# Report 4: Software Design Description

## Design Overview

This document describes the technical and user interface design of MSSC System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.

The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.

The database design describes the relationships between entities and details of each entity.

Document overview:

* Section 2: gives an overall description of the system architecture design.
* Section 3: gives component diagrams that describe the connection and integration of the system.
* Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 5: overview some main user interface of system.
* Section 6: describe fully attributed ERD.
* Section 7: describe in details all algorithms used in the system.

## System Architectural Design

### Choice of System Architecture

The system is developed using MVC architecture, combine with front end – back end RESTful Web service interface architecture.

## Component Diagram

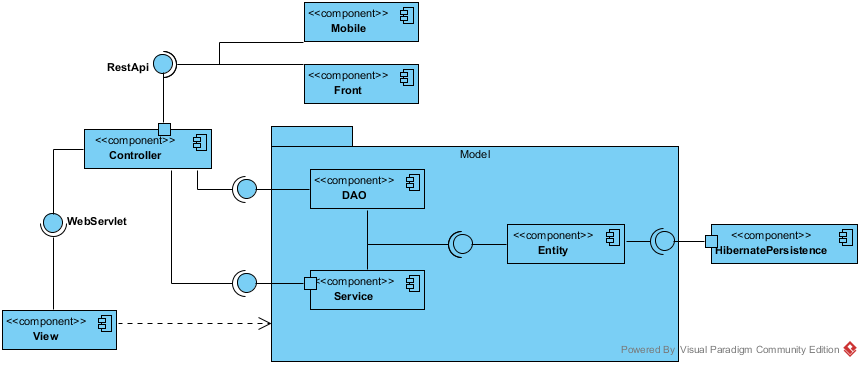


Figure 1: Component Diagram

## Detailed Description Explanation

### Class Diagram

#### Entity Package Class Diagram

Figure 2: Class Diagram

#### Class Diagram Explanation

##### Class 1

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

##### Class 2

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

#### DAO Package Class Diagram

Figure 3: Class Diagram

#### Class Diagram Explanation

##### Class 1

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

##### Class 2

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

#### Service Package Class Diagram

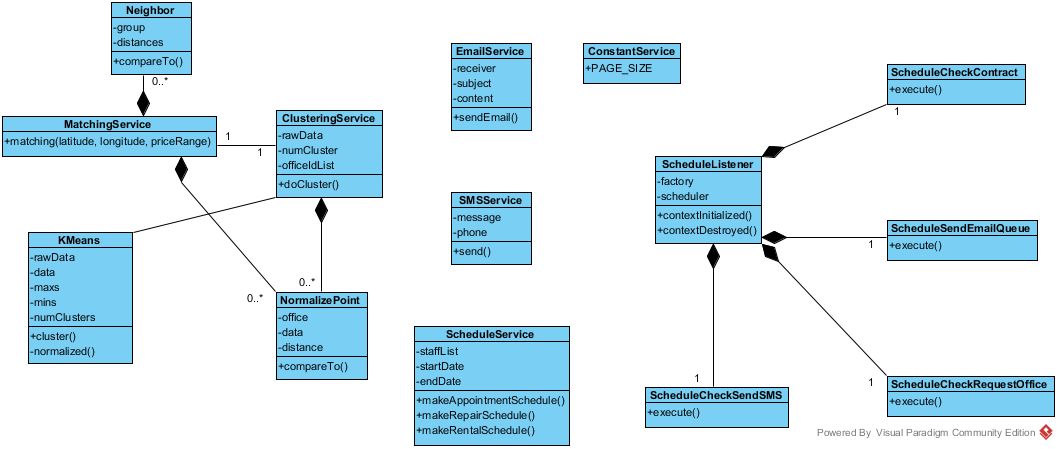


Figure 4: Service Package Class Diagram

#### Class Diagram Explanation

##### EmailService

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| receiver | String | private | The recipient of the email |
| subject | String | private | The subject of the email |
| content | String | private | The content of the email |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| sendEmail | boolean | public | Send the email with the subject and content to the receiver |

##### SMSService

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| message | String | private | The SMS message will be sent |
| phone | String | private | The phone number of the recipient |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| send() | String | public | Send the SMS message to the recipient. Return “SUCCESS” if sent success, otherwise return the error message form the service provider |

##### ConstantService

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| PAGE\_SIZE | int | public | The constant for the page size of the list in admin pages |

##### ScheduleService

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| staffList | List<Staff> | private | The list of all staff for the service |
| startDate | DateTime | private | The start date of the week for the schedule |
| endDate | DateTime | private | The end date of the week for the schedule |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| makeAppointmentSchedule() | Map | public | Return the schedule hash map with appointment id and staff username for the unassigned appointment |
| makeRepairSchedule() | Map | public | Return the schedule hash map with repair id and pre-arranged repair for the unassigned repair |
| makeRentalSchedule() | Map | public | Return the schedule hash map with rental id and pre-arranged rental for the unassigned rental |

##### KMeans

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| rawData | double[][] | private | The raw data for k-means algorithm |
| data | double[][] | private | The normalized data for k-means algorithm |
| maxs | double[] | private | The array for max value of each dimension of k-means |
| mins | double[] | private | The array for min value of each dimension of k-means |
| numClusters | int | private | The number of cluster (k) in k-means algorithm |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| cluster | double[] | public | Do the k-means algorithm. Return the array of group id for each of input data |
| normalized | double[][] | private | Normalize the raw data |

##### ClusteringService

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| rawData | double[][] | private | The input raw data of all office |
| numCluster | int | private | The calculated k-means number |
| officeIdList | int[] | private | List of all office id |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| doCluster | int[] | public | Do the k-means clustering for all office. The result is saved to database also return the array of office group by all office |

##### NormalizePoint

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| office | Office | private | Use to store the office for clustering service |
| data | double[] | private | The normalize data for current office |
| distance | double | private | The distance between the data point and query point in MatchingService |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| compareTo() | int | public | Return the compare result of 2 NormalizePoint |

##### MatchingService

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| matching() | List<Office> | public | Do the matching group for the input office point and the all office data. Return the list of matched office group sort by relevant |

##### Neighbor

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| group | int | private | The index of the office group |
| distances | List<Double> | private | The list of distance of query point and office in group for the KNN matching service |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| compareTo() | int | public | Compare distance between query point and two list of data |

##### ScheduleListener

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| factory | SchedulerFactory | private | The factory for initiate the scheduler |
| scheduler | Scheduler | private | The class contain the instance of the scheduler factory |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| contextInitialized() | void | public | Override context listener when the app initialized |
| contextDestroyed() | void | public | O Override context listener when the app destroyed |

##### ScheduleCheckContract

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| execute | void | public | Execute the schedule for check the contract due date. Cancel the contract if the contract at due date |

##### ScheduleCheckRequestOffice

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| execute | void | public | Execute the schedule for check the Request office. Stored the found office and queue to send an email to customer |

##### ScheduleCheckSendSMS

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| execute | void | public | Execute the schedule for check the appointment date. Send a SMS to user if the appointment date is near |

##### ScheduleCheckEmailQueue

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| execute | void | public | Execute the schedule for check the email in queue to send |

#### Controller Package Class Diagram

Figure 5: Class Diagram

#### Class Diagram Explanation

##### Class 1

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

##### Class 2

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

#### View Package Class Diagram

Figure 6: Class Diagram

#### Class Diagram Explanation

##### Class 1

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

##### Class 2

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

#### Front Package Class Diagram

Figure 7: Class Diagram

#### Class Diagram Explanation

##### Class 1

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

##### Class 2

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
|  |  |  |  |

#### Mobile Package Class Diagram

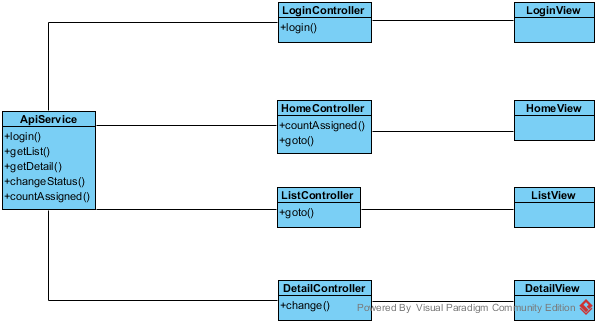


Figure 8: Mobile Package Class Diagram

#### Class Diagram Explanation

##### ApiService

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| login() | function | public | Call the login api and return the result. |
| getList() | function | public | Call the getList api to return the list of job for current staff (appointment/rental/repair) |
| getDetail() | function | public | Call the getDetail api to return the detail of current job |
| countAssigned() | function | public | Call the countAssigned api to return the number of assigned for current job |
| changeStatus() | function | public | Call the changeStatus api to change the status of current job |

##### LoginController

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | string | private | The inputted username |
| password | string | private | The inputted password |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| login() | function | private | Call the login function of ApiService |

##### LoginView

This is the presentation view of LoginController

##### HomeController

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| title | string | private | The title of the job (appointment/rental/repair) |
| badge | Array | private | The array of number of assigned job |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| countAssigned() | function | private | Call the countAssigned function of ApiService to get the job number |
| goto() | function | private | Go to the list page of the job |

##### HomeView

This is the presentation view of HomeController

##### ListController

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| title | string | private | The title of the job (appointment/rental/repair) |
| data | Array | private | The array of the detail of job |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| goto() | function | private | Go to the detail page of the job |

##### ListView

This is the presentation view of ListController

##### DetailController

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| title | string | private | The title of the job (appointment/rental/repair) |
| data | object | private | The detail of job |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| change() | function | private | Call the api to change the status of the job |

##### DetailView

This is the presentation view of DetailController

### Interaction Diagram

#### <Staff> Create office

**Summary:** This diagram used to describe the process of staff create an office

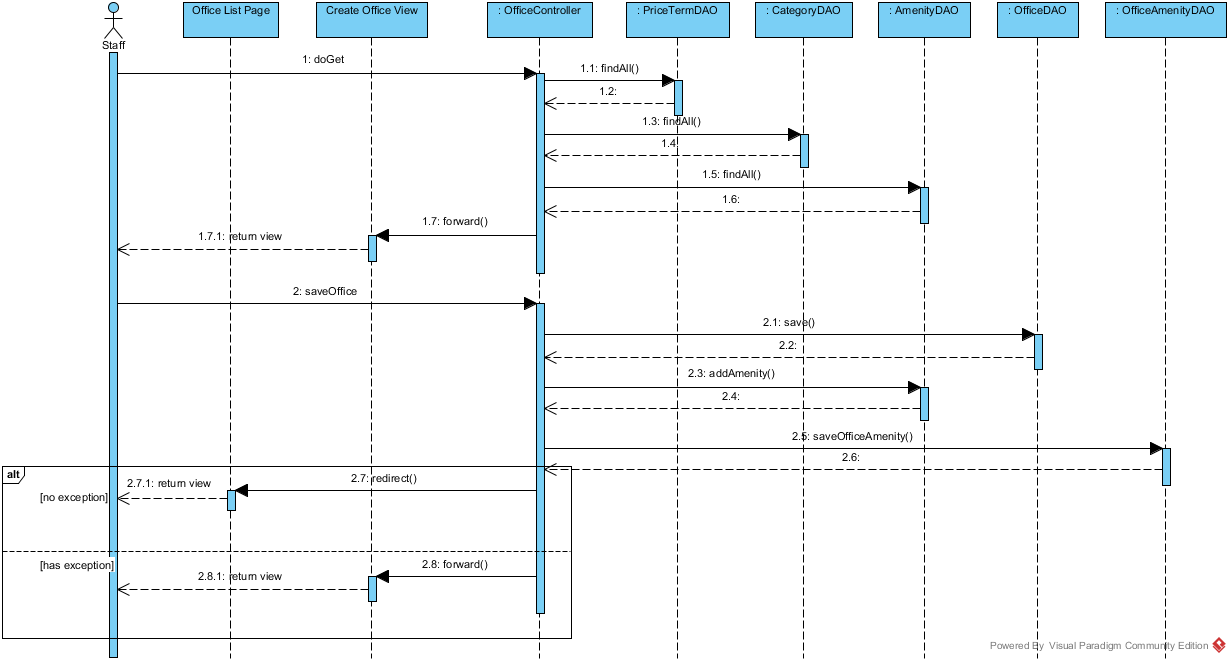


Figure 10: Create office Sequence Diagram

#### <Manager> Create contract

**Summary:** This diagram used to describe the process of create new contract.

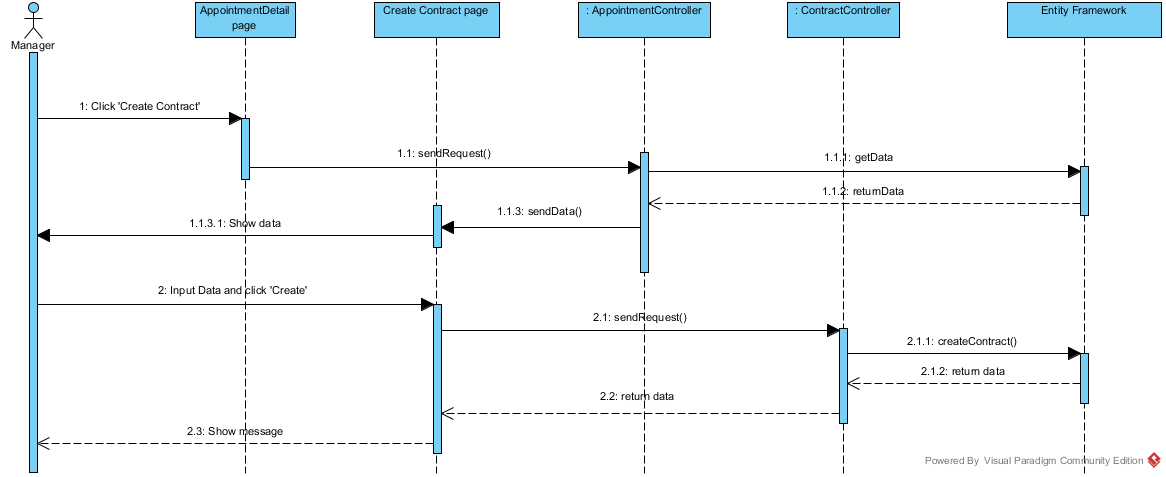


Figure 11: Create Contract Sequence Diagram

#### <Manager> Assign Appointment

**Summary:** This diagram used to describe the process of manager assign an appointment to a staff

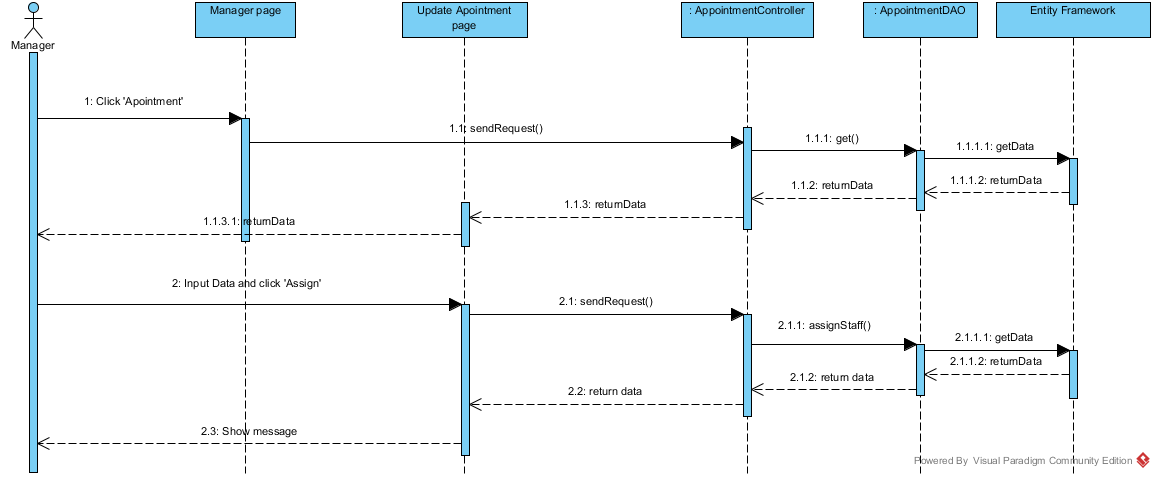


Figure 12: Assign Appointment Sequence Diagram

#### <Manager> Update Appointment

**Summary:** This diagram used to describe the process of manager update an appointment detail

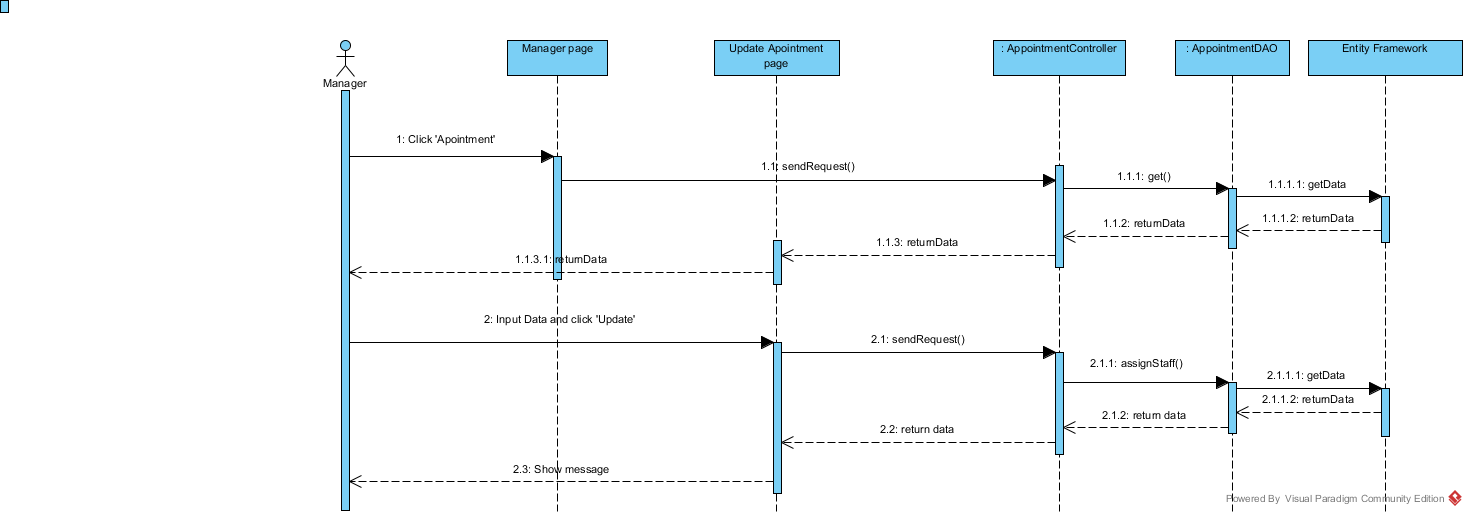


Figure 13: Update Appointment Sequence Diagram

#### <Customer> Get Office List

**Summary:** This diagram used to describe the process of customer request repair office

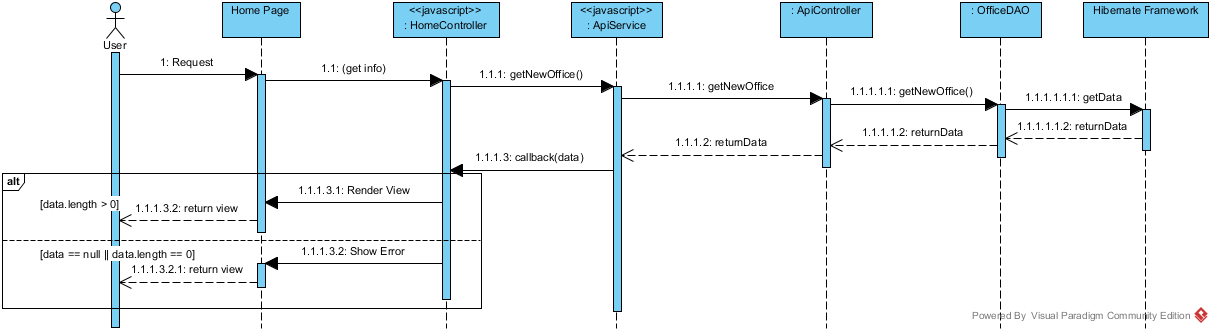


Figure 14: Get office list Sequence Diagram

#### <Customer> Request Appointment

**Summary:** This diagram used to describe the process of customer request an appointment.

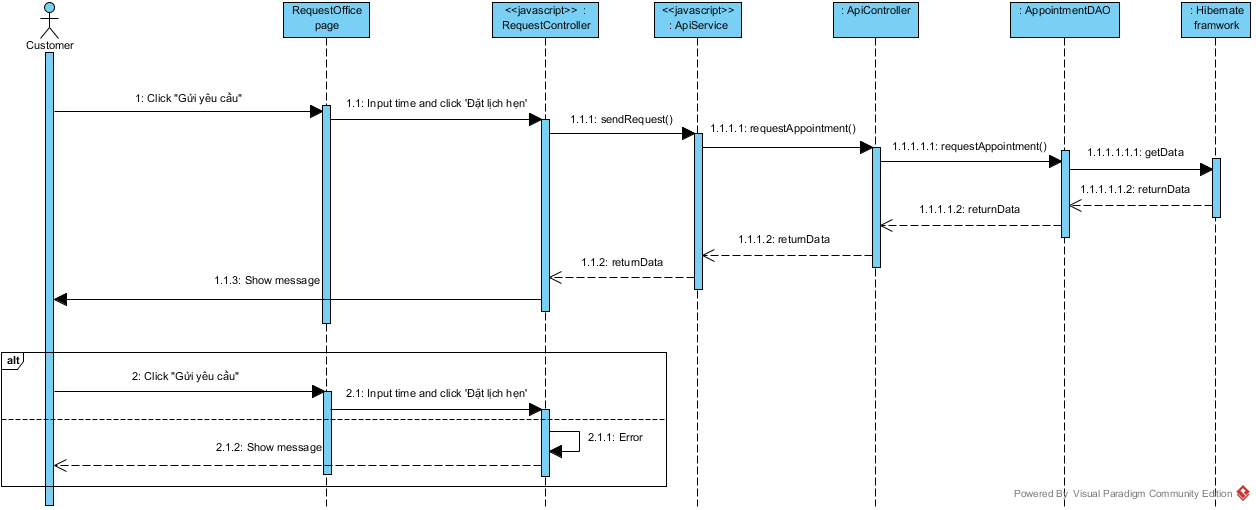


Figure 15: Request Appointment Sequence Diagram

#### <Customer> Request Repair

**Summary:** This diagram used to describe the process of customer request repair office

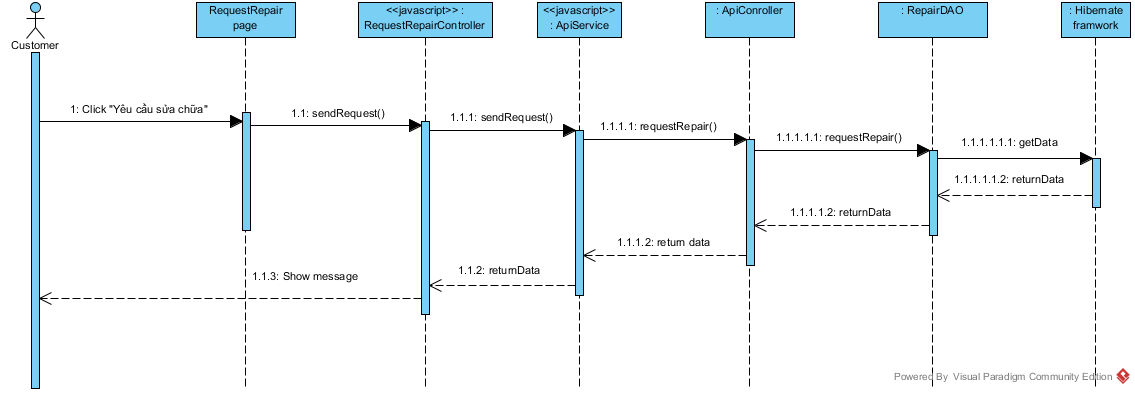
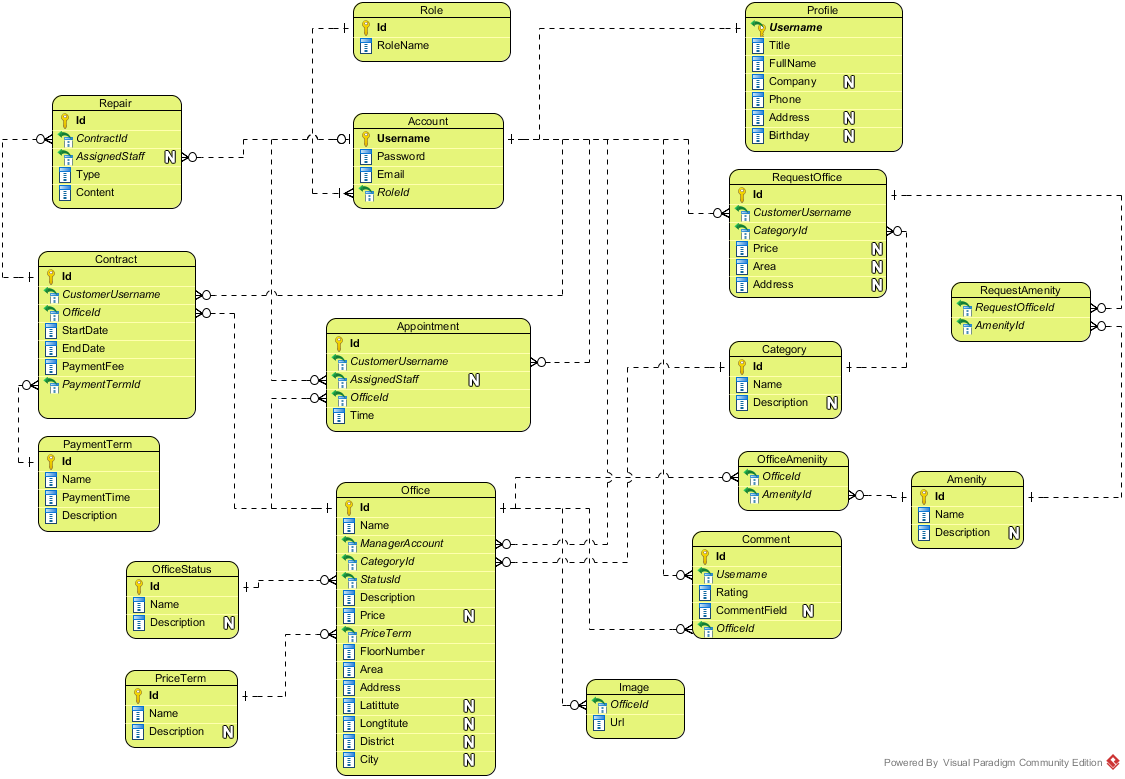


Figure 16: Request Repair Sequence Diagram

## Database Design

### Logical Diagram



**Figure 17: Logical Database Diagram**

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| Account | Describe account of user of system. |
| Staff | Describe account type staff |
| Admin | Describe account type admin |
| Manager | Describe account type manager |
| Customer | Describe account type customer |
| Profile | Describe the contact information for customer |
| Office | Describe information of the office |
| Category | Describe the category of the office |
| Amenity | Describe the amenities of the office |
| Image | List of images of the office |
| Appointment | Describe the information of the appointment of customer for the office |
| Contract | Describe the information of contract between customer and office |
| Request Office | Describe the stored office request of customer |
| Repair | Describe the repair request for each contract |
| Rental | Describe the rental request for contract |
| Rental Item | Describe the rental items which available for rental request |
| Account | Describe account of user of system. |
| Staff | Describe account type staff |
| Admin | Describe account type admin |
| Manager | Describe account type manager |
| Customer | Describe account type customer |

Table 1: Entity Data Dictionary