Home Automation System

A cheap and open-source alternative to control household appliances

Bassam Ruwaida, ruwaida@kth.se, 076-2262492
Toni Minkkinen, tonim@kth.se, 076-2491015

Abstract

This project revolves around creating a home automation system prototype with the main focus being the ability to lock/unlock a door through the internet. The system consists of a central device, a server and an Android application.

The central device is a microprocessor, in this case, a Raspberry Pi that connects to the Internet and receives an order to control a motor which in turn turns the lock with the help of gears. The ability to rotate the motor in both directions is achieved by the use of an H-bridge. The server manages users and devices, and handles the communication between the application and the central device. Users and devices are stored in a database on the server. The application is a frontend which presents the user with a list of devices to interact with.

The main prototype where the Raspberry Pi acted as a central device was abandoned due to time and resource constraints. It was instead used to control the motor directly. This brought up some problems concerning powering the device using batteries. The software of the prototype is mostly working but due to the same time limitations not all planned features could be implemented.

Keywords: Home Automation System, Raspberry Pi, Android, Door lock, Wireless Communication.

Table of contents

Contents

Abstract... 2

Table of contents 3

1. Introduction 5

1.1 Problem Description 5

1.2 Objective 5

1.3 Disposition... 6

2. Background... 7

2.1 Existing Products... 7

2.1.1 Lockitron 7

2.1.2 UniKey Kevo 8

2.1.3 August Smart Lock... 9

2.1.4 Goji Smart Lock...10

2.2 Technology...10

2.2.1 Bluetooth 4.0...11

2.2.2 Arduino 11

2.3 Security 12

3. Theory...14

3.1 Overview 14

3.2 Raspberry Pi...15

3.3 Software 16

3.4 Door Locks 17

3.5 Locking Mechanism...18

3.5.1 Motors...19

3.5.2 H-Bridge...20

3.6 Switch/Sensor 20

3.7 Wireless Connection...21

3.7.1 Zigbee, Z-wave and 2.4 GHz Transceiver 21

3.8 Communication & Security 22

3.9 Prototype Overview 23

4. Method...25

4.1 Hardware 25
4.2 Software 25
4.2.1 Server and database 26
4.2.2 Client application 26
4.2.3 Raspberry Pi Software 26
5. Results and analysis 27
5.1 Hardware 27
5.1.1 Design 27
5.1.2 Circuitry 29
5.1.3 Efficiency 30
5.1.4 Power management 31
5.1.5 Lock Detection 31
5.2 Software 33
5.2.1 Server and database 33
5.2.2 Android application 36
5.2.2 Raspberry Pi 38
5.3 Comparison 39
6. Conclusion 40
7. Future Work 41
References 42
Image references 45

1. Introduction

Today, technology has become an integrated part of people's lives. It has, and continues to influence many aspects of daily life and has allowed better social interaction, ease of transportation, the ability to indulge in entertainment and media and has helped in the development in medicine. The creation of many devices such as mobile phones and computers have caused many people to rely on technology to communicate with their friends, store information such as pictures, movies, documents, and music . The internet has become a common interface that many devices use in order to simplify the daily life of many people. The Internet has given people the ability to search for information, store their own information in the cloud while also giving them better ways of managing information. From the time of its introduction, the amount of people that use mobile phones and the internet to communicate with other people has increased dramatically to become one of the major means of communication.

Smartphones have allowed people to connect to the internet without the need for a computer, while still offering the same functionality but through different means. With the introduction of better hardware and better software, smartphones have become powerful devices and have become an important part of people’s daily lives. A major aspect is how the smartphone is able to connect and communicate with other devices. For example, smartphones can be used as a mouse for a computer, or it can connect to the speakers of cars allowing consumers to play their own music. There are many applications of this sort. A field that is recently gaining popularity is home automation which can also use smartphones as information or functionality hubs.

1.1 Problem Description

Many home devices now have WiFi and can interact with other home devices, smartphone applications and home computers. An issue is that these devices cannot communicate with each other or require an additional device to do so and need an individual application on the smartphone to be controlled. A much better option is to unify these devices into one program/device that controls them. As an example, one can control the lights, microwave, oven, tv, air-conditioning and door locks through one application on the smartphone. This gives the consumer more control of their home, for example, it allows them to set up conditions for when the lights turn on, or if they are on their way home, to preheat the oven before they get home. Therefore, home automation can simplify many manual actions.

1.2 Objective

One home automation application that has recently started to become mainstream is the ability to control door locks using a smartphone application or through the internet (web application). This project aims to develop a prototype of a product capable of locking/unlocking a door, with an emphasis on low cost and open source configurability. The end goal beyond this project would be a product that would hopefully allow people to connect to many other home devices through WiFi.

1.3 Disposition

Section 2 (Background) describes existing products that are similar to this project while also stating their features and the technologies used in making them. **Section 3 (Theory)** talks about the theory behind the different technologies and components used in the project and the reason why we use them. It also describes how the different components interact with each other. **Section 4 (Method)** states how the work and experimentation was approached, the methodologies and tools used throughout the project. **Section 5 (Results and analysis)** describes the results of the project and analyses them. Details about how the different components of the system work and are implemented are also described. **Section 6 (Conclusion)** discusses some problems encountered throughout the project and how they affect the final prototype. While also discussing whether the prototype is functional. **Section 7 (Future Work)** describes how some of the problems that were encountered could be solved and how the prototype could be improved. It also talks about the planned features that could not be implemented in time.

2. Background

In this section, existing products are briefly introduced. The technologies which they are based upon and their security features are discussed.

2.1 Existing Products

Four examples of door lock products that are out on the market or soon to be released are: Lockitron, Unikey Kevo, August Smart Lock and Goji Smart Lock. This section will introduce the functionalities and features of each of these products.

2.1.1 Lockitron

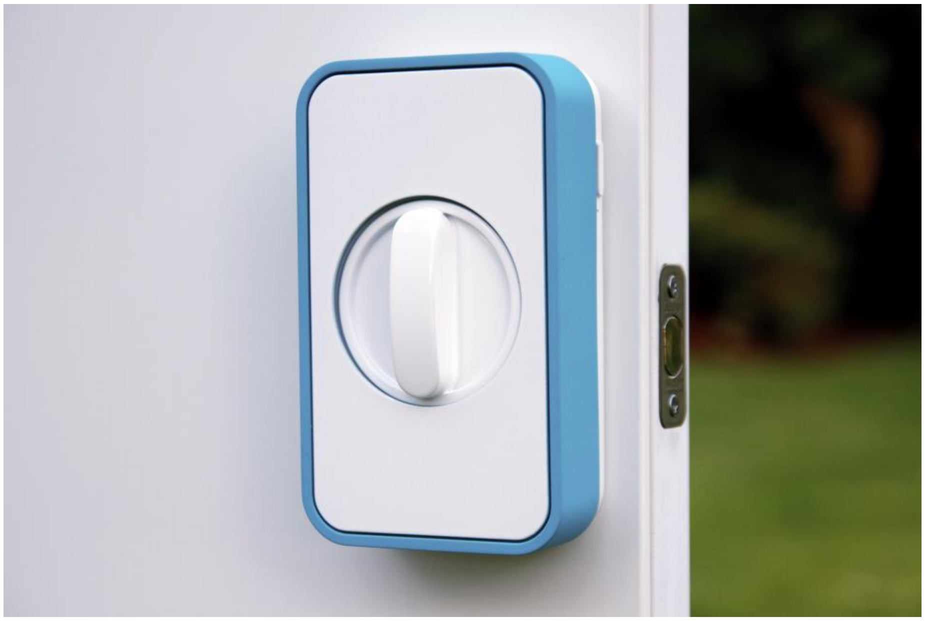


Figure 1: Lockitron [P1]

Lockitron is the product which is most similar to our project and is already on the market [1]. The first iteration of Lockitron replaced the deadbolt (discussed in section 3.4), but the newer iteration which is shown in figure 1, is placed on the door lock from the inside, thus allowing the use of the product even for renters since it can be easily removed and installed elsewhere. An issue would be that there are many different variations of door locks, therefore the user can print out a template to check if Lockitron would fit on the door lock, or send a picture to Apigy, which they would evaluate. Its batteries can last up to one year, and can send a notification when they are running low.

Lockitron allows the user to lock their door from anywhere in the world through WiFi. There are applications for both iOS and Android. It also functions with other mobile phones through the use of simple text message commands. There are multiple ways of unlocking the door lock; through the internet while using the app, by using Bluetooth 4.0 while walking up to the door, or through NFC(Near Field Communication). Bluetooth 4.0 is only available for some currently released Android smartphones and the Iphone 4S and 5 [2]. The user is also able to share access with family and friends, by using their email address or phone number. The consumer also receives notifications when the lock is being used by someone else. Apigy has released an API which allows the user to write their own programs to control the lock through scripts, applications and websites [3].

Lockitron is built upon Arduino [4], which allows it to be open source and easily modified with extra functions such as having secret knocks that unlock the door. It also has an integration with If This Then That (IFTTT) [5], which is a service that allows the user to have “triggers” with “actions” to accommodate them. For example, if the trigger is “door is unlocked”, then the action can be “turn on the lights”. This can be implemented through integration with other IFTTT compatible devices such as the Belkin Wemo [5]. Connecting Lockitron to the user’s WiFi is accomplished through the use of an Electric Imp WiFi module [6] which simplifies getting Lockitron to the network. The consumer is required to enter the WiFi network information into the application and hold the phone’s screen up to the Lockitron, then the information is flashed onto the module, through BlinkUp technology [5]. Lockitron also has integration with the Pebble smartwatch [5], allowing the user to control the door lock through the watch.

2.1.2 UniKey Kevo

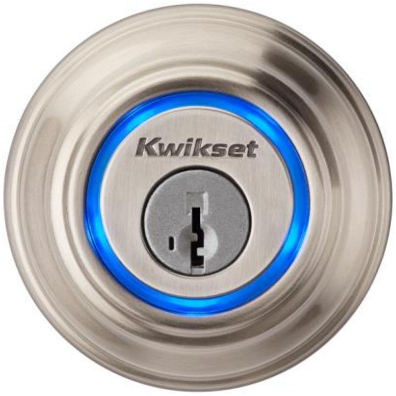


Figure 2: The UniKey Kevo [P2]

The Kevo application uses Location Services and Bluetooth low energy to detect when the user is near the door and a touch on the lock will lock or unlock it [7]. The lock which is shown in figure 2, has a feature to detect if the user is inside or outside of the door to prevent unauthorized access. The control application is only available for iPhone 4S and iPhone 5. UniKey also provides a Kevo Fob for users without a compatible phone. A Kevo Fob or key fob is a small security hardware device with built in authentication mechanisms [8]. With the application the user can manage locks, key fobs; and send, disable and delete electronic keys. The electronic keys can be given to family, friends or visitors.

It uses 4 AA batteries and has a claimed battery-life of one year. The lock has indicators for low battery levels and in the case the batteries are not replaced in time, a standard key has to be used. The lock is a deadbolt replacement and is designed to be easily installed.

2.1.3 August Smart Lock

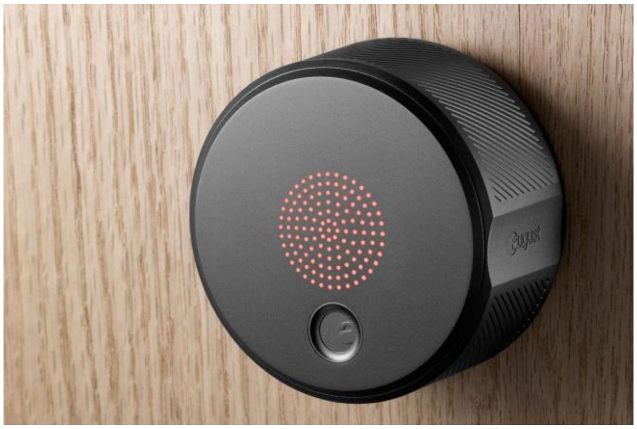


Figure 3: August Smart Lock [P3]

The August Smart Lock which is shown in figure 3, employs only Bluetooth 4.0, hence the device itself is not connected to to the user's home network or to the internet, nor is it connected to any power source [9]. The advantage of this is if the home WiFi network or the power at the user's home is down, the August Smart Lock still functions. Therefore it uses 4 AA batteries, with a lifespan of six months to one year, and notifies the user when the batteries are running low [10].The lock itself is installed on the interior portion of the deadbolt, and the creators boast that it has "90% compatibility" with all deadbolts. It also uses LEDs and chimes to notify the user if the door is locked/unlocked.

August uses Bluetooth to unlock the door when it senses that the phone is near the door, it is unnecessary for the user to take out their smartphone or using the key, thus being handsfree. Though the key can still be used if the device has run out of batteries. It also has the ability to have multiple users and guests by issuing "keys" to them, adding a time limit and taking away permission to any user, which is done through the application itself or through a computer [10].

The Smart Lock keeps a log of the times the door is locked/unlocked and which user has done so while also notifying the main user. As mentioned earlier, the device is not connected to the internet, so it is the application which connects to the server to send notifications. Since each key is issued to a specific person, every time that key is used, the server would notify the main user to which user has used the lock. However, how the device recognizes if the user is outside or inside hasn't been divulged yet by the creators. If the smartphone is lost, then the key can be remotely wiped [10].

2.1.4 Goji Smart Lock

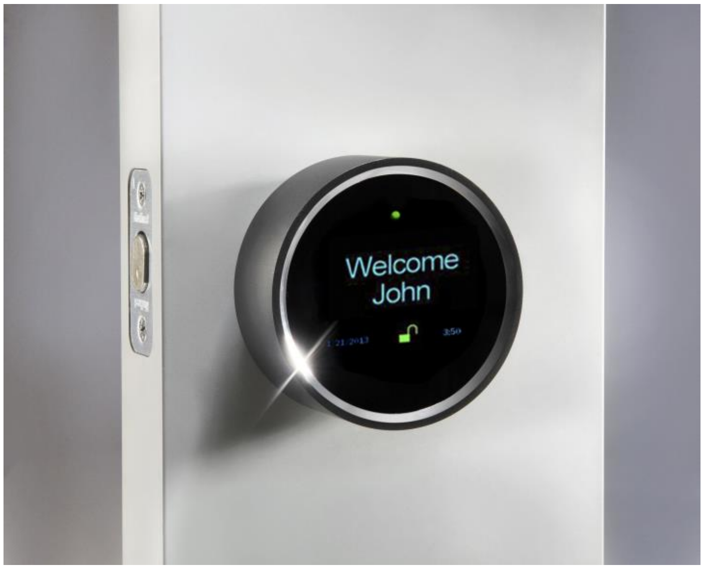


Figure 4: The Goji Smart Lock [P4]

Another product is the Goji Smart Lock [11] which is shown in figure 4. One of the biggest differences between Goji and the rest is that it features a camera. The camera is used to take pictures of who is at the door which are automatically sent to the user's mobile phone. The lock is installed instead of a normal deadbolt and is compatible with most doors that use deadbolts [12].

The Goji Smart Lock is connected to the home network over Wi-Fi and can be locked and unlocked from anywhere in the world. The lock also uses Bluetooth to detect when the user is nearby and automatically unlocks the door when the user approaches the door and locks when the user leaves. If the user doesn't have a Bluetooth low energy compatible phone a key fob is also provided. For now Goji only supports iPhone and Android phones but support for more devices is coming. The lock has support for guest access. The user can specify who has access to the door and at what times. It also logs all activity around the lock and the logs are viewable in the mobile application and in the online account.

2.2 Technology

Since most of the existing products take advantage of Bluetooth 4.0, an introduction about Bluetooth and it's features and what differs between the versions are presented. A brief introduction about Arduino is also presented.

2.2.1 Bluetooth 4.0

Bluetooth technology can be found in many devices ranging from smartphones and home entertainment products to watches and medical devices. One popular use is using your phone to connect to the car to listen to music for example. Bluetooth technology is a short-range communication technology which has a low cost and uses low energy [13]. When two devices connect to each other they can “pair” with each other, as long as they are within each other’s proximity. Afterwards a link is maintained, even if there is no data flow. A feature of Bluetooth wireless technology is the ability to handle data and voice transmissions simultaneously. It also operates in the unlicensed industrial, scientific and medical (ISM) band at 2.4 to 2.485 GHz, using spread spectrum frequency hopping full duplex signal. Frequency hopping spread spectrum is a transmission technique where the frequency used is changed or switched at random time intervals. This causes the transmission to be more secure; since it is hard to intercept and has minimal interference with other transmissions [14].

Bluetooth 4.0 is the newest iteration of the Bluetooth wireless technology and is already implemented in some smartphones such as the Samsung Galaxy S3 and the Iphone 5. It will also be implemented in many more smartphones in the coming year [2]. The most significant characteristic of Bluetooth 4.0 is energy efficiency, thus providing a much better battery life for devices. Therefore, a new protocol was added to the Bluetooth Core Specification which is the Bluetooth low energy (BLE) [15]. BLE was designed for devices that collect small chunks of information frequently, therefore it is not optimized for file transfer or streaming even though it has a data rate of 1Mbps [16]. This design allows a device to be on a button-cell battery and last for many months. A new feature that was added to Bluetooth v3.0 and 4.0 is the compatibility with NFC, thus allowing devices to “pair” through tapping these devices together [17].

BLE is also known as Bluetooth Smart, and the devices that implement Bluetooth 4.0 have two distinct variations. They are divided into Bluetooth smart ready devices which are devices that uses the full range of Bluetooth 4.0, and Bluetooth smart devices, which are devices that gather specific information and sends it to Bluetooth smart ready devices. There are also two different wireless radios, the dual mode radios which are in the Bluetooth smart ready devices. These radios support both classic connections and BLE connections. The Bluetooth smart devices have a single mode radio which allows them to only make Bluetooth low energy connections [18].

2.2.2 Arduino

Arduino is a single-board microcontroller board based on Atmel’s 8-bit microcontrollers [19] and is shown in figure 5. The hardware is open-source which means that the user is allowed to study and make changes to the hardware. All original design files are also available.

The standard Arduino board is the Arduino Uno [20]. It is based on Atmel’s ATmega328 microcontroller. The board has 14 digital input/output pins and 6 analog input pins. There are also other models of Arduino boards available with varying sizes, number of I/O pins and functionality [21]. Some of these are the Arduino Mega [22], which is bigger than the Uno and features 54 digital I/O pins and 16 analog input pins, and LilyPad Arduino [23], which is designed to be wearable and only has 9 I/O pins.

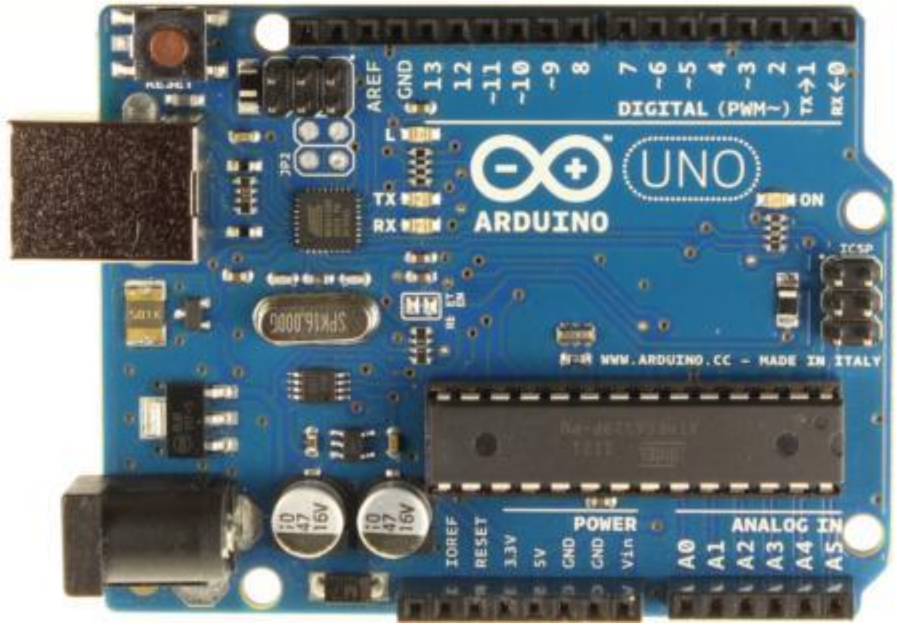


Figure 5: The Arduino Uno board [P5]

For more functionality one can attach add-on modules called “shields” [24] to some of the boards. Some of the functionality shields provide are motor controls, GPS, ethernet, Wi-Fi and LCD displays. The shields are connected to the I/O headers on the board and, depending on what pins are used, can often be stacked on top of each other [25].

The Arduino microcontroller is programmed in C/C++ either by using the Arduino IDE or by using a text editor and manually compiling and linking the source code. The IDE is open-source and is written in Java and is thus also cross-platform [26]. The IDE handles the compiling and linking of the source code and uploads the resulting hex file to the board where it will start running [27].

2.3 Security

All of the locks mentioned in section 2.1 use the built-in security of Bluetooth when using their respective application to unlock the door. All of them, except Kevo, claim to use the same secure communications protocols and data storage security as online banking services [28][29][1], which is AES-128 encryption [30]. Kevo claims that they use “military grade” encryption [31]. All of Lockitrons traffic goes through HTTPS and thus uses TLS (Transport Layer Security) [32] which is a set of cryptographic protocols that provide communication security over the Internet.

As mentioned earlier, devices that use Bluetooth can pair with each other, thus a link is created between them. To secure the link level, four entities are used; the Bluetooth device address which is unique for each Bluetooth device, private authentication and encryption keys, both which are 8 to 128 bits in length, and finally a random number which changes frequently. These entities are used to generate a key or Personal Identification Number (PIN) which is then used between the devices to connect or transfer data [33]. A 4-digit PIN is usually sufficient for most services, but for higher security services, a larger digit code can be used, since a PIN can vary between 1-16 octets. There are three modes of security for Bluetooth, the first mode is non-secure, the second mode is a service level security, the third level is a link level security [33]. The second level can be used to pair with devices such as a headset, while the third level can be used to send/receive data between two mobile devices .

A difference between regular/classic Bluetooth and BLE is in the generation of a Long-Term Key (LTK) instead of a link key. Both these keys perform the same task but instead of both devices generating the same key, the way LTK is established is different. One device determines the LTK and sends it to the other device during pairing [34].

3. Theory

The project revolves around developing a home automation system which allows people to control household applications through a smartphone application. Currently, it is focused on being able to lock or unlock a door lock. This is achieved by using a central device that connects to the door lock.

This section describes the overview of the project, many of the components which might make it to the final product and the technologies used for the software part of the project.

3.1 Overview

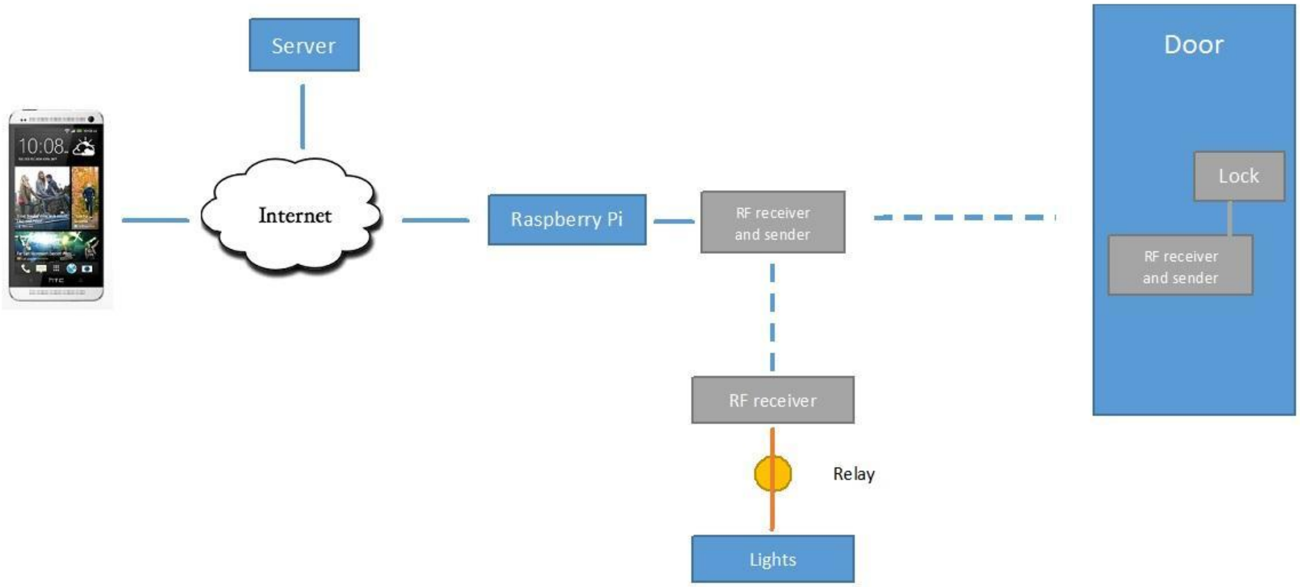


Figure 6: The structure of the planned product.

To develop a home automation system, the developer has to look at the necessary components and software involved from the lock to the users smartphone. Figure 6 represents the structure of the planned product where everything is controlled wirelessly. The product aims at being able to work with many house appliances such as door lock and lights for example, though the lights are a secondary objective.

The first component would be the door lock itself. Door locks differ around the world. There are many lock variations with different locking mechanisms. Therefore the product should be able to work with most of them without problems. Door locks are discussed in section 3.4. Details about locking mechanisms are discussed in section 3.5. A sensor or switch will be used to detect if the door is locked or not, which will be further looked into in section 3.6.

Two microcontrollers are needed to control the hardware, provide the user with a way to communicate with the lock, connect to the internet (through wifi, for example) and allow for further customisation and connectivity. One microcontroller acts as the central unit while the other microcontroller is responsible for controlling the motor which locks/unlocks the door.

Concerning the central unit, two of the most commonly available microcontrollers are Raspberry Pi and Arduino. The Arduino was discussed in section 2.2.2 and the Raspberry Pi is discussed in section 3.2. The component the team intends to use is Raspberry Pi. What makes it more appealing than other microcontrollers is the fact that it consumes less energy/power due to its ARM processor, while it still hosts an entire operating system. This allows it to run several services (e.g. networking, hardware control) simultaneously, and provides an environment on which many high level programming languages such as Python, Javascript, Ruby and Perl can run. Because of this, programming its hardware is drastically simplified. At the door lock a microprocessor such as Arduino or PIC (Peripheral Interface Controller) could be used to control the lock. The wireless communication is discussed in section 3.7.

A server is also needed to host a database which would store lists of users and devices. Considering the fact that this device could potentially become a popular product, one must anticipate scaling up the number of users and devices. Because the database would only hold, at most, four tables, and because there are limited interrelations, partitioning the tables is not considered a major issue -- at least currently. The Raspberry Pi would connect to the server which would just provide a secure frontend to the database and a central point for users' smartphones to connect to. The server and the difference between the web application or a native application are discussed in section 3.3.

With the planned product, the user should be able to, with the application, register and login, get a list of devices (door locks) with their statuses and interact with said devices. Another feature would be to use the phones GPS or WiFi to detect when the user approaches the door to unlock it and lock it when the user is leaves the home area or disconnects from the home WiFi network. The user should also be able to give others access to their lock and grant time limited access for guests. The server should log all actions and present to them to the owner of the lock via the application.

3.2 Raspberry Pi

The Raspberry Pi which is shown in figure 7, is a small single-board computer developed by the Raspberry Pi Foundation. It features a Broadcom SoC (system on a chip) with a 700MHz ARM11 CPU and 512MB of RAM. On the board there are also many interfaces, for example USB, Ethernet, video and audio and 26 GPIO (General Purpose Input Output) pins [35]. Some of the GPIO pins are for power and some have special functions, such as UART (Universal Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface Bus) and I^2C (Inter-Integrated Circuit) [36]. The Raspberry Pi runs the Linux operating system off a SD-card.

Figure 7: Raspberry Pi and SD-card [P6]

3.3 Software

The client application can either be a web application or a native mobile application. The advantages of a web application are that anyone can access it with a web browser, regardless of platform, there is no need for an app store and applications prototypes are faster to develop. The downside is that it is not possible to use the hardware and software capabilities of the phone, for example GPS and notifications.

The advantages of a native application are that it performs better since its written closer to the hardware and the user interface looks and feels more natural compared to those of web applications. The downside of native applications is that the application is platform dependent, meaning the application would have to be written for each platform to reach a larger user base.

The largest mobile phone platforms are Google’s Android and Apple's iOS. The market share for Android is 79.3% with 187.4M units shipped and for iOS 13.2% with 31.2M units shipped during the second quarter of 2013 [37]. In May of 2013 Google announced that there are 900 million activated Android devices [38] while in January of 2013 Apple had sold more than 500 million devices [39]. The cost to register as a iOS developer and distribute the app in the App store is $99/year [40]. To be able to distribute Android applications on the Google Play store the cost is a one time fee of $25 [41]. Since Android has a larger user base and the cost of development is cheaper it was decided that it was the platform the team would use.

A server is to be used to manage communication between users and locks. The server also handles the database which stores all user information, such as login credentials and IDs of their locks. When the user uses the application to unlock the door, the application first authenticates with the server and then the server lets the software on the lock know it needs to unlock.

The software on the lock device could either check with the server at a set interval if it needs to lock or unlock the door, or get notified by the server. The problem with the latter option is that then the user would have to open ports in any eventual firewalls between the device and the Internet. Another problem would be if the user has a dynamic IP address from their ISP (Internet Service Provider). Then they would have to manually change it in the application every time it changes, or use a service that binds their IP to a static URL. After the device has determined if it needs to lock or unlock the door, it sends a signal to the locking mechanism.

3.4 Door Locks

There are many different types of locks with varying security and prices. Security ranges from normal everyday use to very secure which are used for example in banks and places that contain sensitive information. The more secure a lock is the higher the price gets. Locks also work in various methods and have different locking mechanisms. Some of these mechanisms are, pin & tumbler, tubular, mechanical and electronic locks [42]. A pin & tumbler lock is shown in figure 8. The most common types of door locks are deadbolt locks and key locks in doorknobs. The deadbolt is a locking mechanism that slides a metal bolt or latch though a cylinder to secure the door. Therefore a deadbolt lock can be referred to as a cylinder lock [43].

Figure 8: A Pin & Tumbler Lock [P7]

Inside the cylinder there are multiple pairs of pins, upper pins and bottom pins. The upper pins are always the same length, but the bottom pins differ in length depending on the key itself. Each pair of pins lies in a shaft, with springs at the top of the shaft which hold the pins in their place. When the key is not inserted, the bottom pins rest in the cylinder, while half of the upper pins rest in the cylinder, thus preventing the lock from rotating [44].

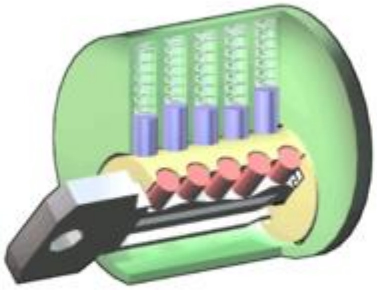


Figure 9: How a cylinder lock unlocks [P8]

The cylinder rotates when the key is inserted, this is due to the fact that keys have notches that match the bottom pins to a determined height such that there is a line between the pins, this line is called the shear line. At this state, the upper pins rest in the shaft above the cylinder. Therefore, when the shear line is clear, the cylinder is able to rotate as shown in figure 9, thus sliding the bolt forward and backward [45]. If the wrong key is inserted, the pins will not align themselves such that the shear line is clear, causing the cylinder to still be blocked, therefore not allowing the door to be unlocked.

The cylindrical lock is a form of mechanical lock. Another type of lock or locking mechanism are electronic locks. Electronic locks can use magnets, solenoids and motors to lock the door. There are varying ways of operating the lock, from using a switch or a keycard, to using biometrics such as fingerprints or retinal scans.

Due to the fact that the project is taking place in Sweden, the project focuses on the majority type of door locks or lock cases which are used throughout the country. Most of the existing products are targeted for use in America, where when the key is used, the turning knob rotates with it, therefore they are connected in this sense, but if the turning knob is used, the cylinder(outer key part) does not rotate, thus the turning knob is independent of it. A difference between Swedish and American door locks is that the fact that in Sweden, the outer part and the inner part of a door lock are independent. This means that if a door is unlocked with a key, the turning knob does not move. This causes a problem with detecting if the lock is locked or not, which will be investigated later in the report.

3.5 Locking Mechanism

This section focuses on how the door lock is locked/unlocked mechanically, when the Raspberry Pi sends a signal. For the locking mechanism, a motor is required to rotate the locking pin. Which in turn moves the latch in and out. Since the locking pin is flat, it would not be possible to rotate it using only the motor. Therefore a gear or a cogwheel is connected or attached to locking pin.

There are two different types of gears that can be used, spur gears and helical gears [46]. Spur gears have teeth that are perpendicular to the gear surface and multiple gears can be used in parallel with each other. Helical gears have teeth that are angled allowing for more contact with other gears. They can be used in parallel or at a 90 degree angle (perpendicularly).

To be able to move the latch in and out, the motor should be able to rotate in both directions. To do so, the polarity of the voltage needs to be changed. If a battery is used, alternating the respective pins will change the polarity and hence the direction, but the requirement of changing the connection is not suitable. Therefore another approach is used. A component or circuit capable of changing the polarity of the motor is the H-bridge.

3.5.1 Motors

This section discusses the different types of motors that could be used for the project. The first motor which is discussed is the DC motor. A DC or Direct Current Motor is an electric motor that converts an electrical current to mechanical energy or force. This is due to the simple fact that when a conductor carrying a current is placed in a magnetic field, it experiences torque and therefore moves. If the current moves in the opposite direction then so will the rotation. The rotation then depends on the current flow, so the motor moves when there is power and stops moving when power is shut off. A DC motor requires an H-bridge to be able to rotate in both directions.

Two other motor types that could be beneficial are servo motors and stepper motors. A servo motor is a type of specialized DC motor mostly used for angular precision control. In addition to the DC motor, it consists of a potentiometer, gear arrangement and circuitry which allows it to be controlled or programmed by the consumer. The angular rotation of a servo motor ranges from half circle (180 degrees) to full circle (360 degrees). It relies on encoders which provide the positioning information. A servo motor is capable of rotating in both directions without the need for an H-Bridge, but in comparison with a DC motor, they are also more costly.

A stepper motor is very similar to a servo motor. The difference is in the way it operates. It contains multiple electromagnets which are positioned around the center of the motor. When these electromagnets are energized, they pulls the gear towards it. This type of rotation is called a step. It requires a lot of power. An H-bridge is also needed.

A DC motor was chosen for the prototype for multiple reasons of which the main reason is a design choice. The motor will be positioned in a way that allows for the door to be locked/unlocked manually from the inside. A servo motor does not fit this specification. Furthermore, a DC motor is much easier to control, if not as accurate, and considerably cheaper compared to servo and stepper motors. For the final product, the use of a stepper motor might be more suitable due to the accuracy.

3.5.2 H-Bridge

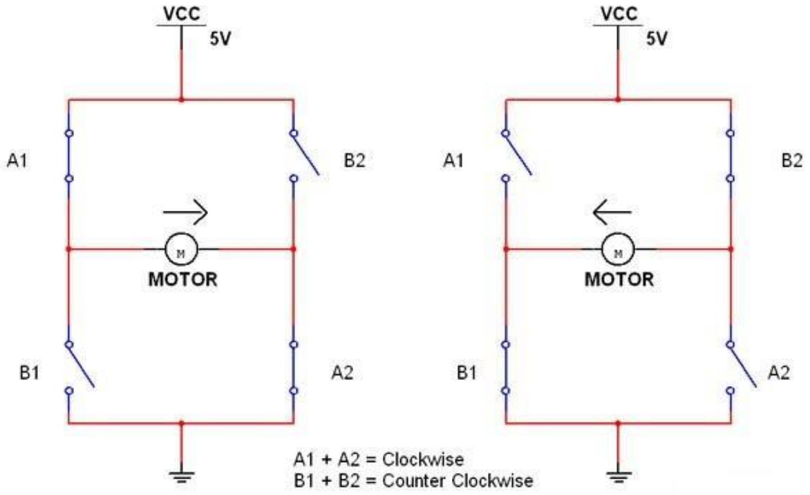


Figure 10: H-bridge theory [P9]

The H-bridge is a circuit capable of accelerating a DC motor in forward and reverse. It is given its name since the circuit has an ‘H’ shape as shown in figure 10. It consists of 4 switches, which when combined differently can lead to different motions or actions. When the A1 and A2 switches are closed, the motor gets power and rotates in the clockwise direction. When the B1 and B2 switches are closed, the motor rotates in the anti-clockwise direction. The A1 and B1, or A2 and B2 switches should never never be closed at the same time since that can lead to a short circuit and damage the H-bridge or other components on the circuit. The switches can consist of 2 NPN-transistors and 2 PNP-transistors or all NPN-transistors. NPN and PNP are a bipolar junction transistors that consist of two types of semiconductors. In the case of PNP for example, the n-type semiconductor which acts as the base, is between 2 p-type semiconductors that serve as an emitter and a collector.

3.6 Switch/Sensor

A sensor with the ability to detect proximity and act as a switch, while sending back information to the Raspberry Pi, is needed. It is used as extra precaution to sense if the door is locked/unlocked. This is the solution to the problem encountered when using swedish locks. Two such sensors/switches are the hall effect sensor and a microswitch.

A hall effect sensor is a transducer that has varied output voltages depending on the magnetic field [47]. These responses can be analog or digital. The binary on-off digital signal is a reason the hall effect sensor is referred to as a switch. Since it has varying outputs, it can be used in a wide variety of products, computers, cars, aircrafts and medical equipment for example. When the hall element is subjected to a magnetic field, it sends out an constant output voltage, which is very small, yet is proportional to the strength of the magnetic field [48]. The output voltage can change from maximum to nearly zero which is useful for digital devices.

The hall effect sensor is made up of 3 pins. One for supply, one for ground and one for the output signal. A supply voltage is needed to produce an output signal and create the switching effect. The sensor can either be “on” or “off” depending on how its circuitry is designed [49].

A microswitch is an electric switch that functions by applying a little force through a tipping point mechanism. When force is applied to the button of the switch, a current is allowed to run through the switch because the wires are connected. The opposite is also possible. So the microswitch can be either always “on” or “off” depending on the different working environments.

3.7 Wireless Connection

An aspect of home automation systems that has been an issue is the ability to control the household appliances wirelessly. The introduction of home automation systems into households and office buildings was obstructed by the distance barrier i.e how far the signals could travel before becoming weak. Nowadays, with the technologies available, mainly the ones that use the Radio Frequency (such as WiFi, Bluetooth and NFC), it is possible to create a wireless system that connects all the devices together.

However this does create a new problem. How does one make sure that all the devices can communicate using a singular technology such that they are compatible with each other. There are some manufacturers such as Zigbee or Z-wave who have created wireless communication protocols that are designed for home automation. Other solutions could be the use of Bluetooth 4.0 which has been discussed earlier in section 2.2.1, or 2.4 GHz transceivers. These technologies will be looked into further below.

3.7.1 Zigbee, Z-wave and 2.4 GHz Transceiver

Zigbee is wireless communication standard which is based on the IEEE 802.15.4 specification. It operates at the 2.4 GHz radio frequency globally, 915 Mhz and 868 Mhz in America and Europe respectively [50]. The wireless technology was designed to be used for low cost and low power control networks and wireless sensors. Therefore devices using Zigbee can have a battery life of many years. The data rate ranges from 20 kb/s in the 868 MHz frequency band to 250 kb/s in the 2.4 GHz frequency band and has a transmission range of 10 meters to 1600 meters, depending on the output power and the environment. Some of the advantages of using the Zigbee protocol are the support for different network topologies, low duty cycle, low latency and 128-bit encryptions [51].

Z-wave is another wireless communication protocol designed for home automation which also benefits from mesh networking which is explained in the next paragraph. It uses simple, reliable and low power radio waves and use the same chip family which allow the household electronics to communicate with each other using the common protocol. A single network can support up to 232 devices, through combining multiple networks [52]. Even non Z-wave devices can join the network by adding a Z-wave module to them. The frequencies used by the Z-wave devices vary depending on the country, and they do no operate at the 2.4 GHz frequency band as other home devices typically do. They have a transmission range of 30 meters indoors and 100 meters outdoors [53].

An important feature of both the Zigbee standard and the Z-wave specification is that they use mesh networks, i.e a network comprised of nodes that are able to communicate with each other dynamically without the need for a central unit [51][52].

To be able to use either of these wireless communication standards with Raspberry Pi, an additional module is required. For Zigbee, the Raspberry Pi requires an Arduino shield and a XBee module [54]. While for Z-wave, a module such as Razberry is required [55].

A transceiver is a combination of both a transmitter and a receiver. It can handle both analog and digital signals. The most common transceiver uses the 2.4 GHz frequency band which lies in the industrial, scientific and medical (ISM) bands. Though there are transceivers that use other frequency bands. A requirement of using a any of these solutions at the door would be the inclusion of another microprocessor which communicates with the Raspberry Pi.

3.8 Communication & Security

To make sure no one else but the authenticated user is able to unlock the door, the system needs to have good security both between the server and phone/device and between the device and lock, if a wireless solution is used here.

For the Internet traffic between the server and phone/device HTTPS could be used instead of HTTP traffic. HTTPS is the HTTP protocol layered on top of the TLS (Transport Layer Security) [56] protocol to achieve secure communication over the Internet. HTTPS provides authentication of the web server that the user is communicating with and thus prevents man-in-the-middle attacks. It also supports bidirectional encryption of data which protects against eavesdropping and tampering with the data by a third party. In practice this means that when the user uses the application to unlock a door, the client only communicates with the correct server, no information (name, password) is sent in plain text and no one else can impersonate that user.

The security for the wireless communication is more complicated. Different communication protocols have their own security implementation while they implement the same basic functionality. Bluetooth security was already discussed in section 2.2.1. Both ZigBee and Z-Wave use AES-128 encryption and an unspecified form of authentication [57][58]. If some other protocol was to be used, that did not have built-in security, some kind of encryption and authentication would have to be implemented between the Raspberry Pi and the lock.

3.9 Prototype Overview

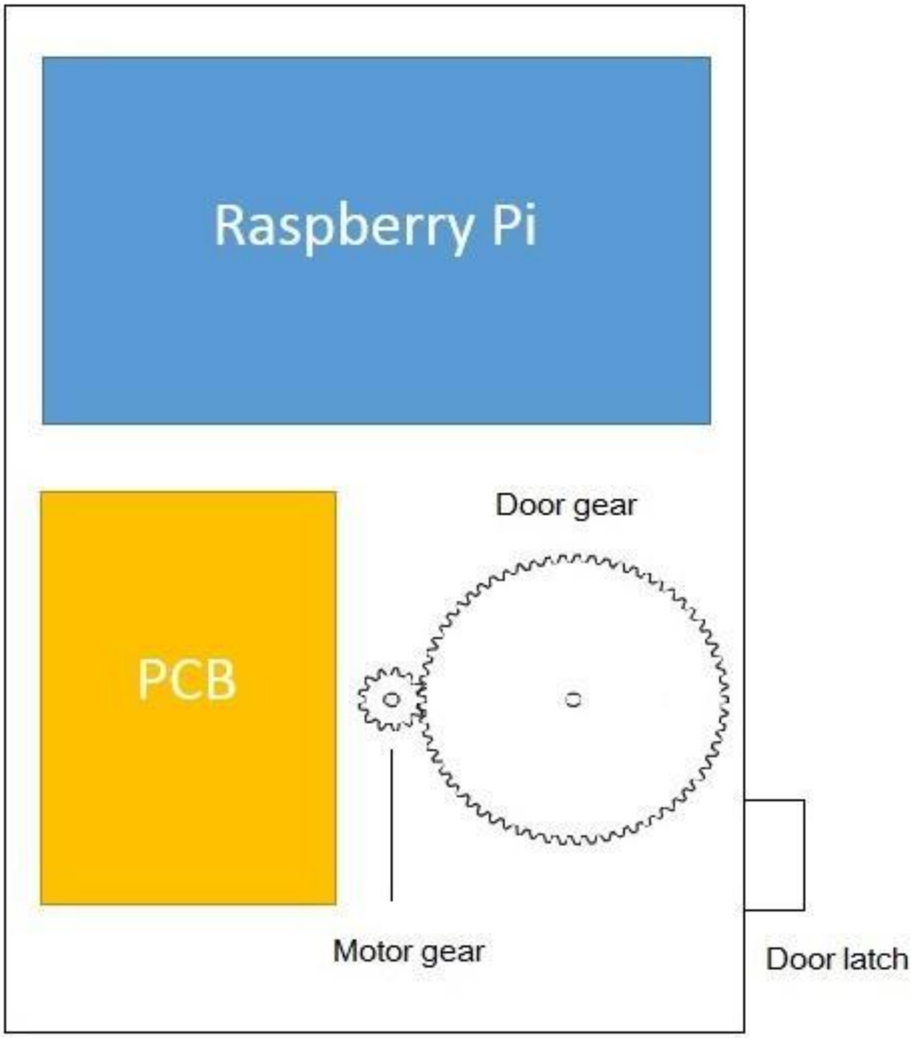


Figure 11: The design of the planned prototype.

The prototype still uses most of the components that the planned product does. Figure 11 presents the design of the prototype. As can be seen, the Raspberry Pi is no longer used as a central unit and is used directly to control the motor itself. The reason for this is due to monetary and time constraints. Though, the team considers that if it is possible to control the motor with Raspberry Pi then it is possible for any other smaller microcontroller to achieve the same result. This would cause the prototype to be much smaller. The Raspberry Pi would be powered by an electrical output close to the door. A base is needed such that the components are not directly connected to the door. This base is then mounted on the door.

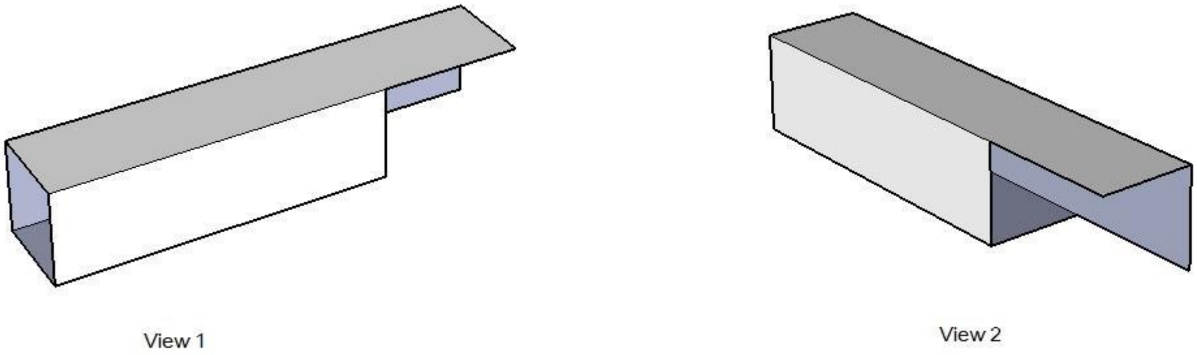


Figure 12: The motor mount.

Since an H-bridge is needed, it and the circuitry needed will be soldered onto a printed circuit board (PCB) and placed under the Raspberry Pi. The motor would be mounted using a small case that is attached to the base. The case, which is shown in figure 12, has two sides that are attached to the base, and two smaller sides that hold the motor in place, yet have a wide enough opening for the motor gear to control the gear placed on the door. The prototype would be covered by a plastic box while leaving enough room for the inner door lock component to be attached over the gear allowing the consumer to lock the door manually.

The prototype will use a sensor as an additional detector to check whether the door is locked or unlocked. Since the team wanted to keep the cost of the prototype low, a DC motor is used. Controlling the DC motor is much easier than a stepper-motor since it is very straightforward.

The software features for the prototype are register/login, retrieve list of devices with their status and interact with the devices. The team decided to focus on these features because they are the most basic yet the most important features.

4. Method

This section describes the process that was followed to achieve the set goals. It describes the methodologies used for the programming while also describing the tools used for working on the Android application and database. It also states the procedures undergone for building the prototype, and the different experimentation of components used to find the best possible solutions.

4.1 Hardware

The team first decided to try building an H-bridge circuit to perform the task of rotating the motor. The NPN-transistors used were the 2N4401H4. An issue that was encountered was that the motor wasn't rotating fast enough, it wasn't using the full power of the 9V battery. In theory, if a small current runs to the transistors base, a larger current is allowed to flow from the collector to the emitter. It was assumed that the current was not strong enough, therefore the transistors were exchanged with the PowerMOS transistor BUK545-60A/B. There was no noticeable change except that the battery started overheating. After a bit of experimentation with the circuit, the reason as to why the motor wasn't getting enough power could not be determined. Afterwards, the L293D H-bridge component was used which is capable of running two motors at the same time.

Furthermore, the center hole of the gear was cut to fit on the locking pin, and the gear on the motor was changed such to match the larger main gear. Plexiglass was used as a mount/base for the prototype. The screws used to attach the inner part of the door lock were used to stabilize the plexiglass against the door.

4.2 Software

The software development methodology used in this project can be described as a mix of the Spiral development model [59] and Evolutionary Prototyping approach [60]. The project was broken into smaller segments (e.g. functionality and features) and for each segment the objectives and constraints were determined. They were then developed, tested and, if needed, refined. Finally the next iteration was planned. By using the Evolutionary Prototyping approach, the system is continually improved and built upon. As an example of the development method, the following paragraph describes the process of creating the login and register functions.

A simple login/register function on the server was built. It used a small table in the database to store, in plain-text, just the name and password of the user. A basic HTML form was used to test the functionality. Once it worked, a Android GUI and necessary methods for communicating with the server were written. Then more features were added to the functions, such as hashing of passwords stored in the database and storing more information about the (email and date of creation).

The following subsections describe the tools used for creating the software and the reasons why they were chosen.

4.2.1 Server and database

The server-side of the software is built on the software platform known as LAMP [61]. LAMP stands for Linux (OS), Apache (HTTP Server), MySQL (database) and PHP (programming language). This software platform was chosen because the team had previous experience in using the included software and programming MySQL based database applications with PHP. Other platforms, databases and programming languages were not considered. phpMyAdmin [62] was used to create and manage the database. The servers response to client and device requests is a JSON (JavaScript Object Notation) object.

4.2.2 Client application

The Android platform was chosen for the client application because of the wide user base, registration and distribution cost, and because the team already had some experience in programming Android applications. The application was written with Eclipse [63]. Eclipse is an Integrated Development Environment mostly used for Java programming, but also supports other programming languages and feature additions by using different plugins. The Android Development Tools (ADT) [64] plugin was used to get support for Android application development. ADT lets the developer set up new Android projects, create user interfaces for the application, debug the application, export the application and run it in an emulator. Messages are sent to the server via HTTP POST and the responses are parsed with the JSON Java library.

4.2.3 Raspberry Pi Software

The software on the Raspberry Pi needs to be able to use the GPIO pins to control the locking mechanism. Many programming languages have libraries that add that functionality, including C/ C++, Java and Python. Python was preinstalled with the RPi.GPIO module [65] on the Raspberry Pi operating system and is the preferred and most used language in the community. Therefore it was decided to use Python for the project. The Python scripts were written in a text editor and executed with the Python interpreter. Messages are sent to the server with HTTP POST and the servers response is parsed with the JSON module from Python Standard Library.

5. Results and analysis

This section presents the acquired results from the project. It describes the final outcome of the prototype, the android application and the server and database and the software on the Raspberry Pi. Analysis of some of the hardware is also stated.

5.1 Hardware

In this section, the design of the prototype is presented and discussed. The circuitry is described. Afterwards the Efficiency of the prototype and the power management are discussed.

5.1.1 Design

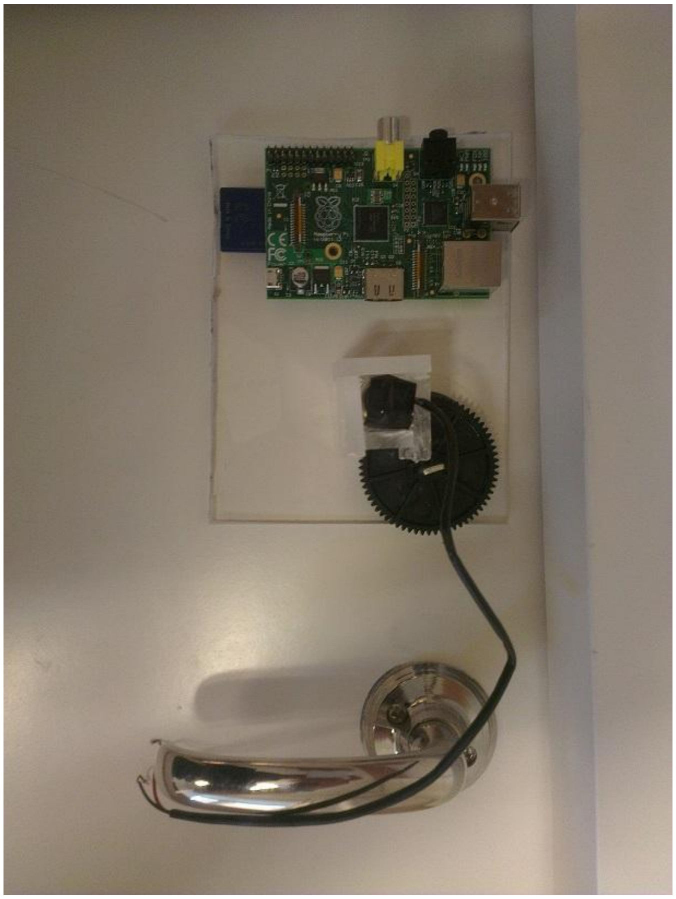


Figure 13: The prototype mounted on a door.

Figure 13 shows the actual prototype. The base used to mount the components is plexiglass. Plexiglass was used for the prototype due its low cost and ability to take the shape needed for the prototype. The dimensions of the plexiglass are 11.5 cm x 15 cm, which easily fits the Raspberry Pi. Due to time constraints, the H-bridge component and wires were not soldered onto a piece of PCB which would have had the dimensions 2 cm x 3 cm. The motor case is also visible and has the dimensions 3 cm x 3 cm and 5 cm high.

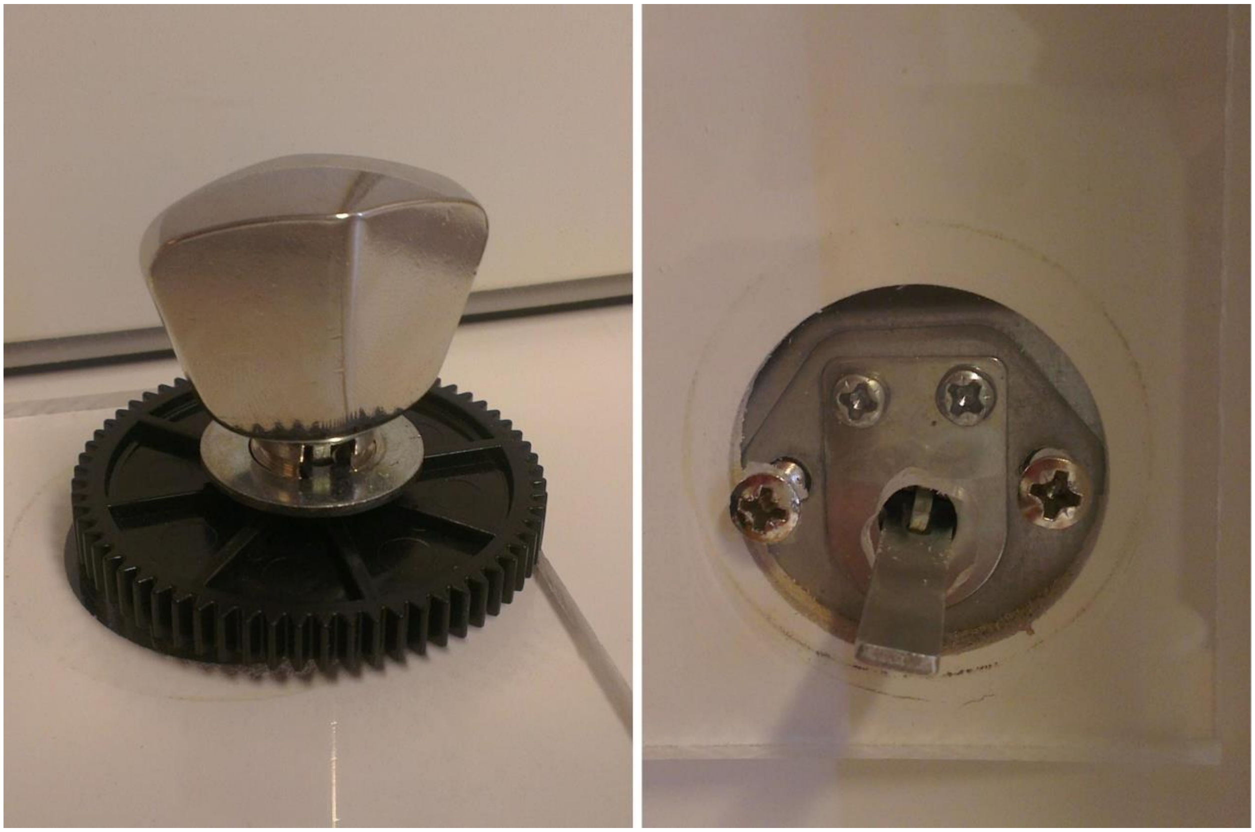


Figure 14: Turn knob and plexiglass attachment.

The two screws used to hold the plexiglass in place are shown in figure 14. It also shows that the turn knob is capable of being attached as long as the locking pin is long enough. If it is not long enough, then it can be easily changed. An issue with the design is the fact that the Raspberry Pi needs an electrical outlet nearby.

5.1.2 Circuitry

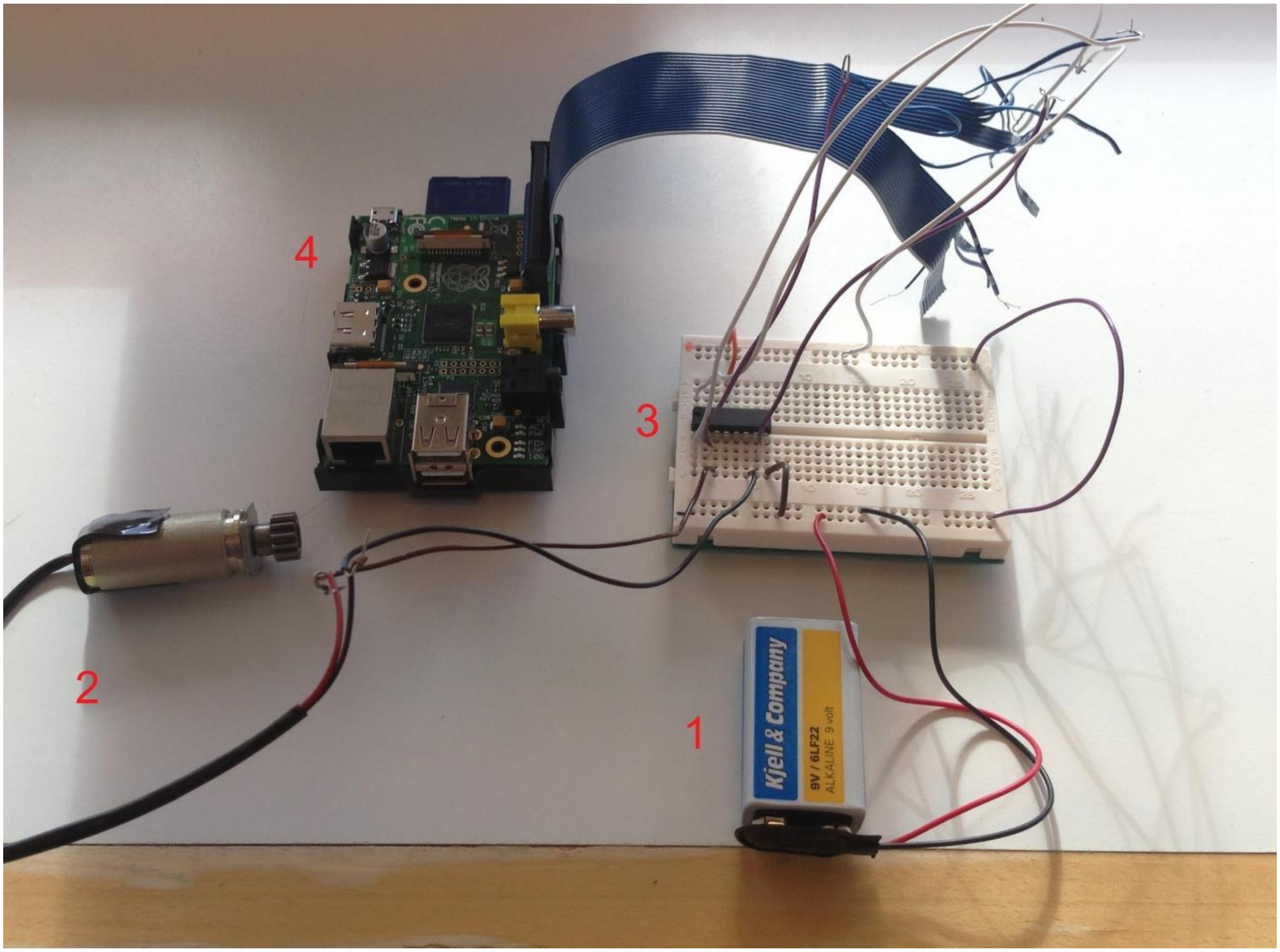


Figure 15: The electronics

The L293D H-bridge chip, also known as the motor controller IC has helped in simplifying the circuitry as shown in figure 15. The components used for the prototype are as such: A 9v battery (1), a DC motor (2), L293D chip (3), and Raspberry Pi (4).

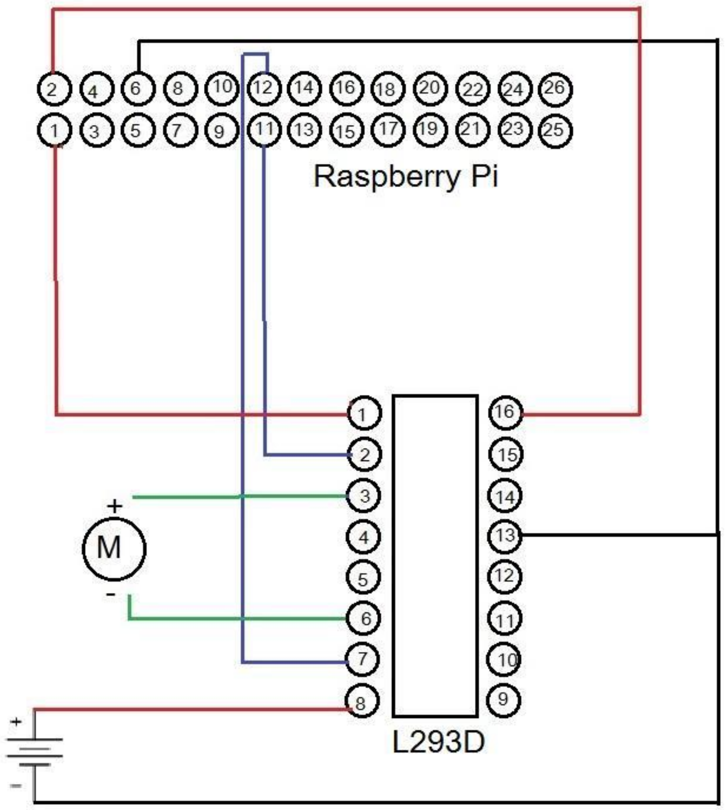


Figure 16: Circuit diagram.

The L293D chip has two power pins. One powers the motor (pin 8) and is powered by the battery. The other, which is for the chip’s logic, is powered by a 5V pin from the Raspberry Pi (**Pi**: pin 2 - **H-bridge**: pin 16). The Raspberry Pi and L293D ground pins are connected to each other (pin 6 - pin 13). The chip is enabled through a 3.3V pin from the Raspberry Pi (pin 1 - pin 1). Finally, there are 2 input/control pins which are controlled by any 2 digital output pins from the Raspberry Pi (pins 11 & 12 - pins 2 & 7). These input pins are responsible for rotating the motor in different directions (pins 3 & 6). The circuit diagram can be seen in figure 16.

5.1.3 Efficiency

The efficiency of locking/unlocking the door has many factors. These factors are the power supplied to the battery and the gears and the resistive force from the door lock itself (deadbolt). The big gear used has 65 teeth, while the smaller motor gear has 13. With 9V and a limited time interval of 10 seconds the motor itself is able to rotate for 2.5 turns (900 degrees). The door lock rotates 0.5 turns (180 degrees). With a 12V supply, the motor itself rotates 3.5 turns (1260 degrees), and the door lock 0.7 turns (252 degrees). Obviously the use of a 12V supply for the motor is more efficient. It would be able to power the motor for a much longer period of time in comparison with the 9v battery.

5.1.4 Power management

Considering the placement of the Raspberry Pi, a question about the possibility of powering it through the use of a battery arose. The Raspberry Pi needs 5V and 700mA to function. The Raspberry Pi is capable of running off of batteries. To feed power to the Raspberry Pi, a voltage regulator is needed between the Raspberry Pi and the batteries to ensure that only 5V gets to the Raspberry Pi. Most voltage regulators need in excess of 6V going into them before they produce 5V [66]. This means that there is a loss of voltage which ends up as heat.

Two different types of regulators that could be used are linear regulators and switching regulators. Each allows the Raspberry Pi to function for different amounts of time and have different efficiency levels. The time achieved by these regulators is as such [66]:

$$\text{Time (Linear Regulator)} = \frac{\text{Battery Capacity}}{\text{Load Current}} \quad (1)$$

$$\text{Time (Switching Regulator)} = \frac{\text{Battery Capacity} * \text{Efficiency} * \text{Battery Voltage}}{100 * \text{Load Current} * \text{Pi Voltage}} \quad (2)$$

Additionally the battery needs to be able to supply 700mA per hour. There are many different battery packs that range from 150mAh to 10,000+mAh. But the more current a battery can supply, the larger and costlier it gets. In conclusion, it is possible to power the Pi by using batteries but it is impractical.

5.1.5 Lock Detection

An issue encountered throughout the project relates to the ability to detect whether the door is locked/unlocked after using a key or the turning knob. Considering that the two sides of the door lock are independent, the prototype is unable to recognize that the door is no longer locked/unlocked while using the key for example. Therefore it does not update the information stated on the application.

Another similar issue arose from deciding on an interval of time for the motor to unlock/lock the door. Independently, this works as it should. But in case the door is unlocked/locked manually, the door locks rotation is at least 180 degrees or more, and the consumer could rotate it to its maximum. This could be a problem since the motor might not be able to lock/unlock the door during the time interval. These problems can be solved using a sensor, but the following paragraph explains why a sensor was not used during the project.

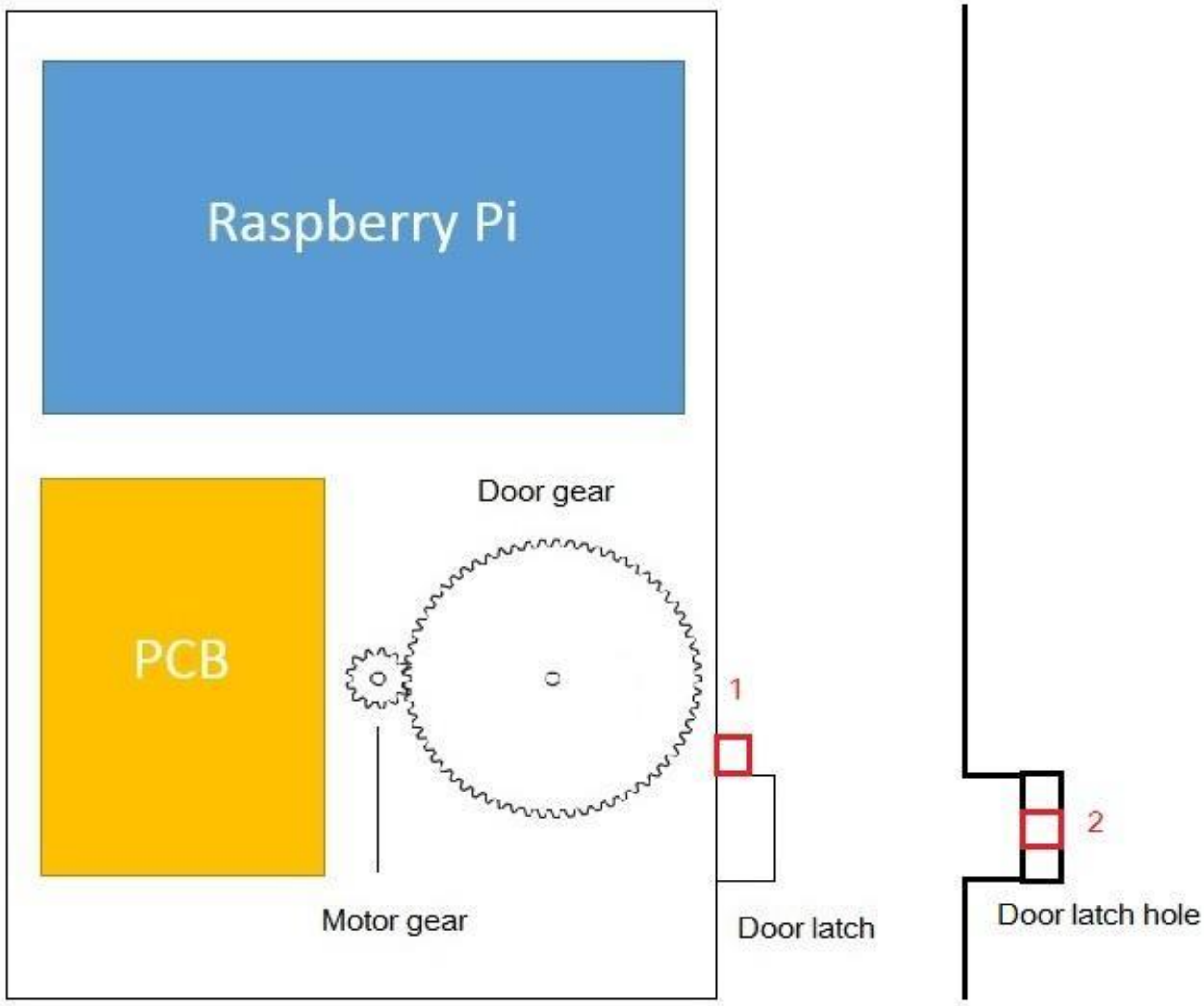


Figure 17: Placement of the sensor.

An issue with the design was the placement of the sensor which would detect if the deadbolt latch is outward or not. Some of the possible positions are shown in figure 17. The problem with the first position (1) is that the space between the door and its frame varies between door to door and the sensor could be damaged. Two door gaps were measured and the distance between the door and the frame tends to be around 3-4 mm, while the width of the sensor is 5 mm. The sensor could be placed vertically (2 mm), but then it is unclear whether it would be able to detect the magnet.

The complication with the second position (2) is the modification of the deadbolt latch compartment. This modification is needed such that the sensor is placed in the compartment in a way that would not damage it. The wires that would be needed to power the sensor would also come in the way of the consumer entering their house. Therefore the concept/idea was dropped, and the use of the key has been classified as a backup in case the phone is lost/out of battery.

5.2 Software

The software consists of three components: the client application, the server and database, and the application on the device. Figure 18 shows the overall flow of communication between the components. This section will describe the communication between the different components, how the components are built, show the database structure and show what the application looks like.

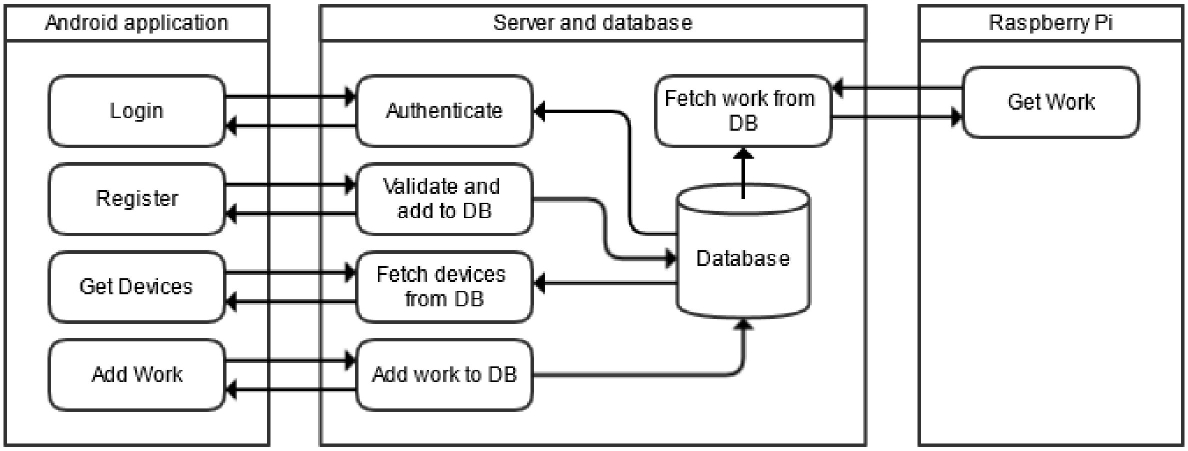
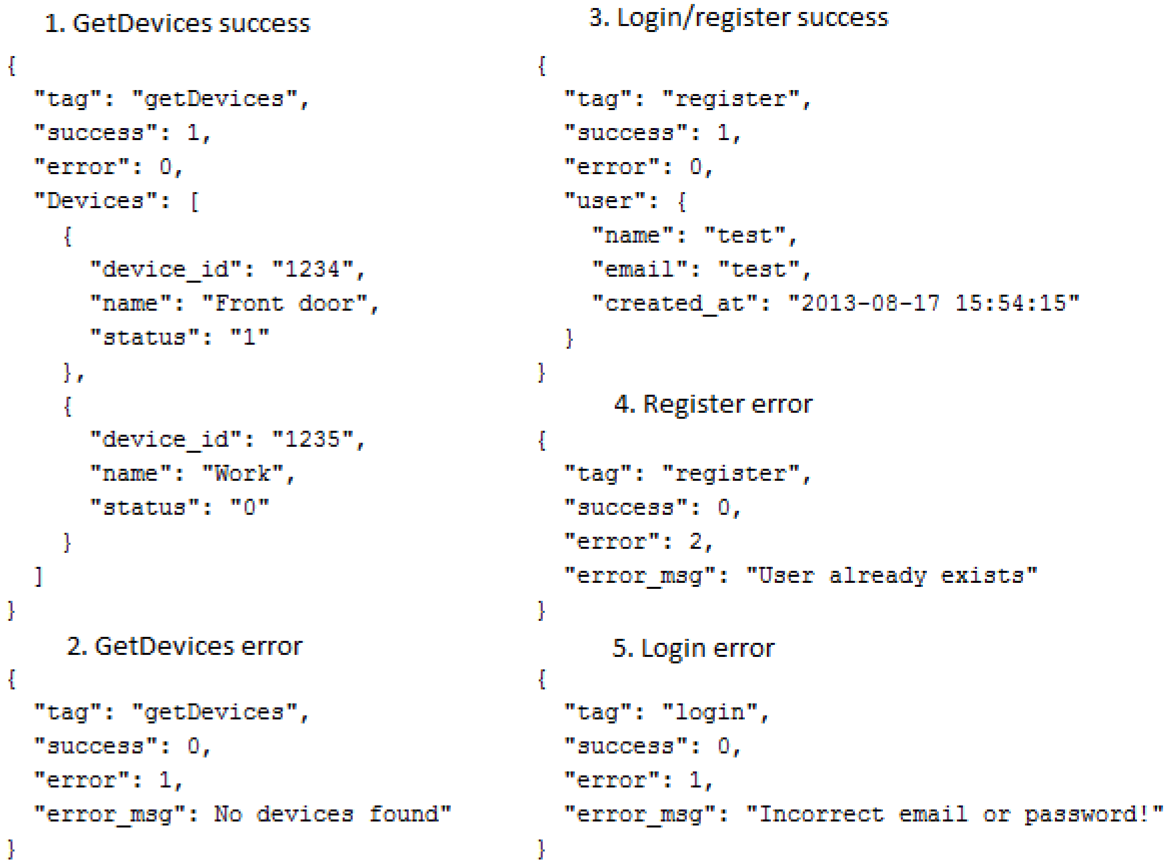


Figure 18: The communication flow between components.

5.2.1 Server and database

The server handles communication to and from both the client and the device applications. The software is split up into smaller parts, PHP scripts, where each part take care of one particular task. These tasks are serving the user, serving the device and managing the database.

The clients and devices communicate with the server by sending HTTP POST messages to it. These messages are retrieved as an array from PHPs $_POST variable. The message contains a tag field, so that the server knows how to process the message, and fields for other information needed. For the client service the tag can be login, register, addWork and getDevices and the other fields can be name, email and password for login/register/getDevices and additionally deviceId and work for the tag addWork. The device service for now only supports the tag getWork which, in addition to the device id and password, checks if a user has used the application to toggle the device.



1. GetDevices success

```
{
  "tag": "getDevices",
  "success": 1,
  "error": 0,
  "Devices": [
    {
      "device_id": "1234",
      "name": "Front door",
      "status": "1"
    },
    {
      "device_id": "1235",
      "name": "Work",
      "status": "0"
    }
  ]
}
```

2. GetDevices error

```
{
  "tag": "getDevices",
  "success": 0,
  "error": 1,
  "error_msg": No devices found"
}
```

3. Login/register success

```
{
  "tag": "register",
  "success": 1,
  "error": 0,
  "user": {
    "name": "test",
    "email": "test",
    "created_at": "2013-08-17 15:54:15"
  }
}
```

4. Register error

```
{
  "tag": "register",
  "success": 0,
  "error": 2,
  "error_msg": "User already exists"
}
```

5. Login error

```
{
  "tag": "login",
  "success": 0,
  "error": 1,
  "error_msg": "Incorrect email or password!"
}
```

Figure 19: Different JSON responses from the server.

The response from the server is a JSON encoded message containing the tag the request came with and the response which varies based on the tag. Examples of these responses can be seen in figure 19. The response for getDevices is a list of devices the user has access to and shows for each device its id, name and status (on/off, locked/unlocked etc.)(1) or an error message (2). Login and register returns the account details (3) or an error message explaining what went wrong e.g. name already exists (4), wrong password (5).

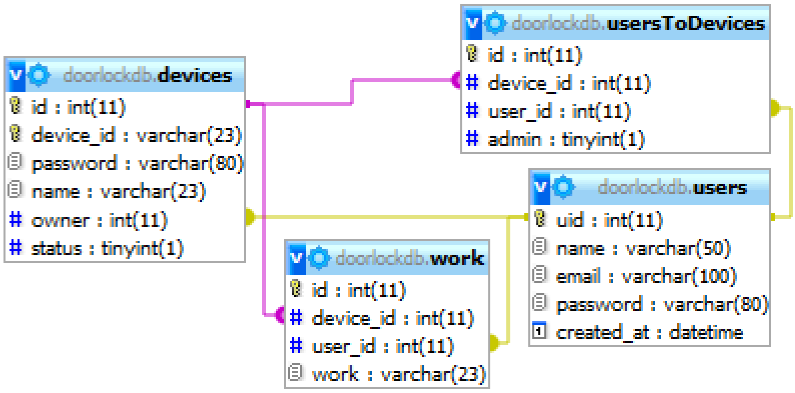


Figure 20: A graphical representation of the database tables and their relations.

The database consists of four tables: users, devices, a relation between users and devices, and work, as can be seen in figure 20. The users table contains an id used internally on the server, login name, email, password and a timestamp when the account was created. Similarly the device table contains id, device_id (serial number), password, a user set name/description, status of the device and id of the owner. The relation table maps users to devices so that each user can have access to multiple devices and a device can have multiple users. The work table contains the device_id for the device the work is to be performed on, the user_id of the user who initiated the work, and the work to be performed (e.g. lock/unlock door).

The passwords are hashed before storing in the database to increase security. A hash is the result of a one-way function that maps data of variable length to data of a fixed length. This ensures that even if someone was to see the data in the database, no passwords would be exposed. The hashing is done by passing the password, along with a “salt”, to the crypt() function in PHP. The salt is a long random sequence of characters to help make weak (short) passwords longer and thus harder to break. It is then hashed with the Blowfish algorithm [67].

5.2.2 Android application

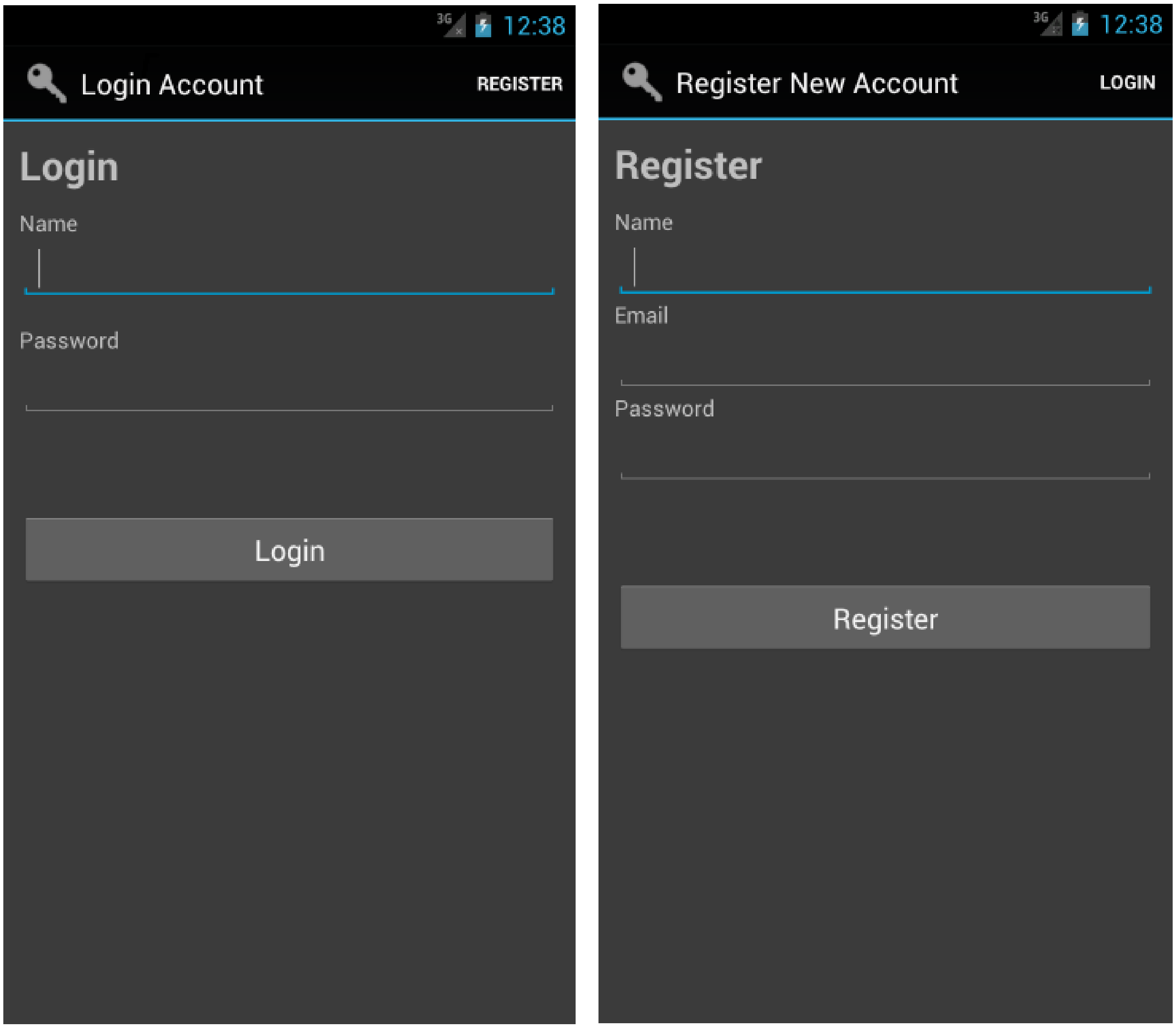


Figure 21: The login and register screens.

When the user opens the app for the very first time a login screen, as shown in figure 21, is presented. Here the user can choose to login or press the register button to get to the registration screen. At the top of the application is a UI element called ActionBar [68]. The ActionBar shows the application icon, where in the application the user is and buttons to interact with the current view.

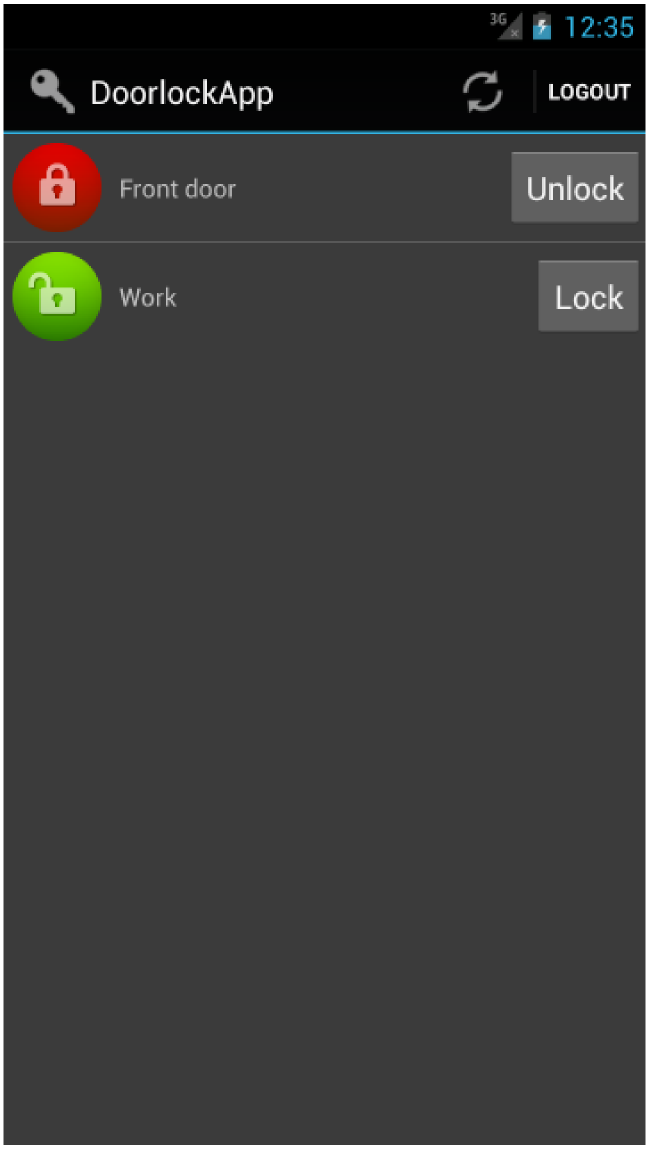


Figure 22: The device list.

Upon logging in or successfully registering, the user details are stored in the built in SQLite database of the application and the user stays logged in until the logout button is used. Logging out clears the data in the database so that the user must log in again, and is thus presented with the login screen. All database interactions are handled by a database handler class. Once logged in, the user is presented with a list of devices to interact with (Figure 22). The list shows the name and status of the device, as well as a button to lock or unlock the door. The ActionBar in this view has a refresh button to update the list of devices and a button to logout of the application and return to the login screen.

When a button is pressed, the application sends a HTTP POST request to the server via a Java class that handles the network connection. The request contains a tag field that corresponds to the action the user made and other information, such as login credentials and lock id. The servers response is parsed into a JSON Java object and is passed back to the function that did the request. If the request was successful the application will change view accordingly, that is, go from login/register screen to the device list, or update the device list if the update/lock/unlock buttons are pressed. If the request resulted in an error message, then that is displayed on the screen (not pictured).

Each view has its own user interface (UI) layout in a XML (Extensible Markup Language) file and an activity class which is written in Java. The XML file contains all the details about the UI elements such as what type they are (textview, text input, buttons), their position and text within the elements. The Java files contain the logic code for the view such as what happens when a button is pressed.

5.2.2 Raspberry Pi

The software on the Raspberry Pi checks with the server at a set time (hardcoded for now, 10 seconds) interval if there is work for it to do by sending a HTTP POST request. The JSON response from the server is either an error message indicating what went wrong or the actual work for the device to perform. The work can be locking or unlocking the door. Once the Raspberry Pi has determined what it needs to do, it will run the corresponding subroutine for either locking or unlocking. The subroutines send a timed signal to the correct GPIO pin that in turn tells the motor circuit to turn the motor. The timing of the signal was measured so that the motor has enough time to completely lock/unlock the door lock.

The interval at which the Pi checks with the server was chosen to be 10 seconds long. The shorter the interval, the faster the door would open but also increase the amount of bandwidth and computation needed for the server. With a longer interval, the server saves on these resources but it takes longer for the lock to open. Since the user can unlock the door while approaching it and considering how long it takes to unlock manually with a key, it was deemed that the interval time is not too long.

5.3 Comparison

The following table compares the features of the planned product with the planned prototype and the final prototype.

Feature	Planned product	Planned prototype	Final prototype
Raspberry Pi as central unit, wireless communication with lock	X		
Raspberry Pi controls lock directly		X	X
Lock is controlled by mobile application	X	X	X
GPS/Wifi on phone detects if user is near door to unlock, locks when out of range	X		
Giving other people access	X	X	
Register/login with application	X	X	X
Give time limited guest access	X	X	
Log of actions on server	X	X	
Get list of devices with statuses	X	X	X
Interact with devices	X	X	X
DC-motor in lock	Or stepper motor	X	X
Sensor to detect if door is locked or not	X	X	

Due to time constraints, not all the planned prototype features were finished. Though the most basic functionalities were completed such that the prototype can perform the most basic actions which is locking/unlocking a door.

6. Conclusion

Throughout the project many problems arose that could not be solved during the designated time period. Two major issues relate to the ability to detect whether the door is locked/unlocked. These issues arise either when using a key or using the turning knob. Such an action is independent of the motor, therefore the system is unable to recognize that the door is unlocked if done with a key. Another case would be that the maximum rotation of the locking pin, where the time interval specified may not be enough to lock/unlock the door using the motor. These issues could be solved with the use a sensor, but as mentioned earlier, this was not possible throughout the project.

Since the Raspberry Pi was used to control the motor directly, some theoretical problems were not answered as well. These theoretical problems relate to using the Raspberry Pi as a central unit and having it communicate with the door wirelessly. As stated in section 3.7, there are many available technologies that could be used for the control of the door lock wirelessly, but they pose more questions. How many home appliances can use them? How much cost would they add to the entire system? What is their transmission radius? Are they secure?

With more time and resources, it is possible to solve the problems encountered throughout this project turning the prototype into an actual product. The final conclusion would be that the prototype is functional, yet requires more work to complete all the functionalities that would be required of a commercial product. The team truly believes that home automation is the next big step in the lives of consumers. The technology is available, most homes have a WiFi service, most consumers have smartphones. What is left is creating a unified home automation system where the home appliances are all connected allowing the homeowner to control every aspect of their functions.

7. Future Work

To ensure that the prototype created during this project can achieve its maximum potential, there are a number of improvements and changes that can be implemented. Also, the problems encountered throughout this project should be addressed.

Foremost, the most prominent would be the Raspberry Pi working as a central unit connecting the consumer to devices in his/her home. Therefore looking into a two way communication between the Raspberry Pi and the device. For the Raspberry Pi, that would be the use of Z-wave putting into account the low price of the attachment module and its compatibility with home appliances allowing the control of them wirelessly. This solution also discards the power management problem since the Raspberry Pi would then be powered from an electricity outlet.

A problem which occurred was the locking detection when manually locking/unlocking the door. The solution to this problem was the use of a sensor but that was not possible, and is explained in section 5.1.5. Therefore anyone attempting to continue this project should spend some time researching the use of a wireless sensor sending information back to the Raspberry Pi directly. This would require choosing a very small microprocessor using a two way transceiver powered by a battery capable of powering these devices for a long period of time.

The prototype can offer more diverse ways of locking/unlocking the door by using other technologies. For example, Bluetooth LE can be used such that the door unlocks when the consumer is close to the door. The door can be unlocked through voice recognition, or using a door pinhole camera with facial recognition abilities. The phone application could use GPS or detect when the user connects or disconnects from the home Wifi network to unlock or lock the door.

Some of the proposed features of the software could not be implemented in time. One of these features was having guest accounts with a limited time access to a lock and a user interface in the application to manage this. Another feature was logging of all user actions on the server side and a way to display it to the user either in the app or via a website. The application also needs to have a way to add a device to the users list of devices and owners of locks need to be able to manage who else has access to their device.

References

[1] Apigy Inc. URL: https://lockitron.com/preorder. Retrieved: 2013-06-03

[2] Bluetooth SIG, Inc. “Smat Devices” URL: http://www.Bluetooth.com/Pages/Bluetooth-smart-devices.aspx. Retrieved: 2013-06-10

[3] Apigy Inc. URL: https://api.lockitron.com/. Retrieved: 2013-06- 04

[4] Apigy Inc. “Lockitron - Keyless entry using your phone”. URL: https://www.youtube.com/watch?feature=player_embedded&v=D1L3o88GKew. Retrieved: 2013-06-06

[5] Apigi Inc. “One day left to reserve lockitron for 149$”. URL: http://blog.lockitron.com/post/34832752186/one-day-left-to-reserve-lockitron-for-149. Retrieved: 2013-06- 03

[6] Apigi Inc. “Under the hood with electric imp”. URL: http://blog.lockitron.com/post/45921265622/under-the-hood-with-electric-imp. Retrieved: 2013-06-04

[7] Kwikset. URL: http://www.kwikset.com/Kevo/default.aspx. Retrieved: 2013-06-05

[8] Margaret Mouse. URL: http://searchsecurity.techtarget.com/definition/key-fob. Retrieved: 2013-06-05

[9] August. URL: http://www.august.com/. Retrieved: 2013-06- 05

[10] Peter Ha. “Here’s your smart lock of the future, today”. URL: http://gizmodo.com/so-its-not-connected-to-any-wifi-router-how-does-it-s-510438338. Retrieved: 2013-06- 06

[11] Bielet Inc. URL: http://www.gojiaccess.com. Retrieved: 2013-06-05

[12] Bielet Inc. URL: http://www.gojiaccess.com/faq.html#door. Retrieved: 2013-06-05

[13] Bluetooth SIG, INC. “A look at the basics of bluetooth wireless technology”. URL: http://www.Bluetooth.com/Pages/basics.aspx. Retrieved: 2013-06- 07

[14] Margaret Mouse. “Frequency-hopping spread spectrum”. URL: http://searchnetworking.techtarget.com/definition/frequency-hopping-spread-spectrum. Retrieved: 2013-06-17

[15] Bluetooth SIG,Inc. “Bluetooth 4.0 with low energy technology paves the way for Bluetooth Smart devices”. URL: http://www.Bluetooth.com/Pages/low-energy.aspx. Retrieved: 2013-06-10

[16] Joe Decuir. “Bluetooth 4.0 : Low Energy”. URL: http://chapters.comsoc.org/vancouver/BTLER3.pdf Retrieved: 2013-06-11

[17] Brian Bennett. “The power of Bluetooth 4.0: It’l change your life”. URL: http://news.cnet.com/8301-1035_3-57389687-94/the-power-of-Bluetooth-4.0-itll-change-you r-life/ . Retrieved: 2013-06-11

[18] Bluetooth SIG,Inc. “Introducing Bluetooth Smart marks”. URL: http://www.Bluetooth.com/Pages/Smart-Logos-FAQ.aspx. Retrieved: 2013-06-11

[19] Arduino. URL: http://arduino.cc/en/. Retrieved: 2013-06-11

[20] Arduino. “Arduino Uno”. URL: http://arduino.cc/en/Main/ArduinoBoardUno. Retrieved: 2013-06-11

[21] Arduino. URL: http://arduino.cc/en/Main/Products. Retrieved: 2013-06-11

[22] Arduino. “Arduino Mega 2560”. URL: http://arduino.cc/en/Main/ArduinoBoardMega2560. Retrieved: 2013-06-11

[23] Arduino. “Arduino Lilypad simple”. URL: http://arduino.cc/en/Main/ArduinoBoardLilyPadSimple. Retrieved: 2013-06-11

[24] Arduino. “Shields”. URL: http://arduino.cc/en/Main/ArduinoShields. Retrieved: 2013-06-11
[25] Jonathan Oxer. ”Arduino Shield List”. URL: http://shieldlist.org/. Retrieved: 2013-06-11
[26] Arduino. URL: http://www.arduino.cc/en/Guide/Introduction. Retrieved: 2013-06-11
[27] Arduino. “Arduino Build Process”. URL: http://arduino.cc/en/Hacking/BuildProcess. Retrieved: 2013-06-11
[28] August. URL: http://www.august.com/faq.html. Retrieved: 2013-06-10
[29] Bielet Inc. URL: http://www.gojiaccess.com/faq.html#secure. Retrieved: 2013-06-10
[30] Peter Ha. “Are smart locks secure or just dumb?”. URL: http://gizmodo.com/are-smart-locks-secure-or-just-dumb-511093690. Retrieved: 2013-06-05
[31] Unikey Technologies, Inc. URL: http://www.unikey.com/#collapseNine. Retrieved: 2013-06-10
[32] Apigy Inc. URL: https://lockitron.com/help/security. Retrieved: 2013-06-10
[33] Juha T. Vainio. “Bluetooth Security”. URL: http://www.yuuhaw.com/bluesec.pdf. Retrieved: 2013-06-12
[34] Jogn Padgette, Karen Scarfone. “Guide to Bluetooth Security”. URL: http://csrc.nist.gov/publications/drafts/800-121r1/Draft-SP800-121_Rev1.pdf. Retrieved: 2013-06-12
[35] Raspberry Pi Foundation. URL: http://www.raspberrypi.org/faqs . Retrieved: 2013-06-14
[36] eLinux.org. URL: http://elinux.org/RPi_Low-level_peripherals#Introduction. Retrieved: 2013-06-14
[37] IDC. “Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains”. URL: http://www.idc.com/getdoc.jsp?containerId=prUS24257413. Retrieved: 2013-08-15
[38] Dana Wollman. “There have been 900 million Android activations, 28 billion app installs to date”. URL: http://www.engadget.com/2013/05/15/900-million-android-activations/. Retrieved: 2013-08-15
[39] Michael Gorman. “Apple: over 500 million iOS devices sold”. URL: http://www.engadget.com/2013/01/23/apple-over-500-million-ios-devices-sold/. Retrieved: 2013-08-15
[40] Apple Inc. URL: https://developer.apple.com/programs/which-program/. Retrieved: 2013-08-15
[41] Google inc. “Developer Registration”. URL: https://support.google.com/googleplay/android-developer/answer/113468?hl=en. Retrieved: 2013-08-15
[42] Spectrum Brands, Inc. “How locks work”. URL: http://www.kwikset.com/How-To-Choose/How-Locks-Work.aspx. Retrieved: 2013-06-13
[43] Carolyn. “Deadbolt Lock - How deadbolts work”. URL: http://www.familyhomesecurity.com/deadbolt-lock-how-it-works/. Retrieved: 2013-06-13
[44] April Sanders. “How does a door lock work”. URL: http://www.ehow.com/how-does_4596951_door-lock-work.html. Retrieved: 2013-06-14
[45] Marshall Brain, Tom Harris. “How lock picking works”. URL: http://home.howstuffworks.com/home-improvement/household-safety/security/lock-picking.htm. Retrieved: 2013-06-14
[46] Gears and stuff. “Types of gears”. URL: http://www.gearsandstuff.com/types_of_gears.htm. Retrieved: 2013-08-19

[47] K&J Magnetics, inc. “Reed switches and hall effect sensors”. URL: http://www.kjmagnetics.com/blog.asp?p=reed-switches-and-hall-effect-sensors. Retrieved: 2013-08-14
[48] Honeywell. “Hall effect sensing and application”. URL: http://sensing.honeywell.com/index.php?ci_id=47847. Retrieved: 2013-08-14
[49] Wells Vehicle Electronics. “Understanding hall effect sensors”. URL: http://www.wellsve.com/sft503/Counterpoint3_1.pdf. Retrieved: 2013-08-14
[50] ZigBee Alliance. “ZigBee Technology”. URL: http://www.zigbee.org/About/AboutTechnology/ZigBeeTechnology.aspx. Retrieved: 2013-08-15
[51] Digi International Inc. “ZigBee Wireless Standard”. URL: http://www.digi.com/technology/rf-articles/wireless-zigbee. Retrieved: 2013-08-15
[52] Z-Wave Alliance. “About Z-Wave”. URL: http://www.z-wave.com/modules/AboutZ-Wave/. Retrieved: 2013-08-16
[53] Ron Fritz. “What is Z-Wave?”. URL: http://compnetworking.about.com/od/homeautomationzwave/a/what-is-zwave.htm. Retrieved: 2013-08-16
[54] Cooking Hacks. “Raspberry Pi to Arduino shields connection bridge“. URL: http://www.cooking-hacks.com/index.php/documentation/tutorials/raspberry-pi-to-arduino-shields-connection-bridge. Retrieved: 2013-08-17
[55] Z-Wave.Me. “RaZberry Project”. URL: http://razberry.z-wave.me/index.php?id=1. Retrieved: 2013-08-17
[56] T. Dierks, E. Rescorla. “The Transport Layer Security (TLS) Protocol“. URL: http://tools.ietf.org/html/rfc5246. Retrieved: 2013-08-17
[57] ZigBee Alliance. “ZigBee 2012 Specification”. URL: https://docs.zigbee.org/zigbee-docs/dcn/07-5299.pdf. Retrieved: 2013-08-17
[58] Z-Wave Alliance. “About Z-Wave Technology”. URL: http://www.z-wavealliance.org/technology. Retrieved: 2013-08-17
[59] Centers for Medicare & Medicaid Services Office of Information Service. “Selecting a development approach”. URL: http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf. Retrieved: 2013-08-20.
[60] Teach-ICT.com. “Evolutionary Prototyping”. URL: http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/prototyping_RAD/miniweb/pg3.htm. Retrieved: 2013-08-20
[61] Dale Dougherty. “LAMP: The open source web platform”. URL: http://www.onlamp.com/pub/a/onlamp/2001/01/25/lamp.html. Retrieved: 2013-08-18
[62] phpMyAdmin. URL: http://www.phpmyadmin.net/home_page/index.php. Retrieved: 2013-08-18
[63] The Eclipse Foundation. URL: http://www.eclipse.org/. Retrieved: 2013-08-18
[64] Android Developers. “ADT Plugin”. URL: http://developer.android.com/tools/sdk/eclipse-adt.html. Retrieved: 2013-08-18
[65] Python Software Foundation. “RPi.GPIO 0.5.3a”. URL: https://pypi.python.org/pypi/RPi.GPIO. Retrieved: 2013-08-19

[66] Dave Akerman. “Running The Raspberry Pi on Batteries”. URL: http://www.daveakerman.com/?page_id=1294. Retrieved: 2013-08-20
[67] Bruce Schneier. “The Blowfish encryption algorithm”. URL: https://www.schneier.com/blowfish.html. Retrieved: 2013-08-20
[68] Android Developers. “Action Bar”. URL: http://developer.android.com/design/patterns/actionbar.html. Retrieved: 2013-08-21

Image references

[P1] Apigy Inc. URL: https://lockitron.com/press-kit/lockitron-unit.png. Retrieved: 2013-06-05
[P2] UniKey Technologies Inc. URL: http://www.unikey.com/press-room/assets/media/kevo-center-lowres.png. Retrieved: 2013-06-05
[P3] August. URL: http://e69d0927611610ce197b-32b77afb4c365d6c74dcde82c2755256.r14.cf1.rackcdn.com/gallery3.jpg. Retrieved: 2013-06-05
[P4] Bielet Inc. URL: http://www.gojiaccess.com/images/side-view.jpg. Retrieved: 2013-06-05
[P5] Arduino. URL: http://arduino.cc/en/uploads/Main/ArduinoUno_R3_Front_450px.jpg. Retrieved: 2013-06-8
[P6] Raspberry Pi Foundation. URL: http://www.raspberrypi.org/wp-content/uploads/2011/07/7513051848_9a6ef2feb8_o.jpeg. Retrieved: 2013-06-14
[P7] Spectrum Brands, Inc. URL: http://www.kwikset.com/Images/Pin-and-Tumbler-cylinder.jpg. Retrieved: 2013-06-13
[P8] LockPickerNetwork. URL:http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Pin_tumbler_unlocked.png/250px-Pin_tumbler_unlocked.png. Retrieved: 2013-06-13
[P9] Oliver Hunt. URL: http://www.hvlabs.com/Images/hbridgefund.jpg. Retrieved: 2013-08- 19