

# **SOFTWARE REQUIREMENTS SPECIFICATION**

## **1) INTRODUCTION**

### **1.1) PURPOSE**

- The purpose of this document is to build an online helpdesk chat system so as to provide anonymous users with one-to-one assistance with help desk staff .The app provides services in 4 major areas : Tech Support, General Enquiries, Sales associate and Cancellation/Refunding. For administrators, the application provides an interface where they can monitor the current assignments of clients to helpdesk users as well as number of rooms that are vacant / occupied.

### **1.1) INTENDED AUDIENCE**

- Clients which can anonymously get one-to-one assistance from helpdesk users in a completely private environment.

### **1.2) PROJECT SCOPE**

- The purpose of this project is to build an online helpdesk application as well as to create a user-friendly, browser-independent application for regular users, which require assistance from experts in various areas.The app currently focuses on 4 major divisions listed as tech support, general enquiries, sales associate as well as cancellation/refunding . The database server hosts all of the credentials for authenticating helpdesk staff members so that a complete secure environment can be maintained.

## 2) OVERALL DESCRIPTION

### 2.1) PRODUCT PERSPECTIVE

A dedicated one-to-one online helpdesk system for anonymous users. The application consists of 3 major categories:

- **Admin Login** : This category gives/provides a helpdesk staff member to login to a specific helpdesk room , where the client has already logged in and is waiting for the admin.
- **Helpdesk Dashboard** : This category provides a dashboard view of all the administrators a list of all of the logged in clients and admins in all of the currently used rooms. An unassigned helpdesk staff member can access this dashboard to see which client is still not assigned a helpdesk staff member and login to that respective room.
- **Client Login** : this category constitutes a dropdown list for the client to login to a respective room using an alias. This could be any name client wishes to communicate with the admin (can also be used as anonymous identity by using anonymous as display name )

### 2.2) PRODUCT FEATURES

- The main features of the online helpdesk application are noSQL mongoDB for faster and efficient authentication as well as socket.io /socket.io-client for interaction between server as well as multiple instances of client.

### 2.3) USER CLASS and CHARACTERISTICS

Users of the system (helpdesk users as well as anonymous users) have a Specific characteristics based on their roles. For an anonymous user, they login as client by providing their login name as well as the room they wish to login. For helpdesk users, they are required to provide credentials to login to a specific Room they have the authority to login to.

The clients should have the following functions:

- Login to any of the 16 available rooms for helpdesk service
- Get into one-on-one interaction with the admin logged into their room.
- As soon as a client occupies a room, it becomes unavailable for the other client to login. Therefore, the system currently provides only 16 room capacity.

The helpdesk users have the following functionalities:

- Login to an existing room based on the credentials they are required to use during the login screen.
- Able to check for existing clients waiting to be assigned a dedicated helpdesk user by logging into the Helpdesk Dashboard system.

## 2.4) OPERATING ENVIRONMENT

The operating environment for the helpdesk application is as listed below:

- noSql
- client/server system
- Operating System : Windows
- Database: mongoDB
- Platform: Express.js/Node.js/Moustache.js/Moment.js

## 2.5) DESIGN and IMPLEMENTATION CONSTRAINTS

- The database instance consists of a single collection credentials.
- Collection consists of documents following the pattern below:

```
{  
  "username" : "admin01",  
  "password" : "pass01",  
  "room" : "Tech Support 1"  
}
```

The collection in total consists of 16 documents, each of them holding the credentials for a helpdesk user to login as admin into that room.

## 2.6) ASSUMPTION DEPENDENCIES

- **4 categories of rooms noted as Tech support, Cancellation/refund, General Enquiries and Sales Associate , each have 4 sub-rooms.** The documents for the collection are such assigned values such that a particular admin is allowed to login to only a certain category of rooms. Example : a sales associate help desk user can login to any of the 4 available sales associate rooms, but he cannot login to tech support room

### 3) SYSTEM FEATURES

- **DESCRIPTION and PRIORITY** - the helpdesk system maintains a private chat system between an anonymous client as well as assigned helpdesk user. It is a high priority as client gets one-on-one interaction with the helpdesk user anonymously.
- **STIMULUS/RESPONSE SEQUENCES** - a new client always gets an updated dropdown list of only the currently available sessions. A helpdesk user on logging to the dashboard, gets an updated list of currently logged in clients as well as the assigned helpdesk users to those clients.
- **FUNCTIONAL REQUIREMENTS** - centralized database on cloud container provides administrator rights to login from any platform. Since, the credentials are stored on cloud database, security is enhanced as the application itself does not contain any content worth security breach.
- **CLIENT/SERVER SYSTEM** - the public domain of the application, which is hosted statically by the express framework consists all of the front-end files. The application starts with the index.html, which provides a drop-down page to the user. The user is there provided with 3 choices-based on each of them the app further progresses.
- **AUTHENTICATION:** authentication works in 2 different scenarios in the helpdesk application:
  1. When a helpdesk user wants to login to the helpdesk dashboard, he has to enter the username-password pair . this combination can be any of the 16 combinations as the user will only able to view the assignments and not login to any other specific room that is not assigned to him
  2. When a helpdesk user wants to login to a specific room based on the knowledge from dashboard. Now, the user is required to login using the 3 -value credentials : username, password and room . They all work in a specific pattern so as to disallow a helpdesk user to login to another room that is not under his privileges.

#### **4) EXTERNAL INTERFACE REQUIREMENTS**

##### **4.1) USER INTERFACES**

- Front-End-Software: HTML, CSS, Moment.js, Moustache.js , socket.io-client , JS
- Back-End-Software: Node.js, Express.js , Socket.io , NoSQL

##### **4.2) HARDWARE INTERFACES**

- Windows 10 Pro
- Visual Studio Code
- Chrome Developer Edition Browser (supports developer Tools and other required features)

##### **4.3) SOFTWARE INTERFACES**

- Operating System : the operating system for the helpdesk system is completely user-independent as the application will be hosted on AWS
- Database: to increase efficiency as well as to remove the complex Entity-Relationship constraints that come with SQL, the database chosen is MongoDB, with noSQL.
- Visual Studio Code: to implement the project , the programming language chosen is javascript with its enhanced features for both front-end(Moustache, Moment etc) as well as the back-end(Express , Node ).

##### **4.4) COMMUNICATION INTERFACES**

- This project supports all types of browsers. There are simple html pages using html forms as well as moustache for dynamically rendering the content on the html page (a part of it is updating the dropdown list of available rooms for the client as well as displaying logged in users in a room in the sidebar)

## **5) NONFUNCTIONAL REQUIREMENTS**

### **5.1) E-R DIAGRAM**

- Since, the database is an instance of noSql, therefore there is no requirement of E-R diagram.

### **5.2) SECURITY REQUIREMENTS**

- Security requirements for the back-end code , especially the part that involves connection to the database is of critical value. If that back-end file is compromised, anyone can gain access to the database and drop all of the documents in the collection, or even worse, modify the credentials of administrators, or even create a super-admin superseding all privileges.
- Since, the instance of database is not stored locally on the server and hosted on the cloud, careful choices are needed to be considered when choosing a vendor for database cloud service in terms of reliability as well as security.

### **5.3) SOFTWARE QUALITY ATTRIBUTES**

- The application in current conditions is able to satisfy a total of 16 concurrent clients(rooms) , each of them with their own respective helpdesk users.