Software Requirements Specification

for

roomi

Version 1.0 approved

Prepared by Jacob Ladan, Denald Demirxhiu, and Marko Javorac

CENG 319

September 27, 2018

1. Table of Contents

Revision History

Name	Date	Reason For Changes	Version

2. Introduction

2.1 Purpose

In this document we will describe the software requirements for the CENG 319 software project, also known as *roomi*. These requirements will provide an outline and starting point for the development process of this application's production.

2.2 Document Conventions

In this document, the general layouts of our application will be given as images mocked up using a program for Mac OS known as Sketch.

2.3 Intended Audience and Reading Suggestions

This document is to be used by developers, project managers, users, testers, and documentation writers. Readers will find that we have outlined the basic layout of our application's user interface, product functions, and software/hardware interfaces.

2.4 Product Scope

This application will work alongside our computer systems project to give users an aggregation of mobility features that will enable them to control various aspects of a residential dwelling, or enterprise workspace. These aspects include temperature control, lighting colour/brightness, and security through the use of an RFID chip/reader.

2.5 References

[1] Android 6 for Programmers: An App-Driven Approach, 3rd Edition Deitel et al, Prentice Hall, 2015, ISBN-13: 978-0-13-428936-6

[2] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

3. Overall Description

This section will give an overview of the products perspective, functions, types of users, operating environment, design, and user documentation.

3.1 Product Perspective

This application will be a new product as it will be an amalgamation of two separate subsystems; room security and automation. The application will provide a *platform* for which users can control room permissions and settings, such as temperature and lighting. In figure 2.1, you will find and outline of the the major components of the overall system, including interconnections and external interfaces.

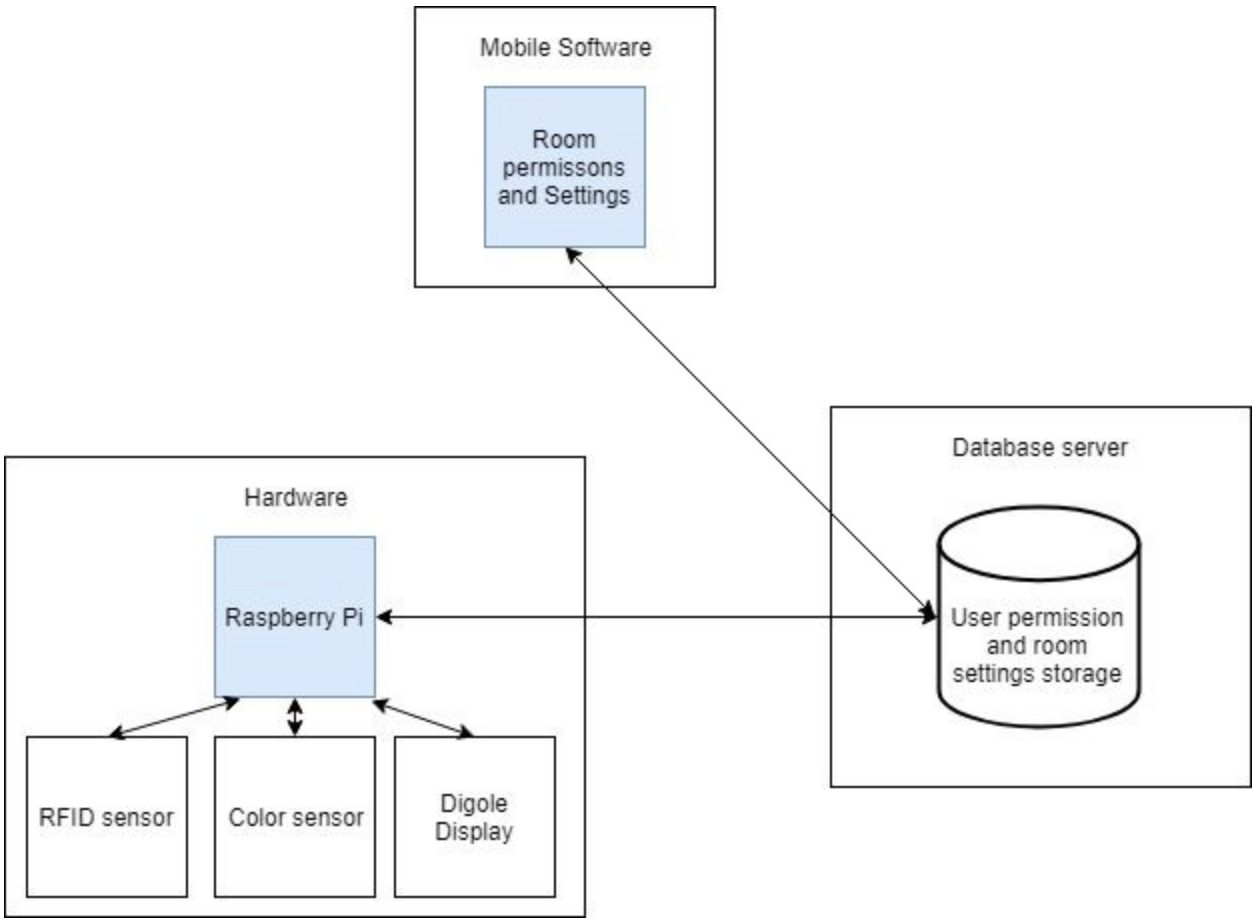

Figure 2.1: Software Interactivity Diagram

3.2 Product Functions

This application will allow users to permit and deny certain employees/residents to connected room. As well, this application will give users the function to adjust room temperature and lighting. Outline in figure 2.2 you will find a user flow diagram outlining the uppermost levels of the application.

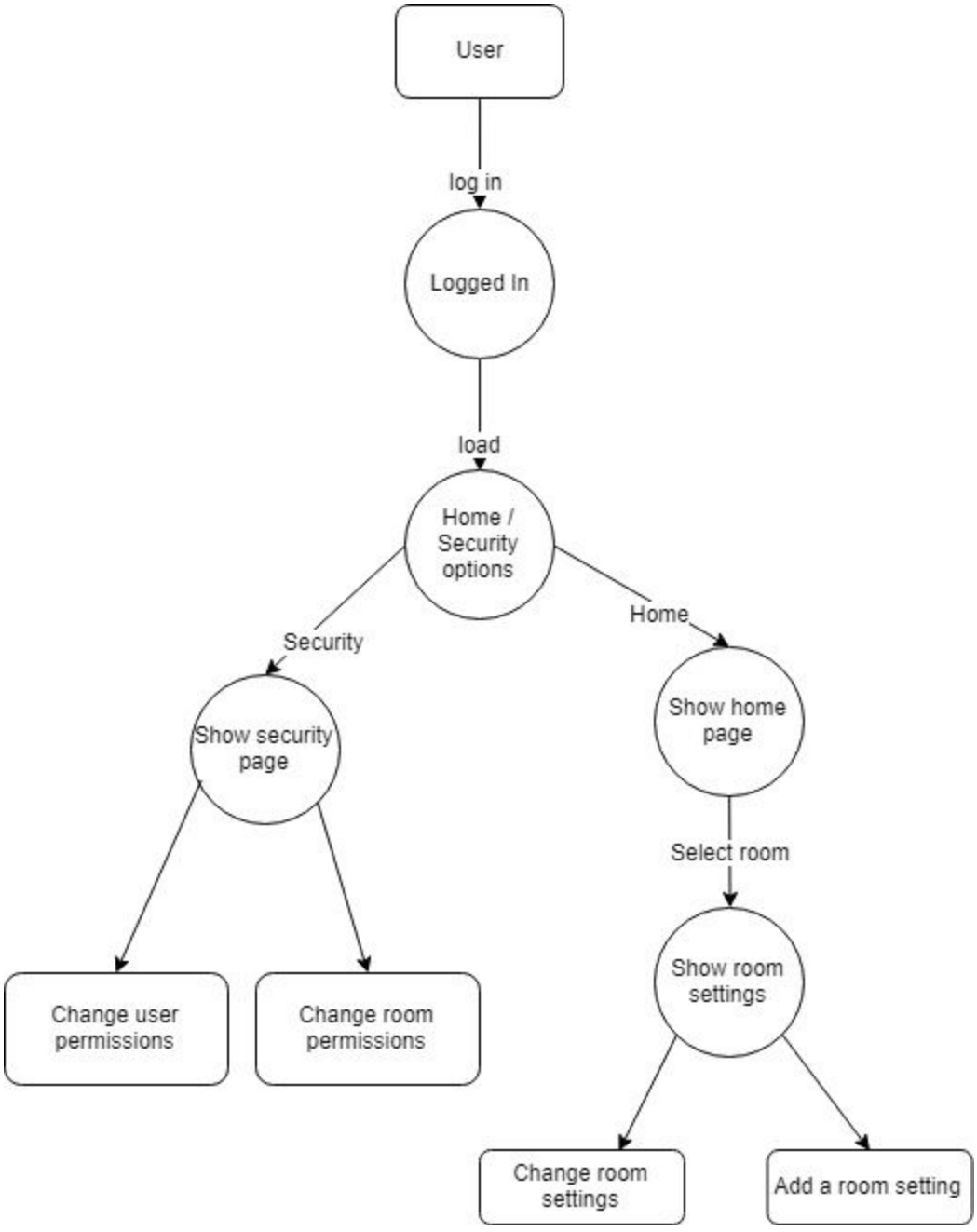

Figure 2.2: User Flow Diagram

3.3 User Classes and Characteristics

Two types of users will interact with this application: home residents and enterprise administration.

Residents will be able to establish connections with desired rooms in order to adjust the room's temperature and lighting brightness/colour. As well as adjust other residents' permissions to certain rooms through the use of the RFID chip/reader.

Administration users will be able to also adjust workspace traits such as temperature and lightning. As well as also permitting certain employees access these areas. Additionally, a subsection of the application will give these users the ability to set up schedules for labs/workspaces.

3.4 Operating Environment

This application will operate on a mobile device running Android OS updated to at least version 5.0 (Lollipop). User settings will be delivered to a remote database (Firebase) through which a Raspberry Pi 3B+ will be also communicating. The Raspberry Pi will perform the necessary actions to control room access/traits.

3.5 Design and Implementation Constraints

As we have three sensors to work with for this project, we will be only fully develop the functions of room access control, and light colour/brightness. However, a prototypal version will be implemented for temperature control. The application will give users the ability to update the database with different temperature values, which can then be used later on if the project continues to implement this functionality.

3.6 User Documentation

Alongside the final product, an about/help page within the application will be available to help users navigate the different functions of the software.

3.7 Assumptions and Dependencies

The most significant dependencies of this mobile application include the server-side database (where the data will be stored), the actual hardware it is going to communicate with, and time as the time constraint we have is restricted to 10 weeks. These constraints do affect the development time and the list of features that are going to be implemented, therefore in the final version some of the features might not be fully integrated and compatible, but the functionality will be implemented. There might be feature drop as a result of the time constraint, which can be implemented at a later version of the application.

4. External Interface Requirements

This section contains all the requirements and specifications of the system. It gives a detailed description of its features and implementations, as well as provides basic prototypes for the user interface.

4.1 User Interfaces

When the user first launches the app it will be presented with a splash screen that includes the name of the application. Immediately after it will be presented with the login screen where the user might log into their own account or create a new account (which might need to be approved by the administrator).

After the user has logged in it will be presented with a greeting and two features that the user might need to control. There's the home feature where the user might add new rooms, control room settings and more. For the security feature the user can go and check room permissions he/she has been assigned, but also change room permissions if they have that privilege.

The users are also supplied with an about page that provides a simple description of the application, as well as a help page that helps them navigate through the application.

4.2 Hardware Interfaces

The mobile application will be built to support devices running Android 5.0 and above. This will ensure access to a majority of user devices. It will need to communicate through the help of a database with a Raspberry Pi 3B+, which contains hardware implementations of the sensors that we will be using in the CENG 317. The RFID sensor will provide a way to enter the room if the user has been assigned permission to that room. The user will use the application to change room settings such as lighting and temperature and the Raspberry Pi will carry that command and perform the actual changes in the specified room. The user might choose to change the color of the lighting by scanning another color using the color sensor, or even choose the color from the presets provided in the application.

4.3 Software Interfaces

The mobile application will need to communicate directly with a database in which it stores user information, privileges, permissions, and room settings. The communication between the database and the application will include both reading and modifying the data, while the communication between the database and the mobile application consists of only reading the data.

4.4 Communications Interfaces

The communication between the mobile application and the database will be handled by several calls that request the information securely. There will not be file transfer between the database and the application. The method of communication is not important because it is handled by the underlying operating system.

5. System Features

5.1 Residential User

ID:FR1

4.1.1.1 User Account Creation

This function allows a user create a new account with the application. The account is required to modify room settings and access. The functionality will be considered High in terms of priority.

4.1.1.2 Rational

In order for the user to login to the application

4.1.1.3 Functional Requirements

The user will navigate to the “create account” page where they will fill a form requiring a username and password. This information will be stored in the database and be used for login verification. Form will be validated at both the frontend and backend.

ID:FR2

4.1.2.1 User Login

Given that the user has created an account, the user will be able to login and access the application

4.1.2.2 Rational

In order for the user to access the core functionality

4.1.2.3 Functional Requirements

The user will enter the login credential on the login page. The application will then communicate with the database to verify. The user will then have access to functionality.

ID:FR3

4.1.3.1 Add Room to “Home”

User will be able to add and label a new room.

4.1.3.2 Rational

Users will need to have a room for IoT devices to be a assigned and connected.

4.1.3.3 Functional Requirements

The user will select the “+” button under the “Home” tab and name the room.

ID:FR4

4.1.4.1 Add IoT device

Users will be able to add and access an IoT Device

4.1.4.2 Rational

Users must add the device before it can be controlled

4.1.4.3 Functional Requirements

The device will have to be compatible with the google home and connected through their framework.

ID:FR5

4.1.5.1 Control IoT device

Users will be able to control and access an IoT Device

4.1.5.2 Rational

Having mobile control of IoT devices is key to a great user experience

4.1.5.3 Functional Requirements

The application will send control commands to the devices using slider bars

5.2 Administration User

ID:FR6

4.2.1.1 Adding people to Access Control List

This function adds a user to the database. The administrator will have to fill out a form containing name, avatar, initial permissions and notes.

4.2.1.2 Rational

Users will need to be added to the database before they can use security functions.

4.2.1.3 Functional Requirements

The administrator will select the ”+” icon under the “people“ tab which will lead them to a form page that must be filled out. Form will be validated at both the frontend and backend. This information will be sent and stored in the database.

ID:FR7

4.2.2.1

This function

4.2.2.2 Rational

Users will need to be added to the database before they can use security functions.

4.2.2.3 Functional Requirements

The administrator will select the ”+” icon under the “people“ tab which will lead them to a form page that must be filled out. Form will be validated at both the frontend and backend. This information will be sent and stored in the database.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

Android Version 5.0 Lollipop: To reach a certain amount of devices, our application must be able to run on at least 5.0 Lollipop and higher builds of Android OS.

6.2 Safety Requirements

At this time, no safety concerns have been brought up with respect to our application.

6.3 Security Requirements

To ensure user's account data is secured properly, our database will be properly secured itself. User's passwords will be properly hashed and salted to prevent account compromisation. Necessary protocols will be in place in order to prevent third party software/users access to the applications main functions of home control and security.

6.4 Software Quality Attributes

Roomi will value ease of use over all other attributes. Our application's main goal is to allow the easier control of various traits of rooms and security throughout homes, offices, and or labs. Among other traits, the application will value testability and reliability. During the development process Roomi will be easily testable as the majority of its testing will involve posting and getting data from our database, which can easily be done with dummy data. Roomi will strive to be as reliable as possible as always having access to these room controls and functionality will be of the utmost importance.

One trait we will not focus on during the process of development will be portability. Roomi's requirement is to be build with Android studio for mobile devices running Android version 5.0 or higher. Meaning, iOS, Windows, and Linux devices will not be taken in to consideration for portability.

6.5 Business Rules

Roomi will be free to download along with the purchase of the corresponding hardware. As such, there will not be any associated subscription costs, or additional payments required. In addition, Google advertisements will not be in place to be used by our application.

7. Other Requirements

Roomie must utilize Google's Firebase in order to save and retrieve data about the user and its room settings. Firebase is a NoSQL database that utilizes a JSON data format. As such our application must adhere to these restrictions.

Appendix A: Glossary

- IoT: Internet of things - A physical device connected to the internet
- JSON: Javascript Object Notation - A lightweight data-interchange format
- RFID: Radio Frequency Identification - Transmitter/Receiver pair used as a secure form of identification
- Raspberry Pi: Micro computer

Appendix B: Analysis Models

To be revised.

Appendix C: To Be Determined List

To be revised.