

Software Requirements Specification

for

Walkera

Version 1.2 approved

Prepared by Team Power Ranger

Team Power Ranger

13 April 2017

Table of Contents

Introduction	4
Purpose	4
Document Conventions	4
Intended Audience and Reading Suggestions	4
Product Scope	4
References	5
Overall Description	5
Overall Product Perspective	5
Database Perspective (Persistent Data)	6
Server API Perspective	6
Product Client Function	7
User Classes and Characteristics	8
Operating Environment	8
Design and Implementation Constraints	8
User Documentation	9
Assumptions and Dependencies	9
External Interface Requirements	10
User Interfaces	10
Hardware Interfaces	12
Software Interfaces	12
Communications Interfaces	13
System Features	14
Login	14
Plan Route	15
View User Information	16
Other Nonfunctional Requirements	16
Performance Requirements	16
Safety Requirements	17
Security Requirements	17
Software Quality Attributes	17
Business Rules	18

Revision History

Name	Date	Reason For Changes	Version
All Members	13/4/2017	Populate All Content	1.0
Chew Zhi Jie	13/4/2017	Updated 2.5 & 2.5 , Deleted additional dependencies	1.1
Chew Zhi Jie	14/4/2017	Added 2.2 & 2.3	1.2

1. Introduction

1.1 Purpose

The purpose of the software requirement specification is to provide supportive information on Walkera. The document will also explain the purpose, features, interface of the application. Furthermore, the document will also present a detailed description about operating environment, design and implementation constraint of walkera.

1.2 Document Conventions

The format of the software requirement specification is simple. Boldface font and indention is being used for general topic title. The remainder of the document will be written using the standard fronts, normal text arial, size 11. Italic font symbolise an important naming or entity for the application.

1.3 Intended Audience and Reading Suggestions

This software requirement document is intended for developers, testers and users.

Developer who can review the application capabilities and easily understand which section of the application should be target in order to add new features or improve the capabilities of the application for future enhancement. Developer can view overall description, system features and other nonfunctional requirements for more understanding of the application.

Tester who want to use this document as a guideline for testing strategy as bugs are easier to discover by using a requirement document. This way of testing becomes more structured and organized. Tester can view product perspective, design and implementation constraint, system features, safety requirement and security requirement for base gathering of testing.

Users who are interested to read the software requirement document of what is the main functionality of the application. User can view product function and user interface for the features of the application.

1.4 Product Scope

The product application implementation is Walkera. Walkera is an application which allows the user to clock their daily required steps based on the selected location or steps set by the user. The purpose of Walkera is to integrate exercise into people's everyday life. The objective of Walkera is to promote a healthy lifestyle. The goal of walkera is to integrate exercise into people's lifestyle despite their busy schedule from work. The corporate goals is to promote Walkera to other countries and not just to singapore by the year 2018. This will helps to expand the popularity of Walkera.

1.5 References

For SRS Documentation, we referred the website information on how to design suitable format and standard. The reference is shown below:

Ubuntu 16.04 - <https://help.ubuntu.com/its/ubuntu-help/index.html>

Apache - <https://httpd.apache.org/docs/2.4/>

MySQL - <https://dev.mysql.com/doc/refman/5.5/en/>

PHP - <http://php.net/docs.php>

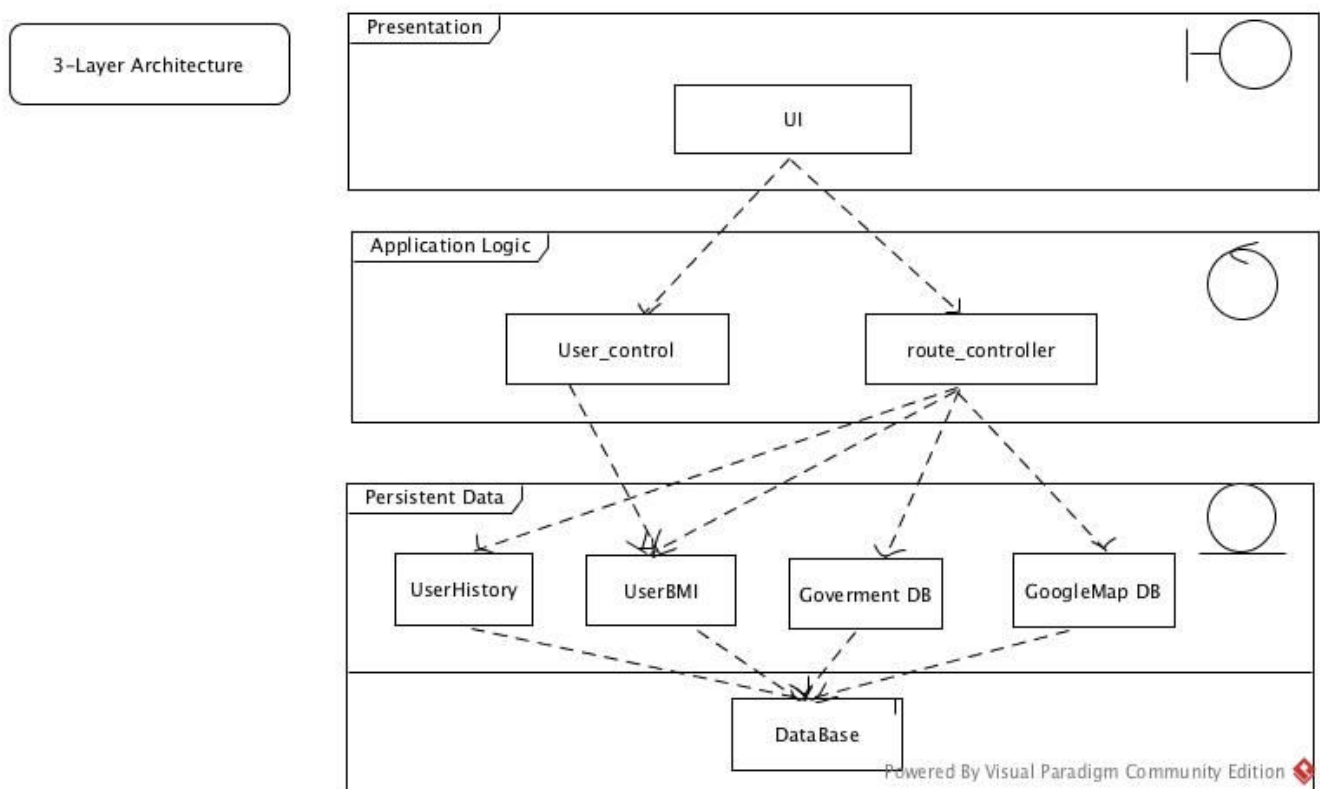
Laravel 5 - <https://laravel.com/docs/5.4/>

Google Map API - <https://developers.google.com/maps/>

Singapore Gov Data - <https://data.gov.sg/>

2. Overall Description

2.1 Overall Product Perspective

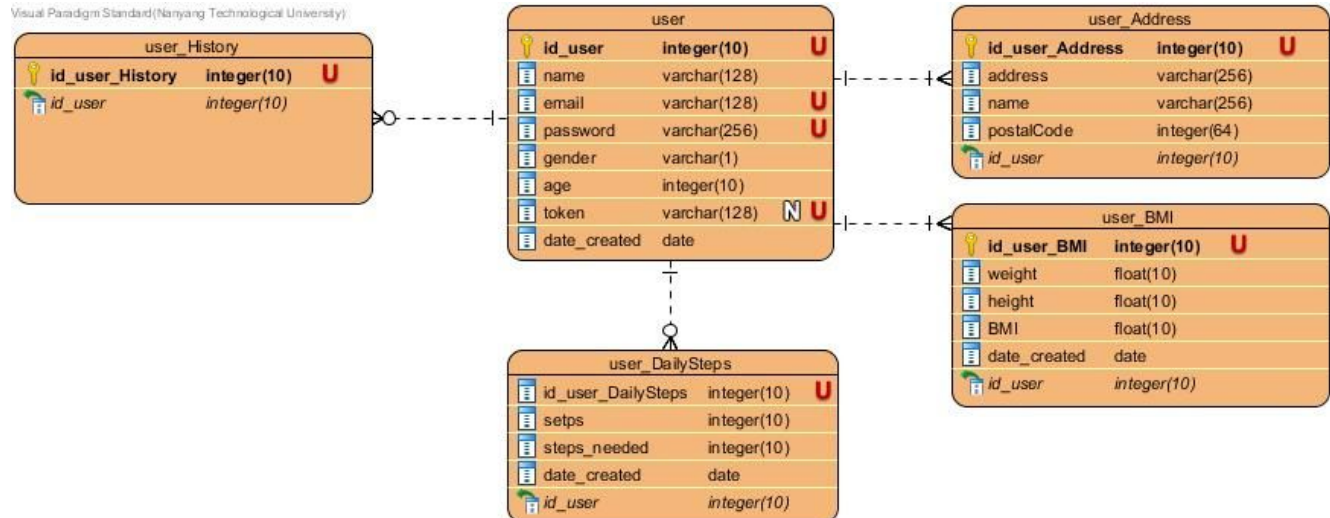


Our product is a self-contained product. But we access Google API to request route information and also access Government Database API to get traffic, weather and etc data. But what we do, is just refer to this information and use our own algorithm to design a most suitable healthy route for the user. To implement this two API we used design pattern so the API become easy and safe to

Copyright ©2017 by Team Power Ranger. Permission is granted to use, modify, and distribute this document.

update. And as you can see at the architecture diagram we designed our own UI, Application Logic and user database.

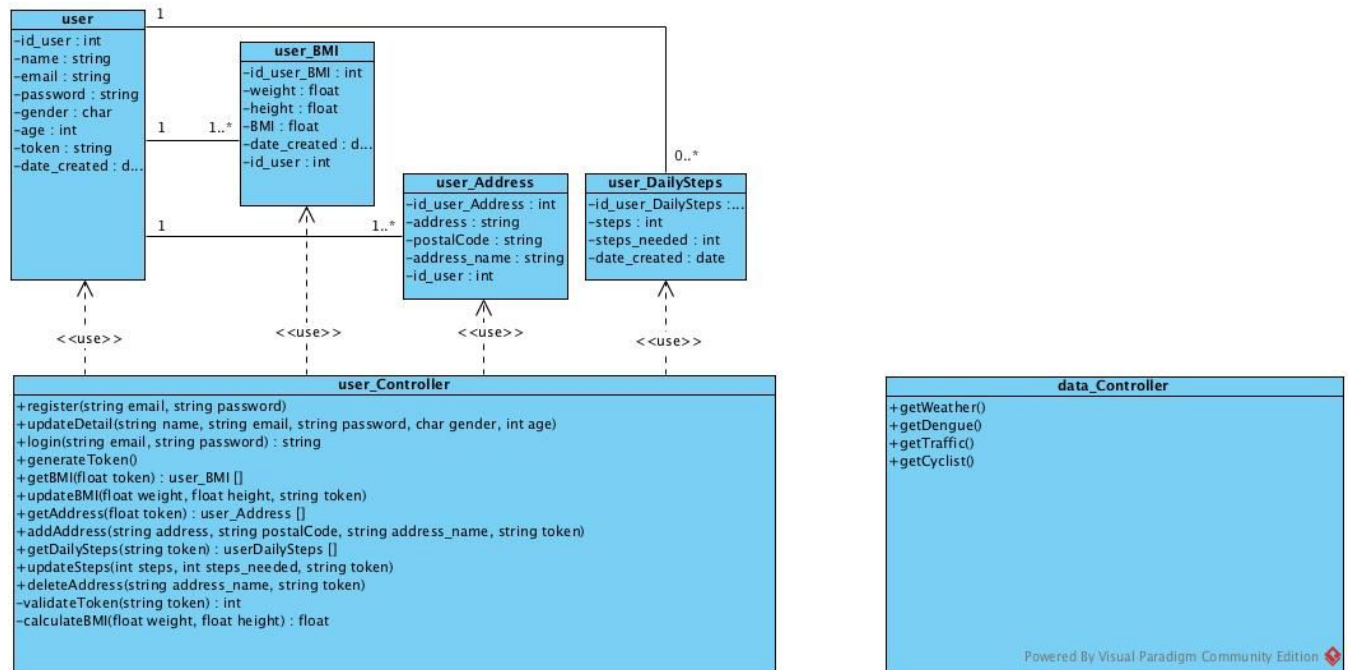
2.2 Database Perspective (Persistent Data)



The relational database design has *User* table as the heart of all data. A *User* consisting of one or more *Address* and *BMI*. The *User* can also have no travel *history* and *daily steps* data.

The above database relation is then implement in Laravel migration which is a type of version control for your database. They allow a team to modify the database schema and stay up to date on the current schema state.

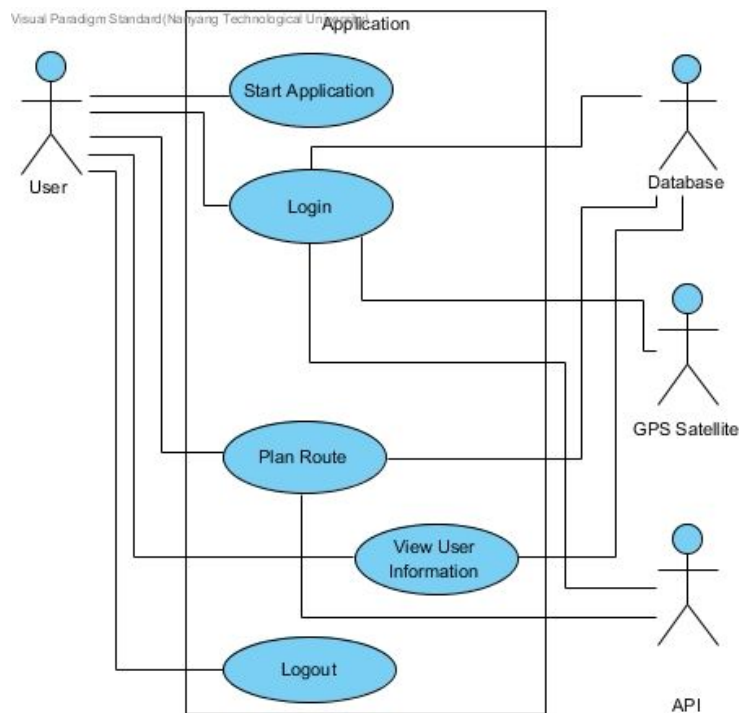
2.3 Server API Perspective



To further simplify the development. Each table entity is then created as a Model Entity for the application controller to manipulate. The underlying Model Entity like User will then manipulate the database directly. This also provides a protection layer for SQL query integrity which mitigates dangerous queries and SQL injection.

The data Controller controls and caches the government Data API which is used to calculate optimal route for the user.

2.4 Product Client Function



1. User Register
 - a. Email format check and password length check
 - b. Update User information to User Database
2. User Login
 - a. Check User is registered
3. Plan Route
 - a. Design Path depend on input start ,destination point and desired step number.
 - b. Access useful API
4. View User Information
 - a. Display Username, Age ,Gender,BMI depend on user database.
5. Progress shown
 - a. Track User steps by pedometer and location by GPS
 - b. Show different label when pre-set goal reaches
6. Logout
 - a. User exit the application

2.5 User Classes and Characteristics

We generate 6 user classes.

1.Obese children and teenagers(under 18):Need exercise but have sports class and parents will control their daily route because of secure issues.

2. Other children and teenagers (under 18): Need less exercise. And have sports activities at school. It's not so safe to walk on the road alone.
3. Obese adult: Most important class. Since they need exercise and they need to work so don't have enough time to exercise. And they can control their daily route freely. No secure issues.
4. Other adult: Need less exercise. Have less time to exercise.
5. Obese Elders: It's better to have more exercise for health issue but the body condition may not support long time exercise.
6. Other Elders: Motivation to exercise is the least one.

2.6 Operating Environment

1. Android for Smartphone Application
 - Minimum SDK Version 19
 - Targeted SDK Version 25
2. Dependencies
 - appcompat-v7:25.0.0
 - design:25.0.0
 - facebook-android-sdk:[4,5)
 - volley:1.0.0
 - support-v4:25.0.0
 - constraint-layout:1.0.2
 - support-vector-drawable:25.0.0
 - androidplot-core:1.4.1
 - play-services-location:10.0.1
 - play-services-places:10.0.1
 - googledirectionlibrary:1.0.5
 - discrete-seekbar:1.0.1
 - gson:2.7
- API Server (Host under Azure A0-Linux VM)
 - Ubuntu 16.04
 - Apache 2.4.25
 - MySQL 5.5.45-log
 - PHP 7.1.3
 - Laravel 5.4

2.7 Design and Implementation Constraints

1. Walkera requires several functional API module to function properly, Walkera uses concrete strategy pattern design which allows simple modulo add and remove. Following are Walker's third party API dependencies:
 - Map API such as Google Map API
 - Route API such as Google Route API
 - Facebook API for login via facebook
2. Walkera requires a RESTful web application that is residing on the cloud to function properly. The RESTful web application act as an API which expose functions via HTTP URL for Walkera user client to manipulate and retrieve relevant data.

2.8 User Documentation

Our user documentation will consist user manual, system reference, on-line help, tutorials and an video description.

To make the structure of the user documentation is clear and readable. Format requirements are:

1. Proper indentation
2. Insert headings, page number and section titles
3. Integrate text fonts and style (Times new roman, 14)
4. Use columns to layout
5. Insert picture for key steps and buttons
6. Use bold to highlight dangerous or invalid operations

Since our users most not have high technical background so the standards are:

1. Simple words and short sentences only
2. More short Verb+Noun to show operations instead of explaining a long process
3. If use professional word or name, paste a link at the end
4. Different languages

2.9 Assumptions and Dependencies

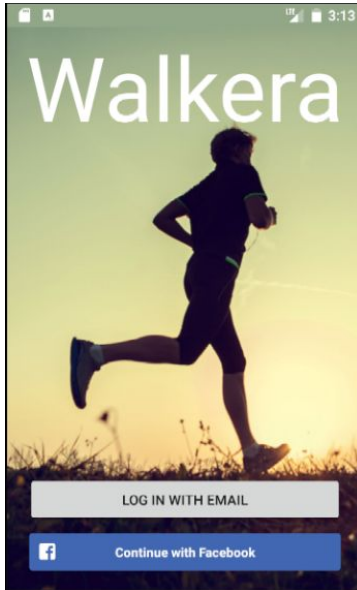
Assumptions: Walkera assume that user are aware of the surrounding while navigating using Walkera. The route showed might sometime be outdated or inappropriate, user is responsible for her own safety when navigating back to the destination.

Besides, user has the integrity to input her personal information such as email, name, age, gender, height and weight correctly for optimal result.

Dependencies: Walkera needs the Google Map and Government Database. We design to API to access these two Actors. For Google Map, after user input start and destination point, we need it to return some routes and we will design an exercise path (walking+transportation/ driving) for user depend on our algorithm. For Government database, we need weather and traffic conditions to choose a most suitable route for user.

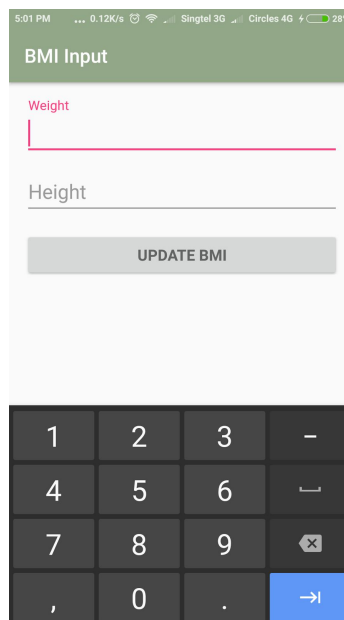
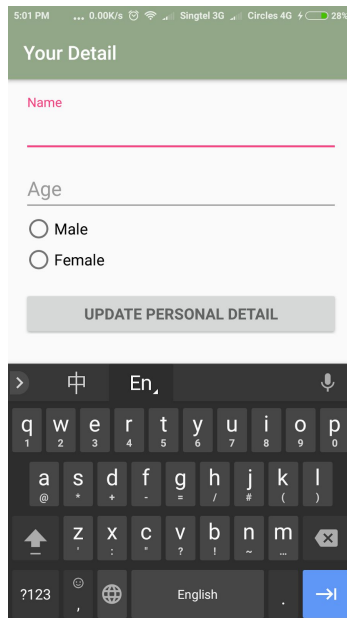
3. External Interface Requirements

3.1 User Interfaces

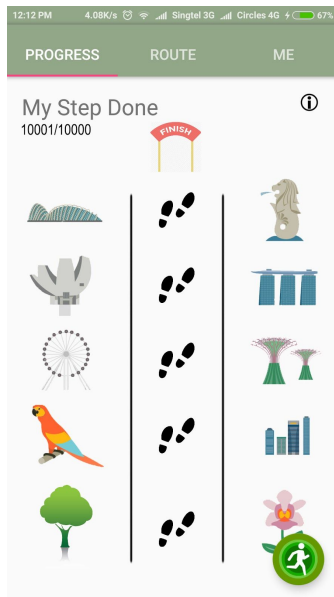


Register Page - A landing page for user who just download Walkera. It consists of Login / Register with email or Facebook.

Personal detail and BMI input page - The Personal detail Page is to provide user a more personal experience when using Walkera. An accurate Weight and Height greatly optimise the result of using Walkera.



Main Landing Page

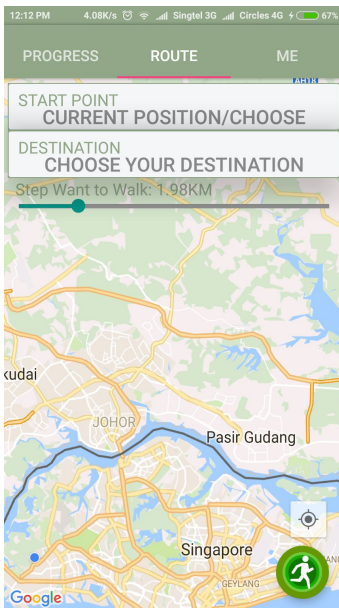


Progress page - Your daily steps achievements using graphical representation which each Singapore's symbolised image will show when user's daily step count reaches certain count.

Currently, it is divided into 2k, 4k, 6k 8k, 10k steps count.

The motivation of creating a graphical representation is to motivate user to try to hit their personal daily steps quota.

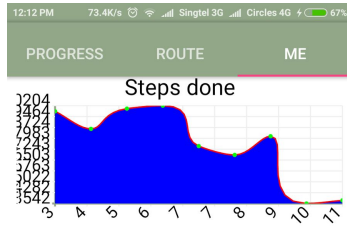
The daily steps quota is unique to each user. It is calculated using your BMI and age.



Route page -

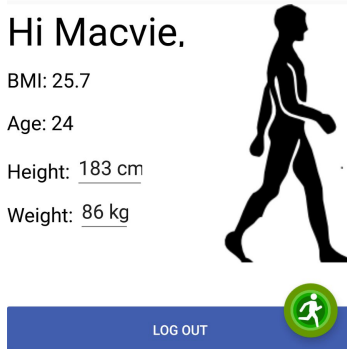
Enable user to either use either their current location or choose their current position as the start point and also choose the destination user desired to go.

The discrete seeking bar below allows user to select the desired distance wanted to walk before taking any transport to reach the destination.



Me page - Show your pass one week steps statistic by using a graph. This gives user a good indication on her walking trend. The steps count is uploaded to the database at 2359 on a daily basis which will reset the steps count at 0000 on the next day.

Below is the main reason Walker asked for user's personal data. This provide user a more personal User Experience.



Walkera will prompt user to update her height and weight on a bi-weekly basis to give user's optimal result. Besides, in the future development, Walkera can show's user BMI trend for user to understand her body condition more.

3.2 Hardware Interfaces

- Android Smartphone - Presentation Layer in 3-Layered Architecture
- Server API (Microsoft Azure) - Application Logical in 3-Layered Architecture
- MySQL Database Server - Persistent Data in 3-Layered Architecture

3.3 Software Interfaces

- Android Smartphone
 - Android Facebook SDK for facebook login
 - Android Core Plot for plotting personal daily steps quota graph plotting
 - Google Direction Library to query for steps count and location to location distance
 - Google Gson to decode Google JSON file
 - Google Map API to show the map on Route page
 - Discrete seekbar for the seekbar that is located in the Route page
 - Android Volley for HTTP POST to communicate with the server API
- Server API
 - LAMP Stack
 - Ubuntu 16.04 Operating System
 - Apache web server software
 - MySQL DB Engine for persistent data
 - PHP
 - Laravel 5 Web API Framework

3.4 Communications Interfaces

- HTTPS Communication (Mobile Application to Server API)
 - HTTP encrypted by TLS which provides authentication of the website and associated web server with which one is communicating, which protects against man-in-the-middle attacks.
 - Additionally, it provides bidirectional encryption of communications between a client and server, which protects against eavesdropping and tampering with or forging the contents of the communication.
 - Mostly HTTP Post function to do functions like login and update.
 - HTTP Get function to retrieve route for route planning
- SFTP to edit and update API's code
 - Login via FileZilla client to access the directory content and files that is reside on the API server itself. Update and add new code to suit Walkera's feature
- MySQL database server connection (Port: 3306)
 - Laravel 5 will open a active DB connection to select, add, update and etc. when a function is called via HTTP route.
 - Access the database via MySQL Workbench to debug web app

4. System Features

4.1 Login

4.1.1 Description and Priority

Executes login procedure to establish connection of account and the server. Retrieve required information from the database. If account has been logged in before, validate login information and move to main page. (High Priority)

4.1.2 Stimulus/Response Sequences

User Action : Enters login information

System Response : Validates login information and asks user to enter personal information

—

User Action : Enters personal information

System Response : Saves personal information entered into the database

—

User Action : Start Application (If not first login)

System Response : Validate login information and retrieve personal information from the database ,GPS information,API information, then move to main page

—

User Action : From the main page after login, press the" route" tab

System Response : Move over to the "Plan Route" page.

—

User Action : From the main page after login, press the "me" tab

System Response : Move over to the "Personal Information" page.

—

4.1.3 Functional Requirements

REQ-1: The application must be able to retrieve and compute information and data within 20 seconds

REQ-2: The application must be able to login within 10 seconds.

4.2 Plan Route

4.2.1 Description and Priority

User enters the “Plan Route” page, where they plan a route to their desired location. The weather, traffic and transport API information previously retrieved will be used to help guide the user to their destination while avoiding heavy traffic and bad weather. (High Priority)

4.2.2 Stimulus/Response Sequences

User Action : Adjust the slider bar that allows user to change distance to walk.

System Response : Adjusts route to accommodate distance that user wants to walk.

—

User Action : Press the start point bar/button

System Response : Allows user to choose to start from either current location or choose their own starting point

—

User Action : Press the destination bar/button

System Response : Allows user to enter their destination

—

User Action : Press the “me” tab

System Response : Move over to the “Personal Information” page.

—

User Action : Press the “progress” tab

System Response : Move over to the “Progress” page.

—

4.2.3 Functional Requirements

REQ-1: A route must be generated within 15 seconds.

REQ-2: The route must use the correct starting point and destination.

4.3 View User Information

4.3.1 Description and Priority

This page allows user to view their BMI,age,weight,height and recent walking trend.

4.3.2 Stimulus/Response Sequences

User Action :Press the logout button

System Response :All personal information is saved to the database and the user is logged out.

—

User Action : Press the “route” tab

System Response : Move over to the “Plan Route” page.

—

User Action : Press the “progress” tab

System Response : Move over to the “Progress” page.

—

4.3.3 Functional Requirements

REQ-1: Information displayed must be consistent with what was entered by user

REQ-2: Logout must be completed within 15 seconds.

4.4 Other Nonfunctional Requirements

4.5 Performance Requirements

- The application system database must be able to store up to a year of user’s path history.
 - Enables users to track their own path history and to select their previous routes.
- The application system database must be able to store 1000000 user accounts personal information.
 - Supports multiple users using the application.
- The application must respond to the user’s input within 2 second.
 - Reduces the waiting time of the users to improve user experience satisfaction.
- The application must be able to support up to 5000 concurrent active query and still respond

under the best user experience time.

4.6 Safety Requirements

- Users not looking out for vehicles while crossing the road
 - The application will show a warning message once the user logs in to remind them to look out for vehicles while crossing the road.
 - The application will use information on the traffic conditions to plan a route for users.
- Users walking into restricted area
 - The application will exclude restricted area when planning of route.

4.7 Security Requirements

- Account
 - Users will have to create an account or use a facebook account in order to use the application.
 - Account will be tagged to the user.
- Account Information
 - Users will only have access to their own personal details and route information.
 - User will only be able to edit their own information.
- Account Password
 - The minimum length for the password will be 8.
 - Users will only be able to change their own account password.

4.8 Software Quality Attributes

- Usability
 - The application must be able to support English language.
- Reliability
 - The application must be able to operate normally after a reset.
 - The application must not be unresponsive for 98% of the usage time.
 - The application must not crash at any point in time at all.
- Portability
 - The application's database must be able to make use of the user's smartphone memory and cloud drive memory for storage of data.
 - The application's database must be able to migrate information to another mobile device easily.

4.9 Business Rules

Anything that consist of private data can only be accessed by the individual user that is tagged by an account. As such, even the administrator will not be able to access the user's' account.

- Application users
 - Access to own account details
 - Edit own account details
 - Plan route
 - View own path history
- Administrators
 - Manage accounts