

Software Requirements Specification

for

EZ Tracker

Version 1.0 approved

**Prepared by Jonas Gamao
Ryan Maynard
Delroy Christie**

JRD Developers

September 27, 2018

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	Error! Bookmark not defined.
2.5 Design and Implementation Constraints	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	4
3.1 User Interfaces	4
3.2 Hardware Interfaces	5
3.3 Software Interfaces	5
3.4 Communications Interfaces	6
4. System Features	6
4.1 System Feature 1	6
4.2 System Feature 2 (and so on)	7
5. Other Nonfunctional Requirements	7
5.1 Performance Requirements	7
5.2 Safety Requirements	7
5.3 Security Requirements	8
5.4 Software Quality Attributes	Error! Bookmark not defined.
5.5 Business Rules	8
6. Other Requirements	

Revision History

Name	Date	Reason For Changes	Version
EZ Tracker Mobile App	27/09/2018	First version	1.0

1. Introduction

1.1 Purpose

The EZ Tracker app is a fitness tracking and monitoring system developed for the elderly, cyclists, and users who find setting things up a waste of time. With simple and advanced modes, this gives users the choice of how much information is provided to them; with simple mode being more focused on easy-to-read information, while the advanced mode focuses on more comprehensive data that is filled with charts and graphs.

The Android application also has a server component, where the user will be able to view the data in the web browser.

The database software will be Google's Firebase which will be filled with the user's information and their activity metrics.

1.2 Document Conventions

The app will remain consistent throughout with the font styles, size and color. The specific font styling is chosen because of the target market. Older folks generally have vision and hand-eye coordination issues, and so the app's user interface is designed with those issues in mind. Button style, size, and position is also consistent throughout the app making for a familiar and easy to navigate experience for the users.

1.3 Intended Audience and Reading Suggestions

This app is developed primarily for people who are not very comfortable using a smart phone or other mobile device, and who are not tech savvy enough to navigate complex interfaces. As such, with its simple but elegant user interface, the app is perfect for the elderly as well as cyclists who use pedal cycles as their primary mode of exercise and who need easy-to-read data.

1.4 Product Scope

The proposed EZ Tracker system will be used to check, track, and monitor users' steps and distance travelled while calculating calories burned during the process. Users will be able to keep track of their progress on a daily, weekly, and monthly basis. The apps that are currently available are tedious and cumbersome, especially for older people and pedal cyclists. The main aim of this system is to reduce the guess work and complexities in manoeuvring the application to obtain the information needed. The time it takes to check data and compare results will be minimal, as all the data will be readily accessible without any hassle. These data can also be automatically emailed to the user's personal email account for additional archive.

This EZ Tracker system will allow the user to view current and past data, mount the system to their bikes, allow the user to set goals for the future, and reward the user when a goal is accomplished. The system shall be able to be used anywhere in the world, even without the use of internet access. It can also be customized to suit the users' needs. Since it is a mobile

app it will be available on both smart phones and tablets, provided they are Android 5.0 and above.

1.5 References

This app shall use external APIs for additional functionalities. For example, google maps is incorporated for helping users to navigate their journey. The link to google login API is found [here](#). The user shall also be able to login using his/her [google](#) or [Facebook](#) account.

2. Overall Description

2.1 Product Perspective

This product was developed out of a specific need of the market. It is a new, self-contained product, the first version in its family. Its goal is to compete against fitness trackers that simply aim to add loads of features just to make them look like they are improving, but as in effect making them more and more complex.

2.2 Product Functions

The major features of this application include:

- **Multiple modes:** this allows the user a choice of how much data is presented to them.
 - Simple mode presents information in easy-to-read text with 1 graph or chart, depending on the information that the user selected to see in more detail.
 - Advanced mode presents information in compact text and multiple graphs and charts.
- **Step counting:** a feature where the app will count the number of steps a user has done. It can be turned on or off, depending on the activity (i.e, turned off for cycling)
- **Distance traveled:** a feature where the distance traveled is calculated based on a start and destination point.
- **Calorie tracking:** a function to calculate the calories burned based on distance traveled and steps taken.
- **Speed:** the feature where the speed at which the user traveled to accomplish his specific goal is calculated.
-

2.3 User Classes and Characteristics

- **Elderly**
 - Simple interface allowing for easy use
 - Large font options for readability
 - Most important user class due to the rising elderly population and with health/fitness on the minds of most people.
- **Power Users**
 - Advanced interface available for the tech-savvy users.
 - Detailed statistics for the avid health tracking user.

- **Cyclists**
 - Robust hardware allows for user to never have to take out his/her phone while cycling.
 - Can monitor basic data on the built-in display.

2.4 Operating Environment

The software will operate in conjunction with a Broadcom development platform (Raspberry Pi) as the main engine, a capacitive touch sensor/effector, an accelerometer, and an OLED monochrome display; powered by the Android 5.0 operating system as the mobile client. On the back end, the system will be linked to Linux based web server and Google's Firebase database for data storage and retrieval.

2.5 Design and Implementation Constraints

The program will be created using the Java programming language with Android libraries. Any smartphone that is capable of running Android 5.0 Lollipop will be able to use this app.

The following constraints can pose a possible threat to the complete functionality of the system:

- Users API's version may not be up-to-date to install the application
- Device may not have enough disk space for the installation (mobile app)
- Google play store may not be available in the user's demographic location
- Other mobile constraints may prevent the installation of the app

2.6 User Documentation

As the app is still in the development stages, all future documentation/instructions will be found at: https://github.com/YamiYukiSenpai/EZ_Tracker/tree/master/documentation

2.7 Assumptions and Dependencies

While the system is fairly simple to use, the assumption can be made that the system may be affected due to certain situations. These include but are not limited to:

- Browser compatibility- the system might not be able to run an outdated browser. Thus, users are being recommended to have the most current version of the browser installed on their device.
- OS Compatibility- The system might not be able to operate on versions of Android that are older than Lollipop, which has been deemed outdated by the developer.
- Internet connection not available.

For the system to be fully functional and effective, it will be dependent on the following:

- Internet Connection- while internet access is not absolutely necessary for the system to function, it does require internet access at some point to interact and retrieve data from the database stored on a web server.

3. External Interface Requirements

3.1 User Interfaces

Figure 3.1 will be the first page a new user will see. **Blue circles indicate clickable buttons/links.**



A mockup of a login screen. It features a 'Login' label above a text input field. Below this is a 'Password' label above another text input field containing seven asterisks. A checkbox labeled 'Remember Me' is positioned below the password field. At the bottom, there are three links: 'Login', 'Register', and 'Forgot Password?'. Each of these three links is highlighted with a blue circle, indicating they are clickable.

Fig 3.1 Login Screen

The user can register for an account if he/she does not already possess one shown in **Figure 3.2** by clicking the Submit button and filling out the required information. The back button at the top right sends them back to the login screen.



A mockup of a register screen. At the top left is a back arrow icon and a 'Register' label. Below the label are several form fields: 'E-mail', 'Full Name', a gender selection with radio buttons for 'Male' and 'Female', 'Age', 'Height', 'Weight', and 'Birthday'. The 'Birthday' field is a date picker showing a calendar grid with months (SEP, OCT, NOV) and years (2011, 2012, 2013). At the bottom right is a 'Submit' button, which is highlighted with a blue circle, indicating it is clickable.

Fig 3.2 Register

From the login screen shown in **Figure 3.1** the user can request a password reset if they have forgotten it by filling out the required information shown below.

The screenshot shows a mobile application interface for resetting a password. At the top, there is a title bar with a back arrow and the text 'Reset Password'. Below this is a 'Login' input field. Underneath is a 'Birthday' label followed by a date picker showing 'SEP 06 2011'. Below the date picker are two input fields: 'New Password' and 'Re-enter Password', both containing masked text (asterisks). At the bottom of the form is a 'Submit' button.

Fig 3.3 Reset Password

3.2 Hardware Interfaces

The EZ Tracker app interacts with a few hardware components to accomplish its tasks. Firstly, system is powered by a Broadcom development platform (Raspberry Pi B+) as its main engine. It also uses a small 128x64 SSD1306 OLED Display for simple, on the fly, statistics. Tracking data is made possible by the LSM303 Accelerometer and Magnetometer, while the MPR121 12-Point Capacitive Touch Sensor offers seamless interaction between the system and the user. Data collected will be recorded to the Trackers local flash SD card storage, and the online database. This will be read to the small built-in screen as well as displayed into the EZ Tracker application. Supported Android devices include Android 5.0 and above.

3.3 Software Interfaces

Connections for the EZ Tracker

- Database: Google's Firebase will be used as the database. Used for storing tracking metrics such as direction, speed, steps, etc. Will only be an outbound connection.

Connections for the EZ Tracker application

- Database: Inbound communication from Firebase. Will display information into the application and display.

- Operating System: Internet Connection (HTTP) to send a request and receive e-mail notifications/instructions for password reset (SMTP).

3.4 Communications Interfaces

EZ Tracker will employ communication through a number of interfaces. The first method being the database between the Tracker and the phone. The logs will be stored on the database from the Tracker and corresponding statistics from the database onto the app (via HTTP/Internet). Secondly, users will have their e-mails tied to their accounts on the app. This will allow email notifications to be received without storing any mobile number information (via SMTP). E-mail notifications can be disabled. A feedback form will also be available from the settings menu for users to contribute any feedback to the developers.

4. System Features

4.1 Simple Mode

4.1.1 Description and Priority

This mode will be the main attraction point for the app. In general, users just want their product to work. It will display the simplest forms of data required to be considered a health tracker. At a glance, users will be able to see all of the information they need to.

Priority Level: High

Risk: 2

Cost: 2

4.1.2 Stimulus/Response Sequences

Preconditions: The user is logged in and the device is connected to the tracker, and the app has not been changed to Advanced Mode.

1. User wishes to see their up to date tracker stats.
2. User opens EZ Tracker and the Simple Mode page is displayed (calories, steps, goals, etc.).
3. User is able to view the basic stats they want to see for their general health/activity.

4.1.3 Functional Requirements

REQ-1: must download and install app from the Google Play Store.

REQ-2: Must be running android 5.0 with at least API 27 on mobile device

REQ-3: User must be registered and logged into the application

4.2 Advanced Mode

4.2.1 Description and Priority

This mode of the app will provide detailed statistics and metrics for the user. It will be mainly for power users as well as any beginner users that become more familiar with the application.

Priority Level: Medium

Risk: 2

Cost: 4

4.2.2 Stimulus/Response Sequences

Preconditions: The user is logged in and the device is connected to the tracker.

1. User opens the app to view their tracker stats.
2. User is curious/wants to see extra metrics/statistics on their activity in the past week/month.
3. User enters the settings menu by tapping the icon in the top right corner.
4. Upon tapping the advanced mode button and pressing apply, the user will be sent to the advanced screen where in depth statistic will be displayed.

4.2.3 Functional Requirements

REQ-1: must download and install app from the Google Play Store.

REQ-2: Must be running android 5.0 with at least API 27 on mobile device

REQ-3: User must be registered and logged into the application

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Upon opening the application on the device. The user will be able to see their up-to-date statistics within 5-10 seconds. The OLED display on the EZ Tracker will continuously update in real time since the numbers are local. In the case of querying the database, connection can depend on the users signal strength and will timeout if not successfully refreshed after 20 seconds.

5.2 Safety Requirements

Database corruption

The information in the database can be back up with the master-slave method in order for users to retain their data in case of data loss.

5.3 Security Requirements

The mobility domain has a privacy sensitive nature, specifically with regards to the location tracking of users. In order to create a viable offering for the user we will build a simple, transparent system that can be understood and trusted by the people that are using it.

In order to build trust with the users of our system, the system shall make use of the following strategies:

- Anonymization & aggregation, so that information may be shared safely without disclosing personal information.
- Encryption, for all data that is privacy sensitive, but must be persisted on the server in order for basic functionality
- Open source / disclose security policies & practices
- Permit the use of anonymous avatars / aliases.
- Give control to end-users over private data (at least a delete private repository option)

5.4 Software Quality Attributes

- **Usability:** checking that the system is easy to use and intuitive for the people not comfortable with current technology.
- **Maintainability:** any crashes regarding the app will prompt the user to send the crash report to the developers.

5.5 Business Rules

Documents or other materials used for this project cannot be used for commercial purposes without the knowledge and consent of the developers.

6. Other Requirements

Currently there are no other known requirements for the project; however, this may change in the event of unforeseen circumstances encountered during the duration of the project.

Appendix A: Glossary

APP – Short form for application

Database – An organized collection of data, stored and accessed electronically.

Android – an operating system designed for mobile devices (i.e. cell phones, tablet computers) by Google, Inc.

Android device – any device running Android. In this document, synonymous to “smart phone running Android.”

Operating System – the software that supports a computer's basic functions, such as scheduling tasks, executing applications, and controlling peripherals.

SSD1306 Monochrome OLED Display – a small display, about 1" diagonal, but very readable due to its high contrast. This display is made of 128x64 individual white OLED pixels, each one is turned on or off by the controller chip.

LSM303 Accelerometer and Magnetometer – a small chip that can determine speed as well as direction (North, South, East, West).

MPR121 12-Point Capacitive Touch Sensor – a touch sensor that can handle up to 12 individual touch pads and can be implemented with nearly any microcontroller.

SD Card – Secure Digital card is a type of memory card typically used in digital cameras and other portable devices.

HTTP – Hypertext Transfer Protocol is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands

SMTP – Simple Mail Transfer Protocol is an Internet standard for electronic mail (email) transmission.