6-13-2014

DigiKey

Maziar Arjomand
Santa Clara University

Michael Hirabayashi
Santa Clara University

Tejender Singh
Santa Clara University

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Date: June 13, 2014

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Maziar Arjomand
Michael Hirabayashi
Tejender Singh

ENTITLED

DigiKey

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Thesis Advisor: Dr. Yi Fang

Department Chair: Dr. Nam Ling

DigiKey

by

Maziar Arjomand
Michael Hirabayashi
Tejender Singh

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 13, 2014

DigiKey

Maziar Arjomand
Michael Hirabayashi
Tejender Singh

Department of Computer Engineering
Santa Clara University
June 13, 2014

ABSTRACT

With the vast majority of university students carrying smartphones, more pieces of technology should take advantage of its availability. Currently, most lock and key systems are outdated and primitive. Our team is proposing a solution to this problem that makes use of the technology that we all carry in our pockets daily. Our solution, DigiKey, will improve security and ease of use along with adding on additional features to the current system with the use of smartphones as digital keys. DigiKey will utilize a database of digital keys that can be used from a smartphone to unlock a DigiKey lock over a bluetooth connection.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Metal locks and keys have been around for thousands of years. In todays world of advancing technology, home security and personal privacy are critical aspects of our society that should be moving forward as well. Specifically, on university campuses, temporary keys are given to students each year. These temporary keys can be troublesome for both the college and the student. Keys can be lost, stolen, or deactivated if the university uses key cards. With new and improving technology, personal security and ease of access of locking systems should also be improving. Currently, opening a lock requires the use of a physical key. These physical keys can either be in the traditional metal form or key cards. In both cases, unlocking doors requires the user to carry an additional item on hand. These physical keys are easily misplaced or forgotten about. Physical space also becomes a concern once multiple keys need to be carried. Additionally, since the keys are physical, it is difficult to make copies or get replacements. Furthermore, if keys are stolen or lost, the security of the lock is compromised and it is often necessary to replace the lock that they key was associated with. Current lock and key systems are insecure, inconvenient, and outdated. Around the world, lock and key systems still use outdated and flawed technology.

1.2 Background

The current security implementations of colleges involve key cards or physical keys are insecure, difficult to manage and keep track of, and a hassle for users. In our project, we update college campus door lock mechanisms to modern standards of security and usability through the use of digital keys hosted on personal smartphones. There are two popular college campus locking solutions in use; physical keys, and key cards. Physical keys are a very old method of key/lock interactions that can easily be broken into with a lockpick or a duplicated key. It also involves handing out physical keys to every student or faculty member, which can easily be lost resulting in financial losses as the college has to cut out new keys. The other popular solution, key cards, has the same issue with physical keys as they can also easily be lost or damaged. Faculty often does not give out physical keys to students due to lack of trust or the mere fact that they themselves need them. This lack of distribution results in limited usage of campus resources by the students. Our solution fixes the problems with current key/lock implementations and adds additional features to improve the usability by both students and college management. Our solution is for the users to use their personal smartphones as digital keys to unlock rooms. Since users will be using their personal devices as keys, the college only needs to distribute software, eliminating the cost of creating and replacing keys for each user. Users are also less likely to lose their smartphone than they are to lose a small key. Since the interactions between the key and lock are on smart devices, the system is also able to provide improved control and monitoring to the college campus and its users. The unlocking mechanism will use encrypted rsa key handshaking, which will be much more difficult to crack than a simple key making the lock much more secure. Users can view logs of when their primary room was accessed and by whom. This feature improves feelings of privacy and control over the users secure space. Users can restrict access to users with a lower security role from their rooms at certain times and feel more secure with their valuables and their room. Colleges are still able to have an administrative override and are also able to gain valuable analytics about users traffic in and out of rooms. Our solution improves the security and privacy provided by current solutions, and adds additional features that benefit both the users and college campus.

1.3 Requirement Analysis

The team and the advisor determined the requirements for smart locking system. The requirements for the project will guide the team in development of the smart locking system throughout the senior design.

The final system must meet the following functional requirements:

- Users/Administration should be able to manage the keys access policy
- Users can unlock their respective room
- Users can only unlock doors for which they have a valid key
- Users can access a door in the case of a lost key

The final system should meet the following non-functional requirements:

- Intuitive and easy to use system for users
- Keys must be secure
- Ability to unlock and lock the door must be reliable
- Process of unlocking a door must be fast

Chapter 2

Use Cases

2.1 Unlocking the door

Actor: Student, Staff, or Faculty

Goal: To successfully unlock a door with a smartphone application

Preconditions:

- User must be logged into the mobile application
- User must be near the door lock

Postconditions:

- User unlocks the door.

Scenario:

1. User unlocks phone
2. User opens the DigiKey App
3. User selects the appropriate key.
4. User touches the Open button.
5. The door opens.

Exceptions:

- User does not have login credentials
  - Warning is displayed on the smartphone
  - Login screen is refreshed

2.2 Reviewing Logs

Actor: Student, Staff, or Faculty

Goal: To successfully review logs for a particular lock.

Preconditions:

- User must be logged into their accont abd have access to the room.
- User has permission to view the logs for the specified room.

Postconditions:

- User can view the logs.

Scenario:

1. User opens the app and selects ”View Logs”
2. Logs are displayed for different rooms sorted by date and time.

2.3 Creating/Sending temporary keys

Actor: Studdent, Staff, or Faculty

Goal: To create a temporary key for another user.

Preconditions:

- User has admin privleges for the specified room.
- User is logged into their account.

Postconditons:

- Second user receives a valid key for the room.

Scenario:

1. User opens app and selects ”Create Key”
2. User enters info about the key and selects another user from contacts.
3. User hits ”send” to push the key to the other user.

Exceptions:

- User does not have administrative privileges.
  - Warning is given.
  - Key is not created.

Chapter 3

Wireframes

The following wireframes are meant to illustrate our functional and nonfunctional requirements into a software prototype. It visualizes the user interface design and sequence of flow for each page in our smartphone application.

In Figure 3.1, the first wireframe shows our landing page when the user login into our application. It contains a large unlock image to make it easier for users to unlock a door. On the top, the tab bar contains key names for all the doors the user have access to. In this homepage, the user can select a key name from tab bar and simply press unlock image to open a door. The second wireframe in Figure 3.1 is a navigation drawer that directs the user to different pages within the application.

In Figure 3.2, the first wireframe contains details about keys linked to a particular user. The second wireframe contains sent keys to other users for temporary access to a particular room. This page contains the list of sent keys user shared with friends, family and guests over time.

In Figure 3.3, the first wireframe allows users to create temporary keys for friends, family and guests. The user can allow access respective room for certain amount of time. In addition, they can add rules to the key for temporary access to other users. The second wireframe shows log history for each room. The user can check details about unlocking history in case the security of door was compromised.

In figure 3.4, the user can customize settings so they can personalize the application.

Figure 3.1: Home Screen/Navigation Drawer Wireframes

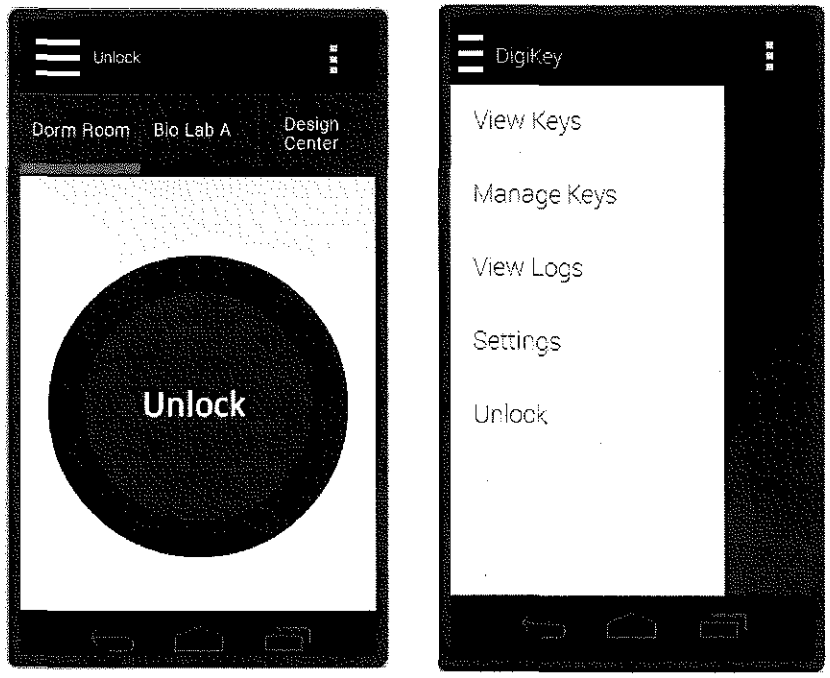


Figure 3.2: Manage Keys Wireframes

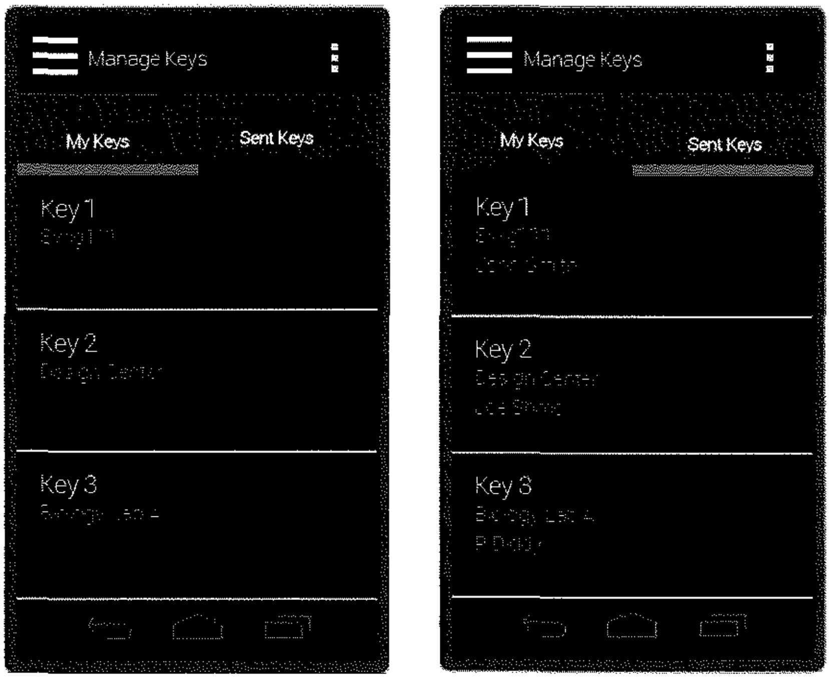


Figure 3.3: Create Key/View Logs Wireframes

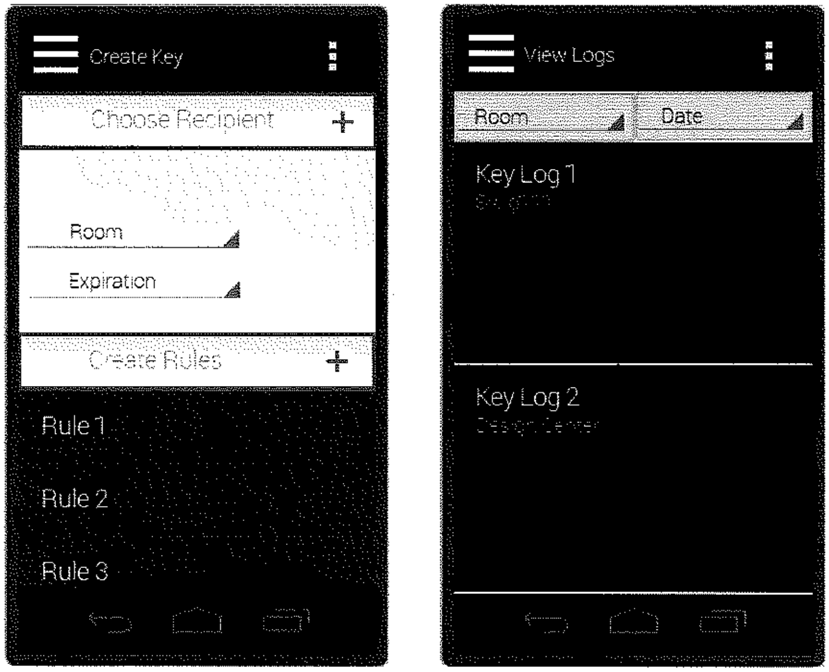
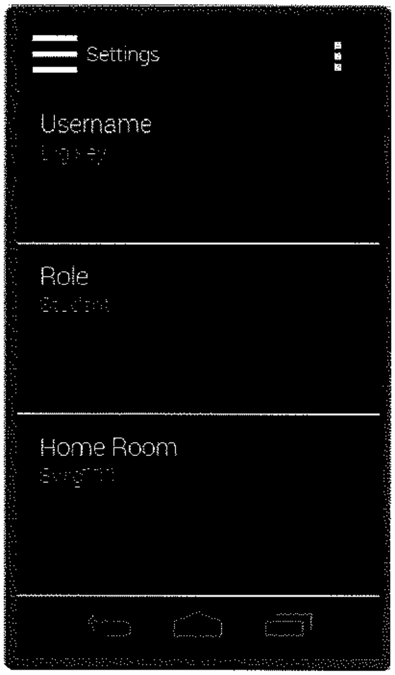


Figure 3.4: Settings Wireframe



Chapter 4

Design Plan

Our Design will utilize 3 main implementation points:

- Intelligent Locking Mechanism
- The Digital Key
- Backend Server Support

Lock

The intelligent locking system will be represented by an electromagnetic lock controlled by a Raspberry Pi. The Raspberry Pi controlling the lock is a small linux system on a chip (SoC). The Raspberry Pi will be running a program that will manage the connections made with Bluetooth devices. The program will accept user information and keys over a Bluetooth connection and determine whether or not to unlock the electromagnetic lock. The electromagnetic lock will be controlled, or turned on and off, by opening and closing the circuit of the electromagnetic lock. The Raspberry Pi will utilize a relay port on the PiFace controller in which the electromagnetic lock will be connected. The relay port will manage whether or not the circuit is open or closed and thus whether or not to provide power to the lock.

Key

The smart locking system will open or unlock based on a valid key being sent over the Bluetooth connection. Keys will be unique for each user and will be validated through the Bluetooth connection to the smart locking system. The digital key will be sent over the Bluetooth connection by a mobile application on Android devices. This application will contain user information as well as all the user's digital keys. The mobile application will store and pull this user information from an online hosted database. User keys will be generated and managed through the use of a backend dashboard.

Backend Support

User keys and other information will be stored in an online database called Parse.com. The database will be easily accessible from both the mobile Android application and our smart locking system through the use of the Parse APIs. Using these APIs, users will be able to change their user information and send the updated information to Parse.com to store. User accounts and their corresponding information can be managed under the Parse.com dashboard.

Chapter 5

Technology Used

In accordance with the specified requirements, the following technologies have been chosen:

Software:

- Android App

  The app will be our client application for interfacing with the lock.

- Android bluetooth libraries.

- Linux Bluetooth Libraries.

  This provides our lock application an interface that we can use to control the bluetooth hardware.

- Python application.

  Python is the programming language we wll use to implement the application that will control the lock and interface with the smartphone applicaiton.

- RSA keys.

  Our project uses RSA keys for secure authentication between the phone and the lock.

- Parse.com

  Parse is a B̈ackend as a Serviceẗhat allows us to host our database and notifications in the cloud.

Hardware:

- Raspberry Pi.

  The Pi is used for controlling the physical lock and hosting our lock application.

- USB Bluetooth Adapter.

  Used to run a bluetooth server on the Raspberry Pi/lock.

- USB Wifi Adapter.

  Provides the Raspberry Pi with network connectivity.

- Android Smartphone

  Used to run the Android application that unlocks the door.

- PiFace controller.

  Provides an interface for the Raspberry Pi that the physical lock can be controlled with.

- Magnetic Lock

  Used for locking/unlocking the door.

Chapter 6

Data Flow Model

In Figure 6.1, the data flow model represents the steps for processing data in each stage of our system. In our system, data flow from smartphone application to lock manager, lock manager to database, database to lock manager, and lock manager to lock. Our smartphone application sends data associated with an account to lock manager. The lock manager parses the data received from smartphone application to create a query for back-end database. The lock manager sends query to request data from database. For example, the lock manager can query data about username Digikey for dorm room. The database then returns requested data to lock manager for decision-making. In this case, it can return key associated with dorm room to verify if the user is authorized. When the lock manager receives all the data, it would compare the data from smartphone application and database to decide whether the user is authorized or not. If the user is authorized, the lock manager request lock to unlock the door. Otherwise the door remains unlocked until correct key is received from smartphone application.

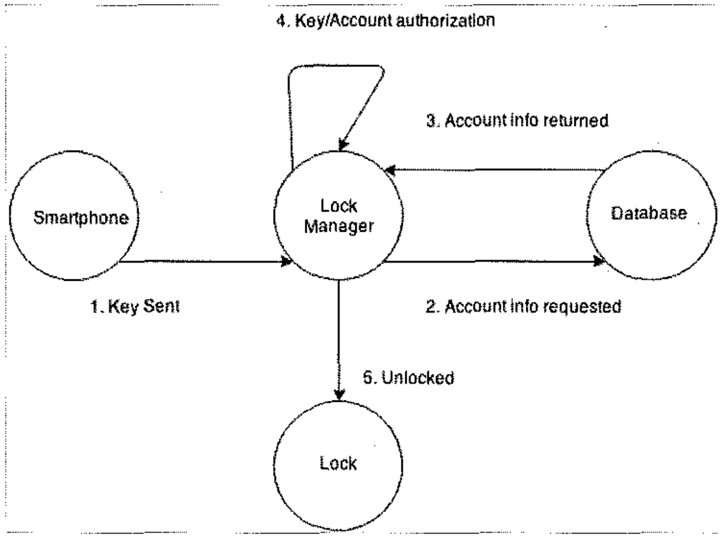


Figure 6.1: Data Flow Model

Chapter 7

Design Rationale

When considering systems for the DigiKey project, our team considered several options for each aspect of the system.

As we designed the system, one of our first considerations was what platform we would use to house the digital key and how to interact with it. We wanted to use a smartphone, as it is an easily accessible piece of technology that is in almost everyones pockets. With 70% of college students using smartphones, we are able to easily distribute our app without requiring additional hardware purchases by our users. We first considered using an iOS app on iPhones to act as the digital key interface for the client side. However, Apples iOS is a smaller share of the mobile user base compared to Android and publishing applications to the Apple App Store is a lengthy process that requires a paid developers license. As a result, we chose the Android platform because it provides us with a larger user base, easier and cheaper distribution, and an environment we are more familiar with because it uses Java.

For the lock aspect of the system, we decided on using a Raspberry Pi to represent the lock. For our development purposes, the Pi gives us a large variety of libraries we used in our design along with several additional parts we could add on to the pi, such as relays and Bluetooth adapters. The Pi gives us a platform that easily allowed for changes in our design later in the process while remaining cheap and easy to develop on.

To communicate between the lock and key, we chose Bluetooth as our networking method because it allows us to easily pair the devices without interrupting other internet connections. One other consideration for this aspect of the system was wifi-direct. However, that would require additional user interaction along with interrupting the existing internet connection. This would result in reduced ease of use, which is a high priority requirement for the system.

We decided to use parse.com, a backend as a service, to host our databases because it is very easy to use and reduces alot of extra work necessary for our project. Instead of having to setup a web hosted database where we needed to create and design database schema, parse.com allowed us to easily register our applications and begin pushing data into tables without any other setup. This decision cut down on the boilerplate code we had to write, and allowed us to focus on the more difficult parts of the project.

In our considerations for designing the system, we placed our priorities on platforms that were cheap, easy to use, and versatile. We wanted to be able to change the design of some aspects of the system as the development process progressed. For example, there might have been an instance where we found out that something we had planned to use turned out to be inaccessible, or we discovered a change in requirements. The Android platform, Raspberry Pi, and parse.com provided us with the flexibility we needed to satisfy our requirements.

Chapter 8

Architectural Diagram

The architectural diagram is meant to illustrate how the technical components of our system interact. Based on the requirements of the project, the following diagram was produced. The user can attempt to unlock the door using smartphone application. The smartphone application uses Bluetooth technology to create a secured network connection with Raspberry Pi. Once the connection is established, the smartphone application can send messages to Raspberry Pi. The message contains user account information, key, and timestamp for security purposes. The Raspberry Pi parses the message and query from back-end database to determine if the user is authorized or not. Once the Raspberry Pi determines if the user is authorized, it would give signal to magnetic lock to open the door. The smartphone application is also connected to our back-end database to pull all the user information when the user attempts to login. Once the user is authenticated, it query data such as user keys, key names, and log history from our back-end database. The smartphone application also query to database to update information about data logs, temporary keys and etc.

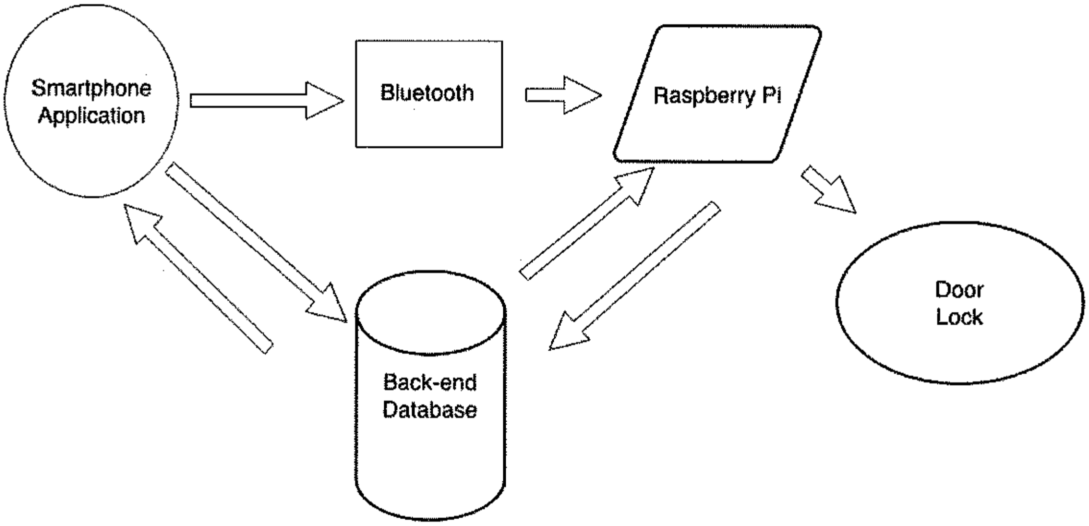


Figure 8.1: Architectural Diagram

Chapter 9

Testing

9.1 Alpha Testers

For our system, our alpha testers would be our team and our advisor. Since most of the team members are involved in the testing phase which is outlined in the development timeline of our project, our advisor was our alpha tester. Our advisor knows our project details closely so he was a great source to get help for alpha testing.

9.2 Beta Testers

For our system, our beta testers are be our friends. Once the system is fully implemented, we asked our close friends to help us test our mobile application and admin desktop dashboard to ensure we catch bug errors. Having people test outside our group gave us better chances of finding those bug errors in our implementation. When a beta tester found a particular bug error, they could notify our team project so we could immediately fix those coding errors. Fixing those coding errors immediately helped our system become more stable and reliable when deployed to a production environment.

Chapter 10

Risk Analysis

Project risks are meant to address situations that could compromise the development cycle of the project. Risks are listed with expected consequences, the probability that they occur, the severity of the risk, its impact (which is calculated by multiplying probability by severity), and ways to address the consequences.

Figure 10.1: Risk Analysis Table

Risk	Consequence	Probability	Severity	Impact	How to Address
Learning curve for technology	- Loss of time - Difficulties in implementation	0.9	6.0	5.4	1- Find efficient learning tools 2 - Learn together as a group
Busy schedule of team members	- Slow progress in development of the project	0.7	4.0	2.8	1. Make a good project timeline 2. Learn time management skills
Changing/Updating Requirements	- Some work needs to be done again - Need more time to develop	0.3	5.0	1.5	1. Make sure to have constant communication within the team to avoid changing requirements.
Team members getting Sick	- Some work not completed - Loss of human resource	0.1	7.0	0.7	1. Other members in the team step-up and work on those incomplete work
Technical problems(HD crash)	- Risk of losing project data	0.001	7	0.007	1. Back-up data in several places

Chapter 11

Development Timeline

In order to properly manage our time and successfully finish the project in time, we are using a Gantt Chart. A Gantt chart lays out the deadlines on a weekly basis and each individuals responsibilities for those deadlines.

Figure 11.1: Fall Development Timeline

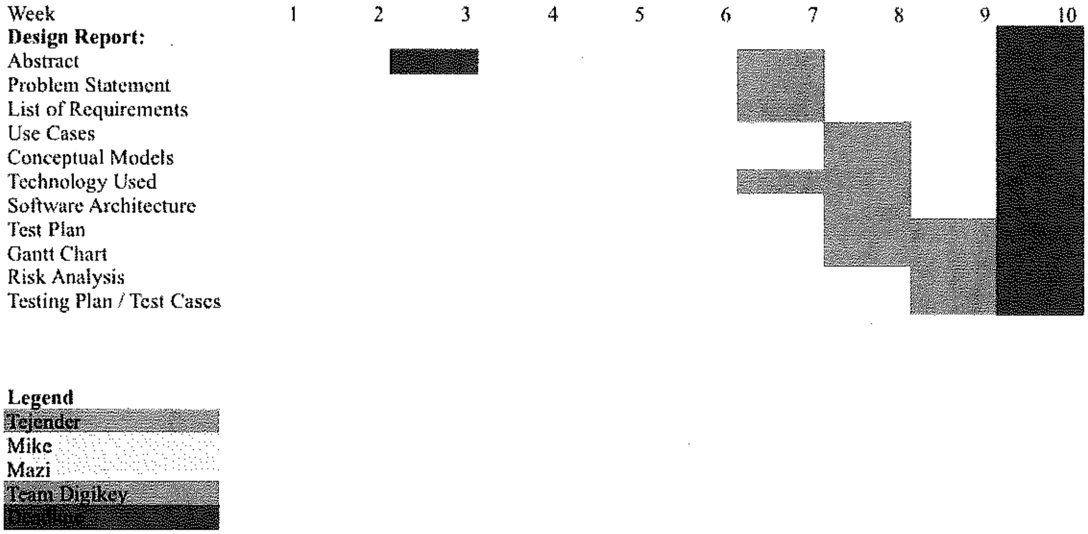


Figure 11.2: Winter Development Timeline

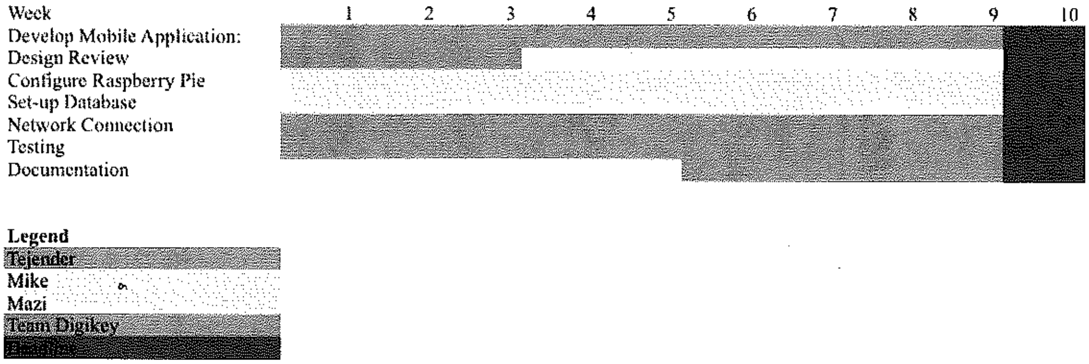
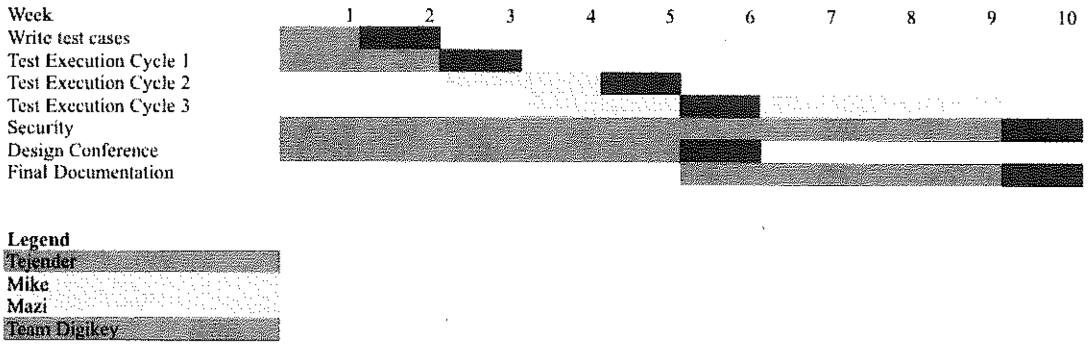


Figure 11.3: Spring Winter Timeline



Chapter 12

Societal Issues

12.1 Ethical

Our project does not create any new dangerous technologies. Instead, our system is using current technologies in a manner that helps create a more secure environment for both the user and admin.

12.2 Social

As a team, we believe our system will have a positive impact on the general public. We are striving to make peoples everyday lives more simple and secure. Our system also is easily incorporated into the public since it is a downloadable mobile application.

12.3 Political

Our team does not believe that our system will have significant political impact as it provides benefits for personal safety and convenience.

12.4 Economic

The mobile application should save Universities or administrators money because it eliminates the use of key cards, card readers, and card writers.

12.5 Health and Safety

Since our project is a security system, it is crucial that our system has necessary security safeguards. Customers will be using our system to protect their belongings and will be expecting our system to work flawlessly.

12.6 Manufacturability

The main component of our system is a mobile application which can be easily distributed. The Raspberry Pi and the PiFace to move the lock are the only parts of our system that would have to be manufactured.

12.7 Sustainability

Since our system is centered around a mobile application, the system is very sustainable. If changes are made to the system, the updates can easily be downloaded without wasting any resources.

Chapter 13

Conclusion

The DigiKey mobile application on the Android platform is an excellent system for personal security. The ability to lock and unlock doors wirelessly while maintaining a higher level security than traditional key and lock systems makes customers lives easier and more secure. Using the Raspberry Pi and a backend as a service allows for a cheap and reliable system that is flexible on the administrative side.

Our project offers multiple advantages over traditional locking systems. DigiKey allows users to easily manage keys with nothing more than a smartphone. Furthermore, using a smartphone application to manage keys provides additional features over physical lock and keys. These additional features include ease of access, organized keys, and increased security. Since our project is account based, users can also access their keys from multiple devices. This access provides users with back up methods for using their keys. The system we designed also allows for user roles with varying degrees of access. These user roles designates keys access policies for each user giving more control and security its users.

Although our project is very stable, reliable and fast, it still has three disadvantages that can affect user experience. Firstly, the system would not function properly if there were no internet service available for Raspberry Pi. Secondly, the user needs to get access to another device in case their mobile application runs out of battery or is stolen. Thirdly, the smartphone device needs Bluetooth available constantly therefore, and cannot have another Bluetooth device connected to it.

Our project holds strong potential for future work because the team can add more functionality to it and eliminate all the disadvantages listed above. At the moment, our project can perform our main function of unlocking a door. For future work, the team can add sharing keys, sending alert messages, and viewing logs, as well as extending this project to other use cases. A sharing keys feature would allow users to give temporary access to friends, family or guests for given amount of time. Sending alert messages would notify the user when friends, family or guests try to access the door. Viewing logs would give the user a dashboard to view all the data about unlocking activities. These logs would allow user to track the date and time when the door was unlocked.

The project is not only limited to college campuses but also can be extended to hotels. Hotels can take advantage of our project to create an interface to allow customers to check in and check out using our smartphone application. For example, the customer can check-in and can get access to their room with the smartphone application. The customer can check in and check out anytime without contacting the hotel, and could potentially. This would streamline the process of booking a room in hotel and could potentially improve the user experience for the customers.