ESBHA001

Software Requirement Specification

Soft

Issue/Version: 0.2

**Document Control Data Sheet**

|  |  |
| --- | --- |
| **Document information** | |
| **Classification** | Confidential |
| **Document Name** | ESBHA001 Software Requirement Specification |
| **Document Issue No.** | 0 |
| **Document Revision No.** | 2 |
| **Document Id** | ATPL/RED/24/01/2020-2 |
| **Document Issue date** | 24-January-2020 |
| **Prepared by** | Veena G |
| **Reviewed By** | Arjun Varma |
| **Approved By** |  |
| **Distribution List** | Alcodex project team |

**Document Amendment Sheet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sl No** | **Issue /**  **Revision** | **Date** | **Change Description** | **Prepared**  **By** | **Reviewed**  **By** | **Approved**  **By** |
| 1 | 0.1 | 21-01-2020 | Initial draft | Veena G | Arjun Varma |  |
| 2 | 0.2 | 23-01-2020 | Update against review comments | Veena G | Sreevidhya |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Contents

[1.0 Introduction 6](#_Toc30756185)

[1.1 Purpose and Scope of Document 6](#_Toc30756186)

[1.2 Project Overview and Scope 6](#_Toc30756187)

[1.3 Assumptions and Constrains 6](#_Toc30756188)

[2.0 Functional Requirements 7](#_Toc30756189)

[2.1 Device Management 7](#_Toc30756190)

[2.2 User Management 7](#_Toc30756191)

[2.3 Data Management 8](#_Toc30756192)

[2.4 Alarm Management 8](#_Toc30756193)

[2.5 Error diagnostics 9](#_Toc30756194)

[2.6 API Module 9](#_Toc30756195)

[3.0 Non-Functional Requirements 10](#_Toc30756196)

[4.0 Interface Requirements 10](#_Toc30756197)

[4.1 User Interfaces 10](#_Toc30756198)

[4.2 Hardware Interfaces 10](#_Toc30756199)

[4.3 Software Interfaces 11](#_Toc30756200)

[5.0 Data Requirements 11](#_Toc30756201)

[6.0 Installation 11](#_Toc30756202)

[7.0 Project Execution Requirements 12](#_Toc30756203)

[7.1 Deliverables and Delivery Dates 12](#_Toc30756204)

[7.2 Operating Environment 12](#_Toc30756205)

[7.3 Testing 12](#_Toc30756206)

[7.4 Acceptance Criteria 12](#_Toc30756207)

[8.0 Appendix 13](#_Toc30756208)

[8.1 Appendix A: Acronyms 13](#_Toc30756209)

# Introduction

## Purpose and Scope of Document

This document provides the software requirements of environmental sensor server for Bhavnagar Municipal Corporation. The scope of this document includes the detailed software requirement specifications of server.

## Project Overview and Scope

ESBHA001 environmental sensor server provides a web application server and a database server for processing and storing data provided by environmental sensors of Envitus IoT Suite by Alcodex.

The server includes different features such as live data monitoring, statistical data calculation, alarm management, user management and error handling. It also has APIs to integrate with third parties, for getting data in digital signage.

**1.2.1 Scope**

In this project we assure the design,development,testing, supply, installation and deployment of Envitus sensor server and its dependencies such as nodeJS and mongoDB.

**1.2.2 Out of scope**

1. Alcodex doesn’t provide any IT Infrastructure for deploying the server application.

2. Supply of servers for application, database or recovery.

3. Operating Systems to install in servers.

## Assumptions and Constrains

#### Assumptions

1. Server shall be available 24\*7 online.
2. Internet shall be always available.
3. Client shall provide hosting space.
4. Data Center environment shall be secure.
5. Configuration of servers shall be based on server sizing document.
6. Self-signed certificate for HTTPS protocol shall be used in the server.

#### Constraints

1. Non availability of network.
2. Installation of server will not be easy if configuration of servers is not based on server sizing document.

# Functional Requirements

## Device Management

**2.1.1 Device Management Table**

There should be a table in device management module, to list all the devices added to the server. The table should include fields such as device Id, device type, family and subtype. There should be also buttons to edit and remove existing values of the device.

**2.1.2 Add, Edit or Delete device**

The application should be capable of adding new devices to the server. There should be fields to include device details (device id, family, subtype), location details (city, zone, landmark, latitude, longitude, time zone) and customer details (customer name, lot number). There should be options to edit or delete each device’s detail.

**2.1.3 Location map**

All the devices should be marked in the map by a location marker using the location details provided in the server.

## User Management

**2.2.1 Add or Delete user**

The application should be capable of adding new users to the system by specifying the details such as name, email id, phone number and user role. Every user should have a unique username and password. By using that credentials, a user can easily login to the application.

Added users can also be deleted using a delete button.

**2.2.2 User roles**

User Roles supported are super admin, administrator, supervisor, and operator. Different privileges can be assigned based on user roles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modules** | **Super Admin** | **Admin** | **Supervisor** | **Operator** |
| Dashboard | **🗸** | **🗸** | **🗸** | **🗸** |
| Home page | **🗸** | **🗸** | **🗸** | **🗸** |
| Live data | **🗸** | **🗸** | **🗸** | **🗸** |
| Alarm management | **🗸** | **🗸** | **🗴** | **🗴** |
| Active alarms | **🗸** | **🗸** | **🗸** | **🗸** |
| Device management | **🗸** | **🗸** | **🗴** | **🗴** |
| API key | **🗸** | **🗴** | **🗴** | **🗴** |
| User management | **🗸** | **🗸** | **🗴** | **🗴** |
| Data Diagnostics | **🗸** | **🗴** | **🗴** | **🗴** |
| API\_FAQ | **🗸** | **🗸** | **🗸** | **🗸** |

## Data Management

**2.3.1 Device status table**

There should a table to display all the devices added in the application. A status indicator should be present to display the current status of the device. If the device is online, the indicator should be in green and if offline the indicator should be in yellow. In addition to that, the table should display the details of the devices such as device Id, family, subtype, city, landmark and customer name.

**2.3.2 Historic data display**

Each and every data should be stored in the application and should be accessible whenever it is needed. The data should be displayed in both tabular and graphical format. There should be an option to select the parameter to plot the graphical chart. Tabular raw data shows the numerical data values of parameters of each device and calculated Air Quality Index (AQI) with respect to time.

**2.3.3 Statistical data of device**

There should be a table to represent the statistical values of each device such as minimum, maximum, average and sample count. Different timeframes such as daily, monthly and yearly should be present. Chart view shows the graphical representation of the statistical data.

**2.3.4 Data download**

Both historical as well as statistical data should be downloaded by specifying a start and end date. The downloaded data should be in .csv format.

## Alarm Management

**2.4.1 Alarm management table**

There should be a table to display all the alarm rules added in the application. The table should have fields such as rule name, type, clearing mode and configurations.

**2.4.2 Add, Edit or Delete alarm rule**

The application should support functionalities such as add, edit or delete alarm rules. To add a new alarm rule, the authorities should include the details such as rule name, clearing mode, device type and configurations. There should be an option to display all the devices under the particular device type, in which authorities can select the required device for that configuration. There should be a list of all the parameters for adding the configuration. Clearing modes used are time based and manual. In time-based clearing mode, there is an option to add the time interval to clear the alarm. In manual mode, the user should manually clear the alarm. There should be options to edit or delete alarms by selecting a particular rule.

**2.4.3 Active alarms**

All the generated alarms should be displayed in an active alarm page. Each data of the device is compared with the threshold value set by the authorities while adding the alarm rule. If the device value exceeds the value of the threshold then an alarm is generated. The table should have fields like alarm name, timestamp and logs.

## Error diagnostics

**2.5.1 Device table**

There should be a table to list all the devices in the application. There should be fields such as device Id, type, family, customer name and device location details.

**2.5.2 Initialization error table**

This table should display the initialization error relating to each sensor. If there is an initialization error with a sensor, then the data corresponding to that sensor should be 1. Otherwise it should be 0.

**2.5.3 Read error table**

This table should display the read error relating to each sensor. If there is a read error with a sensor then the data corresponding to that sensor should be 1. Otherwise it should be 0.

## API Module

**2.6.1 API key generator**

There should be an option to generate API keys by specifying a username and a limit. The given limit will be the maximum limit of that key on that day. The generated key should be displayed in a table with the limit and username.

**2.6.2 API support**

APIs are used for pushing data to the digital signage and it’s also used for third party integration of Envitus server application. There should be 4 types of APIs.

2.6.2.1. Live data API

2.6.2.2. Live data count API

2.6.2.3. Stat data API

2.6.2.4. Stat data count API

# Non-Functional Requirements

**3.1 Performance**

* Memory usage should be normal.
* It should handle more than 1000 concurrent users.

**3.2 Scalability/Capacity**

The system should be scalable with new features.

**3.3 Security Requirements**

* System shall use https protocol for secure communication.
* System shall have accessibility using SSH for secure communication.

**3.4 Localization**

System shall use English interface.

**3.5 Recoverability**

Data can be recovered in the case of any failures. There should be disaster recovery system to store the data of each device in addition to the environmental sensor server. In the case of any failure, the DR should act as DC. Application shall be enabled with an auto-start script mechanism to automatically restart after a reboot.

**3.6 Applicable Standards**

The system should use CPCB guidelines for AQI calculations and colour coding.

# Interface Requirements

## User Interfaces

The web application shall have a very user friendly and attractive user interface which should meet the colour scheme suggested by Central Pollution Control Board (CPCB, India) to display environmental parameters like Air Quality Index (AQI).

## Hardware Interfaces

NA

## Software Interfaces

NA

# Data Requirements

ESBHA001 environmental sensor application accepts JSON data as input.

The sample format is:

*{*

*“device\_Id” : ”abc”,*

*“data” :*

*{*

*“temperature” : 30,*

*“pressure” : 90,*

*“humidity” : 100,*

*“PM10” : 30,*

*“PM2p5” : 20,*

*“PM1”:10,*

*“CO”:0.1,*

*“CO2”:69,*

*“NO2”:0.02,*

*“SO2”:0.05,*

*“O3”:0.01,*

*“noise”:80,*

*“rain”:1,*

*“er\_read\_sensor”:10,*

*“er\_init\_sensor”:40*

*}*

*}*

# Installation

Refer [Installation Guide](https://drive.google.com/file/d/1dzGyhaBNzw1roxB8nzoxNsAtXncAMVd9/view?usp=sharing).

# Project Execution Requirements

## Deliverables and Delivery Dates

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Deliverable** | **Date** | **Responsibility** |
| 1 | Source Code |  |  |
| 2 | User Manual |  |  |
| 3 | Installation guide |  |  |

## Operating Environment

ESBHA001 IoT sensor server consists of application server and database server.

**7.2.1 Application Server**

It should have NodeJS package installing in it. This act as the platform for running the application.

* VM/PM: PM
* Operating System: Ubuntu 16.04 LTS
* Node Version: v8.10 or above

**7.2.2 Database Server**

It should have mongoDB to store and process the data.

* VM/PM: PM
* Operating System: Ubuntu 16.04 LTS
* MongoDB Version: v3.6 or above

## Testing

Testing has to cover the following.

* Unit testing by developers
* Module and integration testing by testing team
* Beta testing by potential users

## Acceptance Criteria

System shall be considered as accepted after running the system for minimum of 1 week

continuously and it measures data correctly during the testing.

# Appendix

## Appendix A: Acronyms

|  |  |
| --- | --- |
| **Term/ Acronym** | **Definition** |
| API | Application Programming Interface |
| AQI | Air Quality Index |
| CSV | Comma Separated Values |
| HTML | Hyper Text Markup Language |
| HTTPS | Hyper Text Transfer Protocol Secure |
| IoT | Internet of Things |
| JSON | JavaScript Object Notation |
| OS | Operating System |
| PM | Physical Machine |
| SRS | Software Requirement Specification |
| SSH | Secure Shell |
| VM | Virtual Machine |