Software and Hardware Requirements Specification

for

Greenhouse Monitor

Version 1.0 approved

Prepared by Kenneth Chen and Princess Hernandez

Humber College – School of Applied Technology

January 24, 2019

Table of Contents

[1. Introduction 1](#_Toc536019284)

[1.1 Purpose 1](#_Toc536019285)

[1.2 Document Conventions 1](#_Toc536019286)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc536019287)

[1.4 Product Scope 1](#_Toc536019288)

[1.5 References 1](#_Toc536019289)

[2. Overall Description 2](#_Toc536019290)

[2.1 Product Perspective 2](#_Toc536019291)

[2.2 Product Functions 2](#_Toc536019292)

[2.3 User Classes and Characteristics 2](#_Toc536019293)

[2.4 Operating Environment 3](#_Toc536019294)

[2.5 Design and Implementation Constraints 3](#_Toc536019295)

[2.6 User Documentation 3](#_Toc536019296)

[2.7 Assumptions and Dependencies 3](#_Toc536019297)

[3. External Interface Requirements 3](#_Toc536019298)

[3.1 User Interfaces 3](#_Toc536019299)

[3.2 Hardware Interfaces 3](#_Toc536019300)

[3.3 Software Interfaces 3](#_Toc536019301)

[3.4 Communications Interfaces 4](#_Toc536019302)

[4. System Features 4](#_Toc536019303)

[4.1 Sign in/Logout 4](#_Toc536019304)

[4.2 Data readings and location 4](#_Toc536019305)

[4.3 Hourly line graph 5](#_Toc536019306)

[4.4 Calling 5](#_Toc536019307)

[4.5 Notification 5](#_Toc536019308)

[5. Other Nonfunctional Requirements 6](#_Toc536019309)

[5.1 Performance Requirements 6](#_Toc536019310)

[5.2 Safety Requirements 6](#_Toc536019311)

[5.3 Security Requirements 6](#_Toc536019312)

[5.4 Software Quality Attributes 6](#_Toc536019313)

[5.5 Business Rules 6](#_Toc536019314)

[6. Other Requirements 6](#_Toc536019315)

Revision History

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

# Introduction

## Purpose

The purpose of this document is to present a detailed description of the software requirements for a mobile application called Air Safety Monitor. It will include system features, interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external hardware.

## Document Conventions

In this document, a mockup diagram is used to illustrate an ideal design of the mobile application.

## Intended Audience and Reading Suggestions

This document is indented for mobile users, workplace management and the developers of the system.

## Product Scope

This software system will be an air monitoring system for all individuals. This system will be designed to raise awareness of the air around us by providing real-time record of air quality like CO and CO2. It provides a feature that allows for manual emergency call when high percentage of dangerous air is detected, otherwise it can be performed manually. For anyone at their workplace and even outside, this system is very efficient to use.

## References

*IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.*

*Davis M A, “Just Enough Requirements Management: Where Software Development Meets Marketing”, New York, Dorset House Publishing, 2005.*

# Overall Description

## *C:\Users\kench\Pictures\chart.png*Product Perspective

The product is have 2 main parts, the mobile app and the raspberry pi. The app show is used to view data on current gas level of a location in real time. The Pi main use is the gather current gas level of it location.

The mobile application will need to communicate with the database to get the data from the pi. The other function is to see the pi on the map and add it to the users list of pi that he/her want to see the data.

The pi is user to communicate with the database by sending data for it to be stored. All of the database communication will go over the Internet. The pi send data once every few mins.

## Product Functions

In the main menu, the user is able to access data recorded, their current location and has the ability to call emergency manually. The application shows whether or not the air is “GOOD” or “BAD” depending on the data recorded at the specific location. The application’s performance allows for calling feature if the user has yet to evacuate the area (if the application is warning the user that the air is “BAD”).

## User Classes and Characteristics

There are 2 types of user for this systems. Users with the mobile application and users that setup there air quality device. This system is for both type of user to be on the same system.

The mobile application let the user see the current data and hourly data on a selected system. The user can check a list of sensor data that is made public or check the data of their own sensor. The main user base is for any person that care about the air quality around them or have breathing problem. The product is for a daily user to track the air quality of the device’s area and report back the data hourly.

The air quality device user add their own device with an approximate locations, a name, is info if it’s indoor or outdoor. It will also ask if the user if they want to make the device viewable to the public. This is for user that add device in their own home and don’t want to make their data public. The user can add or delete their own device anytime.

## Operating Environment

The device operating environment is mostly ideal for indoor but it can also work outdoor. The mobile application can be used anywhere, software is android lollipop and up.

## Design and Implementation Constraints

The Internet connection is a constraint for the application. It isn’t a huge constraint but the application fetches data from the database once in a while or the static wouldn’t update, it is crucial that there is an Internet connection for the application to function. The raspberry pi itself have that constraint due to the need of it sending data to the database as well.

## User Documentation

The user is able to “Sign Up” with their name and email, which will be stored in a remote database – Firebase – to save the user’s personal information. There is a main menu that has the choice of calling for emergency, so that the user can manually call.

## Assumptions and Dependencies

One assumption about the product will be use on a mobile phone that meets the required specs. If the phone does not have the hardware resources required for the application, for example other application has already allocated space, the application may not work as intended.

# External Interface Requirements

## User Interfaces

Please refer to mockup page.

## Hardware Interfaces

Gyroscope detects rotational motion and tracks walking movement. VOC sensor measures the levels of volatile organic compounds that are in the air. Both sensors are integrated together with Raspberry Pi. The completed hardware will be able to be connected with the software in order to receive and store data in a remote database.

## Software Interfaces

The mobile application have alerts that can tell the user of poor air quality. Information is displayed in graphs for each air pollutant per hour. This is saved through the database. The communication between the database and the mobile application consists of operation concerning reading the data. The calling feature utilizes implicit Intent to make a phone call by calling a built-in Phone Call functionality in Android.

## Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. The communication is handled by the operating systems for both the mobile application and database.

# System Features

## Sign in/Logout

4.1.1 Description and Priority

The user should be able to sign up and sign in in order to fill their personal information.

4.1.2 Stimulus/Response Sequences

Given that a user has downloaded the mobile application, then the user should be able to register through the mobile application. The user must provide user-name, password and e-mail address.

4.1.3 Functional Requirements

In order for a user to register on the mobile application. The application will first send the user login name and pass with the database once if it the same, the data will send back a signal to let the user in. If the user get the inputs wrong, then the user wouldn’t be able to login.

## Data readings and location

4.2.1 Description and Priority

This page will show percentage (%) of different toxins in the air and also the location of the user.

4.2.2 Stimulus/Response Sequences

This is the main screen of the applications. This will allow the user to see the chart. The user click on the button that say ‘Current Data’ and it will display the information to the user.

4.2.3 Functional Requirements

In order for this page of the application to work the user must be connected to the internet. The next requirement is the Pi’s sensor must be work and be connected to the internet for the database to received information.

## Hourly line graph

4.3.1 Description and Priority

This page will show a graph of the readings for the day per hour.

4.3.2 Stimulus/Response Sequences

This screen of the applications will show a graph of the gas level by the hour show it change over it time. The user click on the button that say ‘Hourly Data’ and it will display the information to the user.

4.3.3 Functional Requirements

In order for this page of the application to work the user must be connected to the internet but not need for all time since the graph would not change every minute. The next requirement is the Pi’s sensor must be work, then it get the average gas level of the past few min then sends it to the database.

## Calling

4.4.1 Description and Priority

When the air is dangerous and the user is located at the same area the application will warn the user and ask if they want to call their emergency number.

4.4.2 Stimulus/Response Sequences

Inside the setting the user can turn this feature on or off. If on the user can add their own emergency number or let it be the default to 911. If the function is trigger it will ask if the user wants to call their emergency number.

4.4.3 Functional Requirements

The user must have their GPS and this calling feature turn on. The user must be in the same area as the Pi. In the user application there will be a function checking if the gas level still at a dangerous level for more than a few min it will trigger the call function.

## Notification

4.4.1 Description and Priority

This will notify the user if the air is dangerous and is required to get to another area. This messages will pop up in the notification bar when a certain air quality level is met.

4.4.2 Stimulus/Response Sequences

Inside the setting the user can customize their notification. If the function is trigger it will make a sound and make a sound.

4.4.3 Functional Requirements

In the user application if the currently air quality reach a dangerous level is will send a notification. The user must have their GPS user must be in the same area as the Pi.

# Other Nonfunctional Requirements

## Performance Requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

## Safety Requirements

If there are issues with the application itself, the developers are required to troubleshoot it.

## Security Requirements

The user’s information is saved from a remote database – Firebase. The data recorded is saved from a remote database – Firebase

## Software Quality Attributes

The user has the ability to login to their account and check the current air quality within the area. Otherwise, the application will not show anything but a login screen. Information from the databse is only accessible when the user is logged in.

## Business Rules

When the user is at a location, the application shows percentages of different kinds of airborne compounds. Also, when the user moves to another location, the application will show an updated percentage within the location.

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

TBD, no abbreviations or acronyms yet

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

TBD

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

Appendix list unfinished

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>