done and this will be final system.

With regards to the web server, a prototype will be developed that has predefined outputs we expect to get when certain inputs in the form of requests are sent to it. To test server functionality, requests will be sent to it and assess the response we get if it matches the expected response. The server will be tested to see how it handles multiple requests and multiple writes. This will be done by sending multiple requests for the same resource e.g locker from different devices to see which device gets the correct response and what happens to the others.The locking mechanism will also be given inputs to test its performance if it does what is expected. The outputs of the locking mechanism will be printed to screen for comparison with the expected output; the expected output would be an input to the server.

Testing the locking mechanism in isolation from the other components will require sending commands to the embedded system via the terminal to open locks. Stability and durability tests need to be conducted as well to determine if the locks can withstand physical force.

4 Ethical, Professional and Legal Issues

4.1 Legal implications:

- We will not store username and password, instead we will use OAuth to authenticate users on the UCT HTTP service. However we will keep a database of student numbers that are allowed to use the lockers. Other information would be locker number and reservation duration which does not have any legal implications.

- Our system is responsible for protecting user belongings. We will provide a warning to users that belongings are stored at own risk and that our solution is not liable for any theft of property.

- User belongings cannot be accessed by anyone other than themselves via the access protocols provided (RFID, Mobile app, Website). If users fail to collect their belongings after their reservation ends, the locker will remain closed. Users will need the administrator to open the locker for them if the reservation time expires.

4.2 Ethical clearance for experiments:

We need user inputs for UI design. User names will be anonymous. We will use paper for initial prototypes to gather user inputs. We might take pictures of users interacting with prototypes but faces will not be shown. In later user evaluations, we might use a questionnaire to get feedback on iterations, we will not include any personal details of users except for their opinions.

4.3 Intellectual property rights:

The project/idea is proposed by UCT computer science department. Development of the project will be done together with members of the department. Software components to be used are open source e.g reactjs, android studio, java etc. The department also provides hardware to construct the necessary hardware components of our project. The project is for the gain of the university and purely for educational purposes. The resulting software will be publicly available for anyone to view. Our project will not be patented by any team member because it does not belong to any of us.

5 Related Work

Many projects of a similar nature have been done before. Many using a single microcontroller to control the locking mechanism. For this project we are looking at using two devices, namely the Raspberry Pi and the Arduino Uno for our smart locking system. In a project done by Rafid Karim and Haidara Al-Fakhri [1] a smart door lock was achieved through Near Field Communication technology and a Powered over Ethernet (PoE) circuit board. The PoE contained the MSP430 microcontroller which was connected to a NFC reader through the Serial Peripheral Interface (SPI) [1]. This allows the smart lock to verify that the user is close to the lock when the user opens it over a wireless network. However this will only work if the phone provides NFC as well. The system makes use of a cloud server as a middleware between the phone and the PoE circuit board. The phone sends the commands to the cloud server, the server sends it to the PoE and if the user is close to the lock the PoE circuit board processes the command. In another project done by Lia Kamelia et.al [2], a smart locking system is achieved through Bluetooth technology. An Arduino Uno with a Bluetooth module attached is used to control the lock. An android phone is then used to connect to the Arduino over Bluetooth to unlock the door. Pandurang [4] et. al used a similar approach, but they added an extra feature in the form of motion detection technology. A camera was attached to the system and if a user wanted to open the lock the camera had to first detect a human in front of it. All the projects looked at do not allow an array of lock to be controlled from a microcontroller or single device. This project will do just that, it will make use of an embedded system to achieve the project goal. Since an array of smart locks adds more complexity, extra resources are required.

6 Anticipated Outcomes

6.1 System outcomes:

Software's produced are:

1. Code for controlling physical hardware
2. Website for users to interface with locker
3. Website for administrator of locker system
4. Mobile application for users to interface with locker

The key features will be:

- A touch screen next to the lockers running the mobile app.
- A web site for interacting with the lockers.
- A locking mechanism controlled by embedded system.
- A mobile application for interacting lockers.

Major design challenges:

- Debugging the hardware aspect of the project
- Prototyping the hardware
- Designing user interfaces based on inputs from people who might not even be the users of the system. The

students are leaving UCT.

6.2 Project impact:

Expected results:

The resulting system is expected to function as follows:

1. A user will be able to login using their student number and password.
2. The user will be able to book a locker or
3. Open a previously booked locker.
4. When booking in 2, the user can either select a specific locker (provided it is available) or chose to be given a random locker.
5. The administrator can set the maximum number of hours a user can have a locker for.
6. User can have a locker for not more than the hours set by the administrator.

Difference made by results:

- Use of username and password ensures authentication before access, hence provides security.
- User booking allows control over access to lockers. Ensures that everyone gets a chance to use a locker.
- Access to only one locker per username, prevents users from owning multiple lockers.
- Time limit on locker ensures that lockers are free for next user.
- Random allocation of locker, reduces decision time for users. (minimise Paradox of choice by Barry Schwartz)
- Overall the resulting solution provides a safe system that can be used in public environments to protect belongings. The system can be monitored to ensure fair use by the target users.

6.3 Key success factors

Factors to measure project success:

- Schedule: If we finish the project within the set deadline.
- Scope: When the finished system has completed all the items on the scope of our project.
- Budget: if we complete our project within the given budget.
- Team: Every team member finds valuable lessons from the project and completes their section on time.
- User satisfaction: User ratings to give an indication of their satisfaction with the project outcome i.e. how well the outcome meets their security needs.
- Quality of outcomes: the number of tests that our system passes.

7 Project Plan

7.1 Risk and Management

The Risks involved and the mitigation strategies can be found in Appendix A. Members not doing or finishing their part impacts this project significantly.

7.2 Timeline

The Gantt chart outlining the timeline of the project can be found in Appendix B.

7.3 Required Resources

Important Hardware needed:

- Embedded Device
- Locker Cabinets
- RFID reader
- Touchscreen
- Android/IOS Phone
- Solenoid Locks and Latches

Anticipated Software that will be used:

- Phone Gap
- Ubuntu Server
- Apache
- MySQL Database

Honour Computer science students will also be needed to participate in our usability tests.

7.4 Deliverables

The main deliverables for this project is the Cross Platform application, User and admin Websites and the Locking Mechanism. Other important deliverables for the project are:

1. Final Project Report
2. Final Project Code
3. Digital Poster
4. Project Website and Reflection Paper.

7.5 Milestones

The project Milestones are listed on the Gantt chart (Appendix B). The three key milestones are:
Stage One: This stage is the development of the system. Each member will develop their part of the system and test it while it is being developed and once done, integration will take place

Stage Two: This stage involves user tests of our system. This will be the first tests conducted and the first version of the system will be produced

Stage Three: This is the second round of tests with the user. This will be the final user evaluation before the system is produced as a complete product.

Stage Four: This stage involves finalizing the system and putting it together to be presented as a final working system
Other milestones include the final report, website and reflection paper.

7.6 Work Allocation

There will be three sections in the project. Marion will develop the user interfaces, Norman the Webserver and Raees the Locking Mechanism.

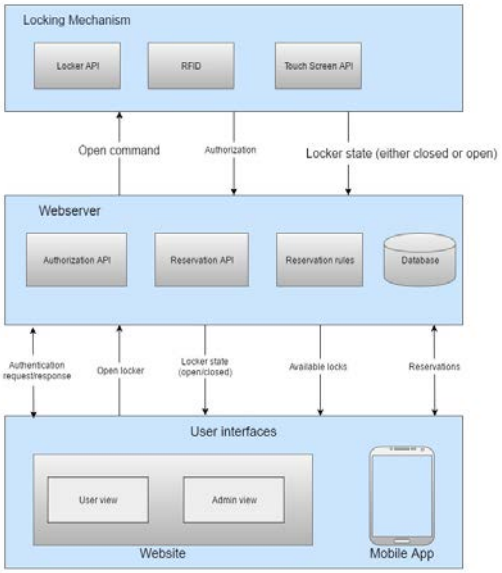


Figure 3: Architecture of system

References

[1] Rafid Karim, Haidara Al-Fakhri, “Smart Door Locks”, December 2013

[2] Lia Kamelia, Alfin Noorhassan S.R, Mada Sanjaya and W.S., Edi Mulyana, 2014, Door-Automation system using Bluetooth-Based Android for Mobile Phone, *ARPN Journal of Engineering and Applied Sciences*,9(10),pp.1759 – 1762

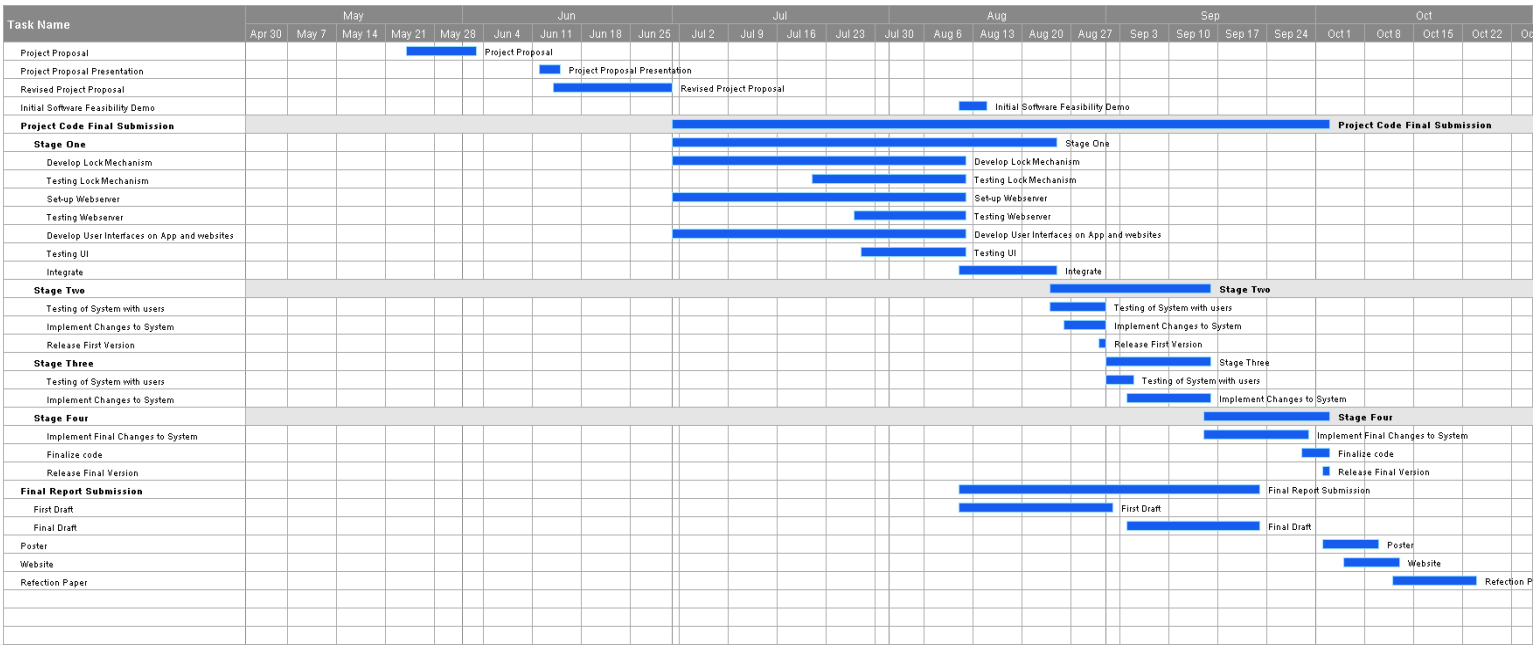
[3] Bhalekar Pandurang, Jamgaonkar Dhanesh, Prof. Mrs. Shailaja Pede, Ghangale Akshay, Garge Rahul, 2016 ,Smart Lock: A Locking System Using Bluetooth Technology & Camera Verification, *International Journal of Computer Applications*, 4(1), pp.136–139.

[4] Bødker, S., 2000. Scenarios in user-centred design—setting the stage for reflection and action. *Interacting with computers*, *13*(1), pp.61-75.

Appendix A

Risk	Impact	Probability	Mitigation
Members unable to complete their section on time.	High	Low	Make sure there is a clear division between tasks that need to be done Help each other if one task proves to be too much for one person.
Hardware issues (Cannot get the specific hardware, wrong hardware bought and broken hardware)	High	Medium	Keep spares. Make sure then buying a piece of hardware that the specs match your proposed system.
Scope of project is too large to complete in the time frame	Medium	Medium	Before adding functionality, discuss the impact it will have on the project. Start tasks early
User interface too complex and not user friendly	Medium	Medium	Conduct user studies before building the user interface. Conduct usability studies once the first prototype of the system is completed
Integration of the different tasks done by each member is unsuccessful	Medium	High	Provide enough time for integration Make sure each member knows exactly how each of their components will fit into the final system

Appendix B


Task Name
May
Jun
Jul
Aug
Sep
Oct
Apr 30
May 7
May 14
May 21
May 28
Jun 4
Jun 11
Jun 18
Jun 25
Jul 2
Jul 9
Jul 16
Jul 23
Jul 30
Aug 6
Aug 13
Aug 20
Aug 27
Sep 3
Sep 10
Sep 17
Sep 24
Oct 1
Oct 8
Oct 15
Oct 22
1 Project Proposal
2 Project Proposal Presentation
3 Revised Project Proposal
4 Initial Software Feasibility Demo
5 Project Code Final Submission
6 Stage One
7 Develop Lock Mechanism
8 Testing Lock Mechanism
9 Set-up Webserver
10 Testing Webserver
11 Develop User Interfaces on App and websites
12 Testing UI
13 Integrate
14 Stage Two
15 Testing of System with users
16 Implement Changes to System
17 Release First Version
18 Stage Three
19 Testing of System with users
20 Implement Changes to System
21 Stage Four
22 Implement Final Changes to System
23 Finalize code
24 Release Final Version
25 Final Report Submission
26 First Draft
27 Final Draft
28 Poster
29 Website
30 Refection Paper
31
32
33