National College of Ireland

BSc (Honours) in Computing BSHCIOT4

2016/2017

Steffan Schmieder

x13757775

steffan.schmieder@student.ncirl.ie

Smart Lock

Technical Report

National
College of
Ireland

Table of Contents

Executive Summary

The idea behind the project is to create an Android application that allows users to control the locking system of a smart lock from their phone. Users who have a smart lock installed in their home have the power to lock and unlock their home from anywhere in the world. It uses your devices Wi-Fi/4G to connect to your home Wi-Fi system which will then connect to the smart lock. This means you don’t have to carry around a set of keys all the time and eliminates the chances of arriving home and being locked out of your house if you lost your keys. You can’t always be at home so opening the lock from anywhere for kids, relatives, friends or cleaners if they don’t have keys or access to the house is appealing for many people. Some other features the application includes is granting or revoking other users access at any time and timed locking/unlocking. This is useful for users who want the locks for their house to automatically lock late in the evening as a safety feature in case they forget to lock the house before going to bed. The application provides resent activity logs to show users who has been entering their home and at what time. This app will give users more freedom and control over their own home allowing them to plan their day more freely. The application initially will only be available on android devices as it is created using Android Studio and is written in Java.

1 Introduction

1.1 Background

IoT is a very new and up and coming technology with more and more products and devices being developed each year. Internet of Things (IoT) is the connecting of physical objects or things to the internet. It is seen as an evolution from Machine-to-Machine (M2M) communication and is built upon cloud computing technologies, relying on networks of various data gathering sensors communicating across the internet. It allows for real time analysis of environments and for actions to occur based on those readings. The technology is used for self-driving vehicles, smart electricity, smart health monitoring systems to smart homes with many everyday tasks being automated for convenience. The concept of the Internet of Things, wasn’t officially named until 1999 but some of the first examples where around in the 1980s. Approaching 2013 the Internet of Things had matured into to a system using multiple technologies from wireless communication to the Internet and embedded systems. The standard fields of automation such as automation of buildings and homes to wireless sensor networks, control systems, GPS and others, all support IoT. IoT can be thought of as any device that can send or receive information that is connected to the Internet. This includes a vast number of devices, ranging from cellphones to building maintenance, cars, kitchen appliances, medical devices such as a heart monitors or biochips in a farm animals, can transfer data over a network and are members of IoT. In the past the equipment was slow and expensive which made it hard for companies to make a successful product that would be affordable to a large market. Now that these devices are becoming smaller, faster, cheaper and more reliable the market has opened up for some very new and interesting ideas.

The products currently out on the market similar to mine are still in the early stages with updates coming to them frequently. There are also some similar products in development that will reach the open market in the coming years. The list of possibilities with wireless locking applications is huge as this is a very new and open market. I feel these kind of products are the future with wireless locking systems becoming the norm for many people. As most people always carry a smart phone with them the convenience of not having to bring keys everywhere you go and the comfort of your door unlocking when you approach it will become an enticing appeal for many people.

Safety is an important issue with new IoT devices as vulnerabilities can be exploited from hackers. Developers need to insure control of these devices do not get into the wrong hands. Importantly data sharing and privacy becomes a big issue when using the Internet of Things. There are growing concerns when billions of devices are interconnected. Many businesses will be faced with storing the large amounts of information these devices will be producing. They will need to use methods of securely storing the data while still being able to access, analyze, and track the huge amounts of it being generated. The applications and devices need to be built from the ground up with security in mind as hackers are becoming smarter and more determined.

1.2 Aims

The scope of the project is to develop an android studio application that allows users to connect their android device to a smart lock. The lock could be used for the doors of your house, car, garage etc.

The application will allow you to lock and unlock the smart lock from anywhere in the world with your android device. The user’s android device will use Wi-Fi/4g as a connection means to connects to the users Smart Lock. Users can also use Bluetooth as a direct connection when available. The aim is to offer a more fixable and convenient way of accessing your belongings by eliminating the need to always carry a set of keys everywhere you go and reduces the chances of you arriving home and not being able to get access into your house because you lost your keys as long as you didn’t lose your phone as well. The application shall provide features allowing owners to grant or revoke access of their Smart Lock to other users. The functionality aims in providing users a ubiquitous presence to the application. Other features the application will include is timed locking and unlocking. Users have the ability to set a time they want their Smart Lock to automatically lock or unlock at. The application would also provide logs showing information on what time the door was locked or unlocked at and the current statue. Creating the application with security in mind is an important aspect I will focus on.

The intended customers are people who like using the latest and modern technology. It targets users who want more control and convenience of their home, as most people always carry a smart phone with them the convenience of not having to bring keys everywhere you go and the comfort of your door unlocking when you approach it will become an enticing appeal for many people. It can be used by people who are away from home but want to open it for cleaners or family members. The application is simple and easy to use, basic knowledge of how to use Wi-Fi, create an account and to browse options would be required.

1.3 Technologies

The program will be written in Java. It will be compatible with the majority of android devices that have version 6.0(Marshmallow) or higher and that have access to the google play store. The project application will be created in android studios, meaning a large amount of time & research will go into learning how to use and create software applications in it. Wi-Fi and Bluetooth will be used to transmit information to and from the android device. For the purpose of this project I am going to be connecting the android device to a Raspberry Pi as the Raspberry Pi will display the function of an operating Smart Lock. I have installed a Dexter Industries operating system onto the Raspberry Pi as this is needed to connect it to my laptop for it to function. There is a kit that comes with the Raspberry Pi called the Grove Pi that has many sensors and accessories that I will use in my project. My android application will connect to the Raspberry Pi and a connected light or electronic switch will represent the opening and closing of a locking system. The Raspberry Pi uses a basic python script that polls the AWS server to check what state the user wants it to be in and makes the switch. The android device will use Amazon web services (AWS) to send and receive information to and from the Raspberry Pi. I've opted to use a Django framework for the server to run on a AWS web instance. Im using a Django REST framework to control the lock. A SQLite3 database backend is used to keep track of the users, the locks associated with them allowing users to give lock permissions to a user and stores the user logs. I have created a EC2 instance in AWS and deployed the Django server on an Apache web server. The Apache web server lives on my AWS instance.

2 System

2.1 User Requirements Definition

Customers want a fast and easy to use app that gives them convenience and peace of mind over their own home. They also want the app to be safe, secure and to work on a variety of android operating systems. Appealing features are timed locking/unlocking, automatic locking/unlocking when coming into and out of range and functionality anywhere in the world.

2.1.1 Login

The system must allow a user to login with a username and password if they have previously set up a profile.

2.1.2 Create Profile

If the user does not have an existing profile, the system must allow the user to create a profile from the start up screen. This profile must accept a username and a password that is chosen by the user. Usernames must be unique and the system must not allow multiple profiles of the same username.

2.1.3 User Interface:

By using graphical design software, a GUI which reflects users flavor and leaves a good impression on them while using the App, has been created. The GUI is simple and user friendly.

2.1.4 Database:

Stores sets of data, each table contains an associated group of information such as user list, locks list, activities, user account information. An SQLite3 database is used that will live my AWS instance.

2.1.5 Retrieve Function:

Let’s user retrieve/reset his or her password/username by providing the associated e-mail.

2.1.6 Add new lock function:

Allows the user to add a new lock that they have bought to their account. They will need to do this in order to have access to the lock.

2.1.7 Open/close lock function:

Let’s the user open and close the lock from anywhere in the world when they are connected to the internet.

2.1.8 Grant Access

This allows owners to grant or revoke other users access to their Smart Lock. This feature can be changed at any time.

2.1.9 Automatic locking/unlocking function:

Allows the user set an option that opens the lock when the user comes into range.

2.1.10 Timed locking/unlocking function:

Allows the user set a time for the lock to open and close at. This can be a onetime function or left on repeat.

2.1.11 Activity’s:

The activity’s section shows all the latest operations of the smart lock. Users will be able to see the recent actions/logs of the lock such as what time & date the lock was opened or closed on and will show whose device used the lock.

2.1.12 Logout function:

Allows the user to logout of their account. Users may have more than one account they want to access.

3 Requirements Specification

3.1 Functional requirements

This section lists the functional requirements in ranked order, which specify all the fundamental actions of the mobile application.

3.1.1 Use Case Diagram

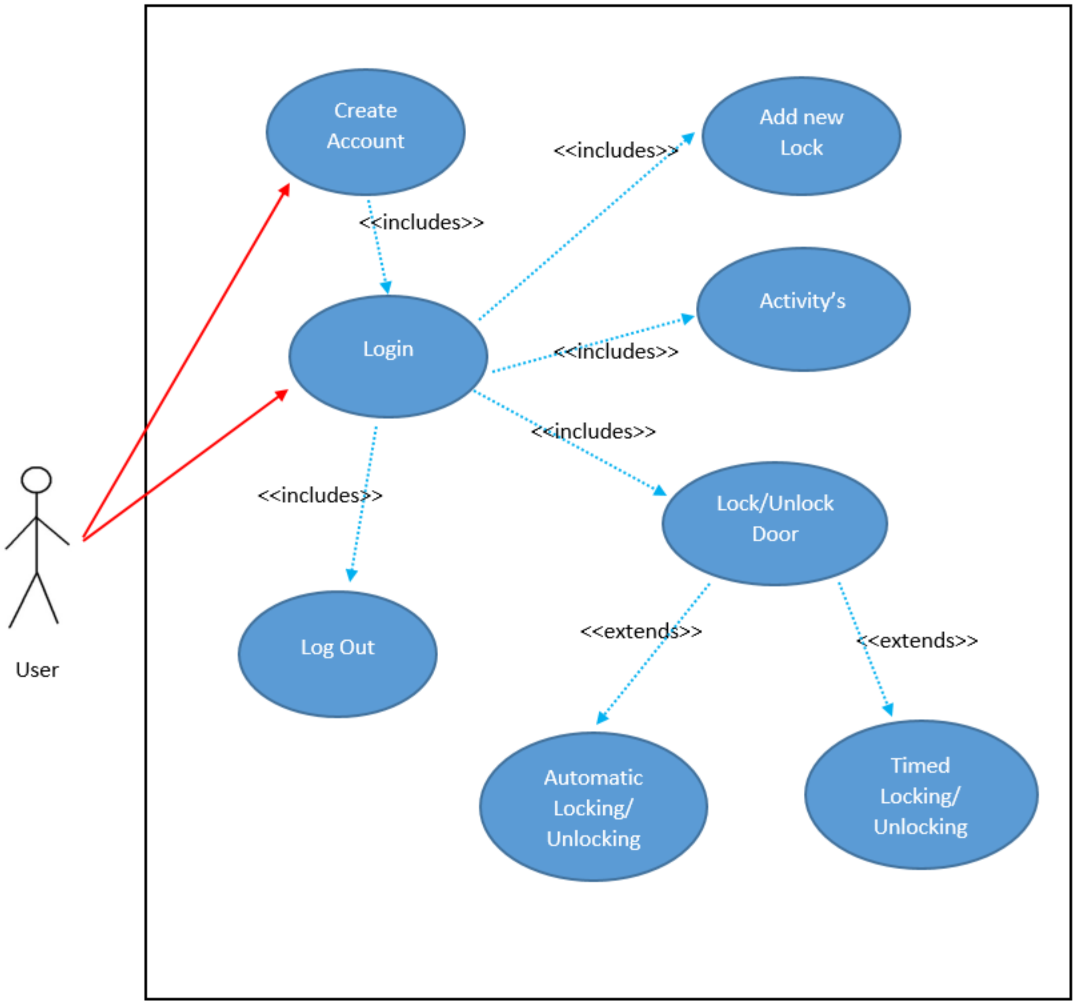


Figure 1 - This use case diagram depicts how the mobile application will interact with the users of the system and the overall behaviour of the system.

3.1.2 Requirement 1 <User Create Account>

3.1.2.1 Description & Priority

The user shall be able to register an account with the mobile application. The user will be asked to provide a username and a password. Users account will be saved to the system and it is essential in order to proceed with the mobile application.

3.1.2.2 Use Case

Use case ID

UC01

Scope

This use case describes how the user registers their unique account by pressing Create Account button.

Flow Description

Precondition

The mobile application is installed on an Android device.

Activation

This use case starts when a user clicks Create Account button.

Main flow

1. The user launches the mobile application.
2. The system displays a screen with a Login and Create Account button.
3. The user clicks Create Account button.
4. The system displays create account screen.
5. The system saves the users details.
6. The system displays the Login screen.

Alternate flow

A1: <No username or password entered>

1.The user does not enter a username or password to create an account.

2. The system displays a message to the user that a username and password must be entered.

3. The use case continues at position 4 of the main flow.

Exceptional flow

E1: <Account already exists>

1. The user enters credentials for an existing account.
2. The system displays a message to the user that the account already exists.
3. The system continues at position 2 of the main flow.

Termination

The system presents the screen that displays the Login button.

Post condition

On successful completion an account is created for the user and then the user will be prompted to Log In. If a failure occurs the user will be prompted to create an account again.

3.1.3 Requirement 2 <User Login>

3.1.3.1 Description & Priority

When a user has registered an account with the mobile application, the user will be able to Login to the mobile application. The user Logs in and their access is either granted or declined accordingly to their account existence.

3.1.3.2 Use Case

Use Case ID

UC02

Scope

The scope of this use case describes how the user logs in to their account by pressing the Login button.

Flow Description

Precondition

The mobile application is installed on an Android device.

The user has an existing account.

The user’s username and password exists on the system.

Activation

This use case starts when a user clicks the Login button.

Main flow

1. The user launches the mobile application.
2. The system displays a screen with a Login and Create Account button.
3. The user clicks the Login button.
4. The user enters their username and password.
5. The system checks the users credentials against the system and allows access.
6. The system displays the home screen.

Alternate flow

A1: <Username or Password is incorrect>

1. The user enters an incorrect username and password.
2. The system displays a message to the user that the account does not exist.
3. The use case continues at position 2 of the main flow.

Exceptional flow

E1: <User account does not exist on the system>

1. The system checks the users credentials and the account does not exist.
2. The system displays a message to the user that the account does not exists.
3. The use case continues at position 5 of the main flow.

Termination

The system presents the home screen.

Post condition

On successful completion the user is logged in and the home screen is displayed. If failure occurs the system displays a message and the user will be asked to enter a username and a password again.

3.1.4 Requirement 3 <Add New Lock>

3.1.4.1 Description & Priority

When the user has logged in they will see an option to add a new lock to their account. They will need to do this in order to have access to the lock.

3.1.4.2 Use Case

Use Case ID

UC03

Scope

The scope of this use case allows the user to add a new lock to their account by scanning for a lock over WIFI.

Flow Description

Precondition

The mobile application is installed on an Android device.

The user has an existing account.

The user’s username and password exists on the system.

Activation

This use case starts when a user clicks the Add New Lock button.

Main flow

1. The user launches the mobile application.
2. The user enters their username and password.
3. The system checks the users credentials against the system and allows access.
4. The system displays the home screen.
5. The user clicks Add New Lock button.
6. The system scans for lock over WIFI.
7. User enters password for new lock.
8. System checks password and adds lock to account.

Alternate flow

A1: <Password is incorrect>

1. The user enters an incorrect password.
2. The system displays a message to the user that the password is wrong.
3. The use case continues at position 7 of the main flow.

Exceptional flow

E1: <No lock found after scanning>

1. The system scans for lock but none found.
2. The system displays a message to the user that no lock was found.
3. The use case continues at position 5 of the main flow.

Termination

The system adds new lock to account.

Post condition

The system adds new lock to the account and tells the user a new lock was added. If failure occurs the system displays a message and the user will be asked to scan for lock again.

3.1.5 Requirement 4 <Open/close Lock>

3.1.5.1 Description & Priority

When the user has added a new lock to the account they will be able to open/close the lock from any in the world once they are connected to the internet or if they are in range of the lock. They will be able to set an option for automatic unlocking of the door when they come into range. They can also set the lock to open or close at a certain time.

3.1.5.2 Use Case

Use Case ID

UC04

Scope

The scope of this use case describes how the user opens/closes the lock and changes options for automatic and timed locking/unlocking.

Flow Description

Precondition

The mobile application is installed on an Android device.

The user has an existing account.

The user’s username and password exists on the system.

The user has added a lock to their account.

Activation

This use case starts when a user clicks the Lock/unlock button.

Main flow

1. The user launches the mobile application.
2. The user logins in.
3. The system displays the home screen.
4. The user clicks the lock/unlock button.
5. The system displays the lock/unlock screen.
6. The user choses to open or close the lock.
7. The system opens or closes the lock for the user.

Alternate flow

A1: <User sets automatic locking/unlocking option>

1. The user clicks the lock/unlock button.
2. The system displays the lock/unlock screen.
3. The user changes the option for automatic locking/unlocking.
4. The system saves the new setting.

Alternate flow

A2: <User sets timed locking/unlocking>

1. The user clicks the Lock/unlock button.
2. The system displays the lock/unlock screen.
3. The user sets the option for timed locking/unlocking and sets a time.
4. The system saves the new setting.

Exceptional flow

E1: <No lock saved in system>

1. The system has no locks saved in the account.

2. The system displays a message to the user that there are no locks saved to the account.
3. The use case continues at position 3 of the main flow.

Termination

The system opens/closes the lock.

Post condition

On successful completion the system will have open or closed the lock and will display the locks current status. If failure occurs the system displays a message and the user will be asked to try again.

3.1.6 Requirement 5 <Activity’s>

3.1.6.1 Description & Priority

Users can access the activity’s menu when they have logged into the app. The activity’s section shows all the latest operations of the smart lock. Users will be able to see the recent actions of the lock such as what time & date the lock was opened or closed on and will show whose device used the lock.

3.1.6.2 Use Case

Use Case ID

UC05

Scope

The scope of this use case describes how the user can view the recent activities of the smart lock.

Flow Description

Precondition

The mobile application is installed on an Android device.

The user has an existing account.

The user’s username and password exists on the system.

The user has added a lock to their account.

Activation

This use case starts when a user clicks the activities button.

Main flow

1. The user launches the mobile application.
2. The user logins in.
3. The system displays the home screen.
4. The user clicks the activity’s button.
5. The system displays the recent activities of the smart lock.

Alternate flow

A1: <No saved activities>

1. The user clicks the activity’s button.
2. The system displays a message telling the user there are no saved activities.

Exceptional flow

E1: <No lock saved in system>

1. The system has no locks saved in the account.
2. The system displays a message to the user that there are no locks saved to the account.
3. The use case continues at position 3 of the main flow.

Termination

The system displays the recent activities of the smart lock.

Post condition

On successful completion the system will be displaying the recent activities of the smart lock. If failure occurs the system displays a message and the user will be asked to try again.

3.1.7 Requirement 6 <User Logout>

3.1.7.1 Description & Priority

When a user has registered an account with the mobile application and has logged in, the user will be able to log out of the mobile application. The users Login details will be saved on the system for the future access.

3.1.7.2 Use Case

Use Case ID

UC06

Scope

The scope of this use case is to allow the user to exit their account.

Flow Description

Precondition

The mobile application is installed on an Android device.

The user has an existing account.

The user’s username and password exists on the system.

Activation

This use case starts when a user clicks the Login button.

Main flow

1. The user launches the mobile application.
2. The user enters their username and password.
3. The system checks the users credentials and allows access.
4. The system displays the home screen.
5. The user decides to log out.
6. The system logs the user out of their account.

Alternate flow

A1: <None>

Exceptional flow

E1: <None>

Termination

The system logs the user out of their individual account and presents the start screen and allows the user to log in.

Post condition

On successful completion the user is logged out and the start screen is displayed.

3.2 Non-Functional Requirements

3.2.1 Performance/Response time requirement

The mobile application should be fast and responsive. The application should switch between display screens instantly. The locking/unlocking function should take no more than 3 seconds.

3.2.2 Availability requirement

The application should be available 24/7 so that the users can access all features of the application whenever they want. Unplanned downtime should be as low as a few seconds a year to avoid disruption.

3.2.3 Scalability requirement

The application can be expanded to support increasing number of users without reducing the applications performance.

3.2.4 Recover requirement

Users will be able to recover their account if they have forgotten their username or password.

3.2.5 Robustness requirement

The application should have robustness requirements so it does not crash at the slightest disturbance. This will allow the app to be resistant against things such as an unexpected flood of requests from an intentional attack.

3.2.6 Security requirement

Security requirements must be put in place in case of accidental loss and corruption, or from a deliberate unauthorized attempt to access or alter the data. The mobile application must interact with the SQLite database in order for the user to log in and to be identified by the application. The information stored in the database must be kept secure.

Only one session should exist. This is a secure session that is only open to users that have created an account with the mobile application.

3.2.7 Reliability requirement

Unplanned downtime should be as low as a few seconds a year to avoid disruption. Data backups will also be used to insure there is no loss of personal data.

3.2.8 Maintainability requirement

The mobile application should be easy to extend. The software will use good design principles that will allow easy system maintenance or modifying it for new functions.

3.2.9 Portability requirement

The mobile application must be available to use on 100% of android devices and should be portable with Android OS 2.2 Froyo and higher.

3.2.10 Extendibility requirement

In the future the application should be supported on iOS and windows phones.

3.2.11 Reusability requirement

The application shall use many of the resources common among Android applications such as zoom etc.

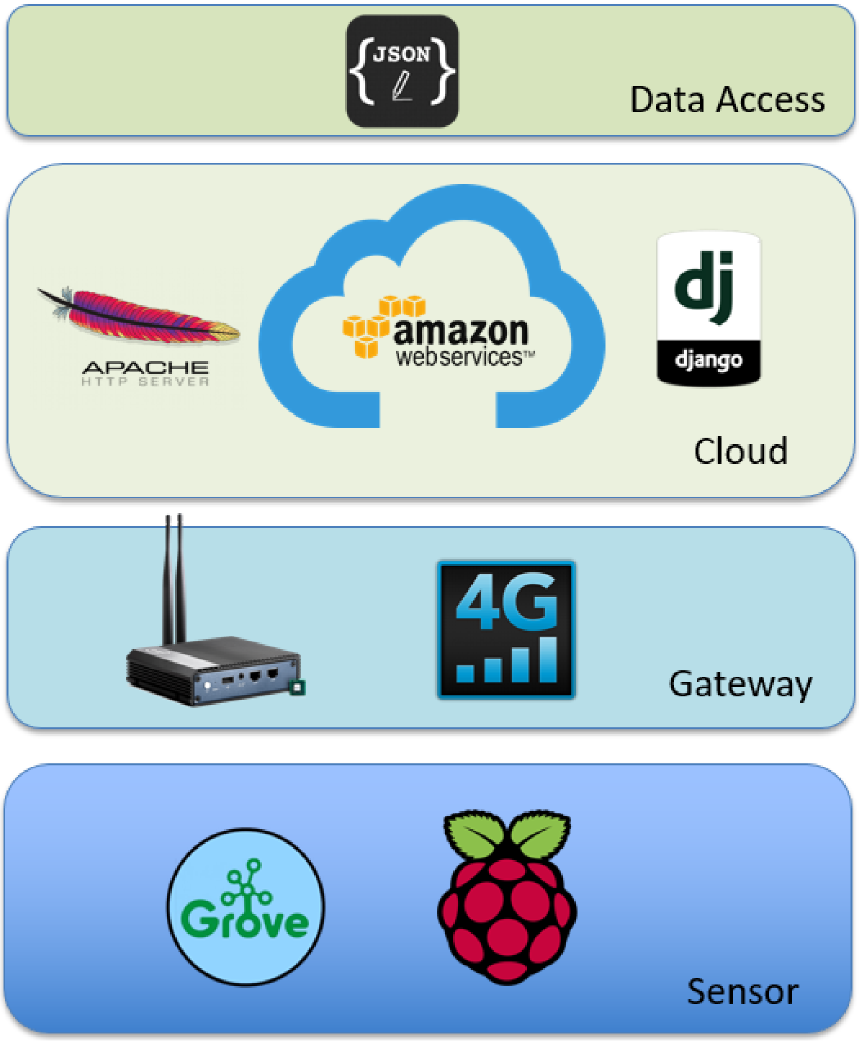
3.3 Usability requirements

- Ease of learning: The system must be easy to learn for both novices and users with experience from similar systems.
- Task efficiency: The system must be efficient for the frequent user.
- Ease of remembering: The system must be easy to remember for the casual user.
- Understand ability: The user must understand what the system does.
- Subjective satisfaction: The user must feel satisfied with the system.

3.4 Design and Architecture

This system architecture is created such as, when a person logs into their account it will load they're custom settings. Also, all saved locks, timed locking/unlocking options, automatic locking/unlocking and activity information will be loaded. The following diagram represents the high-level architecture of the system.

The user logs in with their username and password that gets authenticated by the server. Application contacts server to check statuses of locks associated with the user. When user executes an action the application connects to Amazon web services (AWS) to send the request using JSON data. An Apache web server will run on the AWS web instance. It listens for incoming messages and saves the information and makes the appropriate response. The Raspberry Pi uses a python script that polls the server every one second to check what state the user wants the lock to be in and makes the appropriate switch if necessary. The Raspberry Pi uses the Grove Pi starter kit for connecting sensors.


JSON
Data Access
APACHE
HTTP SERVER
amazon
webservices™
dj
django
Cloud
4G
Gateway
Grove
Sensor

3.5 Implementation

When a user wishes to unlock/lock their smart lock the code used to make this request is located in the AdapterLockUnlock.java class. The class checks the position of the LockId set by the user in order to make the correct request as seen in the following code.

```
@Override
public void onCheckedChanged(CompoundButton compoundButton, boolean checked)
{
   if(checked)
   {
      Log.d(TAG, "Checked, id = " + arrayListLock.get(position).getLockId());
      arrayListLock.get(position).setLocked(true);
      makePostRequest("lock", arrayListLock.get(position).getLockId());
   }

   else
   {
      Log.d(TAG, "Unchecked, id = " + arrayListLock.get(position).getLockId());
      arrayListLock.get(position).setLocked(false);
      makePostRequest("unlock", arrayListLock.get(position).getLockId());
```

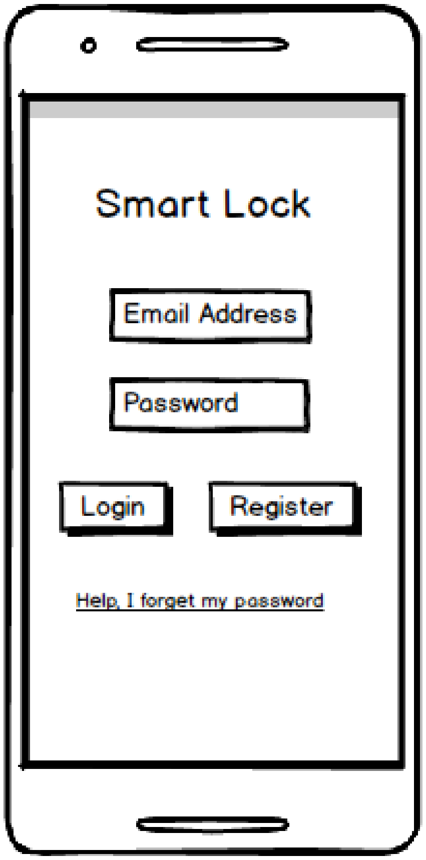
The volley library is used to manage the processing and cashing of network requests that allows the application to make POST requests to the server. AWS is used to send the POST requests and it does this in the form of JSON data. The time, date, lock id and user id is also sent with this request.

```
public void makePostRequest(final String command, final int lock_id)
{
      final String URL = activityLockUnlock.getResources().getString(R.string.SERVER_ADRESS) +
"/lock/";

      StringRequest req = new StringRequest(Request.Method.POST, URL, new
Response.Listener<String>()
      {
         @Override
         public void onResponse(String response)
         {
            VolleyLog.v("Response:%n %s", response);
         }
```

Users can view the resent activity of their lock within the application. This function is provided by the ActivityUserLog.java class. When users select this function a request is sent to the server for this data. The server then responds by sending back the requested data. The time and date of any lock actions is displayed to the user as well as the lock id and who requested this action. The server can save thousands of activity’s for its users. The following code displays the GET request to the server.

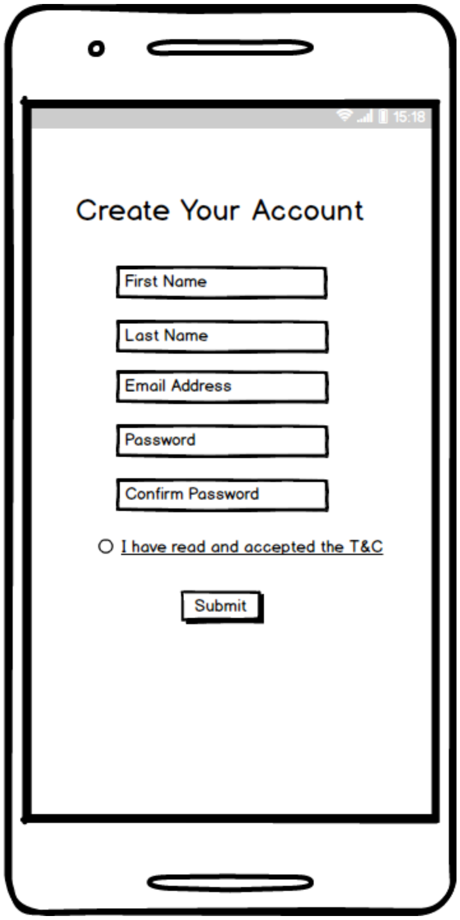
```
public void makeVolleyRequest(final int user_id)
{
    final String URL = “http://ec2-31-248-19-62.eu-west-1.compute.amazonaws.com”
 + "/lock?user_id=" + Integer.toString(user_id);

    StringRequest req = new StringRequest(Request.Method.GET, URL, new Response.Listener<String>()
    {
        @Override
        public void onResponse(String response)
        {
            VolleyLog.v("Response:%n %s", response);
```

3.6 Graphical User Interface (GUI) Layout



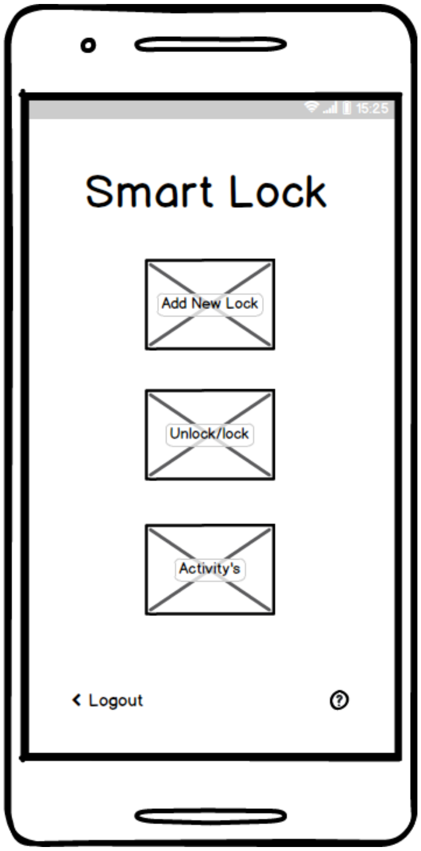
Upon opening the application you'll be greeted with an opening page, which looks along the lines of the mock up seen here. The user enter their login details to enter the app or click "Help, I forgot my password" to reset it.

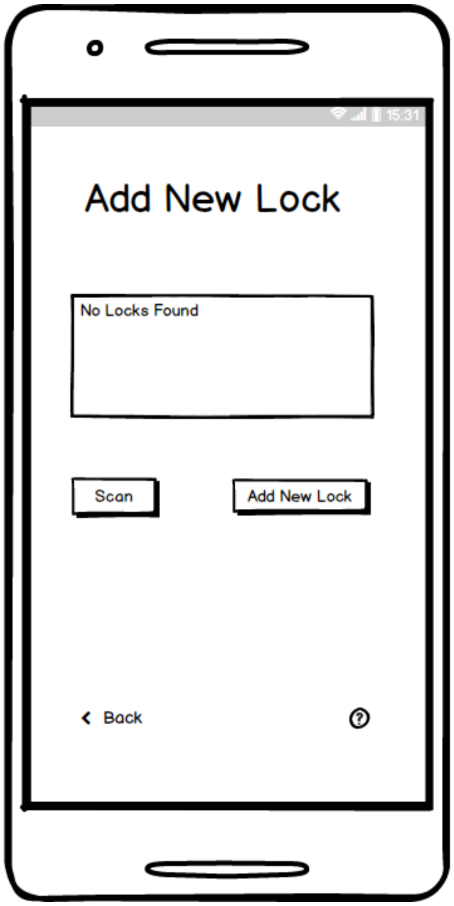
The user can also choose to register an account here.



Here the user can create an account for the application. The user can click on the hyperlink for the T&C that will bring an external page with these details. After they have entered their details and accepted the T&C they can hit submit and an account will be created. The details will be saved to a SQLite database.

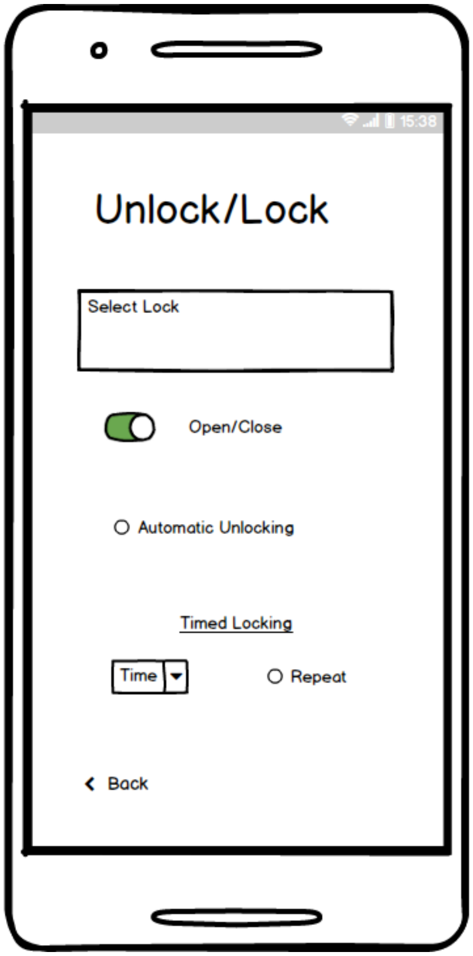
After the user has logged in they will be taken to the main menu. There are three main options they can choose from, add new lock, unlock/unlock and activity's which will be displayed as images. They also have the option to logout and click a help button which will explain information on what is displayed to them.

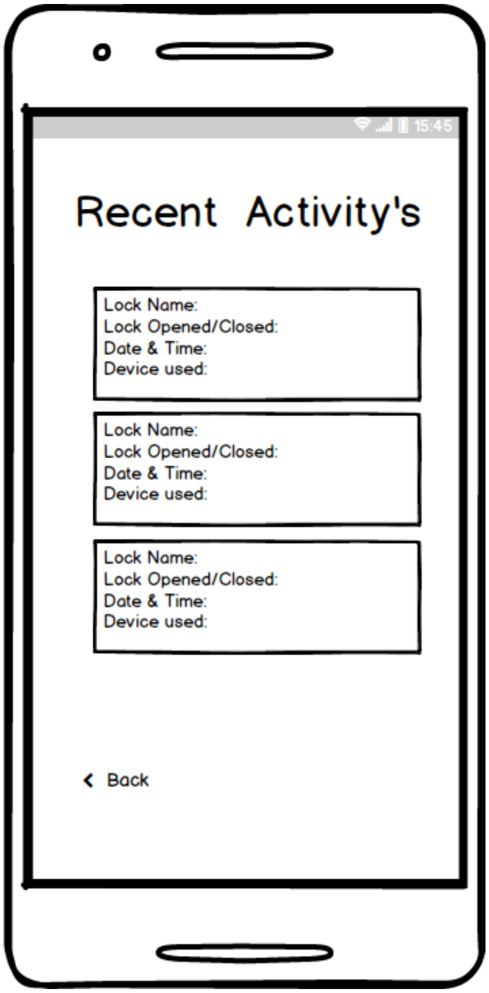




Here the user adds a new lock to their application. They hit scan and the phone uses WIFI to find the new lock. They select the lock and click the add new lock button. Than the user will be prompted with a dialog box to enter the locks password. This is necessary as you don't want anyone to be able to access your lock without your permission. The new lock will now be saved to the database.

Here the user is able to unlock/lock the new lock they have added. They select the lock and use the slider to open or close it. They also have options to turn on automatic locking and use timed locking.





This page shows the recent activity of the users saved locks. It shows the lock name, date & time it was last used on and the device that initiated the operation. This information is all saved in the SQLite database.

3.7 Testing

As the app is going through development I will use android studios to continuously test newly created features in the application. Testing will be done on a regular basis as to ensure low amount of errors during final testing. Android Studio is designed to make testing simple. With just a few clicks, you can set up a JUnit test that runs on the local JVM (Java Virtual Machine) or an instrumented test that runs on a device. I will be testing and running the app on the Raspberry Pi throughout the year.

3.8 Customer testing

For customer testing I will try to make the most of opportunities to show the application to friends and family. Allowing them to use the application and asking them to give constructive feedback will be very useful to me. I will also use the supervisor meetings as an opportunity to show them the product and expand on advice giving at those meetings. I will try to get user feedback from various people on ease of use, simplicity, functionality and design.

3.9 Evaluation

Performing system evaluation is an ongoing process throughout the performance testing effort, it provides more value when conducted early in the test project. The intent of system evaluation is to gather information about the project as a whole, learn the functions of the system, expected user activities, system architecture, and many other details that are helpful in guiding performance testing to achieve the needs of the project. The information provides a foundation for collecting the performance goals and requirements, creating performance-testing strategies and plans, characterizing the workload, and assessing project and system risks.

A great understanding of the system under testing is critical to a successful performance testing effort. The measurements received during the later stages are only as accurate as the models that have been developed and validated in this stage. The evaluation provides an important foundation for determining acceptable performance, specifying the performance requirements of the software, system, components and identifying any risks to the effort before testing will even begin.

When evaluating the system I focused on three main areas. How efficient is the system? Does it operate quickly to user requests? Users will become impatient having to wait for more than a few seconds when changing pages or making actions. Does the system operate smoothly and with minimal waste? An android application must not to powerful and taxing on system hardware.

Is the application easy to use? When creating a product for a wide audience users must be able to use the system easily and effectively. New users must be able to understand and use the system with ease and without training. If users are struggling to operate the application, they will quickly become frustrated and stop using the product. If your product is very complex to use, knowledge of this spreads through word of mouth and bad reviews. Product competitors will quickly take your place and without fast improvements your product can become irrelevant.

How appropriate is the application? The system must be suitable for the market that its sold to. The system must actually meet the needs for the user. If users find the product not to be that useful or it is missing many useful features your product will not succeed.

The system has been checked against the requirements specification. The system must have met all the requirements from the requirements specification document. I ensured that all requirements had been met. I tried to get as much user feedback as possible showing the application to whoever was interested in helping. I had users use the application and get a feel for the product. I asked for as much feedback as possible gathering impressions on whether they liked it, if it made things easier for them and what could be improved. I revived mostly positive feedback from users taking on board any negative aspects or adjustments people wanted to see. Users felt it was an interesting and unique product that if purchased would make life easier. They also noted it was easy to use. I also some spent time using the application ensuring that it was fast/efficient when making requests and navigating through pages.

4 Conclusions

Some advantages of having an application that allows you to lock and unlock your smart lock from anywhere in the world with your android device, means you don't have to carry a set of keys all the time and eliminates the chances of being locked outside and unable to get into your house because you lost your keys. Giving other users access to use the smart lock means you don't have to physically share keys or take them back again. Users won't have to get multiple keys cut for one lock they use. It allows users to plan their days more freely, not having to worry about rushing back home to let people in such as family/friends, plumbers, cleaners, dog walkers etc.

It would also be safer then keys because if keys are mislaid anyone can use them but your mobile phone cannot be accessed as it is password protected. Your mobile phone can also be more easily recovered than keys if lost. With the timed locking feature locks can set to auto lock at certain times, this can be used as a backup safety function. Users will also be able to know who opened their door and when in the activity's screen.

A disadvantage that this project has would be if your phone ran out of charge the application cannot be used. The smart lock would have to be built with key hole and provide a spare key as a backup if this happened. There would also be an online website where users can login and access the features from the app.

The opportunities and limits for this project are huge as this application could be expanded to any type of lock for instance your car, garage, a safe or even your laptop if you have forgotten the password.

5 Further development or research

- Over time the application could evolve with even more features and functionality options for the users. The options are endless but only best, most useful and creative have been chosen for the Smart Lock app.

- Adding notification features such as when the lock has been opened or closed the user receives a notification that the door has been opened. Another option would be if the lock had vibration sensors it could recognize if there was a knock on the door and would send the user a notification.

- Adding a feature that lets a user give temporary access to another user for a small period of time would be another appealing feature that could be implemented in the future.

- NFC technology is becoming more and more popular so allowing the user to unlock their door via NFC by holding their smart phone up to the lock would be another feature that could be implemented.

- Another feature could be providing app in different languages. This would provide the customer group that is using this application to grow. People who struggle with English could have the application in a language they know and understand better providing food chain business more customers.

- Over time this application also could have a website having all the same information that the app provides. To make it even more accessible for the users, also an IOS app and Windows Phone app could be implemented.

- All these features and functions are ideas of how to make this app more efficient, much easier to use for any customer and making it the best in the market.

6 References

Wolpin, S. and Techlicious, L. (2016) 5 best smart locks for your home. Available at: http://www.techlicious.com/guide/5-futuristic-smart-locks-for-your-home/ (Accessed: 15 November 2016).

Unlock bolt from anywhere. Smart, Available at: https://lockitron.com/ (Accessed: 11 August 2016).

Access, G. (no date) 'GOJI Blog', Available at: http://blog.gojiaccess.com/ (Accessed: 2 December 2016).

Brands, S. (2016) Kevo smart lock - control your lock with your phone. Available at: http://www.kwikset.com/kevo/default (Accessed: 23 November 2016).

Available at: http://www.it-c.dk/people/slauesen/Papers/SixStyles.pdf (Accessed: 11 January 2017).

Create an EC2 Instance and Install a Web Server - Amazon Relational Database Service. [ONLINE] Available at:

http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateWebServer.html. [Accessed 04 Feb 2017].

IGCSE ICT - Evaluating the New System | IGCSE ICT. 2017. IGCSE ICT - Evaluating the New System | IGCSE ICT. [ONLINE] Available at: http://www.igcseict.info/theory/8/eval/. [Accessed 09 Feb 2017].

Tom Christie. 2017. Home - Django REST framework. [ONLINE] Available at: http://www.django-rest-framework.org/. [Accessed 16 March 2017].

Amazon Web Services, Inc.. 2017. Cloud Solutions - Amazon Web Services (AWS). [ONLINE] Available at: https://aws.amazon.com/solutions/?nc2=h_ql_ny_livestream_blu. [Accessed 17 March 2017].

Corticon. 2017. Corticon. [ONLINE] Available at: http://documentation.progress.com/output/ua/Corticon/index.html#page/corticon/sample-json-request-and-response-messages.html. [Accessed 09 April 2017].

Raspberry Pi. 2017. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi. [ONLINE] Available at: https://www.raspberrypi.org/. [Accessed 25 November 2016].

Writing your Django app, part 1 | Django documentation | Django. 2017. Writing your first Django app, part 1 | Django documentation | Django. [ONLINE] Available at: https://docs.djangoproject.com/en/1.11/intro/tutorial01/. [Accessed 11 January 2017].

Dexter Industries. 2017. GrovePi Internet of Things Robot Kit. [ONLINE] Available at: https://www.dexterindustries.com/grovepi/. [Accessed 13 January 2017].

August. 2017. August Smart Lock | August. [ONLINE] Available at: http://august.com/products/august-smart-lock/. [Accessed 20 January 2017].

7 Appendix

7.1 Project Proposal

Objectives

My objectives are to build an android studio application that allows you to connect your android device to a smart lock. The lock could be used for the doors of your house, car etc. but for the purpose of this project I will be using a Raspberry Pi to showcase the functions of the smart lock.

The application will allow you to lock and unlock the lock from anywhere in the world with your android device. It will use your devices Wi-Fi/4g to connect to your home Wi-Fi system which will then connect to the smart lock or alternatively it can directly connect to the smart lock. This means you don't have to carry around a set of keys all the time and eliminates the chances of you arriving home and not being able to get into your house because you lost your keys as long as you didn't lose your phone as well. You can also open the lock from anywhere for kids or relatives if they don't have keys or access to the house. Other features the application may include are timed locking and unlocking. This would be useful if you want the locks in your house to automatically lock late in the evening. This is a useful safety feature if you forget to lock the house before you go to bed. The application would also be able to tell what time the door was locked or unlocked. This would allow you to know what time your kids actually came home at.

Background

IoT is a very new and up and coming technology with more and more products and devices being developed each year. In the past the equipment was slow and expensive which made it hard for companies to make a successful product that would be affordable to a large market. Now that these devices are becoming smaller, faster, cheaper and more reliable the market has opened up for some very new and interesting ideas.

The products currently out on the market similar to mine are still in the early stages with updates coming to them frequently. There are also similar products in development that will reach the open market in the coming years. The list of possibilities with wireless locking applications is huge as this is a very new and open market. I feel these kind of products are the future with wireless locking systems becoming the norm for many people. As most people always carry a smart phone with them the convenience of not having to bring keys everywhere you go and the comfort of your door unlocking when you approach it will become an enticing appeal for many people.

Safety is an important issue with new IoT devices as vulnerabilities can be exploited from hackers. Developers need to insure control of these devices do not get into the wrong hands. The applications and devices are being built from the ground up with security in mind as hackers are becoming smarter and more determined.

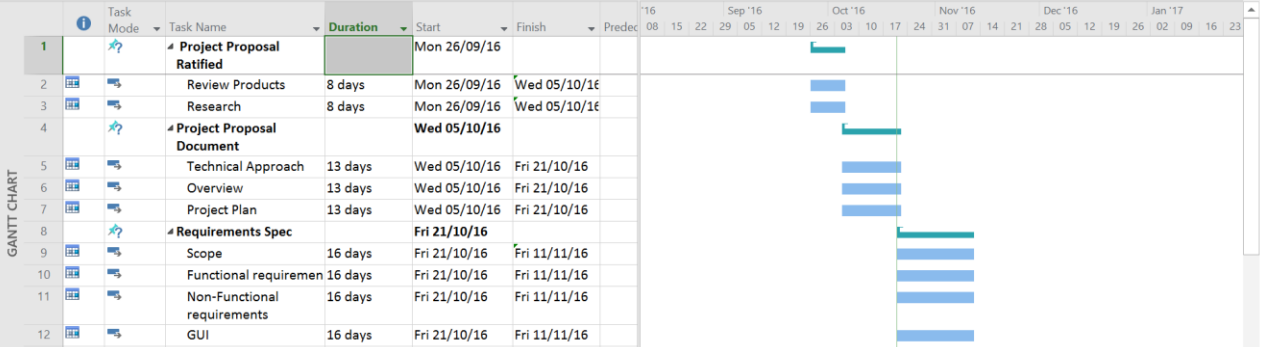
Technical Approach

The project application will be created in android studios so a lot of research will go into learning how to use and create software applications in it. I will research how I can connect android devices over Wi-Fi and Bluetooth. I am going to be connecting the device to a Raspberry Pi. First I have to research how the Raspberry Pi will operate. I must install an operating system onto it and connect it to my laptop for it to function. There is a kit that comes with the Raspberry Pi called the Grove Pi that has many sensors and accessories that I will use in my project. My android application will connect to the Raspberry Pi and a connected light or switch will represent the opening and closing of a locking system.

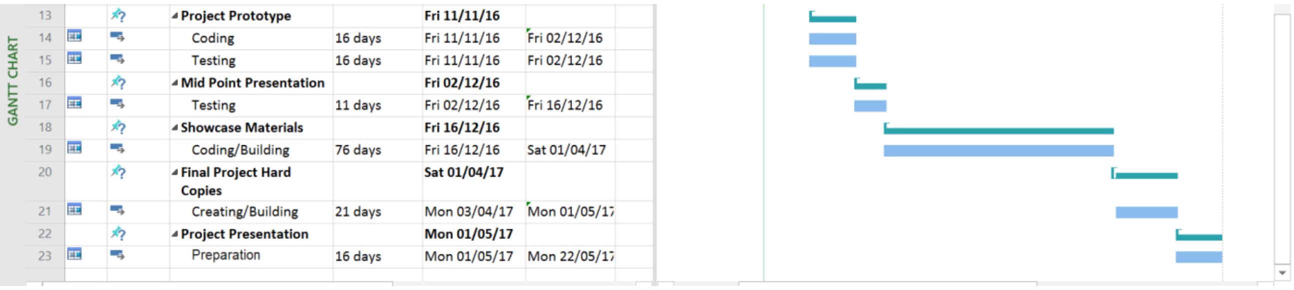
Special resources required

Raspberry Pi, Grove Pi, Android device

Project Plan



		Task Mode	Task Name	Duration	Start	Finish	Predec
1			**Project Proposal Ratified**		Mon 26/09/16		
2			Review Products	8 days	Mon 26/09/16	Wed 05/10/16	
3			Research	8 days	Mon 26/09/16	Wed 05/10/16	
4			**Project Proposal Document**		**Wed 05/10/16**		
5			Technical Approach	13 days	Wed 05/10/16	Fri 21/10/16	
6			Overview	13 days	Wed 05/10/16	Fri 21/10/16	
7			Project Plan	13 days	Wed 05/10/16	Fri 21/10/16	
8			**Requirements Spec**		**Fri 21/10/16**		
9			Scope	16 days	Fri 21/10/16	Fri 11/11/16	
10			Functional requiremen	16 days	Fri 21/10/16	Fri 11/11/16	
11			Non-Functional requirements	16 days	Fri 21/10/16	Fri 11/11/16	
12			GUI	16 days	Fri 21/10/16	Fri 11/11/16	



13	**Project Prototype**		**Fri 11/11/16**	
14	Coding	16 days	Fri 11/11/16	Fri 02/12/16
15	Testing	16 days	Fri 11/11/16	Fri 02/12/16
16	**Mid Point Presentation**		**Fri 02/12/16**	
17	Testing	11 days	Fri 02/12/16	Fri 16/12/16
18	**Showcase Materials**		**Fri 16/12/16**	
19	Coding/Building	76 days	Fri 16/12/16	Sat 01/04/17
20	**Final Project Hard Copies**		**Sat 01/04/17**	
21	Creating/Building	21 days	Mon 03/04/17	Mon 01/05/17
22	**Project Presentation**		**Mon 01/05/17**	
23	Preparation	16 days	Mon 01/05/17	Mon 22/05/17

Technical Details

The application will be created in the latest version of android studios. The program will be writing using Java. It will be compatible with any android device that has version 6.0(Marshmallow) or higher and that has access to the play store.

Evaluation

I will be constantly running my application as I build it with a virtual device available in android studios. I will be testing and running the app on the Raspberry Pi throughout the year. I will also get user feedback from various people on ease of use, simplicity, functionality and design.

____Steffan Schmieder__21/10/2016______

Student Name	Steffan Schmieder
Student Number	X13757775
Course	BSC (Honours) in Computing
Project Title	Smart Lock
Overview of Project I am creating an application that will allow you to lock and unlock the lock from anywhere in the world with your android device. It will use your devices Wi-Fi/4g to connect to your home Wi-Fi system which will then connect to the smart lock or alternatively it can directly connect to the smart lock. This means you don't have to carry around a set of keys all the time and eliminates the chances of you arriving home and not being able to get into your house because you lost your keys as long as you didn't lose your phone as well. You can also open the lock from anywhere for kids or relatives if they don't have keys or access to the house. Other features the application may include are timed locking and unlocking. This would be useful if you want the locks in your house to automatically lock late in the evening. This is a useful safety feature if you forget to lock the house before you go to bed. The application would also be able to tell what time the door was locked or unlocked.	

Meeting 1

Date	26/10/2016
Time	2pm
Duration of Meeting	20mins
Current Challenges Understanding suitable size for project. Working on Prototype	
Goals of Meeting Review software requirements. Better understand project size and workload.	
Goals/Actions for next Meeting Review current progress Expand on project ideas and features	
Learning Agreement **Student Signature**	Steffan Schmieder

Meeting 2

Date	8/11/2016
Time	2pm
Duration of Meeting	30mins
Current Challenges Building on current ideas and project features Understanding full scope of the project Building on requirements specifications document Finishing aims and objectives for the project	
Goals/Actions for next Meeting Reviewing current progress Looking over requirements specifications document Finalising any needed changes Reviewing mock-up GUIs	
Learning Agreement **Student Signature**	Steffan Schmieder

Meeting 3

Date	15/11/2016
Time	2pm
Duration of Meeting	20mins
Current Challenges Building on midpoint technical report Working on functional and non-functional requirements Building on project code Building mock-up GUIs	
Goals/Actions for next Meeting Reviewing current progress Looking at created mock-up GUIs	
Learning Agreement **Student Signature**	Steffan Schmieder

Meeting 4

Date	22/11/2016
Time	2pm
Duration of Meeting	20mins
Current Challenges Creating application code and features	
Goals/Actions for next Meeting Reviewing midpoint technical report Preparing for midpoint presentation	
Learning Agreement **Student Signature**	Steffan Schmieder

Meeting 5

Date	29/11/2016
Time	2pm
Duration of Meeting	20mins
Current Challenges Preparing for mid-point presentation Creating PowerPoint slides	
Goals/Actions for next Meeting Talking over feedback from mid-point presentation	
Learning Agreement **Student Signature**	Steffan Schmieder

Meeting 6

Date	6/11/2016
Time	2pm
Duration of Meeting	15mins
Current Challenges Taking on board all feedback from mid-point presentation and implementing them into the project	
Goals/Actions for next Meeting Discussing overall features and application aspects Discussing technologies used in the project Look over final technical report	
Learning Agreement **Student Signature**	Steffan Schmieder

Meeting 7

Date	8/3/2017
Time	2pm
Duration of Meeting	20mins
Current Challenges Building on project code and communication functions Building application layout and GUIs Continue to work on final technical report Creating my showcase profile	
Goals/Actions for next Meeting Reading and reviewing technical report Discussing overall project completion	
Learning Agreement **Student Signature**	Steffan Schmieder

Meeting 8

Date	19/4/2017
Time	2pm
Duration of Meeting	20mins
Current Challenges Finishing project technical report Receiving as much feedback on project as possible Adding server functions and communication coding Continually building on current progress Creating project poster	
Learning Agreement **Student Signature**	Steffan Schmieder

7.2 Monthly Journals

Reflective Journal

Student name: Steffan Schmieder

Programme: BSc (Honours) in Computing IoT

Month: September

My Achievements

This month, I finalised some of the different ideas and aspects I want to use for my project. I researched similar products out in the market and products many that were still in development. I spent a considerable amount of time on research to ensure my application would be unique and have advantages to other products. I took the best aspects from many different products as well as creating my own concepts to use for my application. I got familiar with some of the different features I will be implementing in my project and prepared for creating my project proposal.

My Reflection

I found it very helpful when researching other products that have been developed. The products websites had many interesting features that helped me come up with my own unique ideas of a product. Many products had YouTube videos explaining how their product functions, its uses, intended customers and much more. I will go back to some of these videos when building my application. I also found it helpful by asking other students on what ideas they had for their own projects.

Intended

Next month, I will start and complete my project proposal document. I must present this proposal to lectures who will review it and ultimately decide if it is suitable to create. I want to be will prepared for this presentation as I feel it is important to get approval to start the project as soon as possible. This will give me time to continue researching and formulating new ideas as oppose to spent my time creating a new project proposal. I will do more extensive research into how complex and large I want to create my application.

I will also try and make a start on the requirements spec document.

Reflective Journal

Student name: Steffan Schmieder

Programme: BSc (Honours) in Computing IoT

Month: October

My Achievements

This month I completed my project proposal document and uploaded it to Moodle. I outlined my objectives, goals for the project and explained my learning outcomes. I have a good understanding on how big the project needs to be so I can complete a large/medium sized project but that I also don't overload myself with too much work and too many features.

I have connected my Raspberry Pi to my laptop and have been working on some tutorials on how it functions. I have connected my Grove Pi kit to the Raspberry Pi and have completed some basic commands to get a LED flashing and a number of sensors working. I need to have a good understanding on how the Grove Pi will work if I want to connect my android application to it. Connecting the Raspberry Pi to the internet was more challenging than I thought but I finally got it working with a Wi-Fi dongle as this will be necessary for my project.

I have created a Gantt chart in Microsoft Project outlining my tasks and uploaded dates.

I have also made a start on the requirements spec document that is due on Friday 11th of November. I will have that finished over reading week and add in more details if anything comes up.

My Reflection

I felted it was very helpful to talk to my fellow classmates on what their project was about. It gave me a good understanding on the scale of their project sizes and features there were going to use. Hearing the details in their projects gave me ideas for different areas or parts of my project and how to improve them. I was able to complete my Gantt chart easily/successfully by hearing what the problems others had and what software they used.

Reflective Journal

Student name: Steffan Schmieder

Programme: BSc (Honours) in Computing IoT

Month: November

My Achievements

This month I completed the initial technical report document and uploaded it to Moodle. I outlined the requirements, future plans for the project and explained my learning outcomes. I spent a lot of time researching many different operations and functions that I can implement into my project. I have finished a full GUI mock-up for the project and have now started looking at the GUI for the main project. The GUI for the main project with a small amount of functionality has to be presented in the Mid-point presentation which is worth 25%. I have also made as start on PowerPoint slides that are needed for the presentation. Much of my time was spent on the technical report as this is a large document with a lot of research needed to complete it to a high standard.

Intended

Next month, I will continue to work on the main project as this will require a lot of time and research to finish completely. The amount of work may not be as much as I would like to complete as I will have to spend most of my time studying for exams in January.

My Reflection

I am glad I completed the requirements specification document to a high standard as a lot of the details and aspects of the technical report had already been completed in the requirements specification document. This helped me with the workload for this part of the project as many of the details just had to be transferred over to the other doc.

Reflective Journal

Student name: Steffan Schmieder

Programme: BSc (Honours) in Computing IoT

Month: December

My Achievements

This month I added more details to my technical report document. I outlined the details on the server I am using in my application. I have researched the different ways I can link my app to the server. One of the ways that I am considering is by connecting it over SSH. I want to insure the application is using at secure connection at all times as this is very important when using an application that has access to the locks of your home. I am also looking at using a REST framework for my application. I have decided that I will be using Amazon Web Services (AWS) with my application. It is a very powerful system with many features that can serve my needs for this project.

Intended

Next month, I will continue to work on the main project as this will require a lot of time and research to finish completely. I will continue to improve the GUI of the application. I want to start working with AWS as this is a complex system that can easily leave a user confused. I want to learn how to use it at an early stage as I could easily run into problems further down the line. It is important to know where I need help as leaving problems to last minute is best avoided.

Reflective Journal

Student name: Steffan Schmieder

Programme: BSc (Honours) in Computing IoT

Month: January

My Achievements

This month I have been continuing to research the sever integration for my application. I have looked at the different options. I am still learning the different aspects and how to use them in AWS. I have been learning what an AWS EC2 instance is and how to set one up. Amazon EC2 is a web service that provides resizable computing capacity. There are servers in Amazon's data centres that you use to build and host your software systems. This solution provides the needs for my application.

Intended

Next month, I will continue to work on the AWS EC2 instance as this requires a lot of work and time to set up correctly. There are high security measures applied when setting this up as it is important only you have access to your servers. I will have to work with public and private keys as one is required to ssh to the AWS instance. A program called putty is used for this function that I will need to download. Researching more details on server integration for the application is another element I intend to work on.

My Reflection

This month was very busy as first semester exams where taking place. The exam dates where 05/01/2017 to /15/01/2017. I focused most of my time studying for these exams. I did not do any work on the project during this time and needed a short break and time to relax after the exams had finished. Semester 2 began on the 23/01/2017. The majority of the progress I made this month was accomplished after the 23rd.

Reflective Journal

Student name: Steffan Schmieder

Programme: BSc (Honours) in Computing IoT

Month: February

My Achievements

This month I set up and initialised the AWS EC2 instance. The instance is set up using the free tier as my application is small and does not have many users. It is not necessary for me to use a paid tier. I choose to install an apache web server on the AWS instance. Apache web server is open source and free to use. It is the most popular web server severing hundreds of millions of websites. It will help deliver content from my application over the internet. I am also continuing to modify and build the android studio application. I've added volley classes to allow POST requests to the server and added server code that allows it to accept POST requests. I've created the python code that will run on the Raspberry Pi. The Raspberry Pi is also able to gather temperature and humidity data and can display this to the user using an LCD screen.

Intended

Next month, I will continue to work the android studio application allowing it to send requests in the form of JSON data. The server should save the requests to a database. The application GUIs still have a basic look and feel to them, so I intend to improve their appearance. I may use Photoshop or Gimp to create image for the GUIs.

My Reflection

I feel I have made some good progress this month and am happy with the application so far. There is still some feature I need to add to make a complete looking application. The project idea is very unique which means when im solving errors or fixing issues there is not much information that is directly related to fixing my issue. I have still been able to find many helpful tutorials that can aid me in this project.

Reflective Journal

Student name: Steffan Schmieder

Programme: BSc (Honours) in Computing IoT

Month: March

My Achievements

My android application is successfully sending request messages to the Raspberry Pi and the server is saving them. I fixed some databases issues so now everything is in working order. The android application is displaying the user logs that are saved in the databases. I have added timed locking and unlocking functionality. I have also made changes to the layout and the design of the app improving the GUIs. I am continuing the work on the technical report document that accompanies this project.

Intended

I intend to add some final aspects and finishing touches to the project in the final month we have left. The technical report has much to be added before it is fully complete. I also have to create a project poster for the showcase presentation. PowerPoint sides must also be created for the final presentation as well.

My Reflection

I am happy I have made good progress so far as the end of semester is approaching and work is piling up. This semester I have no exams as my modules are all graded through projects. I have four on-going projects including this one with three of them due by the end of the month. Much of my time spent at home is completing these projects and writing up reports. One of them has a 5000-word expected report that is going to take a considerable amount time to complete.