**Enterprise Distributed Systems**

**Class Project – Workforce Management System**

**API Design Document**



|  |  |  |
| --- | --- | --- |
| **Team 9** | | |
| **Team Member Name** | **Team Member e-mail** | **SJSU ID** |
| Apoorva Gouni | apoorva.gouni@sjsu.edu | 009540349 |
| Mani Dheep Vipperla | manidheep.vipperla@sjsu.edu | 010093447 |
| Sairam Nutheti | sairam.nutheti@sjsu.edu | 010100545 |
| Satish  Kandimalla | satish.kandimalla@sjsu.edu | 010011612 |
| Sravani Gunti | sravani.gunti@sjsu.edu | 010090405 |
| Surabhi BommenahallyAnanda | surabhi.bommenahallyananda@sjsu.edu | 010114520 |

Contents

[**1** **Abstract** 3](#_Toc416216332)

[**2** **Services** 4](#_Toc416216333)

[**2.1** **Guard Service** 4](#_Toc416216334)

[2.1.1 **API Name: createguard** 4](#_Toc416216335)

[**2.1.2** **API Name: listguards** 4](#_Toc416216336)

[2.1.3 **API Name: deleteguard** 4](#_Toc416216337)

[**2.1.4** **API Name: editguard** 4](#_Toc416216338)

[**2.1.5** **API Name: searchguard** 5](#_Toc416216339)

[**2.1.6** **API Name: displayguard** 5](#_Toc416216340)

[**2.2** **Client Service** 5](#_Toc416216341)

[2.2.1 **API Name: createclient** 5](#_Toc416216342)

[2.2.2 **API Name: deleteclient** 5](#_Toc416216343)

[2.2.3 **API Name: listclient** 6](#_Toc416216344)

[2.2.4 **API Name: searchclient** 6](#_Toc416216345)

[**2.2.5** **API Name: displayclient** 6](#_Toc416216346)

[**2.2.6** **API Name: generateBill** 6](#_Toc416216347)

[**2.2.7** **API Name: viewBill** 6](#_Toc416216348)

[**2.3** **Building Service** 7](#_Toc416216349)

[**2.3.1** **API Name: createBuilding** 7](#_Toc416216350)

[**2.3.2** **API Name: deleteBuilding** 7](#_Toc416216351)

[**2.3.3** **API Name: editBuilding** 7](#_Toc416216352)

[**2.4** **Report** **Service** 7](#_Toc416216353)

[**2.4.1** **API Name: createReport** 8](#_Toc416216354)

[**2.4.2** **API Name: listBuildingReport** 8](#_Toc416216355)

[**2.4.3** **API Name: listClientReport** 8](#_Toc416216356)

[**2.4.4** **API Name: searchReport** 8](#_Toc416216357)

[**2.4.5** **API Name: getReportDetails** 8](#_Toc416216358)

[**2.5** **Alert Service** 9](#_Toc416216359)

[**2.5.1** **API Name: createAlert** 9](#_Toc416216360)

[**2.5.2** **API Name: submitAlert** 9](#_Toc416216361)

[**2.5.3** **API Name: listBuildingAlert** 9](#_Toc416216362)

[**2.5.4** **API Name: listClientAlert** 9](#_Toc416216363)

[**2.5.5** **API Name: searchAlert** 9](#_Toc416216364)

[**2.5.6** **API Name: getAlertDetails** 10](#_Toc416216365)

# **Abstract**

In this project we will design a 3-tier application that will implement the methods of a Workforce Management System for security services such as bay alarm and ADT. We will manage and implement different types of objects in this Workforce Management System like guards, buildings, clients, schedules, reports and alerts. We will implement an associated database schema to represent how a given object should be stored in a relational database. We use web services as the middleware technology. The entities which will be supported in our system are: Person, Guard, Client, Building, Report, and Alert.

# **Services**

Below are the different kinds of services provided.

## **Guard Service**

This service provides the functionalities like creating a guard, deleting a guard, displaying list of guards / guards detail information, editing the guard information.

### **API Name: createguard**

* **HTTP Method:** POST
* **Request: {“**firstName”: firstName, “lastName”: lastName, “address”: address (addressLine1, addressLine2, city, state, zipCode), “phNo”: phNo, “email”: email,”password”:”password”, “roleId”:”roleId”,”guardID”: guardID}.
* **Response:** {Success: ‘Guard Created’ / Error: ‘Failed to Create Guard’}
* **Description:** This method creates a guard with the entities like First Name, Last Name, Address (addressLine1, addressLine2, City, State, zipCode), Phone Number, Email and Guard ID.

**Note:** First Create User in user table and get user id. Create row in userdetails with user id collected in earlier step. Create row in guard table using same user id.

### **API Name: listguards**

* **HTTP Method:** GET
* **Request:**
* **Response:** {Success: Guards /Failure: ‘Failed to Load Guards’}
* **Description:** This method gives all the guards available in the system.

### **API Name: deleteguard**

* **HTTP Method:** **DELETE**
* **Request: { “**guardID**” :** guardID}
* **Response: {**Success : ‘Guard with ID  XXXX is deleted’/ Error: ‘Unable to delete the Guard with ID XXXX’}
* **Description:** This method deletes an existing guard.
* **Note:** Set isDeleted value as 1 in User table

### **API Name: editguard**

* **HTTP Method:** POST
* **Request**: { “guardID” : guardID}
* **Response:**  {Success : ‘Information updated successfully’/ Error: ‘Unable to update the provided information’}
* **Description:** This method updates the information of a particular Guard.

### **API Name: searchguard**

* **HTTP Method:** POST
* **Request**: {“ guardID”:guardID, “firstName”: firstName, “lastName”: lastName}

Here the user can pass either guardID or first name or last name; or can search by giving both first name and last name also

* **Response:**  {Success : ‘List of guards’/ Error: ‘No guard with given ID/name’}
* **Description:** This method searches the entire system and gives the list of guards matching to the search attributes

### **API Name: displayguard**

* **HTTP Method:** POST
* **Request**: { “guardID”: guardID, “firstName” : firstName, “lastName”:lastName}
* **Response:**  {Success : ‘Details of Guard’/ Error: ‘No details found for guard ID/firstName/lastName’}
* **Description:** This method gives all the details like personal information of a guard as well as the details like to which building he responsible for, weekly schedule, weekly is working hours of a particular Guard.

## **Client Service**

This service provides functionalities like creating a client, deleting a client, displaying list of clients, editing the client information.

### **API Name: createclient**

* **HTTP Method:** POST
* **Request:** {“firstName”: firstName, “lastName”: lastName, “address” : address (addressLine1, addressLine2, city, state, zipCode), “phNo”:phNo, “email”:email, “password”:”password”, “startDate”:startDate,”endDate”:”endDate”, “serviceFee”: “ServiceFee”,”roleId”:”roleId”}
* **Response:** {Success : ‘Client created successfully’/ Error: ‘ Unable to create the client’}
* **Description:** This method creates a client with information like client’s first name, last name, client ID, address (address line1, address line 2, city, state, zip code), phone number, email ID and also type of the subscription plan client is interested in.

### **API Name: deleteclient**

* **HTTP Method:** POST
* **Request:** {“clientID”:clientID}
* **Response:** {Success : ‘Client with ID  XXXX is deleted’/ Failure: ‘Unable to delete the Client with ID XXXX’}
* **Description:** This method deletes an existing client.

### **API Name: listclient**

* **HTTP Method:** GET
* **Request:**
* **Response:** {Success: Clients /Failure: ‘Failed to Load Clients}

**Description:** This method gives all the clients available in the system

### **API Name: searchclient**

* **HTTP Method:** POST
* **Request:** {“clientID”: clientID, “firstName”: firstName, “lastName”: lastName}  
  Here the user can pass either clientID or first name or last name; or can search by giving both first name and last name also.
* **Response:** {Success : ‘List of clients’/ Failure: ‘No client with given ID/name’}
* **Description:** This method searches the entire system and gives the list of clients matching to the search attributes.

### **API Name: displayclient**

* **HTTP Method:** POST
* **Request**: {“clientID”: clientID, “firstName”: firstName, “lastName”: lastName}
* **Response:**  {Success : ‘Details of Client/ Failure: ‘No details found for client ID/firstName/lastName’}
* **Description:** This method gives all the client details like personal information, information of building he/she owns and also the subscription plan client signed up.

### **API Name: generateBill**

* **HTTP Method:** POST
* **Request**: {“ clientID”: clientID, “startDate”:startDate, “endDate”:endDate}
* **Response:**  {Success : ‘Bill generated successfully’/ Failure: ‘Unable to generate bill’}
* **Description:** This method generates the bill (calculates the amount based on the subscription type) of a particular client taking the start date and end date attributes.  
  **Note :**Bill is generated only by Admin

### **API Name: viewBill**

* **HTTP Method:** POST
* **Request**: {“ clientID” : clientID, “startDate” : startDate, “endDate”: endDate}
* **Response:**  {Success : ‘bill data’/ Failure: ‘Unable to get the bill’}
* **Description:** This method gives the already generated bill by taking the start date, end date and client ID.

## **Building Service**

This service provides functionalities like creating a building, deleting a building and editing a building information.

### **API Name: createBuilding**

* **HTTP Method:** POST
* **Request:** {“ buildingID”:buildingID, “addressLine1”:addressLine1, “addressLine2”:addressLine2, “city”: city, “state”: state, “zipCode”:zipCode, “clientID”:clientID, “checkinIDs”:checkinIDs}
* **Response:**{Success: ‘Building created successfully’/ Error: ‘Unable to create a building’|
* **Description:** This method creates a building with details like building ID, addressline1, addressline2, city, state, zipcode , clientID who owns the building and also the Checkin point IDs

#### **API Name: Create CheckInPoint**

* HTTP Method: POST
* Request:{“buildingId”:”buildingId’,”description”:”description”,”latitude”:”latitude”, ”longitude’:”longitude”}
* Response:{Success: “checkInId” / Error:”Failed to create Checkin”
* Description**:**  This method created a checkin point with details like checkin description, latitude and longitude.

### **API Name: deleteBuilding**

* **HTTP Method:** POST
* **Request:** {“ buildingID”:buildingID}
* **Response:** {Success : ‘Building with ID  XXXX is deleted’/ Error: ‘Unable to delete the Building with ID XXXX’}
* **Description:** This method deletes an existing building.

### **API Name: editBuilding**

* **HTTP Method:** POST
* **Request:** {“ buildingID”:buildingID}
* **Response:** {Success : ‘Information updated successfully’/ Failure: ‘Unable to update the provided information’}
* **Description:** This method updates the information of a particular Building.

## **Report** **Service**

This service provides the functionalities like creating the reports, listing the reports, searching the reports and displaying the report details.

### **API Name: createReport**

* **HTTP Method:** POST
* **Request:** {“ buildingID”:buildingID, “guardID”:guardID, “incidentDetails”:incidentDetails, “parkingViolationDetails”:parkingViolationDetails, “serviceCallDetails”:serviceCallDetails, “maintenanceCallDetails”:maintenanceCallDetails, “patrolingDetails”:patrolingDetails}

**Note:** Here incidentDetails, parkingViolationDetails, maintenanceCallDetails, serviceCallDetails, patrolingDetails are JSON objects which contains respective fields.

* **Response:** {Success : ‘Report logged successfully’/ Failure: ‘Unable to log the report’}
* **Description:** This method logs a report with details like building name, guard name, guard ID, patrolling details, incident details, parking violation details, maintenance call details and service call details .

### **API Name: listBuildingReport**

* **HTTP Method:** POST
* **Request:** {“ buildingName”:buildingName}
* **Response:** {Success : ‘List of all reports of a building XXX’/ Failure: ‘Unable to fetch the reports for building XXX}
* **Description:** This method retrieves the list of all reports of a particular building.

### **API Name: listClientReport**

* **HTTP Method:** POST
* **Request:** {“ clientID”:clientID(SSN format)}
* **Response:** {Success : ‘List of all reports of a client XXX’/ Failure: ‘Unable to fetch the reports for client XXX}
* **Description:** This method retrieves the list of all reports of a particular client.

### **API Name: searchReport**

* **HTTP Method:** POST
* **Request:** {“startDate”: startDate, “endDate”: endDate, “clientID”: clientID, “buildingName”: buildingName}   
  Here the user can pass either clientID or startDate and endDate or buildingID; or can search by giving multiple parameters.
* **Response:** {Success : ‘List of reports’/ Failure: ‘No report with given search list}
* **Description:** This method searches the entire system and gives the list of reports matching to the search attributes.

### **API Name: getReportDetails**

* **HTTP Method:** POST
* **Request:** {“ reportID”: reportID}
* **Response:** { Success : ‘Report Details’/ Failure: ‘Unable to fetch the report details for report ID XXX’}

**Note:** Here report details is a JSON object which contains fields like building name, guard name, guard ID, patrolling details, incident details, parking violation details, maintenance call details and service call details .

* **Description:** This method displays a report with details like building name, guard name, guard ID, patrolling details, incident details, parking violation details, maintenance call details and service call details.

## **Alert Service**

This service provides the functionalities like creating an alert, listing the alerts for a building, listing the alerts for a client, searching an alert and getting alert details.

### **API Name: createAlert**

* **HTTP Method:** POST
* **Request:** {“ alertType”:alertType}
* **Response:** { Success : ‘Alert created successfully’/ Failure: ‘Unable to create the Alert’}
* **Description:** This method creates the predefined alerts.

### **API Name: submitAlert**

* **HTTP Method:** POST
* **Request:**  {“ AlertDetails”:AlertDetails}

**Note:** Here AlertDetails is a JSON object which contains the fields like alertType, BuildingID, guard details who raised the alert, alert timestamp

* **Response:** { Success : ‘Alert submitted successfully’/ Failure: ‘Unable to log the Alert’}
* **Description:** This method logs the alerts.

### **API Name: listBuildingAlert**

* **HTTP Method:** POST
* **Request:** {“ buildingName”:buildingName}
* **Response:** {Success : ‘List of all alerts of a building XXX’/ Failure: ‘Unable to fetch the alerts for building XXX}
* **Description:** This method retrieves the list of all alerts of a particular building.

### **API Name: listClientAlert**

* **HTTP Method:** POST
* **Request:** {“ clientID”:clientID(SSN format)}
* **Response:** {Success : ‘List of all alerts of a client XXX’/ Failure: ‘Unable to fetch the alerts for client XXX}
* **Description:** This method retrieves the list of all alerts of a particular client.

### **API Name: searchAlert**

* **HTTP Method:** POST
* **Request:** {“startDate”: startDate, “endDate”: endDate, “clientID”: clientID, “buildingName”: buildingName}   
  Here the user can pass either clientID or startDate and endDate or buildingName; or can search by giving multiple parameters.
* **Response:** {Success : ‘List of alerts’/ Failure: ‘No alert with given search list}
* **Description:** This method searches the entire system and gives the list of alerts matching to the search attributes.

### **API Name: getAlertDetails**

* **HTTP Method:** POST
* **Request:** {“AlertID”:AlertID}
* **Response:** { Success : ‘Alert Details’/ Failure: ‘Unable to fetch the alert details for alert ID XXX’}

**Note:** Here alert Details is a JSON object which contains fields like building name, guard name, guard ID, alert type, alert severity and alert description.

* **Description:** This method displays an alert with details like building name, guard name, guard ID, alert type, alert severity and alert description.