Design Manual

**Main Components**

1. Database
2. Authentication
3. Cloud Storage
4. Cloud Hosting
5. Mobile Application
6. Web application
7. Control Unit
8. Mailbox

**Database**

Main technology used: Firebase Realtime Database

Graphical user interface, text, application, email

Description automatically generated

Collections & Sub collection:

* Nodes
  + nodeID(serialNum)
    - init: Boolean
    - nodeId: String

This collection maintains by the system administrator. When a new device is eligible for register, system admin add a separate collection with init field value ‘false’. When the device initialized the init field set to ‘ture’ and push a random nodeId. This nodeId used as the access point for other sub collections.

* initNodes
  + nodeID
    - adminID: String
    - users
      * user0: String

This collection maintains all the registered device information like adminID, users. Separate sub collection can be access using nodeID.

* events
  + nodeID
    - eventID
      * data: String
      * description: String
      * imageURL: String
      * name: String
      * rating: int
      * userType: String
* users
  + usersID
    - admin:Boolean
    - email: String
    - fName: String
    - imgUrl: String
    - lName: String
    - nodeId: String

Users collection holds details about users, token provide by the Firebase auth is used as the

* messages
  + nodeID
    - event: String
    - from: String
    - to: String
    - msgType: String
    - msgUrl: String
    - time: String

**Authentication & Authorization**

Technology Stack: Firebase Auth

Solution has several authentication and authorization steps

1. Device Authentication
2. User Authentication

Device Authentication

The Collection Node, that maintains init details of the Nodes. If the node is initialized database push a deviceID into the collection. That ID take as the access token for backend. In the initialize stage there is a System admin must add separate collection with the serial number for a device. Then user can register their device using register time.

User Authentication

When user make a request on with the email and password, Firebase auth send a access token for the system. That token is used to access database and user can resolve the relevant deviceID using that access token.

Graphical user interface, text, application

Description automatically generated

**Cloud Hosting**

Technology Stack: Firebase Hosting.

Firebase hosting provide a secure web hosting service with a https connection. And the version control mechanism.

Steps:

1. Install the Firebase CLI
2. Initialize the Project
   * Command: *firebase init hosting*
   * During the project initializing project, select the build director as the root directory & select the site as a one-page app.

3. Deploy the site

* + Command: *firebase deploy --only hosting*

Version control: log into the firebase console and navigate to the hosting service, in the hosting service.

Graphical user interface, text, application, email

Description automatically generated

Click on the required version of the web site, there will be a menu show required functionalities for roll back or delete.

Graphical user interface, text, application

Description automatically generated

**Web application**

Technology stack: NodeJS, ReactJS

Required dependencies on NodeJS,

chart.js: 3.5.1

firebase: 8.10.0,

materialize-css: 1.0.0-rc.2,

moment: 2.29.1,

react: 17.0.2,

react-chartjs-2: 3.0.5,

react-dom: 17.0.2,

react-dropzone: 11.4.2,

react-router-dom: 5.2.0,

react-scripts: 4.0.3,

web-vitals: 1.1.2

Web interfaces to implement,

Home Page

Graphical user interface, text, application

Description automatically generated

This window facilitate to navigate to the “Connect Device” , “Sign In” , “FAQ” options. Those options are located top right of the navbar. “Contact” option will redirect to the contact page and the “Download App” can be used to download the latest version of the Mobile app.

Sign In Page

Graphical user interface, application, Teams

Description automatically generated

Sign in page is used for taking Email and password from the user. While validating the username and password. Invalid inputs do not allow to user to proceed on. When user click on the sign In Button, front end sends a login request to the Firebase Auth. Firebase auth issues a access token as response if the request is valid unless it issue a error message. According to the access token, front end redirect to the dashboard page.

Dashboard Page

Graphical user interface, website

Description automatically generated

Dashboard page provides a few services to user,

1. Device States,
   1. Current Operation state (ON/OFF)
   2. Mailbox state (Open/Close)
   3. Event Count

To obtain the device state, web client query the back end with the access token and using that access token, device resolve the DeviceID using the device Collection in the Database. Then using that device ID, that query the state collection and obtain the states.