DEVELOPMENT OF MOBILE APP FOR MONITORING STAIR CLIMBING ACTIVITIES IN A BUILDING

TAN CHEE YANG

A project report submitted in partial fulfilment of the requirement for the award of Bachelor of Engineering (Hons.) Electrical and Electronic Engineering

Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

May 2016

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : __________________________

Name : Tan Chee Yang

ID No. : 11UEB04737

Date : 12/5/2016

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"DEVELOPMENT OF MOBILE APP FOR MONITORING STAIR CLIMBING ACTIVITIES IN A BUILDING"** was prepared by **TAN CHEE YANG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ____________________

Supervisor : Chua Kein Huat

Date : ____________________

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. First and foremost, I have to thank my project supervisor, Mr. Chua Kein Huat. Without his assistance and dedicated involvement in every step throughout the process, this project would have never been accomplished.

Other than that, I would like to thank my project moderator, Dr. Lo Fook Loong for his advice and guidance to improve the project.

I am also grateful to Mr. Ooi Ean Huat, lecturer from Department of Internet Engineering and Computer Science for sharing his expertise and valuable guidance

I would like to thank my parents for the unceasing encouragement, support and attention. I would also like to express my gratitude to my friends who had helped and given me encouragement throughout this venture.

DEVELOPMENT OF MOBILE APP FOR MONITORING STAIR CLIMBING ACTIVITIES IN A BUILDING

ABSTRACT

Climbing the stairs is one of the best exercises that can be done anywhere and anytime. Many health organizations recommend the public to take the stairs instead of waiting for the lifts. This is because there are many health benefits, such as improved cardiovascular system and strengthened muscles and joints, by taking the stairs. In this busy era, most people seldom spend time on doing exercises or sports. Hence, taking the stairs is the exercise that is suitable for those people who always do not have time to exercise. In this project, a mobile application (app) is developed to monitor the stair climbing activities in a building. "Humans as sensors" is the main idea for this development project. The people in the building are like the sensors that provide data for monitoring. The mobile app has the QR code scanning feature to locate the users' location. Then, a simple calculation is done to calculate the calories burnt and energy saved through the stair trip. Moreover, the users are able to track their stair progress. All the stair trips from the users are then sent to the web server for data collection. The development of the mobile app is done by separating each function into a module. The module is then developed and tested individually before being integrated into a single app. All the data collected from the users are then analysed and interpreted to understand the stair climbing activities. The stair usages reach the peak at noon time. In addition, stair usages from the ground floor are higher compared to other floors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

API	Application Programming Interface
App	Application
AVD	Android Virtual Device
CSV	Comma-Separated Values
GUI	Graphical User Interface
IBM	International Business Machines
ICT	Information and Communication Technology
IDC	International Data Corporation
IDE	Integrated Development Environment
JDK	Java Development Kit
JSON	JavaScript Object Notation
NFC	Near Field Communication
OS	Operating System
QR	Quick Response
RAD	Rapid Application Development
SDK	Software Development Kit
UI	User Interface
UTAR	Universiti Tunku Abdul Rahman

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

According to the Smarter Buildings study by International Business Machines Corporation (IBM), lift is the most time wasting issues in the office buildings (Ann Bednarz, 2010). Based on the study, which surveyed about 6000 office workers in 16 United States cities, the cumulative time spent on waiting for lifts for a period of 12 months is about 92 years and the amount of time stuck in the lifts is about 32 years (IBM, 2010). This is one of the main concerns in achieving higher productivity and efficiency in the workplace as the time wasting in the lifts are too long. Time spent waiting for the lifts is one of the most common and undeniable issues nowadays especially in high-rise buildings. The congestion in the lifts is even more serious during peak hour. Moreover, lifts account for eight percent of the building's energy costs (University of Birmingham, 2011).

The alternative to the lift is the staircase, which is one of the oldest building features in architectural history. Before the lift was invented, the staircase is the only way to access different floors in a building. Staircase can be found in every multiple storeys building. Even though lifts are installed in high-rise buildings, staircases are often built next to the lifts to provide an alternative path. In the case of emergency such as fire, the staircases will come into use to evacuate the people. This is because the lifts are not recommended to use to reduce the risk of people trapped in the lifts. Thus, the staircase is generally seen as the evacuation path during emergency only and people tend to use lifts in daily life instead of the staircase.

A recent research shows that climbing the stairs can slow down the aging of the brain (Nicole Stinson, 2016). The research was conducted by the researchers from Concordia University in Canada and they found that the brain age can be decreased by 7 months, by taking 20 stair steps per day (The Siver Times, 2016). Besides, taking the stairs for about 20 minutes can burn more calories compared to running (Best Health, 2011). There are many health benefits which most of the people are not aware of, such as improved cardiovascular health, strengthened joints and muscles, and high calories burnt. Other than that, as compared to the lift, taking the stairs can save about 15 minutes every day (Shah et al., 2011). Instead of waiting for the lifts, taking the stairs can save more time and in the meantime, it can strengthen and improve overall body health. Thus, this is one of the best exercises that is free and it can be done almost anytime and anywhere.

The idea of "humans as sensors" is to build a social network as a sensor network (Wang et al., 2014). If an individual represents a sensor, then a large group of people can form the sensor network. Thus, the stair data are contributed by the people who use the staircases. Nowadays, due to advances in mobile phone technology, mobile phones or also known as smartphones, are able to perform computation and code identification. Therefore, a mobile application (app) is developed to monitor the stair activities in the building. By taking advantage of code identification feature, the mobile phones are used to detect the Quick Response (QR) code and identify the floor to collect the data of the stair usage. These collected data can be used to analyse the frequency of stair being used and the stair activities pattern later.

The main development tool is the Android Studio, which requires Java programming skill. Besides, this mobile app requires external library for the QR code detection. The mobile app must be able to perform the features such as displaying the stair usage of the user, analysing the stair usage, and social network sharing. Moreover, the mobile app must be able send the stair usage to a web server, so that data analysis can be done later. The app is also designed to be interactive and user friendly. The two blocks building of Universiti Tunku Abdul Rahman (UTAR) Sungai Long Campus, namely KA block and KB block are used to collect the data for this project.

1.2 Objectives

The objectives of this project are as follows:

- To examine a mobile app for monitoring stair climbing activities
- To develop a mobile app for monitoring stair climbing activities
- To evaluate the performance of the developed app

1.3 Project Scopes

The main focus of this project is to develop a mobile app for monitoring stair climbing activities. Studies on the platforms, development environments and existing technology are also conducted before developing the mobile app. In addition, the development of this mobile app requires some basic understanding about the web server hosting. Android Studio is the main development tool in this project, which requires Java programming skill. The mobile app is to have user authentication to verify the users. Next, the mobile app has several features such as the QR code scanning, analysis of data and social network sharing. Each feature is designed as a module first. After ensuring all the modules are functional, the modules are integrated together. Moreover, the app is designed to be interactive and user friendly. Lastly, the performance of the app will be evaluated after integrating all the modules into a single application.

1.4 Project Plan

Timeline of this project is shown in Figure 1.1. The project was started on 8 June 2015 and was completed on 15 April 2016. Most of the time was allocated in developing, testing and troubleshooting the app. This is because these tasks are complicated and challenging. Thus, the time spent in these tasks is more than the other tasks. Moreover, the stair climbing data is collected for 51 days, starting from 21 February 2016.

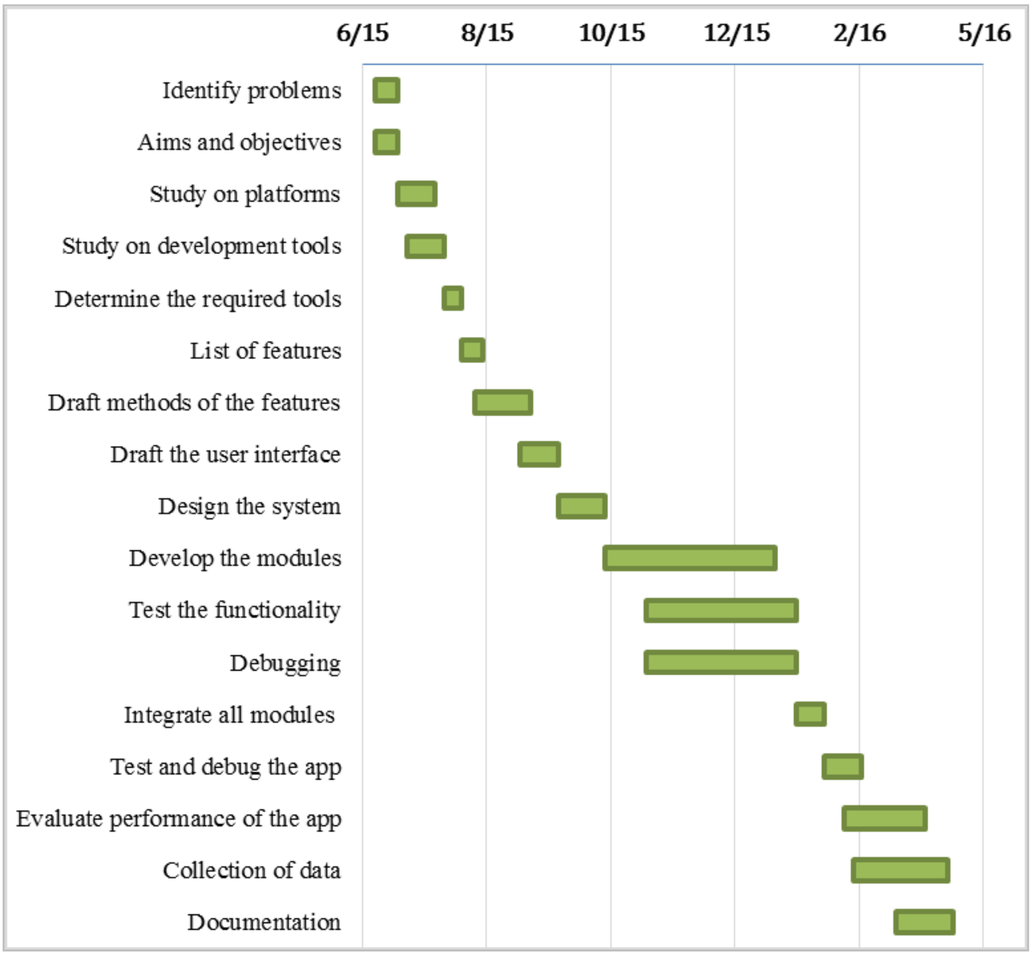


Figure 1.1: Gantt chart of the project

CHAPTER 2

LITERATURE REVIEW

2.1 Development Method

In the past, computation and web browsing are only able to be done by using a computer. This creates difficulties to remote monitoring and controlling. In recent years, due to advanced Information and Communication Technology (ICT), most of the mobile phones, which is also known as smartphones, can replace the computer to perform simple data computation and web browsing. For this reason, remote monitoring and controlling can be easily done by using mobile phone. There are two methods to develop user interface of remote monitoring and controlling for mobile phones.

2.1.1 Mobile Application

Mobile application (app) is a software program that is used in the mobile devices, such as smartphones and tablets. Mobile app is usually developed specifically for a specific platform and the app must be downloaded to the device before using it (Huy and vanThanh, 2012). The available platforms in the market are Android, iOS and Windows Phone. Mobile app able to perform computation and has a better user experience in graphical resolution and interactivity (Chilivumbo, 2015). Moreover, mobile app has the advantages of using the features in the mobile devices, such as camera, sensors detection and Bluetooth.

2.1.2 Mobile Website

Mobile website can provide the information and services through the web browser of mobile devices. The mobile website can be developed by using HTML5, which is the programming language used to develop website. Mobile website is accessible by most of the mobile devices regardless of the mobile devices' platform. Mobile website is generally same as the standard website, but smaller in size in order to fit the mobile devices' screen. Computation is done on the server side which takes longer time to process the information due to the transmission of data.

2.1.3 Comparison of Development Methods

Table 2.1 shows the comparison between two different development methods for mobile devices.

Table 2.1: Comparison between Mobile Application and Mobile Website

Development	Mobile App	Mobile Website
Compatibility	Specific Platform	Cross Platform
Performance	Fast	Depend on Connection Speed
Distribution	Application Store	Website
Connectivity	Offline / Online	Online

After evaluating advantages and disadvantages for both methods, mobile app is chosen as the development method. Mobile app is chosen because it can perform faster in computation which is required in this development. Besides, mobile app is able to operate in both online and offline mode that allows the users to use the app in both modes. Moreover, mobile app can access the other features of the device such as camera. This is the important function that mobile website cannot perform. Due to only specific platform will be targeted, therefore the most commonly used platform is chosen for this development. Hence, this can target more compatible mobile devices that able to use this mobile app. However, there are software that provide cross platform feature for mobile app that developed on a specific OS, and it can be used on different OS as well.

2.2 Studies on Mobile Operating System

Mobile operating system (mobile OS) is the operating system that runs on mobile devices such as mobile phones, smartphones, tablets, computers and other handheld devices. Mobile OS is essential to provide the platform for other application programs to run on the mobile devices. In recent years, Android, iOS and Windows Phone are three main players in the mobile OS.

2.2.1 Android

Android is a mobile OS that based on the Linux-based platform and developed by Google. Android is targeted touch input devices, which include the smartphones and tablets. In recent years, Google has expanded Android OS to other touchscreen devices, such as wrist watches (Android Wear) and televisions (Android TV) which can be seen in CNET's video entitled "*Google will control Android TV, Android Auto and Android Wear video"* (2014). Most of the smartphone manufacturers in the market, such as Samsung, HTC, Lenovo, Sony and Xiaomi use Android as their smartphone platform, which can explain why there is over a billion monthly active Android users (Justin Kahn, 2014). Interface customisation, apps installation from anywhere and expandable storage are the advantages of Android (Florence Ion, 2014). Besides, Android has a large active community and open sources that provide help and support for the development of Android apps. Android Studio is the official Integrated Development Environment (IDE) to develop Android apps.

2.2.2 iOS

iOS or formerly known as iPhone OS is the mobile OS developed by Apple Inc. iOS is designed to run on Apple's devices such as iPad, iPhone and iPod Touch. User interface for iOS is direct and simple to use because of a lack of interface customisation. iOS is well known for good user experience, where apps can be operated smoothly without causing any troubles. This is because Apple fully utilises the hardware and software performance of the devices to optimise the apps. Other than that, iOS provides better security compared with other OS. This is because all the applications published in the App Store are strictly monitored and controlled by

Apple (Adam Dachis, 2011). This is to ensure third-party apps are safe for the users to use without causing a leak of data. IDE to develop iOS apps is XCode, which is developed by Apple.

2.2.3 Windows Phone

The computer OS leader, Microsoft, is also involved in developing the mobile OS, which is known as Windows Phone. Since 2011, smartphone manufacturer, Nokia, formed a partnership with Microsoft and Windows Phone OS is mostly used in the smartphones manufactured by Nokia (BBC News, 2011). Samsung also manufactured a few smartphones that operate using Windows Phone. Windows Phone design is simple and easy to use. One of the good features is the integration of Microsoft Office with Windows Phone, which provides ease of access to view and edit the files like, Excel, PowerPoint and Word, without using the computer. The apps can be developed by using the Microsoft Visual Studio. At the end of July 2015, Microsoft released a new OS, Windows 10, which is a single platform for all the devices including computers, tablets and smartphones (CNET, 2015). The name for the mobile device OS is Windows 10 Mobile.

2.2.4 Comparison of Mobile Operating System

The comparison among the three mobile OS are shown in Table 2.2.

Table 2.2: Comparison of Android, iOS and Windows Phone

Mobile OS	Android	iOS	Windows Phone
Developer	Google	Apple	Microsoft
SDK	Android Studio	XCode	Visual Studio
Main Language	Java	Objective C	C Sharp
Application Store	Google Play	App Store	Windows Store
Latest Version	6.0 Marshmallow	iOS 9	Windows 10 Mobile
License	Open Source	Closed Source	Closed Source

According to the International Data Corporation (IDC), the market shares for Android, iOS and Windows Phones in year 2014 are 81.5%, 14.8% and 2.7% respectively, where the remaining 1% is other OS (International Data Corporation, 2015). As mentioned in the previous section, the OS with the highest market shares will be chosen as the platform to develop the mobile app. Thus, Android is chosen to be the platform for the mobile app. Other than that, the large community support and open source policy provide some advantages in the mobile app development.

2.3 Studies on Code Displaying Methods

In order to identify the floor location of the user, a unique code is assigned to each floor of the building. Input is required for the app to identify the floor and thus, computation can be performed later. The direct input from the user is excluded because it is inconvenience for the user to enter the code while walking and user might input the wrong code accidentally. Hence, other fast and easy methods to input the code is examined in this section.

2.3.1 Quick Response Code

Barcode is the simplest and fast way to hold the information. The information held can only be read by using a barcode reader. A Quick Response (QR) code is a two dimensional barcode that has larger capacity than the traditional barcode, which is in one dimensional. QR code can hold different types of data such as text, addresses, website, images, phone numbers, marketing information, and other types of binary data (Hou A-Lin et al., 2011). Nowadays, the recognition of barcode can be done easily by using the smartphones because of the image processing capability (Liu et al., 2008). The QR code is scanned by using the camera of the smartphones. Then, the patterns of the QR code is interpreted by the decoder software to extract the information stored in the QR code. Due to maturity in the QR code technology, the QR code can be created easily from some of the websites, which is free of charge.

2.3.2 Near Field Communication Tag

Near Field Communication (NFC) is a type of wireless communication that does not require Internet connection. The common applications of NFC are the contactless payment, files transfer and contact sharing. The prerequisite to use this technology is that the smartphone must have NFC feature. When a smartphone is brought close to a NFC tag or another NFC device, the information can be transferred to the smartphone through this contactless communication (NearFieldCommunication.org, n.d.). A NFC tag is the small electronic device that able to store many types of information including commands for phone (Riekki et al., 2012). It can be reprogrammed and encrypted by using the NFC app in the smartphone. Thus, the smartphone is usually a NFC reader and NFC encoder at the same time.

2.3.3 Comparison of Code Displaying Methods

The comparison between the QR code and NFC tag is shown in Table 2.3.

Table 2.3: Comparison of QR code and NFC tag

Methods	QR Code	NFC Tag
Cost	Low	High
Encoding	Any QR code website	NFC Apps
Scanning Distance	Fair	Proximity
Security	Low	High
Mobile Phone Requirement	Camera	NFC

After examining the advantages and disadvantages for both methods, QR code is chosen to display the codes for each floor. This is because the cost of QR code is lower compared to NFC tag, which the only cost for QR code is the printing cost. Besides, the QR code can be generated easily in any QR code website, whereas the NFC tag requires special NFC app to reprogram it. In order to produce a user friendly app, the scanning distance of QR code is definitely an advantage. This is because the users can scan the QR code at a fair distance while moving. Unlike QR code, NFC tag requires a close distance detection, which would be difficult if the stairs are crowded. Furthermore, only the camera feature of the mobile phone is needed for the QR code detection, which can target a larger group of users. This is because only some of the mobile phones are equipped with NFC feature. However, the disadvantage of the QR code is that it is not secure. The code encoded can be replicated and read easily. While the security of NFC tag is higher, it provides encryption that only allow specific person to modify the information.

2.4 Studies on QR Code Decoders

Recognition of the QR code involves complicated algorithms and it would take some time to develop a QR code scanner. Due to the time limitation, a study on the open-source QR code decoders is done to determine the suitable QR code library for the mobile app. There are several open-source QR code decoder libraries available in Internet. However, only the two decent QR code decoder libraries, ZXing and Mobile Vision, are examined for this project.

2.4.1 ZXing

ZXing or “zebra crossing” is a popular open-source barcode image processing library for Android app (ZXing, 2016). It is programmed in Java and able to use in other languages by using port. This library supports the decoding of barcodes for one dimensional and two dimensional barcodes such as QR code. The first release of this library is on March 2008 and its latest release version is 4.7.5. Furthermore, this library includes barcode encoder to encode into different formats of barcode.

2.4.2 Mobile Vision

Google’s Android Developers announced the new Mobile Vision Application Programming Interfaces (API) are added into the Google Play Services 7.8 on August 2015 (Laurence Moroney, 2015). Mobile Vision APIs consist of two sub-packages, which are barcode API and face detection API (Google Developers, 2015). The barcode API supports the decoding of different formats of barcodes. Barcode API is usually updated with the release of new version of Google Play Services.

However, the users are required to have the latest version of Google Play Services installed in their devices.

2.4.3 Comparison of QR Code Decoders

The comparison between the two QR code decoders is shown in Table 2.4.

Table 2.4: Comparison of ZXing and Mobile Vision

Decoder	ZXing	Mobile Vision
Developer	Third Party	Google
Supported 2D Barcode Formats	QR Code Data Matrix PDF-417 (beta) Aztec (beta)	QR Code Data Matrix PDF-417 Aztec
Supported Mobile OS	Multiple OS	Android
First Release (year)	2008	2015
Latest Version	4.7.5	Google Play Services 8.4

Mobile Vision is chosen as the QR code decoder, although Mobile Vision were released recently and is newer than ZXing. This is because this mobile app is targeting the Android users and thus, Mobile Vision is more appropriate, as it is the official barcode decoder API. Therefore, compatibility issues would be lesser, and the documentation and samples are adequate.

2.5 Literature Analysis of Mobile Application

Health and fitness mobile apps are common nowadays, especially for tracking workout and running activities. However, there are only one mobile app that related to stair activities, which is known as StepJockey. This existing stair mobile app is discussed in the following.

2.5.1 StepJockey

StepJockey is a mobile app that is developed by StepJockey Ltd. in United Kingdom (StepJockey, n.d.). This mobile app is part of the commercial product to promote the use of stairs in office buildings. The mobile app able to perform QR code scanning and NFC detecting, which named as StepJockey Smart Signs. However, the QR codes and NFC tags are required to customise and purchase from the company. The app requires the users to register before using it. It is able to track the users' progress and set targets. Besides, it has challenges for all the users to compete. A screenshot of the list of past records of StepJockey is shown in Figure 2.1.

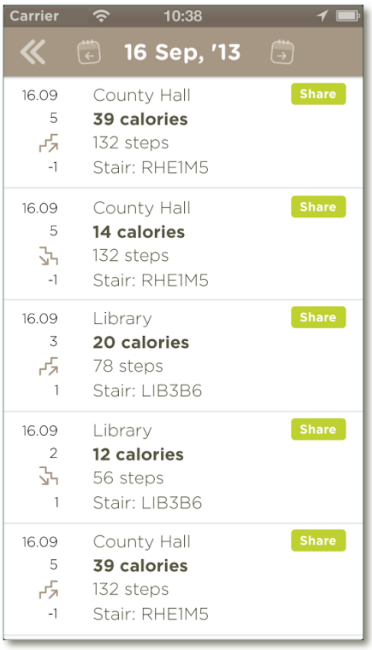


Figure 2.1: Screenshot of StepJockey

CHAPTER 3

METHODOLOGY

3.1 Development Tools

Android Software Development Kit (SDK) is the main development tool that is used in this project. Java Development Kit (JDK) has to be installed before installing Android Studio. The Android Studio can be used to develop different Android applications such as Android Wear, Android TV module, Glass module, and Phones and Tablets Applications. There are various tools in the Android Studio such as Android SDK Manager, Android Virtual Device (AVD) Manager, Android emulator, and Android Monitor. The Android SDK Manager is to download, update and install the packages, platforms and other components. The AVD Manager is to define the specifications of the Android devices. The examples of specifications are the device type, screen resolution, memory size, and etc. Then, the defined characteristics can configure the Android emulator, which can operate and simulate the device in the computer. The developed mobile app can also be tested on an actual Android device by connecting to the computer. Lastly, the performance of the app can be examined by the Android Monitor. Optimization and debugging can be done easily with the aid of Android Monitor.

3.2 Mobile App Development Method

A software development model known as Rapid Application Development (RAD), is chosen as a guideline to develop the app for this project. RAD is suitable to be used because it emphasis on shorten the development time. Furthermore, the RAD approach breaks down a problem into several small tasks, which can be solved separately by prototyping during the early stage. This is to reduce the risk of developing a malfunction software application at the end of the project. In addition, the high flexibility of RAD model, which can adapt the project develops instead of rigorously defining specifications and plans correctly at the beginning stage. Figure 3.1 shows the workflow in this mobile app development project.

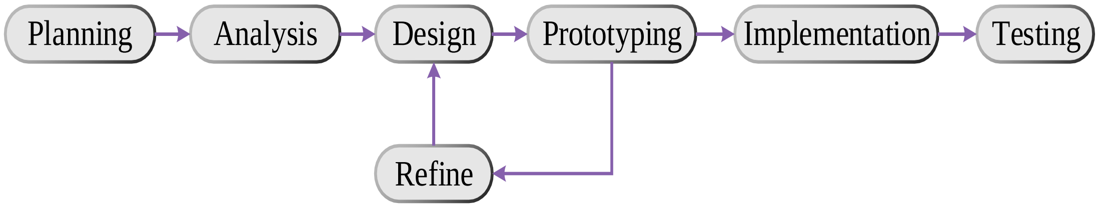


Figure 3.1: Flowchart of Development Process

Planning phase is the initial phase to define problems and objectives. The possible problems in monitoring the stair climbing activities are identified, so that the purposes of this project can be achieved. The aims and objectives are then defined to give a clear vision on the purpose of this project. After defining the objectives, all the required technologies and tools are examined and studied such as platforms, app development tool and web development tool. Besides, the gathering of data such as the number of steps between each floor and the amount of calories burnt are done in this phase. In addition, the QR codes are also generated for each floor.

In analysis stage, functions required in this mobile app are analysed cautiously and each function is developed as a module. The modules are QR code scanner, usage trends and analysis, log record, login authentication and registration. Each module is developed individually before it is merged into a single app. Then, each module is further analysed to choose a more effective method to perform the required tasks. For instance, the trends analysis includes several small tasks such as getting the highest record, daily and monthly analysis, and arrangement of the data that need to be analysed, in order to identify a better way to organize and present the data. Lastly, all the required development tools are identified at the end of this stage.

Next stage is the design stage. The specifications of the mobile app are established before designing the system and user interface (UI) of the module. All the specifications are listed down as the minimum requirement for the app. The appearances and visual effects are also considered in the UI design in order to give a better user experience. The UI is designed to display the data in a meaningful way. Other than that, the approaches to establish the connection between the mobile app and the web server are examined in this stage.

In the prototyping stage, a prototype is developed for each module. The prototype is then tested and examined. A simple test such as functional test is to test the functionality of the module. This is to ensure the module functions correctly based on the predefined specifications. If the module works accordingly, then the refinement stage can be skipped and proceed to the next module design.

If the prototype does not function accordingly or the outcome is undesired, then the ineffective prototype needs to go through a refinement stage. The main function of this stage is to examine and resolve the problems to ensure the developed mobile app is functional. Flaws in the prototype are identified and the solutions to the problems are also determined in this stage. After resolving the issues, a revised version of prototype is developed. This design-prototype-refinement cycle will continue until the desired outcome is achieved.

After completing develop all modules, the modules are integrated together in the **i**mplementation phase. This phase is only executed, after every feature of the app is tested and functional. Then, a complete app is only developed to produce the final prototype. All modules are linked together, so that these modules are able to communicate with each other. This final prototype must be able to perform the functions as stated in the previous. If the final prototype does not function according to the requirements, then the prototype needs to be refined in the refinement stage.

Finally, the final app is developed and tested in testing phase. In this phase, the app is tested with different tests to evaluate the reliability and functionality of the app. The app is also tested to ensure the accurateness and correctness during the extraction of information. However, any issues encountered in this stage have to be resolved before deploying the app. After completing the testing and debugging, the app is ready to be deployed.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 Design Preparation

Preparation involves assumptions, considerations and data collection are done before designing the mobile app. This is to make sure the flow of the design process is not interrupted

Firstly, number of steps of the staircase between each floor for two different buildings, KA block and KB block, are counted. Table 4.1 and Table 4.2 show the number of steps between each floor for KA and KB blocks.

Table 4.1: Number of steps between each floor in KA block

Start Floor	End Floor	Steps
LG	G	23
G	M	23
M	1	24
1	2	24
2	3	23
3	4	24
4	5	24
5	6	24
6	7	24
7	8	24

Table 4.2: Number of steps between each floor in KB block

Start Floor	End Floor	Steps
B	SB	27
SB	G	27
G	1	57
1	2	38
2	3	38
3	3A	27
3A	5	27
5	6	27
6	7	27
7	8	27
8	9	27
9	10	27

In order to calculate the calories burnt, energy saved and amount of carbon emission avoided, some assumptions are made based on the information gathered through research. Generally, the calories burnt for step up is in the range of 0.11 calories to 0.17 calories, whereas only 0.05 calories is burnt for stepping down. Other considerations, such as speed and weight of the user, are not taken into account in calculating the calories burnt.

Some assumptions are made to calculate energy saved by taking stairs as shown in Table 4.3. For a lift to move up and down twenty floors, it consumes about 100 Wh and thus, on average, energy for one floor per round trip would consume 5 Wh. In the previous, the calories burnt for stepping up is about 77% of the total calories burnt for a round trip, whereas only 22% accounts calories burnt for the stepping down. Thus, by making assumption that the percentage of energy used for upward direction is equivalent to the calories burnt for stepping up, the amount of energy required for moving up one floor is 3.86 Wh. Same as the previous, the energy for moving down one floor is 1.14 Wh. An average flight of stairs has 12 stair steps, and hence, the energy per step up and energy per step down are 0.32 Wh and 0.09 Wh.

Table 4.3: Assumptions made for energy saved

Assumptions	**Wh**
Energy for 20 floors per round trip	100
Energy for 1 floor per round trip	5
Energy for 1 floor (step up)	3.86
Energy for 1 floor (step down)	1.14
Energy per step up	0.32
Energy per step down	0.09

The carbon emission conversion factor for electricity is 0.511 kgCO_2/kWh, which is obtained from Sustainable Energy Authority of Ireland (2014). Table 4.4 shows all the assumptions made for the calculation.

Table 4.4: List of assumptions made

Assumptions	
Calories burnt per step up (cal/step)	0.17
Calories burnt per step down (cal/step)	0.05
Energy saved per step up (Wh/step)	0.32
Energy saved per step down (Wh/step)	0.09
Amount of CO_2 avoided (kgCO_2/kWh)	0.511

The equations to calculate the calories burnt, energy saved, and amount of carbon emission avoided are shown below.

$$Calories\ Burnt = no.of\ steps\ \times calories\ per\ step \tag{4.1}$$

$$Energy\ Saved\ = no.of\ steps\ \times energy\ per\ step \tag{4.2}$$

$$Amount\ of\ CO_2\ avoided = energy\ saved(kWh)\ \times kg_{CO_2}/kWh \tag{4.3}$$

QR codes are generated for all the floors. Text encoded in the QR codes are randomly created text. The only difference between the two blocks is that the starting character of the text in KA block is represented with alphabet “z”. In this project, QR codes are generated from a free website, QR Code Generator (The QR Code Generator, n.d.). Each floor is also assigned to a unique floor ID. These are the reference numbers that used by the web server. The text encoded in QR codes for all the floors in KA block and KB block are shown in Table 4.5 and Table 4.6.

Table 4.5: Text encoded in QR code and floor ID for each floor in KA block

Floor	Text Encoded in QR code	Floor ID
LG	z604ksh99	13
G	z501ahy92	14
M	z922kyj99	15
1	z306pca90	16
2	z121jrw81	17
3	z329kth80	18
4	z210sne94	19
5	z103ksh95	20
6	z113ghr91	21
7	z221ssw94	22
8	z222hhj87	23

Table 4.6: Text encoded in QR code for each floor in KB block

Floor	Text Encoded in QR code	Floor ID
B	66jsj0210	0
SB	76kjk0425	1
G	72yjs0814	2
1	79hdh0820	3
2	78khg0224	4
3	81sjh0815	5
3A	85lks0714	6
5	90iya0530	7
6	89kty0309	8

7	91sjh0628	9
8	94bsj1010	10
9	91pcr0303	11
10	94jsj1024	12

4.2 Design Requirements

Design requirement is a comprehensive description of aims for a development project. It contains all the functionality and details required for specific project and is to be achieved at the end of the project. It can be divided to functional requirements and non-functional requirements. Besides, the mobile app is developed for the mobile devices that supports Android 4.0.3 and above.

4.2.1 Functional Requirements

Functional requirements describe what the mobile app needs to do, which are related to the main features and the purpose of the app. Functional requirements of each module are detailed in Table 4.7.

Table 4.7: Functional requirements for each modules

Modules	Functional Requirements
QR Code Scanner	• Scan QR codes • Check validity of QR codes • Choose action before second scan • Scan twice before performing calculation • Calculate usage such as calories burnt, steps, energy saved, and duration • Save the record into CSV file with correct formatting • Send the record to the server when Internet access is available • Store the record to temporary file when Internet access is unavailable

Login Authentication	• Allow user to enter email and password • Encode user's email and password as query, and send to server for verification • Receive response from server • Read response and verify the user identity • Notify user due to unsuccessful login • Store the receive data upon successful login
Registration	• Allow user to input the information details • Verify the validity of the details • Notify user for invalid details • Encode details as query, and send to server for verification and registration • Receive response from the server • Read the data and verify the registration status • Notify the user regarding the registration status
Trends and Analysis	• Show the correct chart • Organize data in monthly and daily • Allow user to choose type of data to display • Show the records in monthly and daily • Calculate the grand total usage • Calculate the sum of today's usage • Identify the highest usage based on trip, day, and month
Logger	• Read from CSV file • Display the past records in descending order

4.2.2 Non-Functional Requirements

The behaviour of the app is described in the non-functional requirements. The non-functional requirements are described in terms of performance and usability as shown in Table 4.8.

Table 4.8: Non-functional requirements of the app

	Non-Functional Requirements
Performance	• Detect the QR code in less than 5 seconds when selected • Show the result right after the second scan • Display the list of past records in less than 3 seconds when selected
Usability	• Have a consistent navigation menu throughout the entire app • Easy to use user interface

4.3 Graphical User Interface Design

Graphical User Interface (GUI) is an interface that uses graphical icons and indicators to interact with the users. It is important to provide users a simple and yet, efficient interface that able to direct the users. Consistent interface is also part of the GUI design consideration.

When the app is first launched, user authentication page is shown to register or verify the user first as illustrated in Figure 4.1. By changing the types of user, the email domain will change automatically. The user needs to register in Register Tab and key in the required information as shown in Figure 4.2.

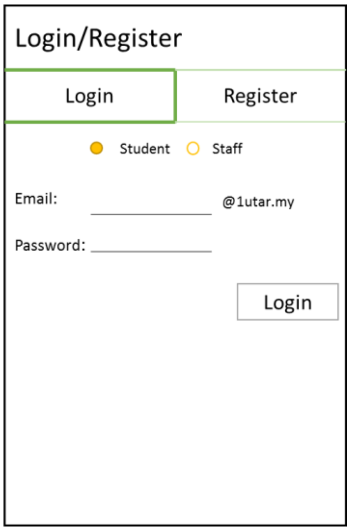


Figure 4.1: GUI of Login Authentication

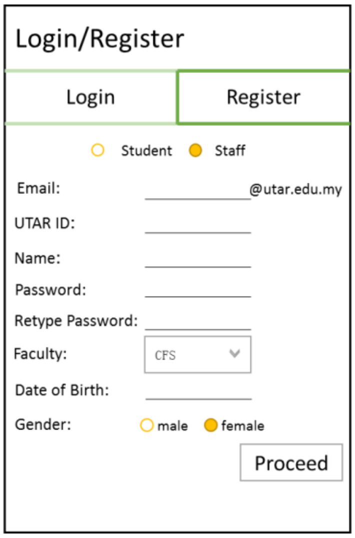


Figure 4.2: GUI of Registration

After verifying the identity of user, the homepage of the app is shown as illustrated in Figure 4.3. The homepage is named as "Summary" and it has two tabs. Today's date, total usage of the day, and the most recently stair trip are shown in the Summary tab, while the total usage and highest usage are shown in All tab. Users can choose to display the today total usage in terms of calories, energy or steps.

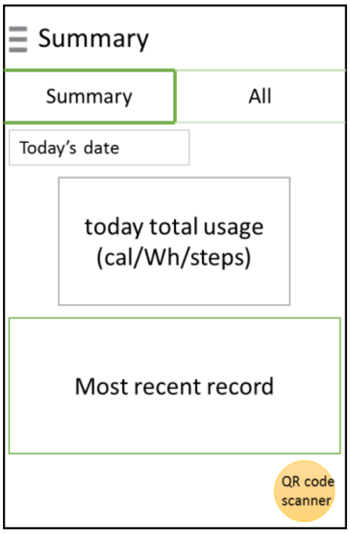


Figure 4.3: GUI of Homepage

Next, the trends module is basically represents by GUI of Trends as shown in Figure 4.4. Two tabs that allow users to switch and view the data in monthly and daily. About half of the screen is used to display the chart and the other half is used to display the records in words. User can toggle the data to be shown in the chart.

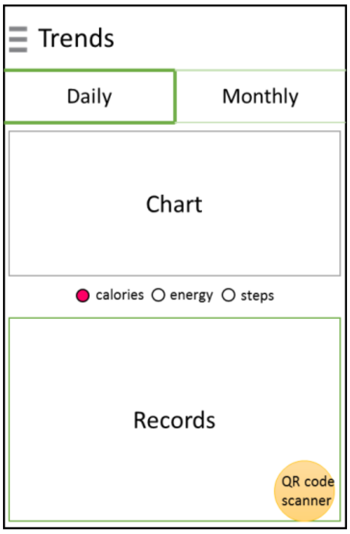


Figure 4.4: GUI of Trends

The following GUI is the stair trip result which is shown in Figure 4.5. The stair trip result is designed to display as a result card. This is to make the result simple and easy to understand. Graphical icons are used to represent the name instead of wording. The result card will show the date, time, calories burnt, energy saved, steps travelled, duration, building, and locations of the starting floor and exiting floor.

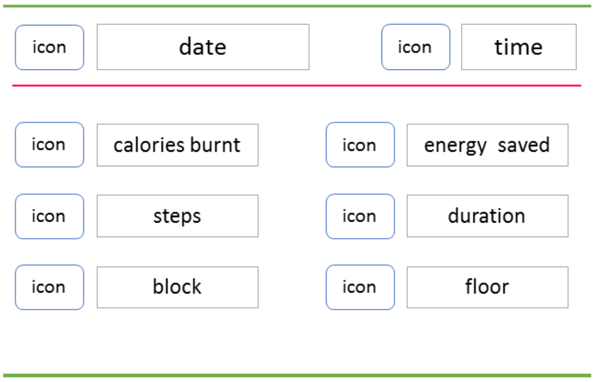


Figure 4.5: GUI of Stair Trip Record

Lastly, the GUI of log module is shown in Figure 4.6. Total usages for calories burnt, energy saved, and number of steps are shown at the top. The past records are sorted from newest record to oldest record. Each result card represents a stair trip. Besides, QR code scanner button can be accessed in any page of the app after login to the app. This allow the user to quickly and easily click the scan button, without having to search in the app.

Figure 4.6: GUI of Logger

4.4 Implementation

The codes of the mobile app discussed in this section. Each module is explained separately and some of important codes are highlighted. Before starting to code, the libraries for some of the special features are added into the dependencies in the gradle file as shown in Figure 4.7. Those special features cannot be used without these libraries. Other than that, permissions required for the app to access the mobile phone features are added in to the Manifest.xml, which is shown in Figure 4.8. This is to allow the mobile app to use the other features in the mobile phone such as camera.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:design:23.1.1'
    compile 'com.android.support:support-v4:23.1.1'
    compile 'com.google.android.gms:play-services-vision:8.4.0'
    compile 'com.android.support:recyclerview-v7:23.0.+'
    compile 'com.android.support:cardview-v7:23.0'
    compile 'jp.wasabeef:recyclerview-animators:1.3.0'
    compile files('libs/mpandroidchartlibrary-2-1-6.jar')
    compile 'com.facebook.android:facebook-android-sdk:[4,5)'
}
```

Figure 4.7: Dependencies in the gradle

```xml
<uses-feature android:name="android.hardware.camera"/>
<uses-feature android:name="android.hardware.camera.autofocus"/>

<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Figure 4.8: Permissions required in Manifest.xml

4.4.1 Registration Module

User is required to register as a registered user before using the app. This is to make sure the target users are students or staffs from UTAR. The flowchart of this module is shown in Figure 4.9. User is required to register using the university email address which the email domain is @1utar.my for student or @utar.edu.my for staff.

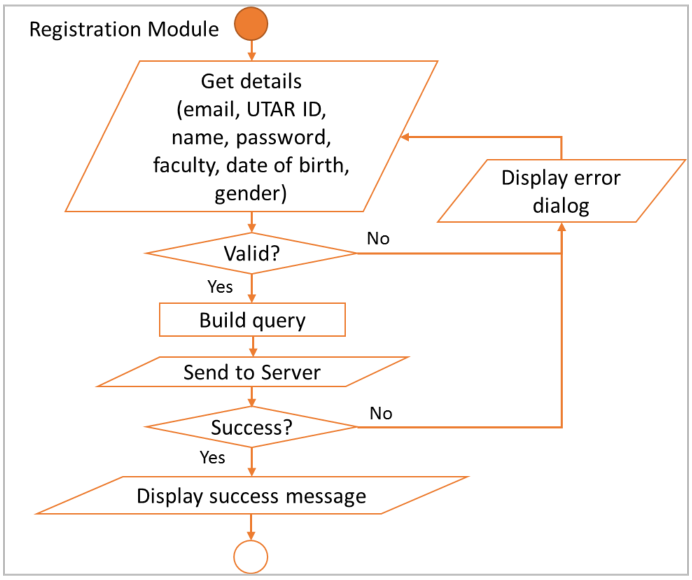


Figure 4.9: Flowchart of registration module

User is also required to enter information such as UTAR ID, name, password, date of birth, faculty, and gender. Before proceeding to registration, all information is passed to the method validation() to check the validity of information. For instance, UTAR ID is checked by using the codes shown in Figure 4.10, which 7 digits number for student and 5 digits number for staff.

After verifying the information, all the information are built as encoded query and sent to the server. The InternetActivity class is called to connect to the web server and the codes for establishing server connection is illustrated in Figure 4.11. The app will receive response from the server upon successful sending the query.

```java
private boolean isValidID() {
    boolean isValid = true;

    try {
        int y = Integer.parseInt(utarID);

        if (utarID.trim().length() != 7 && isStudent)
            isValid = false;

        if (utarID.trim().length() != 5 && !isStudent)
            isValid = false;

    } catch (NumberFormatException e) {
        isValid = false;
    } catch (NullPointerException e) {
        isValid = false;
    }
    return isValid;
}
```

Figure 4.10: Section of code to check validity of UTAR ID

```java
try {
    //Establishing connection
    httpconnection = (HttpURLConnection) url.openConnection();
    httpconnection.setDoOutput(true);
    httpconnection.setDoInput(true);
    httpconnection.setChunkedStreamingMode(0);
     //Set connection timeout in 5 sec
    httpconnection.setConnectTimeout(5000);
    httpconnection.connect();
    //Send query to server
    Log.d("tcy", "Sending query to Server");
    OutputStream os = httpconnection.getOutputStream();
    BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os, "UTF-8"));
    writer.write(query);
    writer.flush();
    writer.close();
    os.close();

    if (httpconnection.getResponseCode() == HttpURLConnection.HTTP_OK) {
         //Connection ok
         //Receive data from server
         Log.d("tcy", "Retrieving data from Server");
         InputStream is = httpconnection.getInputStream();
         reader = new BufferedReader(new InputStreamReader(is, "UTF-8"));
         String line;
         while ((line = reader.readLine()) != null) {
              result.append(line);
         }
    }
} catch (SocketTimeoutException e) {
     //Throw connection timeout
    isTimeout = true;
    e.printStackTrace();
    Log.d("tcy", "Connection timeout");

} catch (IOException e) {
    e.printStackTrace();

} finally {
    if (httpconnection != null) {
         //Close connection
        httpconnection.disconnect();
    }
    Log.d("tcy", "Disconnected from " + url.toString());
    if (reader != null) {
        //Close reader
        try {
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Figure 4.11: Section of code for server connection

The response from web server is in JavaScript Object Notation (JSON) format. If user registered successfully, server will response with the codes shown in Figure 4.12.


```
{
  "success": true
}
```


Figure 4.12: JSON of server response for successful registration

On the other hand, response for unsuccessful registration is shown in Figure 4.15.


```
{
  "success": false,
  "errors": {
    "email": [
      "Email \"cheeyang@1utar.my\" has already been taken."
    ],
    "utarId": [
      "Utar ID \"1104737\" has already been taken."
    ]
  }
}
```


Figure 4.13: JSON of server response for unsuccessful registration

4.4.2 Login Authentication Module

This module is designed to verify the user identity before entering the app. Figure 4.14 shows the flowchart of verifying the user. User is required to key in email and password before proceeding to login. Then, the email and password are encoded into query as shown in Figure 4.15.

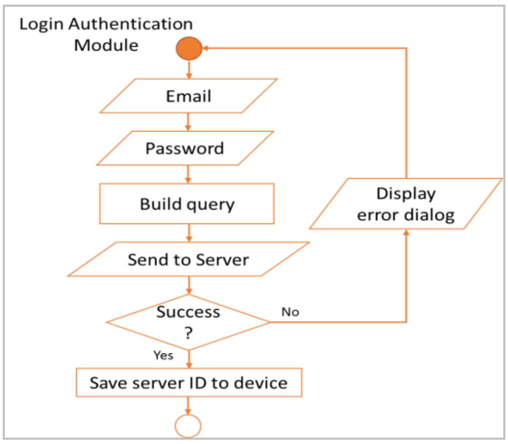


Figure 4.14: Flowchart of login authentication module

```java
private String buildQuery() {
    Uri.Builder builder = new Uri.Builder();

    builder = builder.appendQueryParameter("Participant[email]", email)
            .appendQueryParameter("Participant[password]", password);

    return builder.build().getEncodedQuery();
}
```

Figure 4.15: Section of code to build query

Then, the query is passed to the InternetActivity class to establish connection to the server. If the user is a registered user, a JSON response is received from the server with all the user information as shown in Figure 4.16. The "id" in the JSON string is the unique database reference number for the server, which is needed later in the app. Thus, the "id" is saved into a temporary file, which can be retrieved later. Other than "id" and email, other information are not saved into the device because they are not required to use in the app.


```json
{
  "id": 15,
  "email": "cheeyang@1utar.my",
  "utarId": "1104737",
  "name": "TAN CHEE YANG",
  "gender": "M",
  "birthDate": "1993-09-30",
  "faculty": "E",
  "participantType": "T"
}
```


Figure 4.16: JSON from server for successful login

4.4.3 QR Code Scanner Module

QR code scanner module is the most important module in this app and the flowchart of this module is shown in Figure 4.17. Google Mobile Vision sample is integrated into this module to make use of the QR code scanner function. This module is designed to get two scans, which are Scan In at the entering floor and Scan Out at the exiting floor.

When QR code scanner button is selected, the app will start the BarcodeCaptureActivity class. QR code is captured automatically by the app when the QR code is in sight. The QR code is decoded by using the codes shown in Figure

4.19. Before Scan Out, user has option to discard the scan in data or continue to scan out and the screenshot of the prompt dialog is shown in Figure 4.19. This allows user to discard the previous scan if user forgot to scan out for that trip.

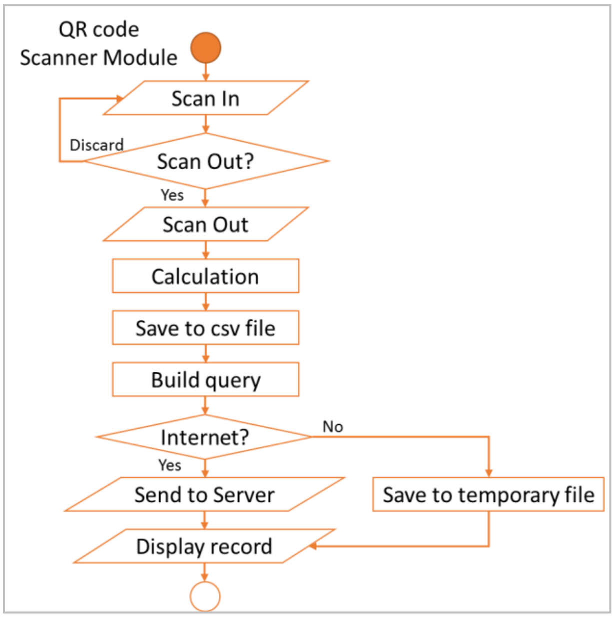


Figure 4.17: Flowchart of QR code scanner module

```
Barcode barcode = data.getParcelableExtra(BarcodeCaptureActivity.BarcodeObject);
String data = barcode.displayValue;
```

Figure 4.18: Section of code to decode QR code

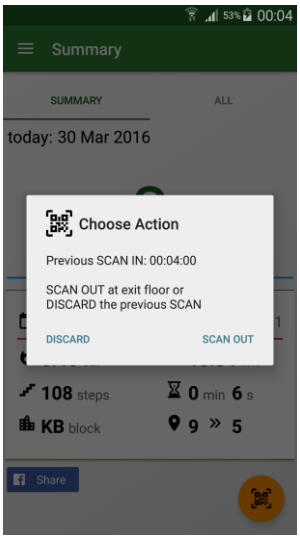


Figure 4.19: Screenshot of prompt dialog for discarding previous scan

After each scan, the validity of data is checked by ensuring the QR code captured is valid. This is to ensure the correct QR code is captured and the stairsIdentity class is responsible to check the QR codes. Besides, after the scan at the exit floor, both start data and exit data are compared to verify that the user scanned the QR codes in the same building.

Then, the CalculationFunc class is called to perform the calculation for calories burnt, energy saved, steps travelled, and duration. The record is then saved into device as Comma-Separated Values (CSV) format file and in temporary file. The record in CSV file is used as the log record file, whereas the temporary file serves as most recently record to be retrieved and displayed in the homepage. Codes for saving into temporary file is shown in Figure 4.20.

```
final String MY_PREF = "MyPref";
SharedPreferences myPreferences;
SharedPreferences.Editor editor;

myPreferences = this.getSharedPreferences(MY_PREF, MODE_PRIVATE);
editor = myPreferences.edit();
editor.putString("calories", array[0]);
editor.putString("energy", array[1]);
editor.putString("duration", array[2]);
editor.putString("step", array[3]);
editor.putString("direction", array[4]);
editor.putString("time", array[5]);
editor.putString("block", array[7]);
editor.putString("from", array[9]);
editor.putString("to", array[11]);
editor.putInt("day", day);
editor.putInt("month", month);
editor.putInt("year", year);
editor.putInt("dayofweek", day_of_week);
editor.apply();
```

Figure 4.20: Section of code to save record in temporary file

In order to log the result to the server, few parameters are required such as “id”, enter floor ID, exit floor ID, enter time, and exit time. The “id” is the reference number received from server in the login module. The floor ID is referred to Table 4.5 or Table 4.6. These parameters are then built as encoded query and sent to server. However, if internet is not available, the record will save into another temporary file. Method to check for network availability is shown in Figure 4.21. When Internet is available, all the records in temporary file will be uploaded to the server.

```java
public boolean hasNetwork() {
    boolean success;
    ConnectivityManager connectManager;

    connectManager = (ConnectivityManager) ctx.getSystemService
            (Context.CONNECTIVITY_SERVICE);
    NetworkInfo activeNetworkInfo = connectManager.getActiveNetworkInfo();
    if (!(activeNetworkInfo != null && activeNetworkInfo.isConnected())) {
        //No network connection
        success = false;
    } else {
        //Has network connection
        success = true;
    }
    return success;
}
```

Figure 4.21: Section of code to check network availability

4.4.4 Trends and Analysis Module

This module is to display and analyse the trip records. In order for the users to keep tracking their stair usage, the records are to display in daily and monthly. The highest track records are also displayed in terms of trip, daily and monthly. Besides, total stair usage is also calculated for the users to know how much they have contributed to the stair usage. The flowchart of the trends and analysis modules are shown in Figure 4.22 and Figure 4.23.

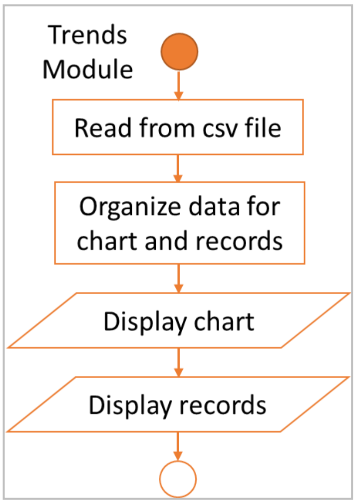


Figure 4.22: Flowchart of Trends Module

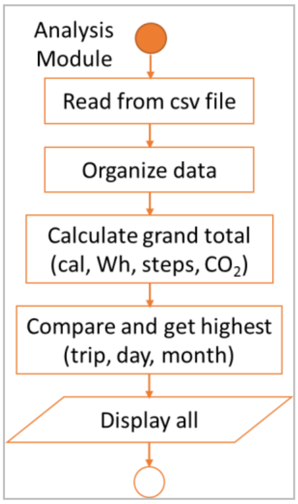


Figure 4.23: Flowchart of Analysis Module

In order to display data in chart, the data needs to be read from the CSV file first. All the data are then passed to the dataOrganizer class to reorganize data in daily or monthly. Then, the reorganized data is displayed in chart and in words. The interface of the trends module that displayed data in daily, is illustrated in Figure 4.24. The calories, energy and steps are differentiate with three different colours to improve the users' visual experience.

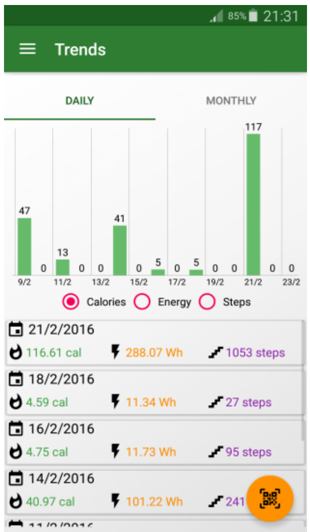


Figure 4.24: Screenshot of the Trends module

Next, the analysis module also has to retrieve the data from the CSV file and passed it through the dataOrganizer class. Then, grand total of calories burnt, energy saved and total steps travelled are calculated through the CalculationFunc class.

Besides, the total avoidable carbon emission is also calculated. This is to inform the user on the amount of carbon emission avoided through stair climbing. In addition, the highest usage of the trip, day and month are also calculated and displayed in this module. The screenshot of the grand totals and the highest stair usages is illustrated in Figure 4.25.

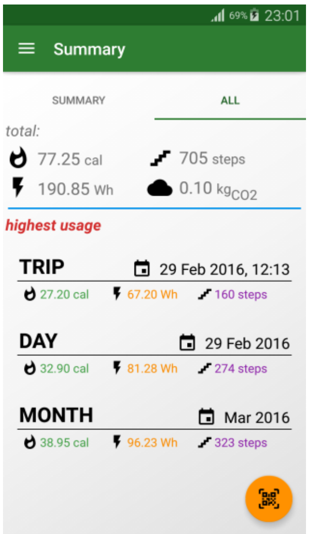


Figure 4.25: Screenshot of the Analysis module

4.4.5 Logger Module

All the stair trip records are displayed in this module and the flowchart of this module is shown in Figure 4.26. The records are first retrieved from the CSV file. Then, the records are rearranged from the newest record to the oldest record. The records are displayed in the list and scrolling effect is added to enhance the user experience.

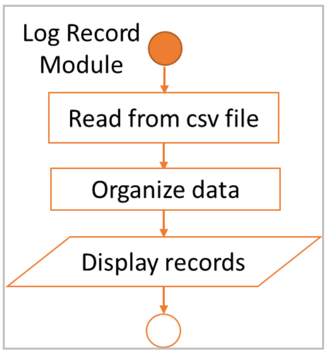


Figure 4.26: Flowchart of Logger Module

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Collection of Data

The mobile app is deployed to the Google Play Store on 21 February 2016. There are 164 registered users as of 12 April 2016. The data of the stair activities are collected from these 164 samples for about one and half month. All the data collected are to analyse the two campus buildings of UTAR Sungai Long.

5.2 Analysis of Data

Figure 5.1 shows the cumulative stair usage from 7 a.m. until 7 p.m. The graph shows the stair trips in total and for different building. The stair usages reached the peak at noon, which mainly due to the lunch break. Lunch break has the higher flow of people because most of the people in the buildings left the buildings to have their lunch.

Furthermore, the stair usage is observed for the two campus buildings, which are KA block and KB block. Based on the pie chart in Figure 5.2, the stair usage in KB block is 79% of the total usage, while the KA block only occupied 21% of the total stair usage. One of the possible reasons could be that there are more people in the KB block. Other reason could be that the congestion at lifts is much serious in KB block and thus, stair usage is higher in KB block.

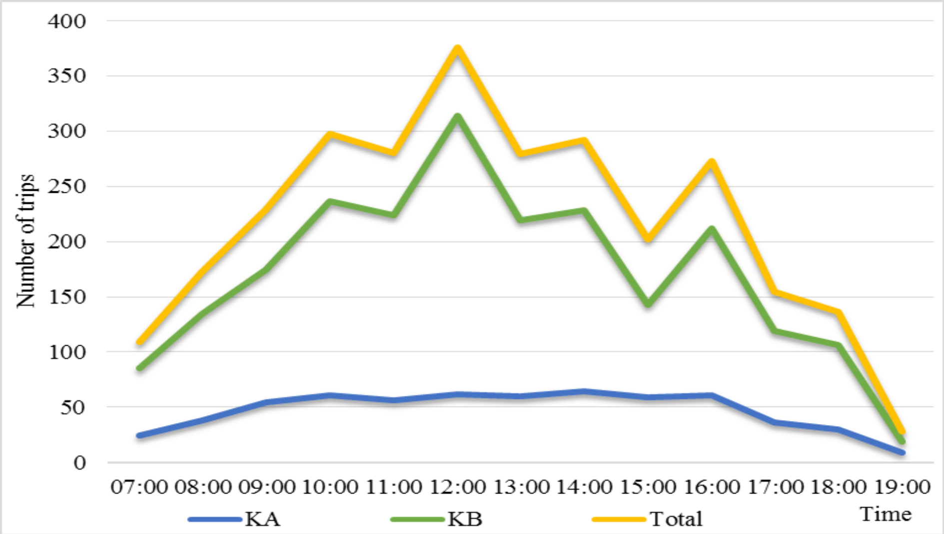


Figure 5.1: Cumulative stair usages at different hours

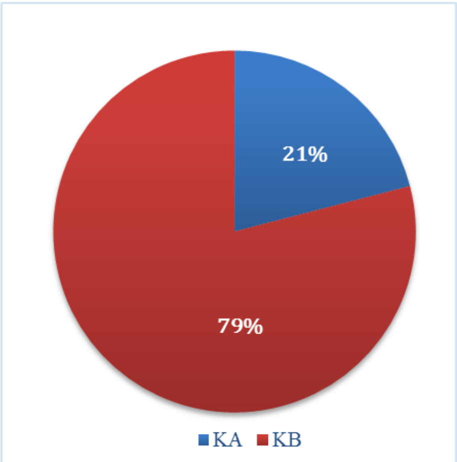


Figure 5.2: Pie chart of percentage of stairs usage for different buildings

In addition, the stair climbing activities are also analysed for both blocks. The data of stair climbing activities for KB block is shown in Figure 5.3. Only the stair climbing activities for ground floor, first floor, second floor, third floor, and fourth floor are analysed. Users tend to climb the stair at the ground floor to reach other floors and the stair usage is highest for ground floor to first floor. Besides, users are most likely to use the stairs to reach the floor above.

As for the KA block, the statistic of the stair climbing activities is shown in Figure 5.4. Only five floors, which are ground floor, mezzanine floor, first floor, second floor, and third floor, are observed. The highest usage recorded is from the ground floor to fifth floor. This is because most of the tutorial rooms and lecture rooms are located in fifth floor and fourth floor. The stair climbing activities at ground floor is higher compared to other floors, which is similar to the KB block.

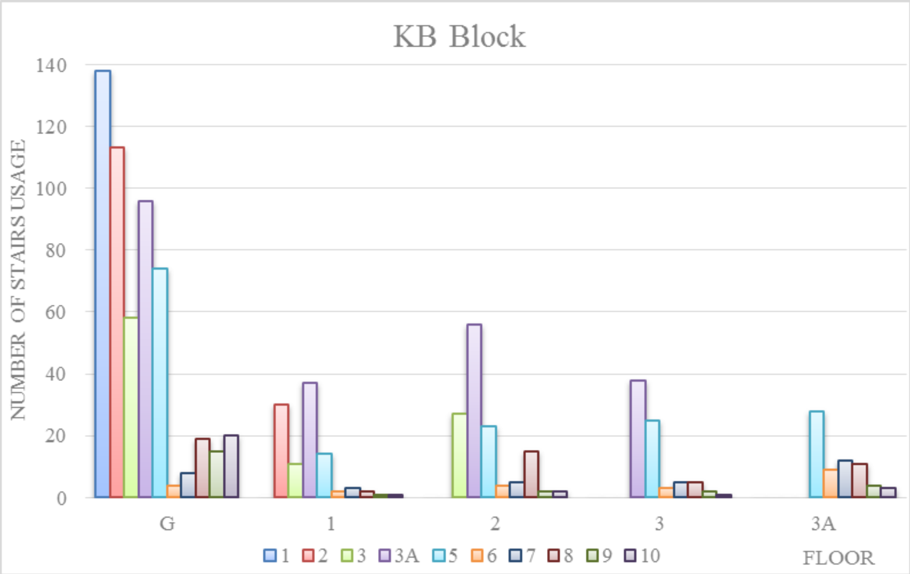


Figure 5.3: Stair climbing activities for KB Block

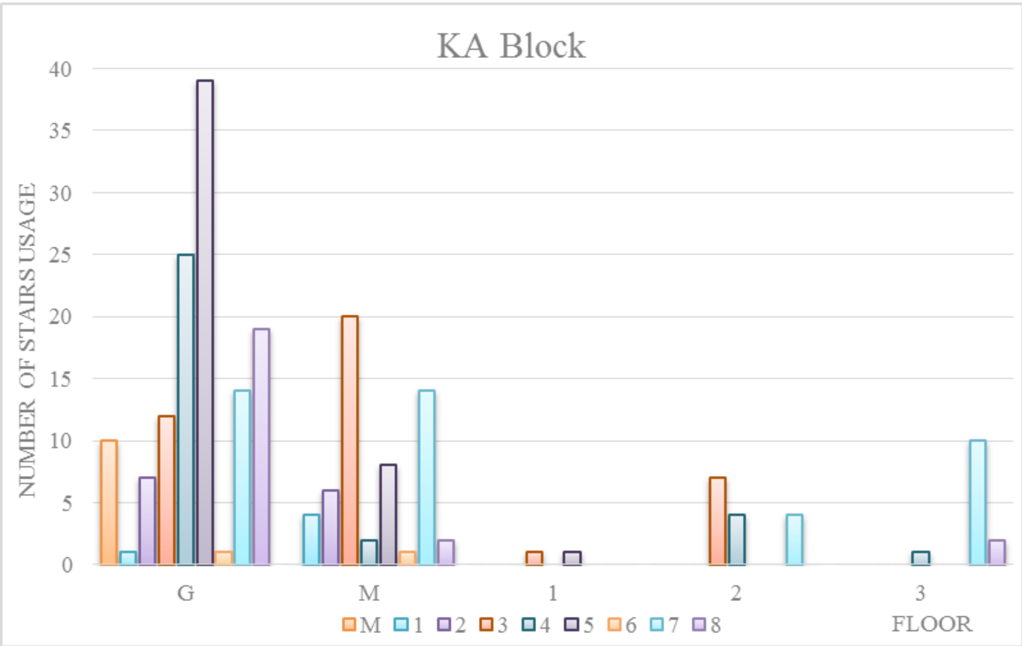


Figure 5.4: Stair climbing activities for KA Block

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

Staircases are always a part of the features in a multi-storey building, whether lifts are installed in the building or not. Stair climbing has many health benefits such as improved cardiovascular fitness and strengthened musculoskeletal system. In addition, stair climbing burns more calories compared to walking. However, the use of the staircases are relatively low compared to lifts in most buildings.

In this project, a mobile app is developed to monitor the stair climbing activities in a building. The aim is to collect and analyse the data of the stair usages from the users. QR code detection is integrated into the mobile app to input the location floor of the user. The mobile app consists of features such as, track and analyse the progress of the user, user authentication, and record to web server. Data collected in the web server can be used to analyse the stair usages.

The stair data collection from 164 samples is done at the two campus buildings of UTAR Sungai Long. The data collection period is about one and a half month. Based on the analysis of stair usages, stair usages are higher in the KB building and the peak hour of the stair usage is at the noon. Moreover, the users are more likely to climb the stairs from ground floor to other floors. However, due to the small sample size, the data collected cannot represent all the population in the buildings.

6.2 Limitations and Recommendations

The developed mobile app has some limitations and some improvements can be done to improve the accuracy of the app. Firstly, the app does not consider factors, such as weight, height and speed of the user into the calculation of calories burnt after taking the stairs. Thus, the calories burnt calculated in the app may not be the actual calories burnt of the user. Further research is required to study the effect of these factors on the calories burnt while taking the stairs. Besides, the energy saved from taking the stairs is only an assumption. A real study on the energy consumption of the lifts in the two buildings is necessary, in order to calculate the actual energy saved. Hence, the accuracy of the calories burnt and energy saved can be improved by conducting more research. Other than that, the app only can read the historical records that are stored in the devices. Thus, if the user clears the app data, all the historical records will be deleted. This can be overcome by reading a copy of the historical records from the web server.

For future improvements, a ranking system can be included in the server side to rank the stair usages of the users. This can create competition and the users with the highest usage will be rewarded. Therefore, this could encourage more people to take the stairs instead of using the lifts.

REFERENCES

Adam Dachis, 2011, *Top 10 Ways iOS Outdoes Android* [Online]. Available at: http://lifehacker.com/5804230/top-10-ways-ios-outdoes-android [Accessed: 5 August 2015].

Ann Bednarz, 2010, *Time spent waiting for elevators? 16 years for NYC office workers | Network World* [Online]. Available at: http://www.networkworld.com/article/2208412/data-center/time-spent-waiting-for-elevators--16-years-for-nyc-office-workers.html [Accessed: 12 April 2016].

BBC News, 2011, *Nokia and Microsoft form partnership* [Online]. Available at: http://www.bbc.com/news/business-12427680 [Accessed: 5 August 2015].

Best Health, 2011. Fitness Trend: Stair-climbing your way to the top. *Reader's Digest*. Available at: http://www.besthealthmag.ca/best-you/fitness/fitness-trend-stair-climbing-your-way-to-the-top/ [Accessed: 12 April 2016].

Chilivumbo, C., 2015. Mobile e-learning: The choice between Responsive/Mobile Websites and Mobile Applications for Virtual Learning Environments for increasing access to Higher Education in Malawi. *IST-Africa Conference, 2015*. May 2015 pp. 1–15.

CNET, 2014. *Google will control Android TV, Android Auto and Android Wear video,*

CNET, 2015, *Microsoft's Windows 10 finally has a release date: July 29* [Online]. Available at: http://www.cnet.com/news/microsoft-windows-10-finally-has-a-release-date-july-29/ [Accessed: 5 August 2015].

Florence Ion, 2014, *10 Android features that still make it better than iOS 8* [Online]. Available at: http://www.greenbot.com/article/2686006/10-android-features-that-still-make-it-still-better-than-ios-8.html [Accessed: 5 August 2015].

Google Developers, 2015, *Barcode API Overview* [Online]. Available at: https://developers.google.com/vision/barcodes-overview [Accessed: 16 April 2016].

Hou A-Lin, Feng Yuan and Geng Ying, 2011. QR code image detection using run-length coding. December 2011 IEEE, pp. 2130–2134.

Huy, N.P. and vanThanh, D., 2012. Evaluation of Mobile App Paradigms. *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*. MoMM '12. 2012 ACM, New York, NY, USA, pp. 25–30.

IBM, 2010. IBM Smarter buildings survey: Consumers rank their office buildings. Available at: http://www-03.ibm.com/press/attachments/IBM_Smarter_Buildings_Survey_White_Paper.pdf [Accessed: 12 April 2016].

International Data Corporation, 2015, *Android and iOS Squeeze the Competition, Swelling to 96.3% of the Smartphone Operating System Market for Both 4Q14 and CY14, According to IDC* [Online]. Available at: http://www.idc.com/getdoc.jsp?containerId=prUS25450615 [Accessed: 5 August 2015].

Justin Kahn, 2014, *Google shows off new version of Android, announces 1 billion active monthly users* [Online]. Available at: http://www.techspot.com/news/57228-google-shows-off-new-version-of-android-announces-1-billion-active-monthly-users.html [Accessed: 5 August 2015].

Laurence Moroney, 2015. Barcode Detection in Google Play services. *Android Developers Blog*. Available at: http://android-developers.blogspot.com/2015/08/barcode-detection-in-google-play.html [Accessed: 16 April 2016].

Liu, Y., Yang, J. and Liu, M., 2008. Recognition of QR Code with mobile phones. *2008 Chinese Control and Decision Conference*. July 2008 pp. 203–206.

NearFieldCommunication.org, *About Near Field Communication* [Online]. Available at: http://www.nearfieldcommunication.org/about-nfc.html [Accessed: 14 April 2016].

Nicole Stinson, 2016. Want to keep your brain young - try these two things. *Daily Express*. Available at: http://www.express.co.uk/life-style/health/652070/brain-health-linked-stairs-education [Accessed: 12 April 2016].

Riekki, J., Sanchez, I. and Pyykkonen, M., 2012. NFC-Based User Interfaces. *2012 4th International Workshop on Near Field Communication (NFC)*. March 2012 pp. 3–9.

Shah, S., O'Byrne, M., Wilson, M. and Wilson, T., 2011. Elevators or stairs? *CMAJ : Canadian Medical Association Journal*, 183(18), pp.E1353–E1355. Available at: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3255141/pdf/183e1353.pdf [Accessed: 12 April 2016].

StepJockey, *About StepJockey* [Online]. Available at: https://www.stepjockey.com/about-us [Accessed: 16 April 2016].

Sustainable Energy Authority of Ireland, 2014, *What are the carbon emission factors used?* [Online]. Available at: http://www.seai.ie/Your_Business/Public_Sector/FAQ/Energy_Reporting_Overview/What_are_the_carbon_emission_factors_used.html [Accessed: 17 April 2016].

The QR Code Generator, *The QR Code Generator* [Online]. Available at: https://www.the-qrcode-generator.com/ [Accessed: 15 April 2016].

The Siver Times, 2016. Climbing stairs would maintain the brain. *The Siver Times*. Available at: http://sivertimes.com/climbing-stairs-would-maintain-the-brain/18609.

University of Birmingham, 2011, *Stair climbing reduces energy bills* [Online]. Available at: http://www.birmingham.ac.uk/news/latest/2011/03/07Mar-Stair-climbing-reduces-energy-bills.aspx [Accessed: 17 April 2016].

Wang, D. et al., 2014. Using humans as sensors: An estimation-theoretic perspective. *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. April 2014 pp. 35–46.

ZXing, 2016, *Official ZXing ("Zebra Crossing") project home* [Online]. Available at: https://github.com/zxing/zxing [Accessed: 16 April 2016].

APPENDICES

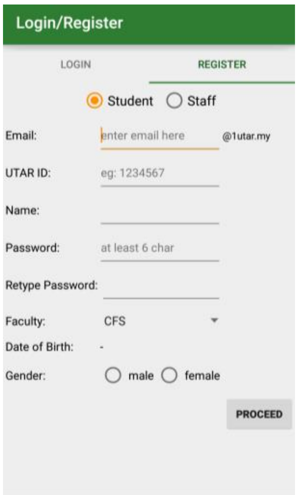
APPENDIX A: User Manual

The hardware requirements to use this mobile app:

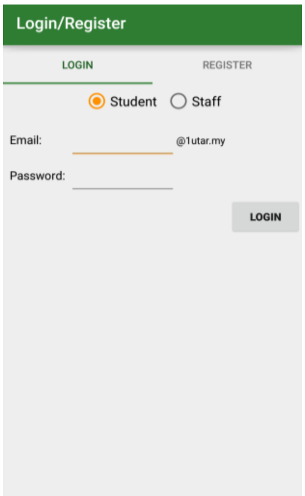
- Android 4.0.3 and above
- 512 MB RAM and above
- Minimum 1 GHz Processor

Getting Started

1. Fill in your details and register as user.



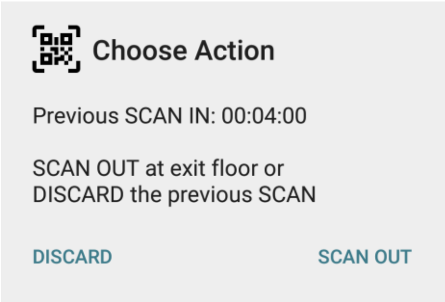
2. Login using your registered email.



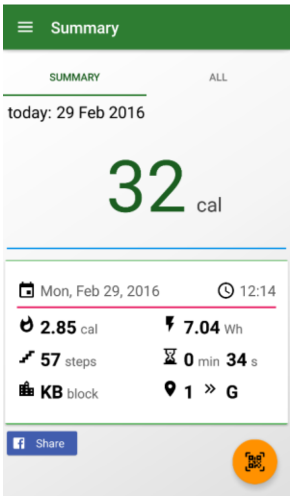
Record Stair Trip

1. Select the QR code scanner button to start recording.

2. When reach your exiting floor, select again the QR code scanner button.
3. You can choose your action from the pop up screen.
   - Discard is to cancel the entering floor scan.
   - Scan Out is to complete the stair trip by scanning the QR code at exiting floor.

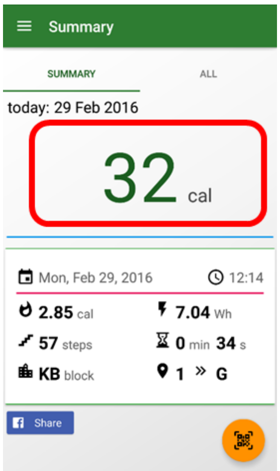


4. Your latest stair trip record is shown at the homepage.

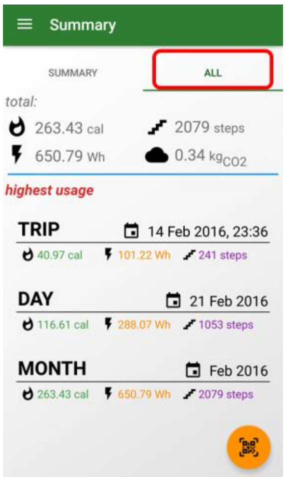


Track Stair Usage

1. Your today stair usages.



2. Select the All tab can get your total stair usage and highest usages.



3. Select the Trends from the navigation menu.
4. Stair usages can be displayed in daily or monthly by selecting the respective tab.

5. Type of data to display can be toggled.