

SRS

Software Requirements Specification

Andre Chia Wei Song
32179197

Diego Miguel Lozano
32179194

Jae Wook Lee
32153492

Jeong Young Hwan
32154231

Yong Jin Kim
32150881

8th October 2019

Mobile Open Source SW Utilization
Mobile Systems Engineering

TABLE OF CONTENTS

1. Introduction	1
1.1 Purpose	
1.2 Project Scope	
1.3 Intended Use	
1.4 Definition & Acronyms and Abbreviations	
2. Overall Description	3
2.1 Product Functions (Functions in our Application)	
2.2 Operating Environment	
2.3 Design and Implementation Constraints	
2.4 Assumptions and Dependencies	
2.5 User Characteristics	
3. System Features	6
3.1 User Interface Diagram	
3.2 Hardware Interfaces	
3.3 Software Interfaces	
3.4 Use Case Diagram	
4. Functional Requirements	8
4.1 Normal User	
4.2 Common Functions	
5. Nonfunctional Requirements	8
5.1 Performance Requirements	
5.2 Security Requirements	
6. Conclusions	9

1. Introduction

The introduction will be implying information about the purpose, project scope, intended usage, definition, acronyms, and abbreviations.

1.1 Purpose

The purpose of this Software Requirements Specification is to describe the general outline this project, which is making “Miniature IoT Home Devices”. This document will be introducing the overall description, system features, nonfunctional requirements.

1.2 Project Scope

Our project is aimed at not only just making a miniature IoT house but also developing a mobile application to control the devices and interconnect 3 different platforms (server, application, Arduino). By using our application, the different IoT devices present in the house can be controlled. The application can be easily navigated through an intuitive User Interface, in order to control the common household appliances that are connected with Arduino modules.

1.3 Intended Use

The project that we have planned consists of a small scale version of a miniature home. In a larger scale, if we were to improve our application, it could be applied to TVs, refrigerators, doors and curtains, etc.

1.4 Definition & Acronyms and Abbreviations

Arduino

Arduino is a microcontroller designed to make our new ideas new easier. It can be connected to various electronic components such as sensors, motors, displays, and wireless communication modules to help us build our project.

Encryption

As we will encrypt the password that is stored in the database, this section will be writing about the definition of database encryption. Database encryption can generally be defined as a process that uses an algorithm to transform data stored in

a database into “ciphertext” that is incomprehensible without first being decrypted. Therefore, the purpose of database encryption is to protect the users’ data stored in the database from being read by unauthorized individuals with potentially “malicious” intentions. The technique we have chosen for our encryption is hashing the passwords in the database.

Hashing

Hashing is used in database systems and security systems worldwide as a method to protect sensitive data such as passwords. However, it is also used to improve the efficiency of database referencing. Inputted data is manipulated by a hashing algorithm. The algorithm converts the inputted data into a string of fixed length that can then be stored in a database. Hashing systems have two crucially important characteristics which are “unique and repeatable” and “hashing algorithms are not reversible”. This ensures that if an unauthorized attacker somehow gains access to the database, the attacker would not be able to read the data.

Internet of Things (IoT)

IoT is a technology that connects the internet by embedding sensors and communication functions into various devices. IoT is a technology that connects various things through wireless communication. Artificial Intelligence technology is used to receive and provide information to users.

Server

A server is a computer system that provides information or services to a client through a network and refers to a computer program or a device.

Database

Database is a storage of data that consist of one system. Server will link application and database, it is because every data that are used to operate application are stored in the database.

Firebase

It provides users with cloud storage services and can function as a back-end. Firebase offers remote cloud servers which makes it easy to store and retrieve data from the server.

2. Overall Description

This section provides information about the overall description of our project. It describes product perspectives, functions of our product, operating environment, design and implementation constraints, assumptions and dependencies.

2.1 Product Functions (Functions in our Application)

- House illumination (2 LEDs/light bulbs, one for a bedroom and the other for the living room).
- Fan for the air circulation.
- Security - Motion Sensor

2.2 Operating Environment

Arduino

- Arduino IDE.
- Language: C/C++.

Server & Application

- Firebase, Android Studio
- Language : JAVA/ JAVA script

2.3 Design and Implementation Constraints

Resource constraints

Limited budget. For example, we could build a custom PCB board to connect all our Arduino-related electronic devices. As this would be more expensive, the components will be soldered together with regular cables.

Size constraints

Our project will just include a simplified miniature model of a hypothetical house. In a real house, some implementation aspects might have to be adapted. For instance, we are using a Bluetooth model, so we would have to take into account the Bluetooth range inside the house.

Our team members' experience is limited in certain aspects

- Server deployment.

- Data encryption.
- Platform integration. Our project will integrate three different platforms (server, application and Arduino). This poses some technical difficulties that will need to be tackled.
- Building the miniature house.

2.4 Assumptions and Dependencies

Assumptions

We are assuming that the implementation of our system in miniature would be the same in a real house, but this might not be entirely true. As IoT device uses a sensor or collect data and give the user information about it, we do not have that function. We are just using Bluetooth in order to turn the light on and off, operate fans.

Dependencies

- **Server & Application:** when the user logs in the application, the username and password are compared with the data stored in the server. Since the password is hashed in the server, we do not ever know the plaintext of the user's password. We are only comparing hashes when the user logs in. Only if they match, the user will have access to the app features.
- **Application & Arduino:** the application and Arduino module are dependent on each other, being connected via Bluetooth. In that way, the user is able to control Arduino by using the app from their smartphone.

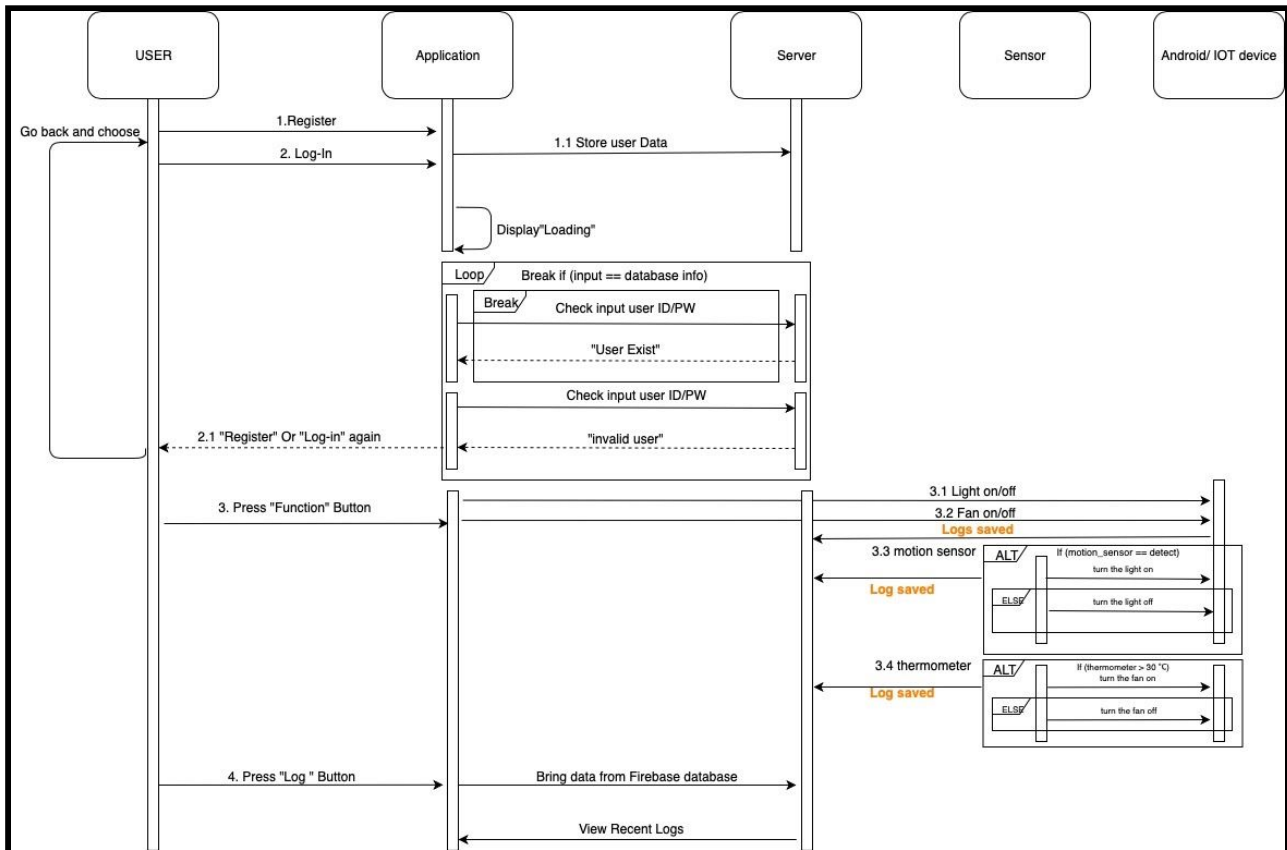
2.5 User Characteristics

The users (students or teacher) should use an android based mobile device in order to use the application. They will be able to control home appliances that are connected with Arduino modules.

(continued on the next page)

3. System Features

3.1 Sequence Diagram



3.2 Hardware Interfaces

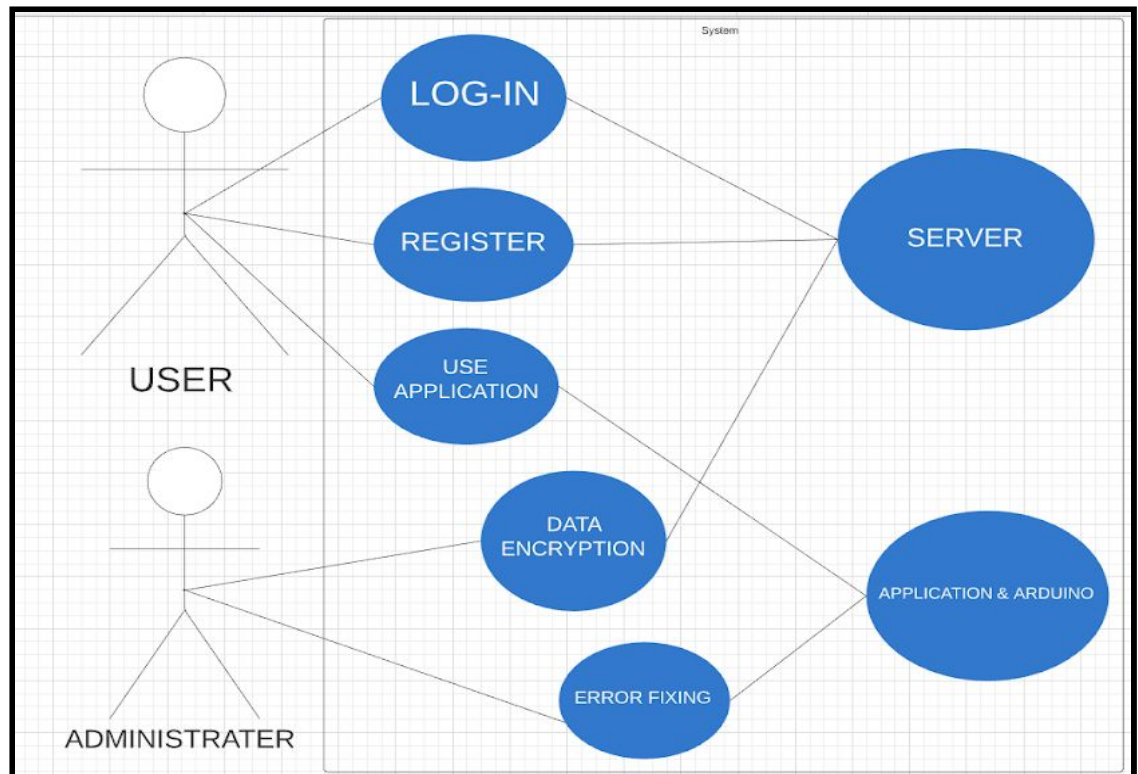
- 1 x Arduino UNO.
- 1 x Breadboard.
- 1 x JY-MCU Bluetooth Slave Module.
- 2 x LEDs.
- 2 x 200 Ohm Resistors (as many as LEDs).
- Breadboard cables.
- Miniature house.
- Battery Supplier
- 1 x Temperature/ humidity sensor DHT-11
- 1 x Infrared sensor HC-SR501
- 1 x Arduino LCD IIC2C 16 x 2 size

3.3 Software Interfaces

- Android Studio is our main software interface. By using android studio we will construct the server of our own, and design our application.

- Arduino is our sub-using software interface. Arduino will be used in controlling household appliances with our application.
- The main User Interface should be clean and motive. It should not be too cold.
- The required function should be added, Functions, Log-View.

3.4 Use Case Diagram



4. Functional Requirements

4.1 Normal User

When someone wants to use our application, the user would need to register with their email address. The personal information will be safely encrypted and stored in a Firebase Server.

4.2 Common Functions

Register

FR01: Before using the application the user have to make his/her ID and Password. The data will be stored at the firebase server database.

Log-in

FR02: The users have to register and then log-in in order to use application. The user will be using e-mail.

FR03: After log out, the application remembers the ID and auto-log in should be implemented

Database save

FR04: The user's important data will be stored at the firebase server.

Encrypt the Data

FR05: The stored data will be encrypted by using "hash" technique. To prevent hacking.

Application Page

FR06: After logging in the application, the user will be able to use "function" and "Log view" category.

Function

FR07: By pressing function button, the user will be able to use IOT service. It will be implemented by arduino.

Log-View

FR08: By Log-View page the server will send the logs that is stored at database.

5. Non-functional Requirements

5.1 Performance Requirements

NFR01: The application should have an interface with buttons in order to move on

NFR02: The application should perform all the functions that is required

NFR03: The application, server, arduino should be linked all together in order to use the application.

5.2 Security Requirements

NFR04: The server should get the user's information and store at database by hashing the passwords.

NFR05: By logging-in at the application, other user won't be able to control the application while other people is using.

5.3 Maintainability Requirements

NFR06: the server should be opened while using the application.

NFR06: the miniature house and arduino have to be operable while using the application.

5.4 Scalability Requirements

NFR07: the system can be expanded into real house

NFR08: The system can put more sensors.

5.5 Usability Requirements

NFR09: the software should be used at mobile.

6. Conclusions

As this SRS report is about writing the outline of the project we have written an introduction, overall description, system features, functional requirements, non-functional requirements. The goal of our project can be divided into two. One is making an IoT device for the miniature house and the other is linking 3 different platforms (server, application, arduino). As we have never attempted to link different platforms before, we are expecting difficulties while finishing this project. Nonetheless, we still plan to have a successful miniature home model that works well with Arduino.