Software Requirements Specification

for

Greenhouse Scale Model

Version 1.0 approved

Harshit Walia, Dhyey Farasram, Jainam Shah

Humber College

Oct 3rd,2019

Table of Contents

[1. Introduction 1](#_Toc20205211)

[1.1 Purpose 1](#_Toc20205212)

[1.2 Product Scope 1](#_Toc20205213)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc20205214)

[1.4 References 1](#_Toc20205215)

[2. Overall Description 1](#_Toc20205216)

[2.1 Product Perspective 1](#_Toc20205217)

[2.2 Product Functions 2](#_Toc20205218)

[2.3 User Classes and Characteristics 2](#_Toc20205219)

[2.4 Operating Environment 2](#_Toc20205220)

[2.5 Design and Implementation Constraints 2](#_Toc20205221)

[2.6 User Documentation 2](#_Toc20205222)

[2.7 Assumptions and Dependencies 2](#_Toc20205223)

[3. External Interface Requirements 3](#_Toc20205224)

[3.1 User Interfaces 3](#_Toc20205225)

[3.2 Hardware Interfaces 3](#_Toc20205226)

[3.3 Communications Interfaces 3](#_Toc20205227)

[4. System Features 3](#_Toc20205228)

[4.1 System Feature 1 3](#_Toc20205229)

[4.2 System Feature 2 (and so on) 4](#_Toc20205230)

[5. Other Nonfunctional Requirements 4](#_Toc20205231)

[5.1 Performance Requirements 4](#_Toc20205232)

[5.2 Safety Requirements 4](#_Toc20205233)

[5.3 Security Requirements 4](#_Toc20205234)

[5.4 Software Quality Attributes 4](#_Toc20205235)

[5.5 Business Rules 5](#_Toc20205236)

[6. Other Requirements 5](#_Toc20205237)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

Greenhouse Scale App is made keeping the needs of green house employees in the Humber Arboretum in order keep a track of the Temperature, and Moisture of their respective greenery but is not limited to them and can even be used by general public in their home gardens as well. It will also allow the user to supply water to their habitat based on the Moisture Level. This document will cover detailed description about the application which will be useful for both the developers and collaborators in this project.

## Product Scope

The Model is built to keep a data of the Temperature readings as well as the Moisture level readings of the plants so the user can retrieve it from the cloud server. It will also allow the user to provide water to their nursery if the Moisture level is below ideal percentage. The main advantage of this model is that it is cloud based and user will be able to monitor and nourish the plants at their convenience. So, the user will be able to access the data from any location. It makes the product different from others because of its automatic irrigation feature which waters the plants whenever the moisture level drops below a certain limit.

## Intended Audience and Reading Suggestions

The document is intended for developers, project collaborators, users and testers. The rest of the document contains overall description of the project, its requirements and features. The users can go over system features if they don’t want to go into the technical details, the testers can review the interface requirements and other nonfunctional requirements in order to test the app based on the requirements listed and can also test the system features by referring to section 3. The suggested reading sequence for tester would be to read through section 1 to 3 then 5, 6 and at last the section 4. Users can go over sections 1 and 4 to avoid going into technical details.

## References

Main Project idea was listed on riipen :

<https://app.riipen.com/projects/7324/details>

# Overall Description

## Product Perspective

The app along with hardware will be a self-contained product which will be using firebase database. The Firebase Realtime Database is a cloud-hosted NoSQL database that lets us to store and sync data between users in Realtime.

The app will receive the following information via cloud-based service:

* Reading from the moisture and temperature sensors and the state of water pump if turned on/off.
* Customer information.

The app will send the following information to the hardware via the cloud-based service:

* Commands to manually override the automatic irrigation system.

## Product Functions

* Loading screen.
* Sign in page:
  + Username and Password.
* Various readings coupled with interactive visuals.
  + Temperature Reading of 1 week (a graph)
  + Moisture Level Reading of 1 week (a graph)
* Notification/Alert System
  + When Moisture Level is below certain limit.
* Water pump turn on/off indication
  + Water Supply until ideal moisture level is reached
* Allows user to share data on various Social Media platforms
* Log- Out Screen.
* Help/Legal Information / Setting Pages

## User Classes and Characteristics

The app is designed keeping a simple yet effective interface in mind and will feature equal privileges to all users enabling them to keep a record of their plants whether it be in a green house or a house garden.

## Operating Environment

Our app will be able to operate on smartphones running on android OS.

Operating Environment for the app is as listed below.

* Distributed database.
* Client/Server system.
* Operating System (OS): Android KitKat and above supported
* Database: Firebase (NoSQL).
* Coding Language: Java

## Design and Implementation Constraints

Design Implementation:

The app has made considered the languages spoken primarily in Canada and will offer support and usage in both English and French along with the capability to add support in other languages in future as well.

Software Constraints:

The app database will only hold previous reading upto a maximum of last 60 days.

Hardware Constraints:

The sensors used in the model will be able to measure the Temperature of range 0 ̇C- 50 ̇C with an error percentage of +- 2C.

The moisture sensor will only be able to measure up to 1 foot deep.

## User Documentation

User manual and documentation will be provided in the Help section of the app.

## Assumptions and Dependencies

Hardware Dependencies:

The user has properly installed the hardware component and connected it with internet and always ensure that power is provided to the hardware for real time updates.

Water reservoir is provided in order to implement the automatic irrigation features.

# External Interface Requirements

## User Interfaces

The app will consist of the following pages: main loading page, login page/ sign up page for new users, temperature, moisture and water supply data pages along with interactive graphs. The app interface will have a side navigation bar to switch between pages which will be accessible from all pages and have settings to change language and units of measurements.

## Hardware Interfaces

The temperature sensors and soil moisture sensors will be connected to the Raspberry Pi and the water supply pump will be connected to the pipe through the reservoir which will be controlled by a relay module. It will be a small setup connected with the plant which will give send data to the cloud.

The app will receive all its input readings through the cloud service. The water supply pump can be manually overridden by user through the app.

## Communications Interfaces

* The user will be asked to sign-up using their email address.
* Communication standards that will be used will be HTTP.
* The user’s passwords will be encrypted using salting and hashing techniques.

# System Features

## Interactive Graphs

4.1.1 Description and Priority

Graphs for temperature and soil moisture readings received from server. This is one of the most important features which allows used to monitor the plants health.

4.1.2 Stimulus/Response Sequences

Based on the readings if the soil moisture level is low the user can manually over-ride the irrigation system, and water the plants.

4.1.3 Functional Requirements

In order to get the graphs and readings from the server, both the app the hardware must be connected to internet

## Self-Irrigation System

4.2.1 Description and Priority

The water pump automatically turns on and waters the plants when the soil moisture falls below a certain limit. It is the second most important feature.

4.2.2 Stimulus/Response Sequences

When the moisture goes under the pre set value by the user, the water pump turns on.

4.2.3 Functional Requirements

In order for the automatic irrigation system to work the water reservoir should be filled with water and a reliable power supply should be always connected with the motor to power it up.

# Other Nonfunctional Requirements

## Performance Requirements

The app should fire up after the client taps on app symbol the primary screen ought to be appeared in several seconds, best case scenario subsequently no defer clock circles will be in the dispatch screen code. The app won't require heaps of information and graphical substance. Energized pictures will be kept at a minim or not in the least. Reaction time regarding time taken to advance to another page or to get information ought not be in excess of a two second pause. This will dispense with coding to supply clients with criticism messages of the holding up procedures and information recovery occasions. At the point when the app is simultaneously with different apps, it ought not influence the exhibition in any huge manner from different apps. At the point when the app keeps running out of sight and is recovered, it ought to stay in a similar state as it was before to counteract loss of basic information. The quantity of calls to the server will be kept at least to consider less memory utilization and power utilization number of calls. At the point when a parcel is lost the app the solicitation for the data ought to be hate and discretionary alarms utilized in like manner. The app ought to be tried primarily on Wi-Fi systems for the extent of this undertaking. It is the made system that the sensors will use to get data to the firebase cloud. To constrain power and time utilization confinement on reserved pictures might be considered.

## Safety Requirements

On the off chance that and at whatever point the server is down, or is in upkeep a reinforcement server ought to be accessible to for switch over. Through constant replication and synchronization with the primary server. Since the database configuration is particular with various information bases overseeing various frameworks the information will be duplicated and put away on different servers in other interfacing framework. This makes a safeguard for potential down time of the server effectively getting information from the sensors through the app (firebase cloud database).

## Security Requirements

The security of the information sent from the app to the firebase cloud database is significant. The availability might be depended on the work of private tending to, advancing to NAT to get to the cloud. The upside of utilizing object-arranged plan in android will be augmented by separating basic usefulness and coding into private (covered up) and ensured strategies. The visitor and the head supervisors will have separate login get to entryways to take into consideration some usefulness not to be available to a client relying upon how they were validated into the framework. The WIFI systems visitors interface with will be vigorously reliant on the particular cafés' WIFI accounts arrangement strategies. It is normal that visitors would interface with an alternate WIFI profile than what is utilized by the sensors and other framework gadgets. HTTPS administrations will be utilized once accessible on server since portable app will more than likely utilize open system profiles.

## Software Quality Attributes

Modifiable – the product will have the option to manage changes in information powerfully since the firebase cloud database framework doesn't require as a lot of upkeep of tables and customary database structures and cost will be less since a physical server isn't there for support.

Ease of use – the clients of the app won't encounter a lot of issues being used as stacking time between screens will be kept at the very least and reaction times to get data limited in the code. Changing between the various states will be effective since moving huge measure of information will be limited.

Unwavering quality – the product will have the option to create low mistake rates over a predetermined time interim as unique data in regards to the readings got from server.

## Business Rules

This will be a free to use app but any copyright infringement will lead to serious consequences.

All users of this app are to use it for plant monitoring purposes only.

# Other Requirements

Third Party Services:

The firebase database will a major part of this project as it will be responsible to keep the record of all readings from the sensors and will provide them to app users.

Legal Information:

Reuse or implementation of any ideologies used in this project would require permissions of the developers otherwise serious consequences would be taken to the extreme of suing under copy right infringement law. This app might be free to use but is not an open- source project.