
Software Requirements Specification

for

Mini Explorer

Version 1.0 approved

Prepared by Jordan Pulido, Duc Nguyen

Humber College

9/26/2018

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	3
2.3 User Classes and Characteristics	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	5
2.6 User Documentation	5
2.7 Assumptions and Dependencies	5
3. External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces	8
3.3 Software Interfaces	9
3.4 Communications Interfaces	9
4. System Features	9
4.1 System Feature 1	Error! Bookmark not defined.
4.2 System Feature 2 (and so on)	10
5. Other Nonfunctional Requirements	11
5.1 Performance Requirements	11
5.2 Safety Requirements	11
5.3 Security Requirements	11
5.4 Software Quality Attributes	11
5.5 Business Rules	12
6. Other Requirements	Error! Bookmark not defined.

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to give a full description of the “Mini Explorer” project. This document will provide the purpose and building process for the development of the software. Additionally, the document will explain user interface and interactions within a third party application.

1.2 Document Conventions

In this Software Requirement Specifications, no document conventions are used.

1.3 Intended Audience and Reading Suggestions

The intended audience are Humber faculties, engineering students (CENG students particularly), and potential companies/investors/individuals that may use this project for commercial or non-commercial uses.

Engineering students are recommended to read the whole SRS carefully and the references as well, while others should focus on Chapter 4 and 5.

1.4 Product Scope

The Mini Explorer is designed to travel in linear surfaces and send environmental readings (temperature, humidity) to the user. The software allows users to control the ME and store ER into the database. This product can assist, but not limited to, environmentalists to take ER in particular areas that are hard to reach; firefighters to measure whether the area is safe to enter.

1.5 References

Deitel, P. J., Deitel, H. M., & Wald, A. (2016). Android 6 for programmers: An app-driven approach. Upper Saddle River, NJ: Prentice Hall.

2. Overall Description

2.1 Product Perspective

The system will consist of three parts: The hardware, the mobile software and the online database.

The hardware is made of the RPi 3B+ with motors (PCA9685 16-channel PWM driver), a distance sensor (VL53L0x ToF distance) and a temperature/humidity sensor (HDC1008). The RPi will be the main body of the ME, with the driver and sensors to travel and read ER.

The mobile software will be compatible with Android OS and connect to the RPi through Bluetooth. This connection allows the phone app to receive real-time readings and control the ME in 4 options: Go forward, go backward, turn left and turn right. In order to be able to do so, each user must have an account with encrypted password.

The database is created on Firebase for many uses. First, it is used to store users' credentials. Users' information can include, but not limited to, username, password, and email. Secondly, it is used to store the ER. Last but not least, any crash logs or status reports of the ME can also be found here.

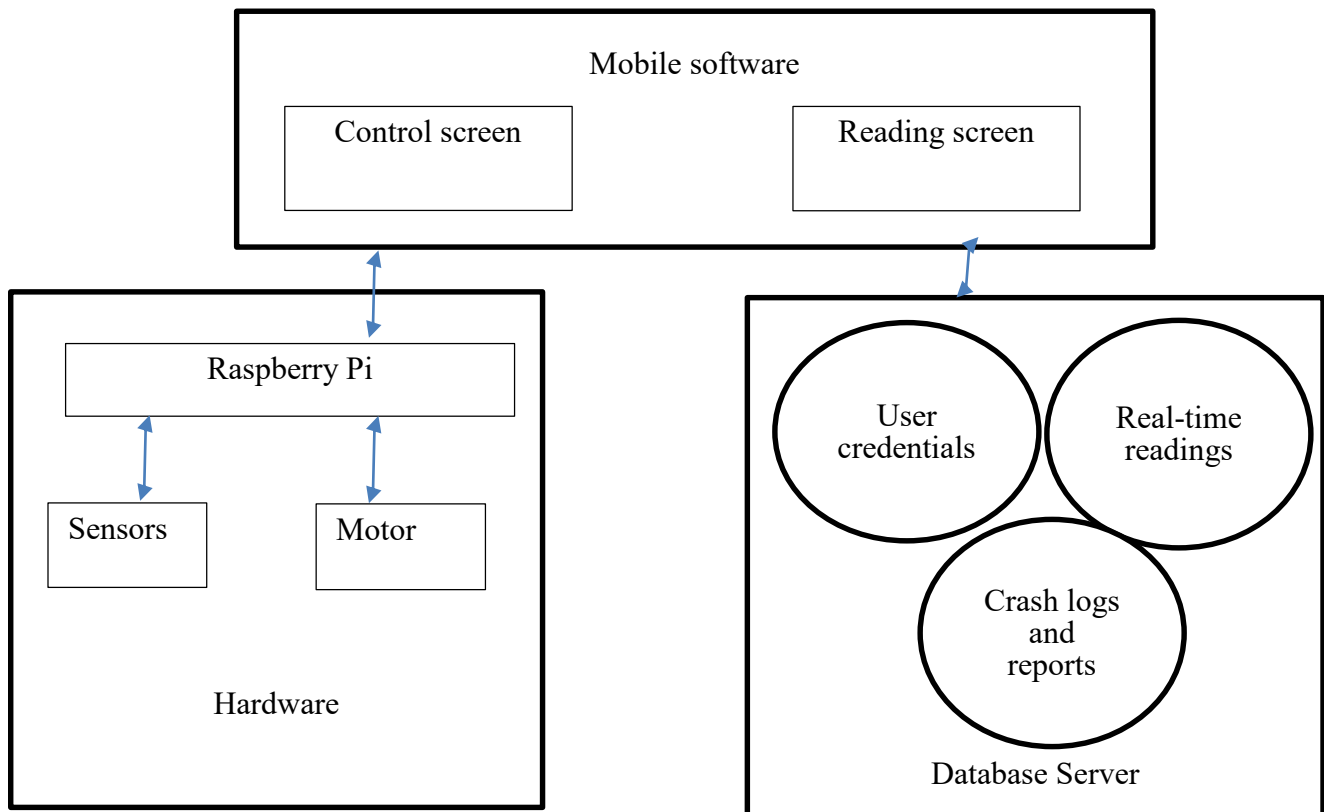


Figure 1 – Block diagram

2.2 Product Functions

- Ability to travel in small areas that may not be accessible by humans
- Read real-time humidity and temperature
- Wirelessly send and receive data, orders from the user

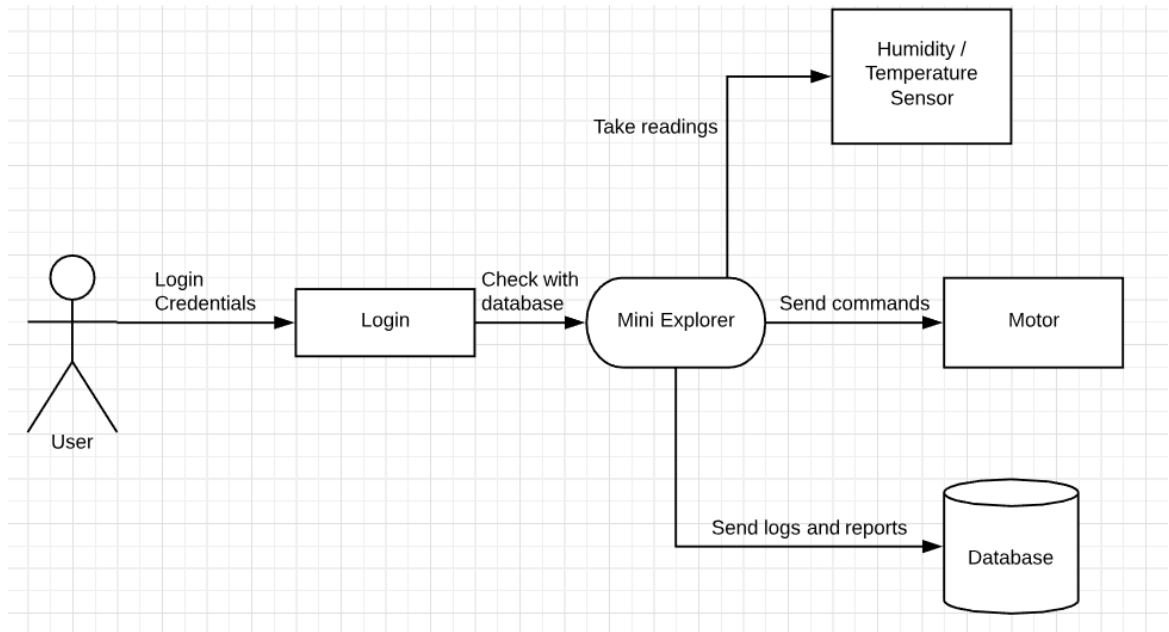


Figure 2 – Data Flow Diagram of the Product Functions

2.3 User Classes and Characteristics

There are two types of users that interact with the remote control car.

The mobile app users interact with the software by controlling the ME, and monitoring the ME's details. They can be casual users or people with expertise in particular scientific fields.

The administrators can interact with the live support screen and software overall. These people have the highest privilege, and are experienced with the device.

2.4 Operating Environment

The hardware part of ME is based on the RPi 3B+, which is compatible to many sensors. The software is written on Android Studio and is compatible with any devices that have Oreo. And the database is created on Firebase.

2.5 Design and Implementation Constraints

The software has a huge dependency on the hardware components. If the hardware is faulty or damaged, the software will not have much uses. Also, troubleshooting the ME requires people with experiences and it is hard to do from distance – physical interactions between the device and the troubleshooter is needed.

The maximum range between the user and the ME can also be a constraint. Since the connection is based on Bluetooth, the distance between both ends should not exceed 100 meters.

Due to the limited experience of the developers, high-level or optional functions/features (GPS, Webcams, ...) can be quite challenging to add in. There can be issues with integrating the app to older Android versions.

2.6 User Documentation

- User manual to control the remote control car using the phone
- On-line help will be accessible in the app
- Tutorials will be provided in the app via lab support

2.7 Assumptions and Dependencies

One major assumption is the hardware will work all the time. If the ME is put into undesirable environments (too hot, too cold, dusty, ...) in a long period of time and/or without proper maintenance, the hardware will likely to encounter several issues and may affect the whole system.

Another assumption is the app will work on every Android phone. Users need to check carefully if their phones are compatible with API 28, and if the app has updates.

3. External Interface Requirements

3.1 User Interfaces

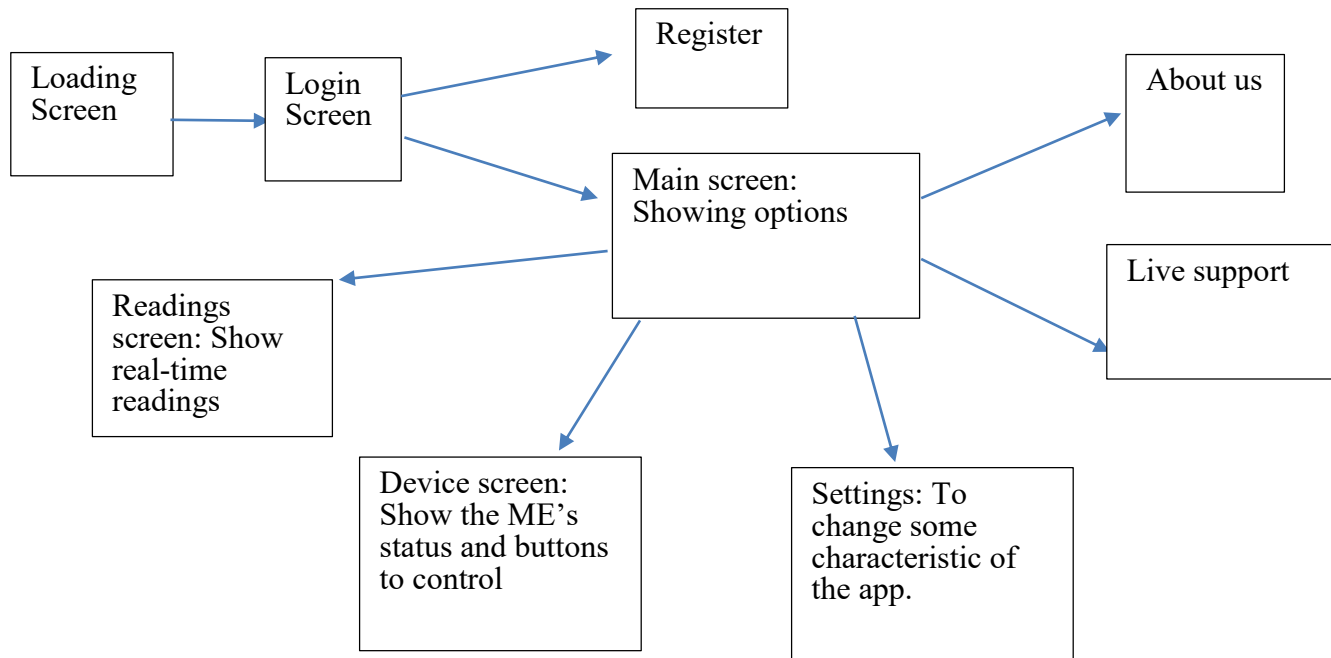


Figure 3 – Block diagram of the User Interface

After loading the app, the user will be introduced to the login screen. If not created, there will be an option to register an account in the application. Once logged in, there will be a main screen showing corresponding icons. When the icons are clicked, another screen will pop up. The drive screen is the remote control for the Mini Explorer with 4 buttons and shows battery status. The Real-time Readings screen will record the readings of the weather surrounding the ME. The options screen will change the settings of the application. The live support screen will provide online help for the users' needs. The last 2 icons are the logout button and the about us page. Here are some demonstrations of the UI, the rest are contained in Screen Mockups.



Figure 4 – Login Screen

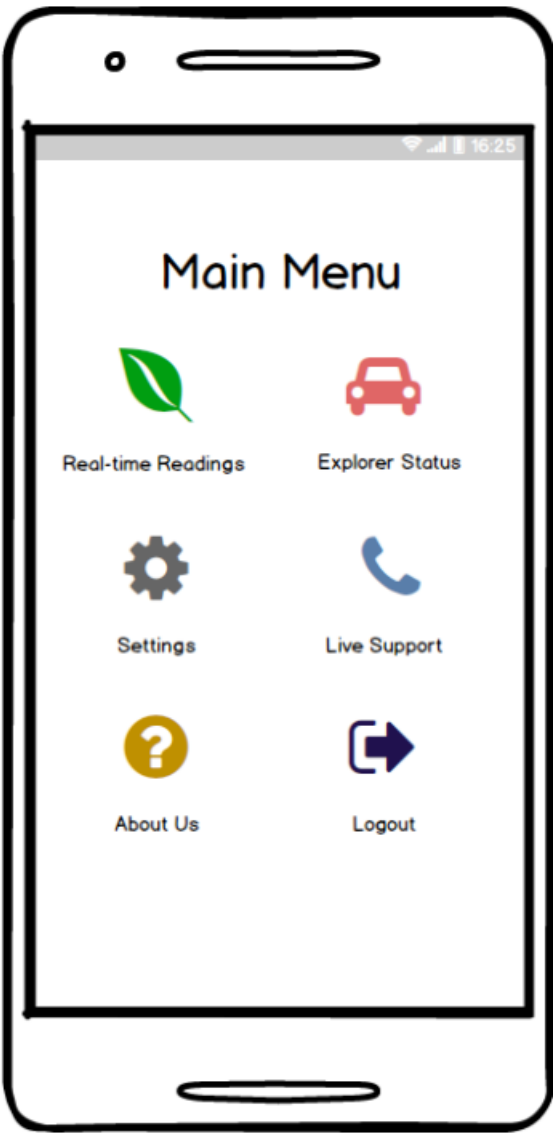


Figure 5 – Main Menu Screen

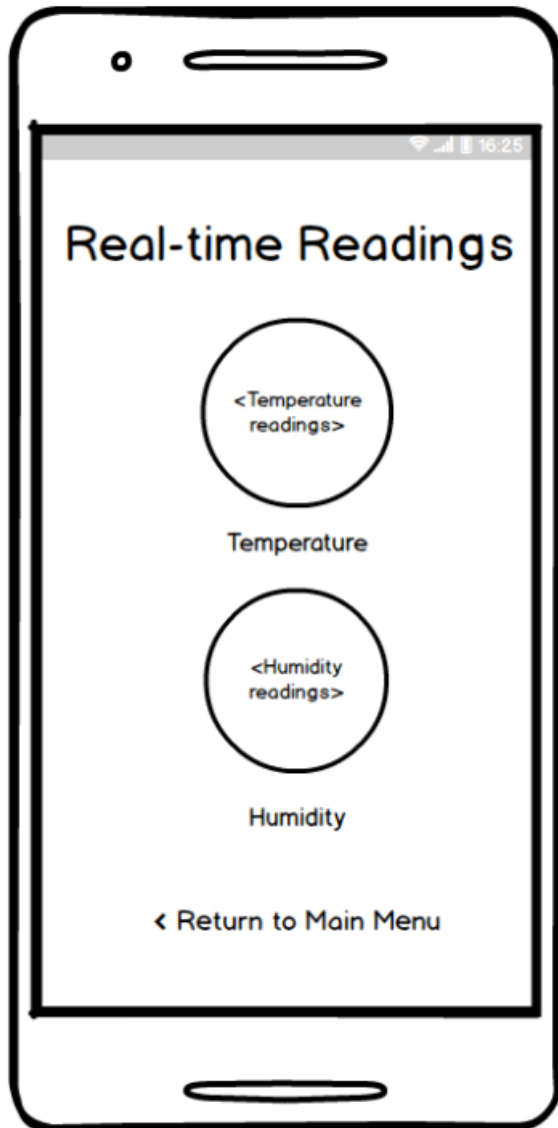


Figure 6 – Real-time Readings

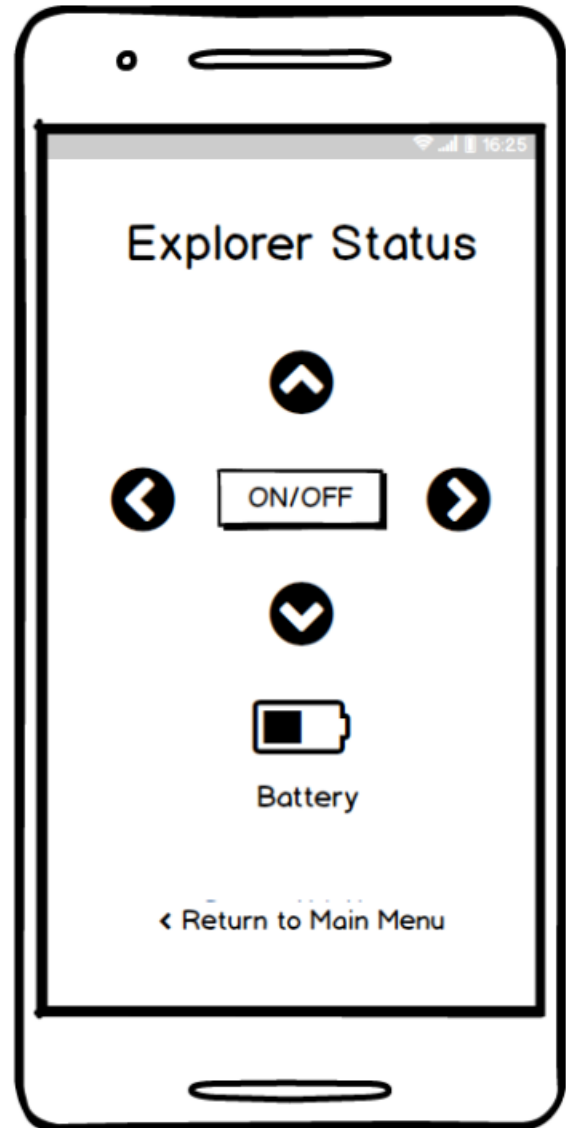


Figure 7 – Explorer Status

3.2 Hardware Interfaces

The RPi 3B+ is contained in a case that resembles the car and must have Bluetooth connection package installed. Attached to the RPi are the motors (PCA9685 16-channel PWM driver), a distance sensor (VL53L0x ToF distance) and a temperature/humidity sensor (HDC1008). Wheels are attached to the motors as well. A Bluetooth module is used for connection between the hardware and the software.

3.3 Software Interfaces

The mobile application is developed on Java and required to install necessary libraries. It is currently compatible with API 28 on Android only. The data between the hardware and software are the real-time readings and responses of each side of the control orders. Data between the software and database are the real-time readings and the status logs.

3.4 Communications Interfaces

The required communications interfaces are Bluetooth and Wireless internet (Through Wi-fi, 3G, 4G, ...) There are no requirements for the connection standards (such as IEEE P1901, ANSI C12.10, HomePlug, ...)

One particular communication characteristic needs to be mentioned, is the password of the user, which will be encrypted by using SHA256 Hash.

4. System Features

4.1 Ability to travel

4.1.1 Description and Priority

Users should be able to control the car by pressing buttons on the app. This is high priority.

4.1.2 Stimulus/Response Sequences

When users press on the “Up” / “Down” button, the ME should receive the signal immediately and move forward/backward. Combined with “Left”/ “Right” buttons, the car should be able to steer left/right while driving.

4.1.3 Functional Requirements

REQ-1: The Bluetooth connection between the ME and the phone is established.

REQ-2: The app is working properly.

REQ-3: The motors are working properly and are connected to the RPi 3B+.

4.2 Receive real-time ER

4.2.1 Description and Priority

Users should receive the surrounding environmental readings that will be shown on the app. This is high priority.

4.2.2 Stimulus/Response Sequences

When the device is up, it will activate the HDC1008 to automatically and simultaneously read the temperature and humidity in desired intervals.

4.2.3 Functional Requirements

REQ-1: The Bluetooth connection between the ME and the phone is established.

REQ-2: The app is working properly.

REQ-3: The HDC1008 is working properly and are connected to the RPi 3B+.

4.3 Send out data to database

4.3.1 Description and Priority

The database should be able to hold users' credentials and previous readings and/or status logs.

4.3.2 Stimulus/Response Sequences

After every registration or a status log is generated, the ME will automatically upload that to Firebase. This has moderate priority.

4.2.3 Functional Requirements

REQ-1: The Raspberry Pi has access to Internet.

REQ-2: Firebase is working properly and has capacity.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The app should be able to load and transfer between screens spontaneously. Whenever a command is sent from the app, the ME should be able to receive and process immediately.

5.2 Safety Requirements

The ME may malfunction due to extreme conditions of heat or cold temperature. The car's wheels might become flat due to overuse of the product, or running over sharp objects that are on the ground. With improper controls or use, the car may collide with the user, or any other objects surrounding the device.

5.3 Security Requirements

Firebase data should only be accessed by users with administration privileges. If there's a security breach, actions must be taken immediately in order to protect users' privacy. Users should only be able to control their own ME(s), not others'.

5.4 Software Quality Attributes

The software itself has a very straightforward and easy to use UI. All options and features are clearly listed on the main menu screens. In the future, the software should be able to be used in older versions of Android, and should be compatible to other versions of Raspberry Pi as well. While haven't been tested, the software should reach an acceptable level of portability, reliability and reusability, as it is developed on one single platform (Android Studio), by using one language (Java), and can be easily debugged or maintained.

5.5 Business Rules

Business can purchase the product for their own purposes, but by no means third parties can have access to see others' data.

Appendix A: Glossary

SRS: Software Requirement Specifications

CENG: Computer Engineering Technology

RPi 3B+: Raspberry Pi 3B+

ME: Mini Explorer

ER: Environmental Readings

Appendix B: Analysis Models

Block Diagram and Data Flow Diagram have been used to complete this SRS.

Appendix C: To Be Determined List

Currently, there are no To Be Determined List.