Environmental Monitoring System

Software Requirements Specification

For Database Design

Version 2.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/10/2015 | 1.0 | Basic document | BitLab Solutions |
| 16/10/2015 | 1.1 | refine document | BitLab Solutions |
| 17/10/2015 | 1.2 | Added figures and appendices | BitLab Solutions |
| 18/10/2015 | 2.0 | refined according to IEEE Citation Reference | BitLab Solutions |

Table of Contents

[1. Introduction 4](#_Toc432931691)

[1.1 Purpose 4](#_Toc432931692)

[1.2 Scope 4](#_Toc432931693)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc432931694)

[1.4 Overview 4](#_Toc432931695)

[2. Overall Description 5](#_Toc432931696)

[2.1 Product Perspective 5](#_Toc432931697)

[2.2 Product functions 5](#_Toc432931698)

[2.3 User characteristics 6](#_Toc432931699)

[2.4 Constraints 6](#_Toc432931700)

[2.5 Assumptions and dependencies 7](#_Toc432931701)

[3. Specific Requirements 7](#_Toc432931702)

[3.1 Functionality 7](#_Toc432931703)

[3.1.1 User login 7](#_Toc432931704)

[3.1.2 Receive different type of data from different sensors 7](#_Toc432931705)

[3.1.3 View the locations of all sensors on a map 7](#_Toc432931706)

[3.1.4 View the readings of each sensor 8](#_Toc432931707)

[3.1.5 Get detailed reports 8](#_Toc432931708)

[3.1.6 Analyse and predict near future environment condition 9](#_Toc432931709)

[3.1.7 Alert, notify the maintenance crew 9](#_Toc432931710)

[3.1.8 Add maintenance details 10](#_Toc432931711)

[3.1.9 Alert about hazardous environmental conditions 10](#_Toc432931712)

[3.1.10 Maintain a user activity log 10](#_Toc432931713)

[3.1.11 Manage users 10](#_Toc432931714)

[3.2 Non Functional Requirements 11](#_Toc432931715)

[3.2.1 Performance 11](#_Toc432931716)

[3.2.2 Usability 11](#_Toc432931717)

[3.2.3 Reliability 12](#_Toc432931718)

[3.2.4 Security 13](#_Toc432931719)

[3.2.5 Maintainability 13](#_Toc432931720)

[3.3 Design Constraints 13](#_Toc432931721)

[3.3.1 System must use SQL database 13](#_Toc432931722)

[3.3.2 System must use java language which Object Oriented supported. 13](#_Toc432931723)

[3.4 Assumptions 13](#_Toc432931724)

[3.4.1 Hard drive space 13](#_Toc432931725)

[3.4.2 Application memory usage 13](#_Toc432931726)

[3.5 Technologies 13](#_Toc432931727)

[4. Supporting Information 14](#_Toc432931728)

[4.1 References 14](#_Toc432931729)

[4.2 Appendices 14](#_Toc432931730)

[4.2.1 Appendix A 14](#_Toc432931731)

[Figure 1 - Block Diagram 6](#_Toc432931732)

[Figure 2 - Log in page 13](file:///C:\Users\Malith\Dropbox\Database\version%202\Software%20Requirements%20Specification%20G8.docx#_Toc432931733)

[Figure 3 - Sensor Readings 13](file:///C:\Users\Malith\Dropbox\Database\version%202\Software%20Requirements%20Specification%20G8.docx#_Toc432931734)

[Figure 4 - Sensor Locations 13](file:///C:\Users\Malith\Dropbox\Database\version%202\Software%20Requirements%20Specification%20G8.docx#_Toc432931735)

Software Requirements Specification

# Introduction

## Purpose

The purpose of this **Software Requirement Specification document** is to capture all the requirements for the Environmental monitoring system. It will explain the scope of the system in the beginning. Then it will describe the functional requirements, non-functional requirements, the constraints under which it must operate, the interfaces of the system and finally the security, reliability, quality factors and limitations of the system.

## Scope

This software system will be a database system for the Colombo city mayor. This system will be designed to maximize the efficiency of mayor’s office by ***assisting in* providing information for reporting and analysis of the environmental condition in the city**. By maximizing the mayor’s work efficiency, the system will meet the mayor’s needs.

More specifically, this system will be designed to collect data from a set of sensors situated around the city. Different sensors will have different attributes. Multiple types of sensors can be located at the same location. The software will facilitate communication between those sensors and the central post which will be situated at mayor’s office via GSM technology. The system will contain a relational database with a list of sensors, newly added sensors and a collection of every bit of data which is being sent from each sensor.

## Definitions, Acronyms, and Abbreviations

* + EMS – Environmental Monitoring System
  + SRS – Software Requirements Specification.
  + Mayor - The highest-ranking official in a municipal government such as that of a city or town.
  + Sensor – The equipment which are used to capture the data from the environment.
  + OODP- Object Oriented Design Pattern
  + Central post - Mayor’s office
  + GSM – Global System for Mobile communication
  + Backup – Save a copy of the data so it may be used to restore the original after a data loss event
  + Server - A computer program or a machine capable of accepting requests from clients and responding to them
  + Database – A collection of information that is organized so that it can easily be accessed, managed, and updated.
  + Cloud - a model of data storage where the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company

## Overview

Next, the Overall Description section, of this SRS document describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand.

The third chapter, Specific Requirements section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of users that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented

## Product Perspective

This system will consist of a desktop application for the mayor’s office and a set of sensors around the Colombo city.

This desktop application will be used to view the data collected from sensors over a certain time period as reports and to analyse those data to get a detailed idea about the environmental condition of the city. Sensors will communicate with the central post (mayor’s office) with GSM signals.

Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. This database server machine will be placed at the mayor’s office. To increase the reliability of the collected data, the database will be backup daily to a cloud system from the local server.

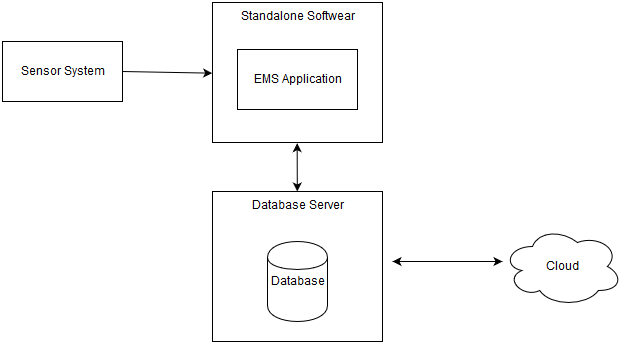


Figure 1 - Block Diagram

## Product functions

This system facilitates in getting readings of following environmental properties.

* Temperature
* Rainfall
* Wind speed & direction
* Quality of the air (oxygen, carbon dioxide percentages) [1]

From the desktop application a user will be able to view the environmental condition in different places.

First, user may select a location and select the type of environmental condition (temperature, rainfall, quality of air, etc…) that he/she wants to check and get a detailed report which is precisely described with graphs and figures.

When installing a new sensor, information about that sensor can be entered to the system. Different type of information about a sensor should be entered to the system (e.g. - date of installation, type of the sensor, etc…).

When a sensor just got repaired that information must be entered to the system (date of repair, nature of repair). If the sensor is completely broken and not functional at all it should be also indicated in the system.

Considering the installation day of a sensor system will provide a rough prediction about when that sensor should get repaired, so the maintenance crew can repair the sensor according to that prediction.

User can get the reports about the environmental condition of one place or an area. Most importantly, the system will produce a set of reports at the end of the month to show the variations of environmental conditions during the month. User can also use the data about environmental condition to get an analysis of the variation of environmental condition during some time period, by analysing those data.

Users with higher access privileges will be able to add and remove users to/from the system.

## User characteristics

There will be multiple user levels with different levels of access privileges. According to the level of access some users will not be able to use some of the functions. So different types of users will have different types of characters in the system. The users and their characters in the system are described as below.

* Mayer:

Mayer will have the highest access privilege. He will be capable of use all the functions in the system that means the mayor will act as the system administrator.

* Clerk:

Clerk will be able to get some reports and use the system to see the information about environmental conditions. So clerk has read only access, not adding, editing access.

* Maintenance manager:

Maintenance manager will be able to enter details about sensors (adding, removing and repairing) get the predictions about the sensors need to be repaired.

## Constraints

For the system to be functional without any breakdowns there are some constraints.

* Continuous functioning of sensors:  
   Sensors should be activated 24\*7, unless sensors will not be able to send data so the reports and analysis will be less accurate.
* Continuous and uninterrupted GSM connection:

Since sensors are sending data using GSM connection, connection should be continuous and uninterrupted, otherwise some data will be loss during the transmission.

* Database server should be up and run 24\*7:

Since sensors are sending data 24\*7 database server at central post should also be up and alive 24\*7 to collect data sent by sensors.

* Well-functioning of sensors:

Sensors should be functioning very well because otherwise data sent by sensors will be erroneous and reports and analysis obtained using those data will also be erroneous.

## Assumptions and dependencies

* Sensors are powered 24\*7:  
   No data loss
* GSM connection is continuous and un interrupted:

No data loss during transmission

* Database server is up and run 24\*7

All the data sent by sensors are stored in to the database

* Sensors are functioning fine

Sensors are sending accurate data

# Specific Requirements

## Functionality

This section includes the requirements that specify all the fundamental actions of the software system.

### User login

In order to use the system, a user should be logged in to the system

**Scenario: Successful log-in**

Given the user wants to log in

When the user logs in providing correct user name, password

Then the user should be logged in successfully

**Scenario: Retrieve password**

Given the user wants to log in And has lost the password   
When the user enters his/her email address in the “Retrieve password” form And submits the form   
Then the user should receive an email containing the password

### Receive different type of data from different sensors

Different type of data from different sensors will be transmitted periodically to the central post. So the system should be able to steer and store each type of data to relevant collections in the database.

* Temperature readings
* Rainfall readings
* Wind speed readings
* Wind direction reading
* Air quality percentage values

### View the locations of all sensors on a map

All the sensors positioned in the city should be illustrated on a map with their summarised details.

**Scenario: View by type of sensor**

Given the user wants to select the viewing type

When the user selects *view by type* as the viewing type

Then map will show sensors by its type

**Scenario: View newly added sensors**

Given the user wants to select the viewing type

When the user selects *view newly added sensors* as the viewing type

Then map will show newly added sensors only

**Scenario: View broken sensors**

Given the user wants to select the viewing type

When the user selects *view broken sensors* as the viewing type

Then map will show broken sensors

**Scenario: View sensors which has reached its routine repair time**

Given the user wants to select the viewing type

When the user selects *view sensors which has reached its routine repair time* as the viewing type

Then map will show sensors which has reached its routine repair time

### View the readings of each sensor

User should be able to query all readings from beginning day to current day for each sensor.

**Scenario: View data as a list**

Given the user wants to select the viewing type

When the user selects *list* as the viewing type

Then it will show data as a list

**Scenario: View data as a graph**

Given the user wants to select the viewing type

When the user selects *graph* as the viewing type

Then it will show data as a graph

**Scenario: View for a specific date**

Given the user wants to select the viewing type

When the user selects *view for a specific date* as the viewing type

Then the user should be able to select a date and view data for that day.

### Get detailed reports

User should be able to view data as detailed reports other than raw data.

**Scenario: get report by sensor**

Given the user wants to select the report generating type

When the user selects *by sensor* as the report type

Then the user will be able to choose a sensor and get a report generated for that sensor

**Scenario: get report by environmental property**

Given the user wants to select the report generating type

When the user selects *by environment property* as the report type

Then the user will be able to choose the type of property and get a report generated for that property

**Scenario: get report by location**

Given the user wants to select the report generating type

When the user selects *by location* as the report type

Then the user will be able to select a location and get a report generated for that location for available types of readings.

**Scenario: get report for entire city**

Given the user wants to select the report generating type

When the user selects *for entire city* as the report type

Then the user will get to input some parameters and get a full report generated for the whole city

### Analyse and predict near future environment condition

The system should be able to predict the near future environmental conditions in the city by analysing the previous data.

**Scenario: predict rainfall**

Given the user wants get the prediction for rainfall

When the user asks to predict

Then the user will get a forecast of each sensor locations in the city as a list or a map

**Scenario: predict wind speed and direction**

Given the user wants get the prediction for wind speed and direction

When the user asks to predict

Then the user will get a forecast of each sensor locations in the city as a list or a map

**Scenario: predict temperature**

Given the user wants get the prediction for temperature

When the user asks to predict

Then the user will get a forecast of each sensor locations in the city as a list or a map

**Scenario: predict quality of the air**

Given the user wants get the prediction for quality of air

When the user asks to predict

Then the user will get a forecast of each sensor locations in the city as a list or a map

### Alert, notify the maintenance crew

**Scenario: notify the routine maintenance of a sensor**

Given the routine maintenance day of a sensor has arrived

When the maintenance crew is available

Then the maintenance manager will get a notification about the sensor which has to be maintained

**Scenario: give an alert about a broken sensor**

Given a sensor has broken (if the central post does not receive data from a sensor for multiple periods, that sensor will be labelled as a broken one)

When the maintenance crew is available

Then the maintenance manager will get a notification about the sensor which has to be fixed

### Add maintenance details

The records of every maintenance must be added to the system by the maintenance manager.

**Scenario: a repair of a sensor (routine/ emergency)**

Given a sensor is repaired successfully by the maintenance crew

When the maintenance manager selects to records that repair

Then the maintenance manager get the wizard to add that record

**Scenario: a replacement of a sensor**

Given a sensor is replaced by the maintenance crew

When the maintenance manager selects to record that replacement

Then the maintenance manager get the wizard to add that record

### Alert about hazardous environmental conditions

System should be able to predict emergency flooding, air poisonings, storms, and high temperature situations. It would help the mayor to react immediately and do a better disaster management in the city.

**Scenario: predict a flooding situation**

Given a sensor records highly growing readings of the rainfall

When the system detects that the sequence will lead to a flooding situation

Then the mayor is alert about a flooding situation at that sensor’s location

**Scenario: predict an air poisoning**

Given a sensor records low oxygen or high carbon dioxide percentage for some certain amount of periods

When the system detects that the percentages are over the standard values for human health

Then the mayor is alert about a air poisoning situation at that sensor’s location

**Scenario: predict a high temperature situation**

Given a sensor records highly growing readings of the temperature

When the system detects that the readings are over the standard values

Then the mayor is alert about a high temperature situation at that sensor’s location

**Scenario: predict a storm**

Given a sensor records highly growing readings of the wind speed

When the system detects that the sequence will lead to a storm

Then the mayor is alert about a storm situation at that sensor’s location

### Maintain a user activity log

The system should record every activity of each user. The administrator can review these records. This is to increase the reliability of the system and for legal needs.

### Manage users

In order to keep track of the users, the administrator should be able to manage the users

**Scenario: register an new user**

Given the administrator is logged in

When the administrator create a new user

Then the new user should be created

**Scenario: Edit an existing user’s information**

Given the administrator is logged in

When the administrator edits an existing user

Then the user information should be updated

**Scenario: Delete/Inactivate an existing user**

Given the administrator is logged in

When the administrator deletes an existing user

Then the user should be deleted

## Non Functional Requirements

This section specifies criteria that can be used to judge the operation of the system, rather than specific behaviours. This should be contrasted with functional requirements that define specific behaviour or functions. [2]

Broadly, this section defines how the system is supposed to be. [2]

### Performance

Since this is going be a large, continuously updating system performance of the database server machine will be a vital factor for the well functionality of the system. Multiple sensors can be concurrently trying to transmit data to the central post so the server machine at the central post should be able to handle those data. Those data should be kept for years for analysing and predicting future environmental conditions.

So the server machine at the central station should have kind of large hard disk capacity, and other high level hardware components.

### Usability

This system is designed for multiple levels of users, so the each user in every user level should have some amount of computer literacy to work with this system. So for their convenient those facts are considered when designing the system.

Even if they don’t have the knowledge, a few hours of training will make them to be able to use the system. The system will be designed in such a manner. That training will be provided by our company when deploying the system.

#### User interface:

System should have a friendly user interface. As mentioned before even a person that does not have a great knowledge about computers should be able to work with the application. For that, the user interface should be simple and smooth.

Simple indications of user interfaces will be like this.

Figure 2 - Log in page

* Admin
* Manager
* Clerk

Username

Password

Forgot password?

Log in

Sensors

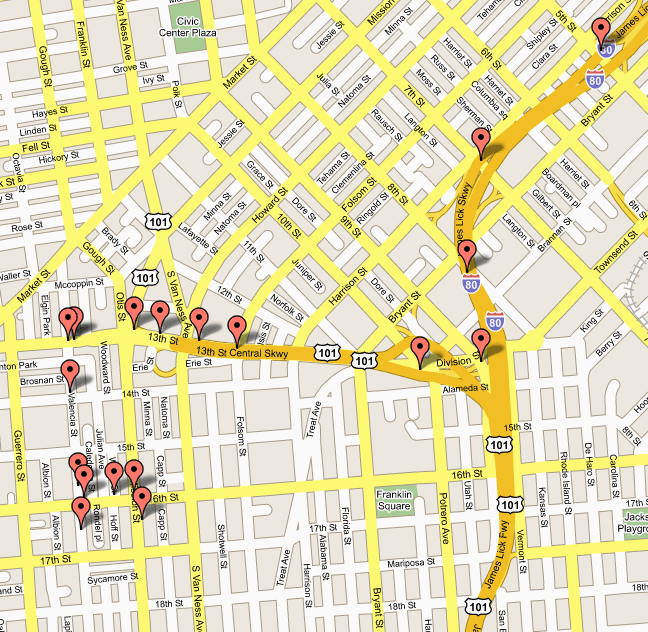


Figure 3 - Sensor Readings

Date & Time

Temp

Figure 4 - Sensor Locations

Figure 2 - Current Details

Figure 1 - Log in

A first-time user of the application should see the log-in page when he/she opens the application, see Figure 2. User can log-in as administrator (Mayor), manager or the clerk. If user wants to log in but has forgotten the password he/she will be able to change it using “Forgot password” option.

After the user logged in he/she should be able to see the current readings of the sensors on the page. Using the date and time option on the page the user can see the previous readings as well. It can be either graphs or on a map, see Figure 3.

Also the user must be able to see the sensor locations and their conditions on a city map, see Figure 4. These functions are operated according to each user’s access levels.

#### User manual:

User manual that is describing the functionality of the application should be provided to the users at the system deployment phase. Then the users will be able to solve the problems that occur to them using the user manual. In this user manual, functionality of the system should be described as simple as possible.

User manual is provide as hard and soft versions.

### Reliability

After the system is deployed data coming from sensor are recorded in the system. With any circumstances system must be well functioning and do its purposes. Therefore following reliability requirements should be fulfil for to make sure that no any data or record is lost.

#### Data consistency:

Since the system is going to handle large amount of data, the database going to use for this system should be able to handle that amount of data.

#### Availability

Since this system is continuously updating system, system should be available 24\*7 without failure except for unavoidable reasons like natural disasters. Even though system failed due to that kind of unavoidable reason, system can be put back into 100% functionality within 12 hours using the help of data that is backed up into the clouds.

#### High accuracy

Reports and analysis provided by the system should be considerably accurate. Therefore there cannot be any losses in the data base, and also data coming from the sensors should also need to be accurate.

### Security

Security of the data inside the database should be guaranteed, unauthorized parties should not be able to access to the system, and some functions of the system should be prohibited to some users depending on the level access privilege they have. To fulfil this requirement, access to the system is password protected and different types of users will be identified in the login.

### Maintainability

Basically our company will support to maintain and do the major improvements to the system in first 12 months. Even after that whenever user needs to have assistance for the maintenance of the system, they can contact our company.

## Design Constraints

This section indicates design constraints on the database system being built. As we are given the freedom to use development tools, we used most of open source development tools and libraries.

### Use of SQL database

To use SQL query language we use MySQL as the database development tool. Because it is a free product which come with SQL query language. Otherwise we have to use development products like SQL Server, Oracle Database, Sybase and Informix.

### Use of java language which supports Object Oriented development.

As we asked to comply with Object Oriented Design pattern for development, we use java language for development purposes. Because it is the language that support OODP with less complexity.

## Assumptions

These are the assumptions we came about the performance of the of the server machine and other components

### Hard drive space

The server machine should have at least 10GB space hard drive for store and execute the database.

### Application memory usage

The amount of Operation System memory occupied by the application can be 80MB. Therefore server machine should have at least 80MB free memory capacity (RAM) in all the time.

## Technologies

The system is implemented as a windows based standalone application. Therefore we should use Windows Operating System on server machine with supportive java interfaces.

### Development tools

Java 1.8 runtime interface use to develop user interface and back end application. As the back end database development tool we use MySQL query language.

### Communication tools

For the system to be accessed data from the multiple sensors which located in varies locations we need to use GSM technology. By using GSM modules we can transmit data from sensors to central application through SMS.

### Data Storing

We decide to use a cloud server to backup databases because store on own local server might not be reliable for the data.

### Hardware components

We should use varies digital sensors to read values of the environment.

Also we should suggest Mayor to buy a server to use as local server because the system need high performance in server machine.

# Supporting Information

## References

|  |  |
| --- | --- |
| [1] | “Carbon monitoring,” [Online]. Available: https://en.wikipedia.org/wiki/Carbon\_monitoring. |
| [2] | “Non-functional requirement,” [Online]. Available: https://en.wikipedia.org/wiki/Non-functional\_requirement. |
| [3] | “IEEE Citation ref,” [Online]. Available: http://www.ieee.org/documents/ieeecitationref.pdf. |
| [4] | “Environmental monitoring,” [Online]. Available: https://en.wikipedia.org/wiki/Environmental\_monitoring#cite\_note-1. |
| [5] | “Innovations In Environment Monitoring,” [Online]. Available: http://www.environment.ucla.edu/reportcard/article1506.html. |
| [6] | “Monitoring Certification Scheme,” [Online]. Available: https://en.wikipedia.org/wiki/Monitoring\_Certification\_Scheme. |

## Appendices

### Appendix A

Project Assignments and Descriptions - DatabaseProjects2013Batch.pdf   
(<http://online.mrt.ac.lk/mod/resource/view.php?id=20975> )

Topic 4 - Environmental Monitoring System

Design a database system for the Colombo city mayor. Colombo city mayor wants to setup an environmental monitoring system which collects data from a set of sensors situated around the city. Different sensors will have different attributes. Multiple types of sensors can be located at the same location. Database should assist in providing information for reporting and analysis (temporal base, type based etc…) of the environmental condition in the city, keep track of newly added sensors etc.