**D. Report No. 4 Software Design Description**

**1. Design Overview**

- *This document describes the technical and user interface design of Hospital Portal 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.*

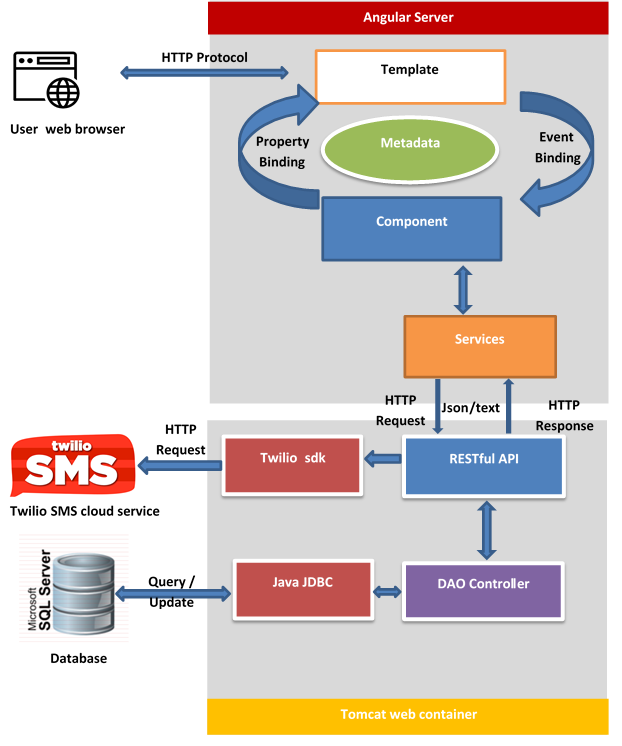
*- The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.*

*- 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: describe screens design.*
* *Section 6: describe a fully attributed ERD.*
* *Section 7: describe algorithms.*

**2. System Architectural Design**

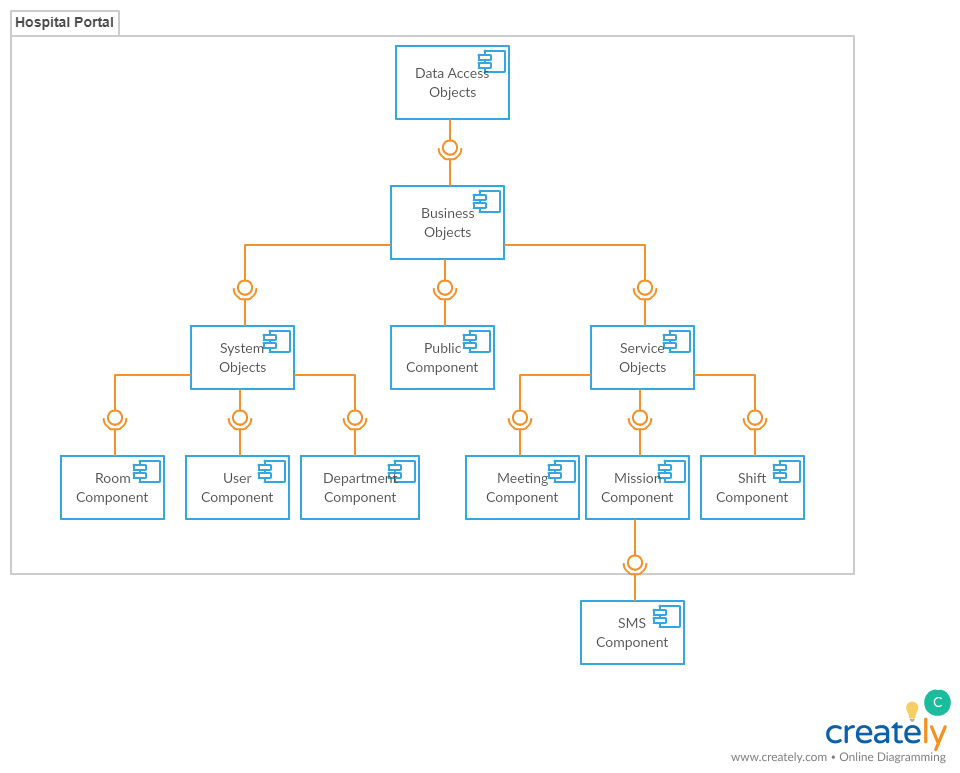
****

**Figure 9 System architecture design**

**2.1 Web application architecture description:**

In Web Application, the system was built based on MVC, while the front end and back end are separated and communicate using the HTTP protocol. With Modal and Controller at the back end and View at front end. We chose a separate structure because of the following advantages:

* Back end to support J2EE compatible MVC and REST API JDBC structure and easy use familiar.
* Front end with Angular 5, support for binding data, one page design, and compatible modules on the mobile interface.
* Separating the controller and model from the view will facilitate future multi-interface and multi-language application development.
* Due to the unique nature of the hospital, the database will sometimes have to be stored locally at the hospital. Separating two servers will ensure that the hosting server on cloud still meets the requirements.
* Technology is selected based on the skill and experience of the developer team.

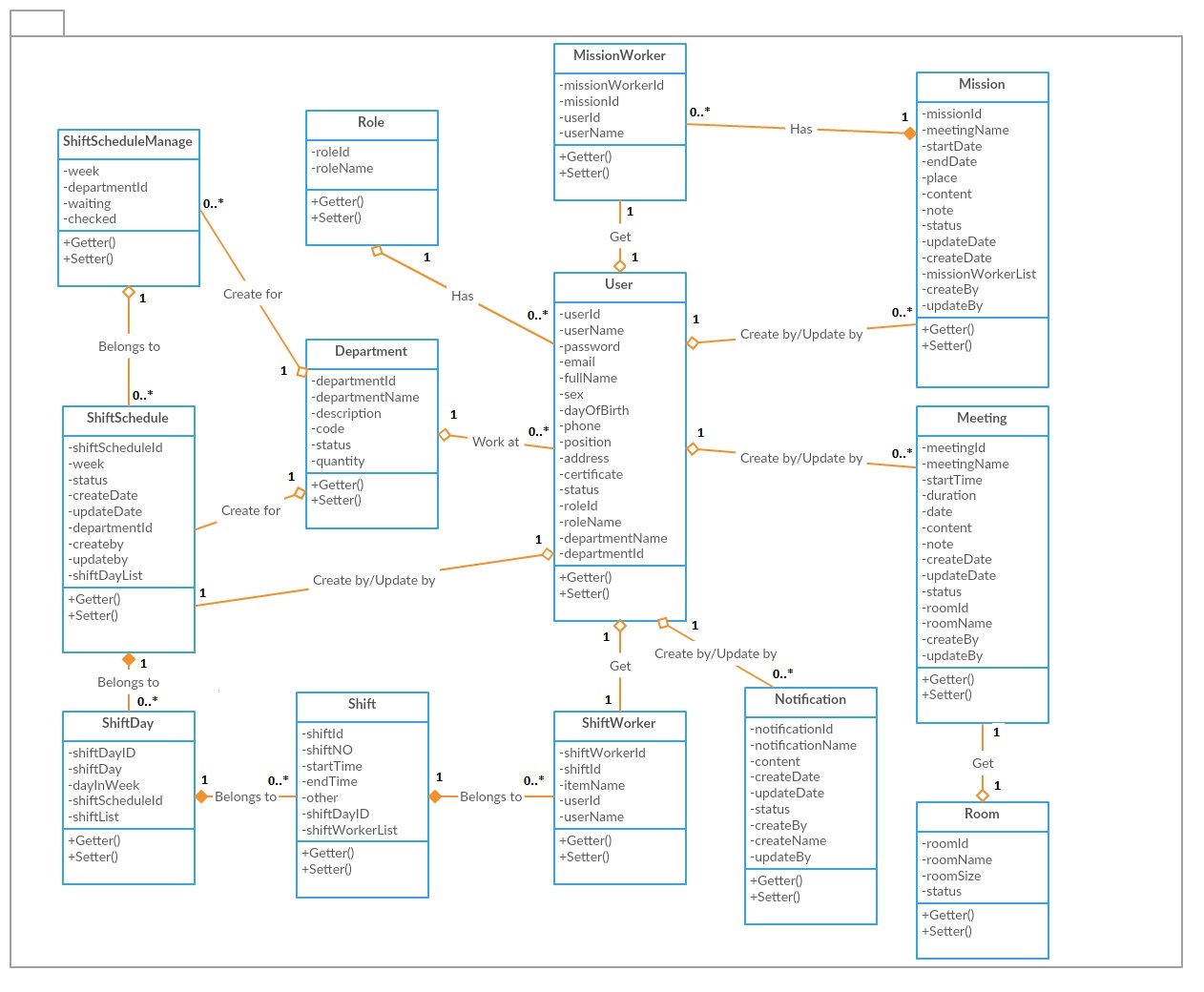
**3. Component Diagram**

**Figure 10 Component Diagram**

**Table 10 Component Dictionary**

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| Data Access Objects | Component to handle interaction between the system and database |
| Business Objects | Common objects to handle domain business operations for each component |
| System Objects | Common objects to handle domain operations for each system component |
| Public Component | Component to handle guest activities in the system |
| Service Objects | Common objects to handle domain operations for each service component |
| Room Component | Component to handle room resource in the system |
| User Component | Component to handle account in the system |
| Department Component | Component to handle department in the system |
| Meeting Component | Component to handle meeting scheduler in the system |
| Mission Component | Component to handle mission scheduler in the system |
| Shift Component | Component to handle shift scheduler in the system |
| SMS Component | Handle SMS process with Twilio SMS API |

**4. Detailed Description**

**4.1 Class Diagram**

**Figure 11 Class Diagram**

|  |  |
| --- | --- |
| *Class dictionary: describe Class* | |
| **Class Name** | **Description** |
| Department | Contain the department information |
| User | Contain the user information |
| Meeting | Contain the meeting information |
| Mission | Contain the mission information |
| Room | Contain the room information |
| Notification | Contain the notification information |
| Mission worker | Contain the mission worker information |
| Shift worker | Contain the shift worker information |
| Role | Contain the role information |
| Shift | Contain the shift information |
| Shift schedule manage | Contain the shift schedule manage information |
| Shift schedule | Contain the shift schedule information |
| Shift day | Contain the shift day information |

Table 11: Class dictionary

**4.2 Class Diagram Explanation**

***4.2.1 Role***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| RoleId | int | Private | Unique identifier of a role |
| RoleName | string | Private | Role name |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.2 Department***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| DepartmentId | int | Private | Unique identifier of a department |
| DepartmentName | string | Private | Department name |
| Description | string | Private | Description of department |
| Status | int | Private | Status of department |
| Code | string | Private | Code of department |
| Quantity | int | Private | Number of employee in department |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.3 User***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| UserId | int | Private | Unique identifier of a user |
| UserName | string | Private | Username |
| Password | string | Private | Password of user |
| Email | string | Private | Email of user |
| FullName | String | Private | Full name of user |
| Sex | int | Private | Sex of user |
| DayOfBirth | date | Private | Day Of Birth of user |
| Phone | string | Private | Phone of user |
| Position | string | Private | Position of user |
| Address | string | Private | Address of user |
| Certificate | string | Private | Certificate of user |
| Status | int | Private | Status of user |
| RoleId | int | Private | RoleId of user |
| RoleName | string | Private | Role name of user |
| DepartmentName | string | Private | Department name of user |
| DepartmentId | int | Private | DepartmentId of user |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.4 Shift schedule manage***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| Week | string | Private | Unique identifier of Shift schedule manage |
| DepartmentId | int | Private | DepartmentId of Shift schedule manage |
| Waiting | int | Private | Waiting status |
| Checked | int | Private | Checked status |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.5 Shift schedule***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| ShiftScheduleId | int | Private | Unique identifier of Shift schedule |
| Week | int | Private | Week of Shift schedule |
| Status | int | Private | Status of Shift schedule |
| CreateDate | datetime | Private | Create date of Shift schedule |
| UpdateDate | datetime | Private | Update date of Shift schedule |
| DepartmentId | int | Private | DepartmentId of Shift schedule |
| CreateBy | int | Private | Create by |
| UpdateBy | int | Private | Update by |
| ShiftDayList | list (ShiftDay) | Private | Shift day list |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.6 Shift day***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| ShiftDayId | int | Private | Unique identifier of Shift day |
| ShiftDay | string | Private | Shift day |
| DayInWeek | string | Private | Day in week |
| ShiftScheduleId | int | Private | Shift schedule Id |
| ShiftList | list (Shift) | Private | Shift list |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.7 Shift***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| ShiftId | int | Private | Unique identifier of shift |
| ShiftNo | int | Private | Shift no of shift |
| StartTime | time | Private | Start time of shift |
| EndTime | time | Private | End time of shift |
| Other | string | Private | Other |
| ShiftDayId | int | Private | Shift day Id |
| ShiftWorkerList | List (shift worker list) | Private | Shift worker list |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.8 ShiftWorker***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| ShiftWorkerId | int | Private | Unique identifier of shift worker |
| ShiftId | int | Private | Shift Id of shift worker |
| itemName | string | Private | Item name of shift worker |
| UserId | int | Private | UserId for shift worker |
| UserName | string | Private | Username for shift worker |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.9 MissionWorker***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| MissionWorkerId | int | Private | Unique identifier of mission worker |
| MissionId | int | Private | MissionId |
| UserId | int | Private | UserId for mission worker |
| UserName | string | Private | Username for mission worker |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.10 Mission***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| MissionId | int | Private | Unique identifier of mission |
| StartDate | date | Private | Start date of mission |
| EndDate | date | Private | End date of mission |
| Place | string | Private | Place of mission |
| Content | string | Private | Content of mission |
| Note | string | Private | Note of mission |
| Status | int | Private | Status of mission |
| UpdateDate | date | Private | Update date of mission |
| CreateDate | date | Private | Create date of mission |
| MissionWorkerList | List (MissionWorker) | Private | Mission worker list |
| CreateBy | int | Private | Create by |
| UpdateBy | int | Private | Update by |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.11 Meeting***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| MeetingId | int | Private | Unique identifier of meeting |
| MeetingName | string | Private | Meeting name |
| StartTime | time | Private | Start time of meeting |
| Duration | time | Private | Duration of meeting |
| Date | date | Private | Date of meeting |
| Content | string | Private | Content of meeting |
| Note | string | Private | Note of meeting |
| Status | int | Private | Status of meeting |
| UpdateDate | date | Private | Update date of meeting |
| CreateDate | date | Private | Create date of meeting |
| RoomId | int | Private | RoomId for meeting |
| RoomName | string | Private | Room name for meeting |
| CreateBy | int | Private | Create by |
| UpdateBy | int | Private | Update by |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.12 Notification***  
*Attribute*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| NotificationId | int | Private | Unique identifier of notification |
| NotificationName | string | Private | Notification name |
| Content | string | Private | Content of notification |
| Status | int | Private | Status of notification |
| UpdateDate | date | Private | Update date of notification |
| CreateDate | date | Private | Create date of notification |
| CreateBy | int | Private | Create by |
| CreateName | string | Private | Create name |
| UpdateBy | int | Private | Update by |

*Method*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

***4.2.13 Room***  
*Attribute*

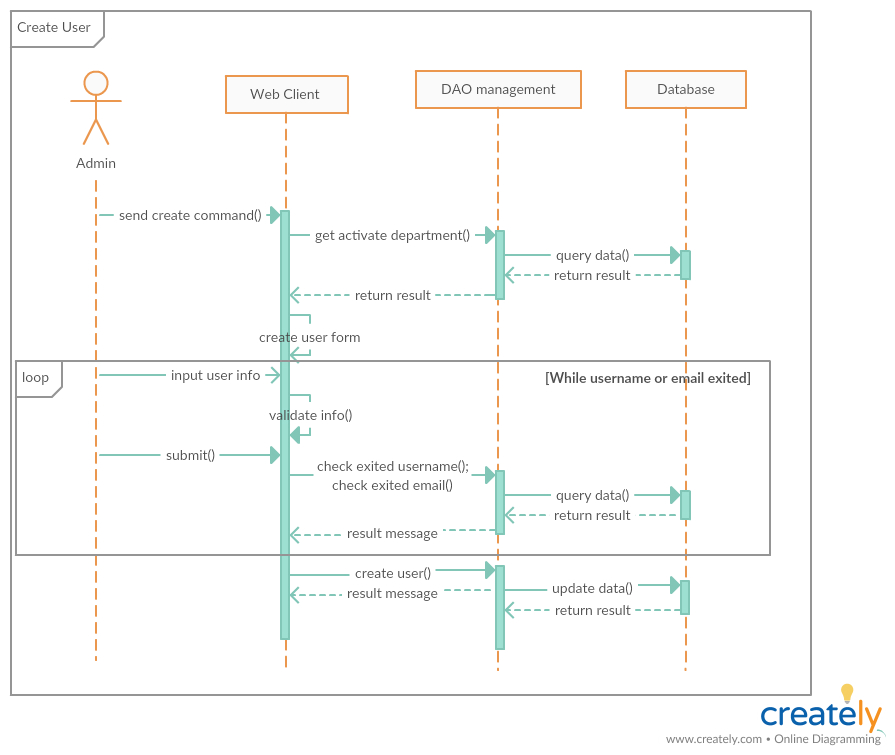
|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| RoomId | int | Private | Unique identifier of room |
| RoomName | string | Private | Room name |
| RoomSize | int | Private | Number of seats in room |
| Status | int | Private | Status of room |

*Method*

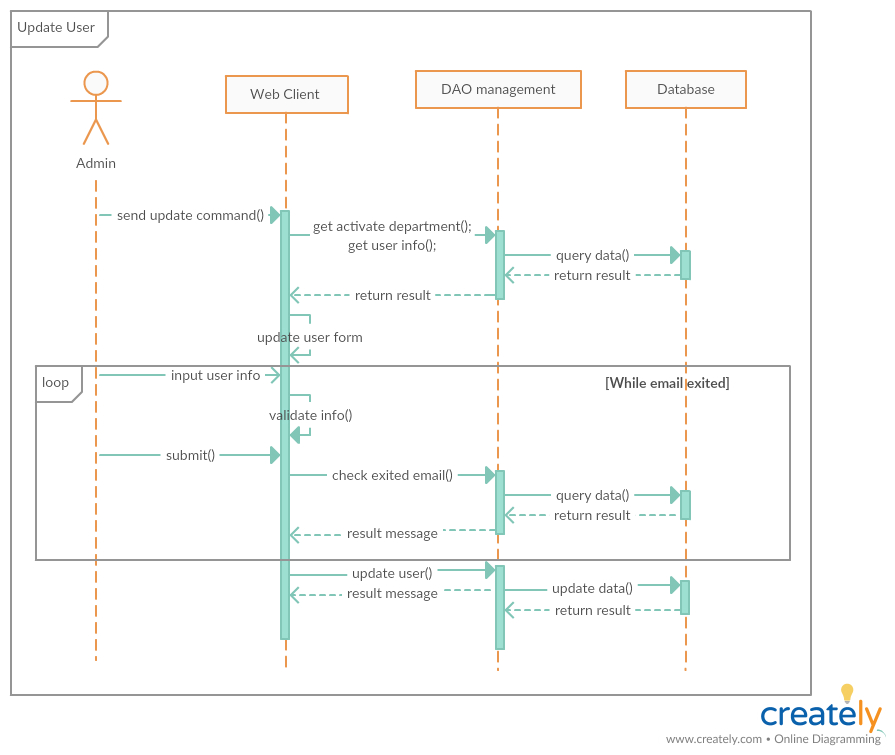
|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

**4.3 Interaction Diagram**

***4.3.1 Create user***

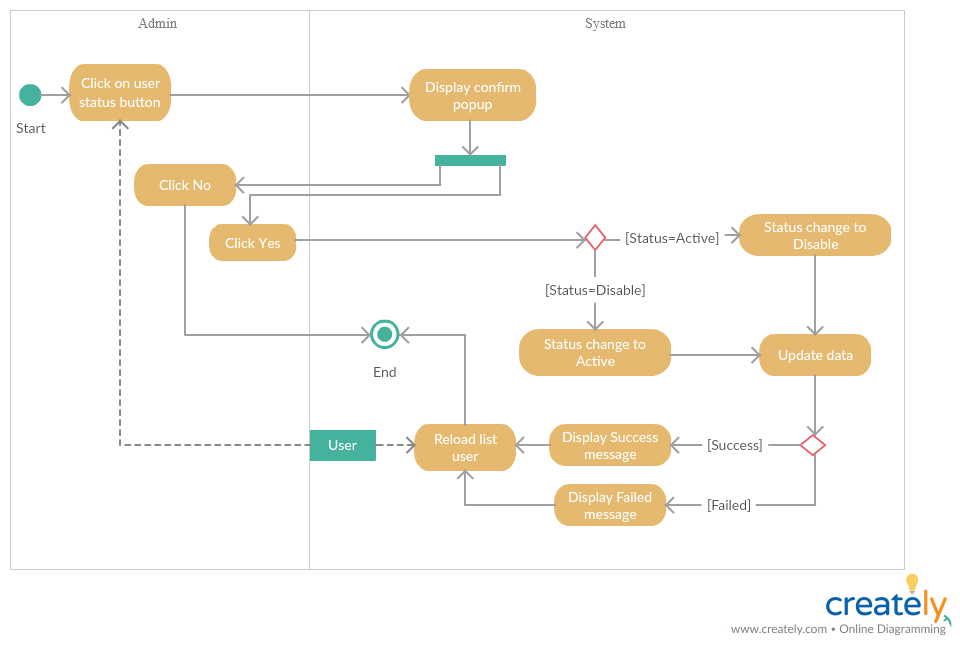
*****Summary: this diagram show process of admin creates new User.*

***4.3.2 Update user***

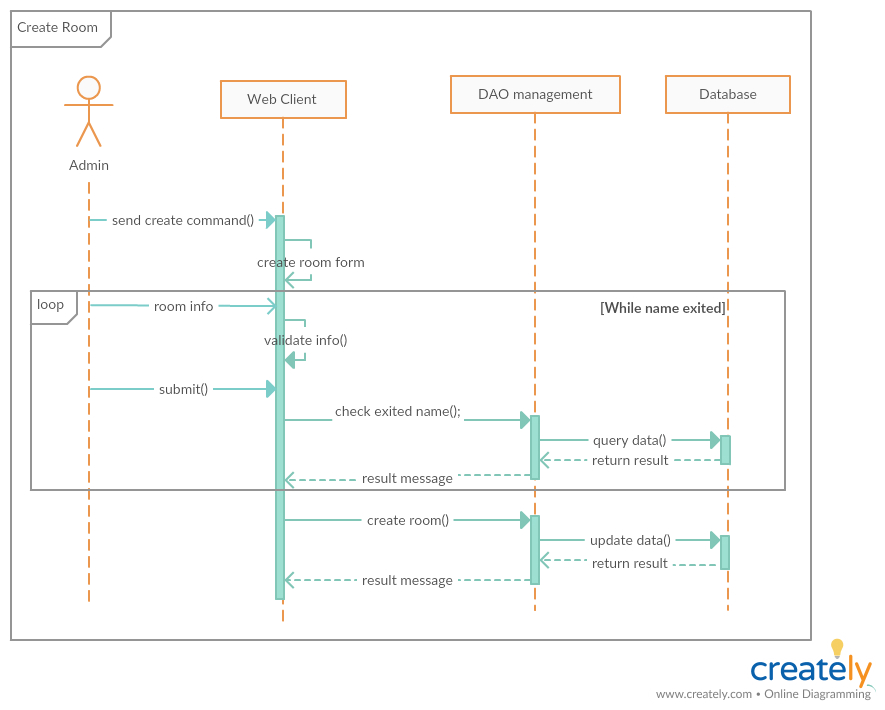
*****Summary: this diagram show process of admin update User*

***4.3.3 Activate/Deactivate user:***

*Summary: this diagram show process of admin Activate/Deactivate user.*

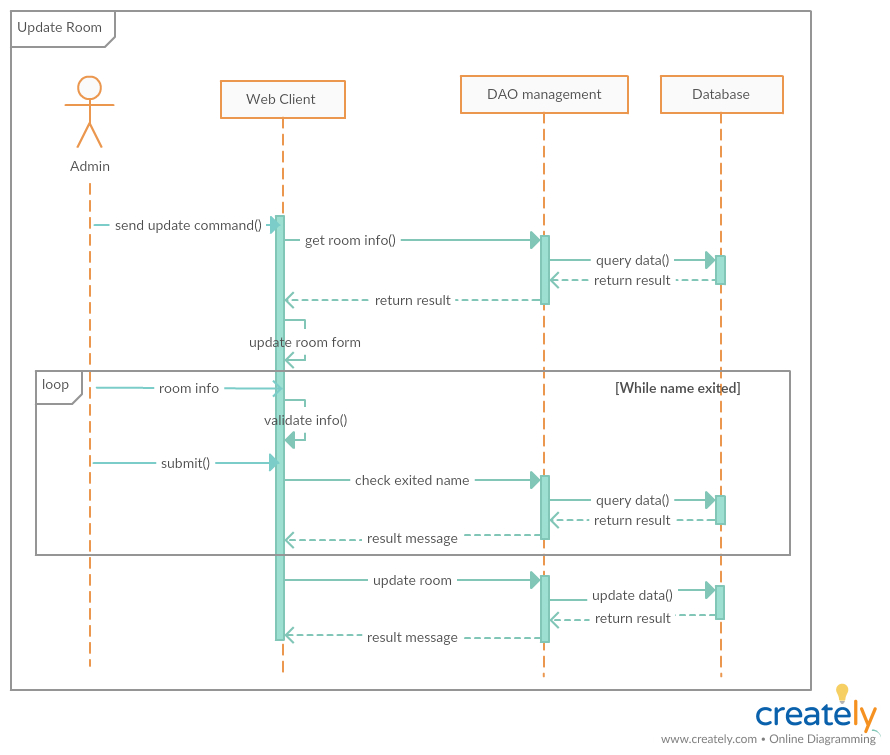
******

***4.3.4 Create room***

*Summary: this diagram show process of admin creates new Room.*

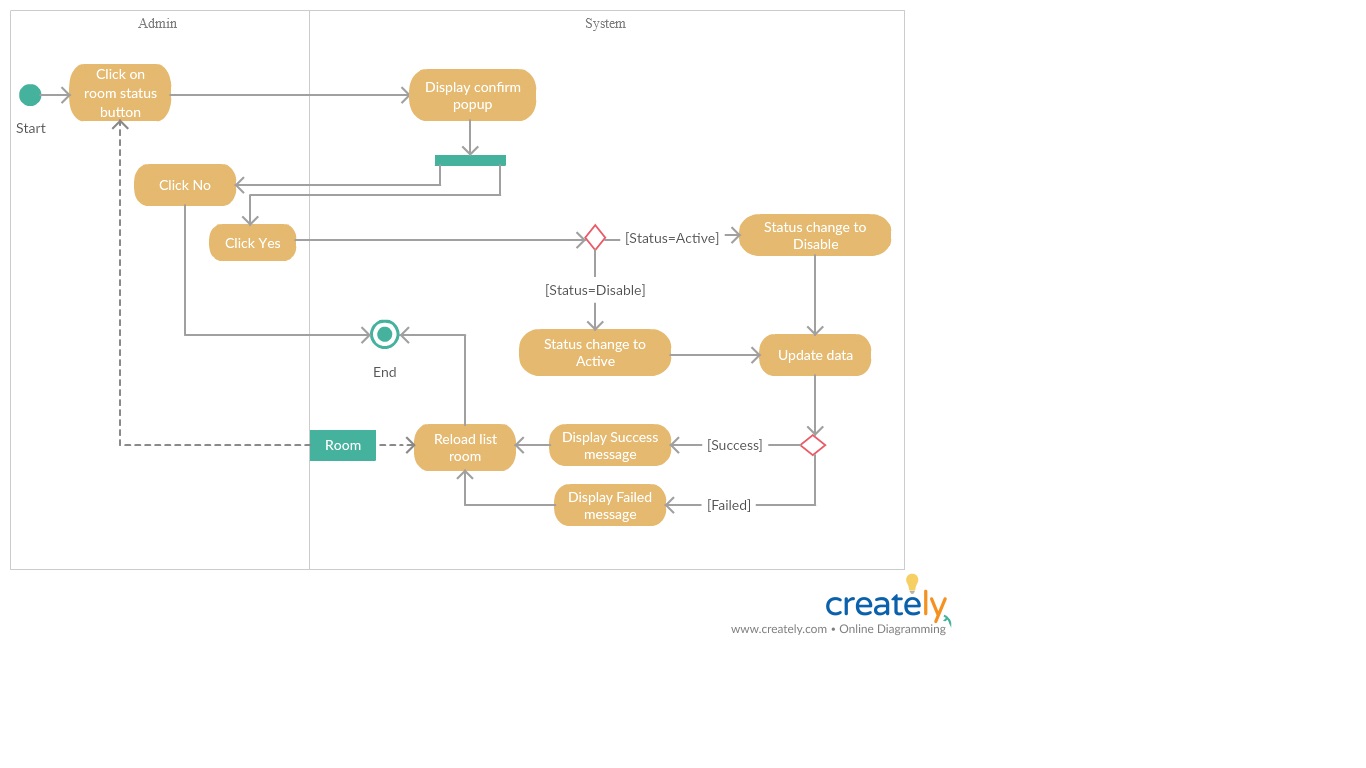
***4.3.5 Update room***

*Summary: this diagram show process of admin update Room.*

******

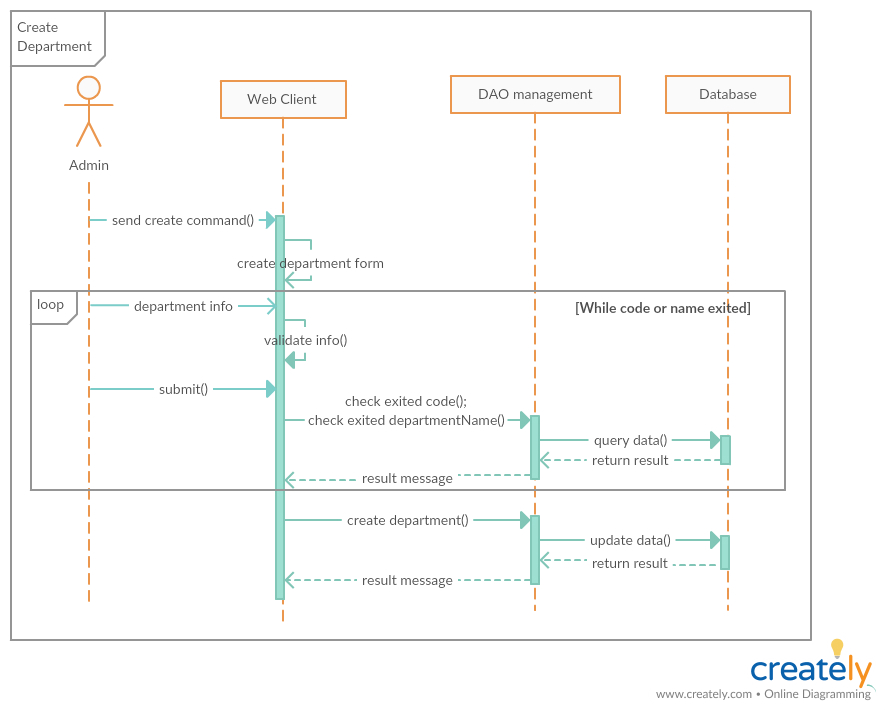
***4.3.6 Activate/Deactivate room:***

*Summary: this diagram show process of admin Activate/Deactivate room*

**

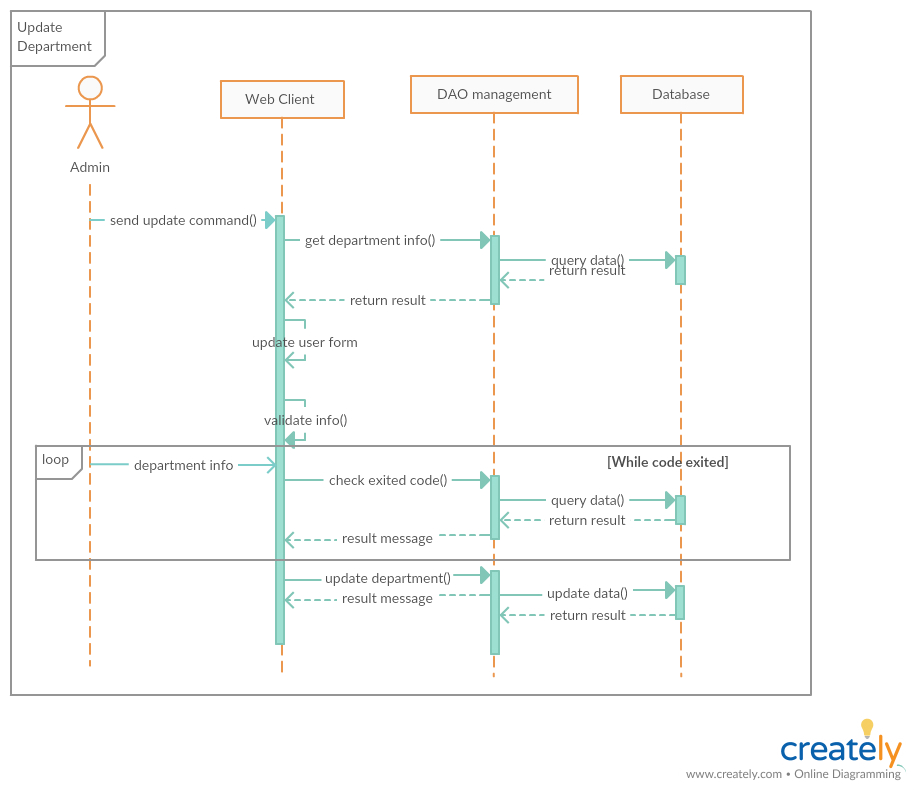
***4.3.7 Create department***

*Summary: this diagram show process of admin creates new Department.*

**

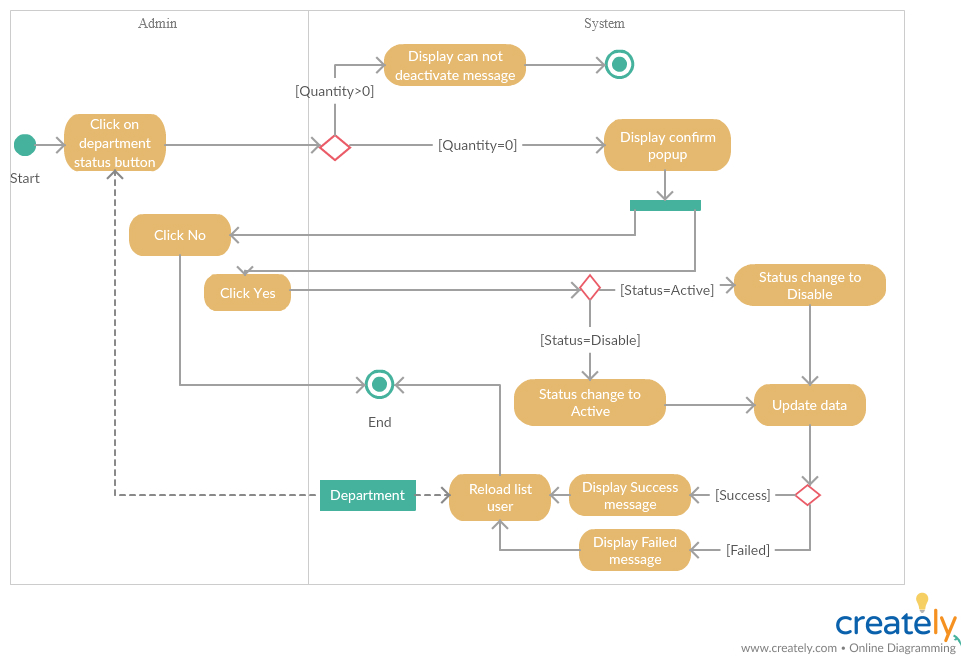
***4.3.8 Update department***

*Summary: this diagram show process of admin update Department.*

**

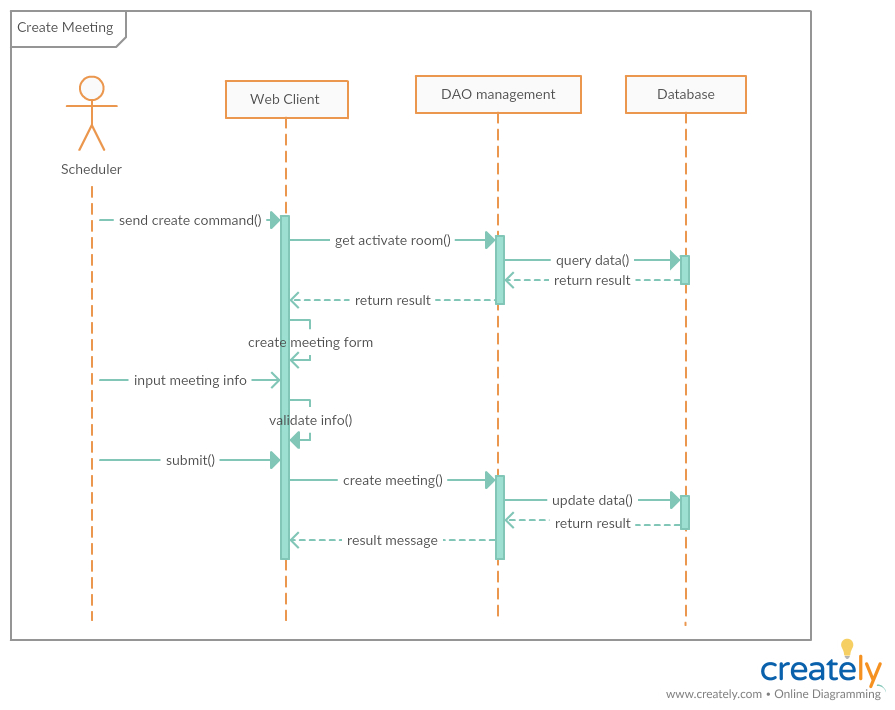
***4.3.9 Activate/Deactivate department***

*Summary: this diagram show process of admin Activate/Deactivate Department.*

****

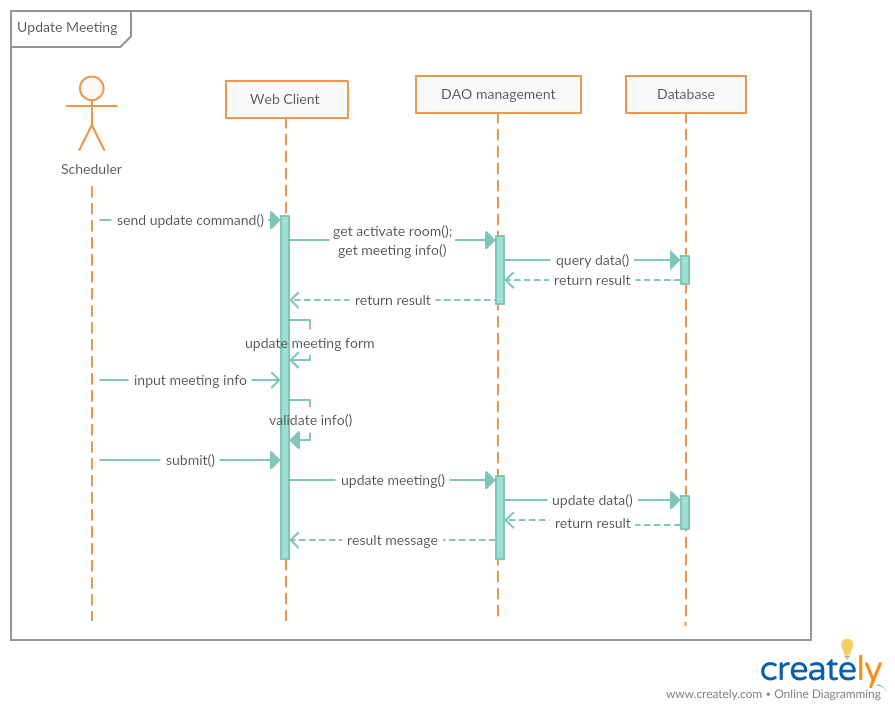
***4.3.10 Create meeting***

*Summary: this diagram show process of scheduler creates new meeting.*

**

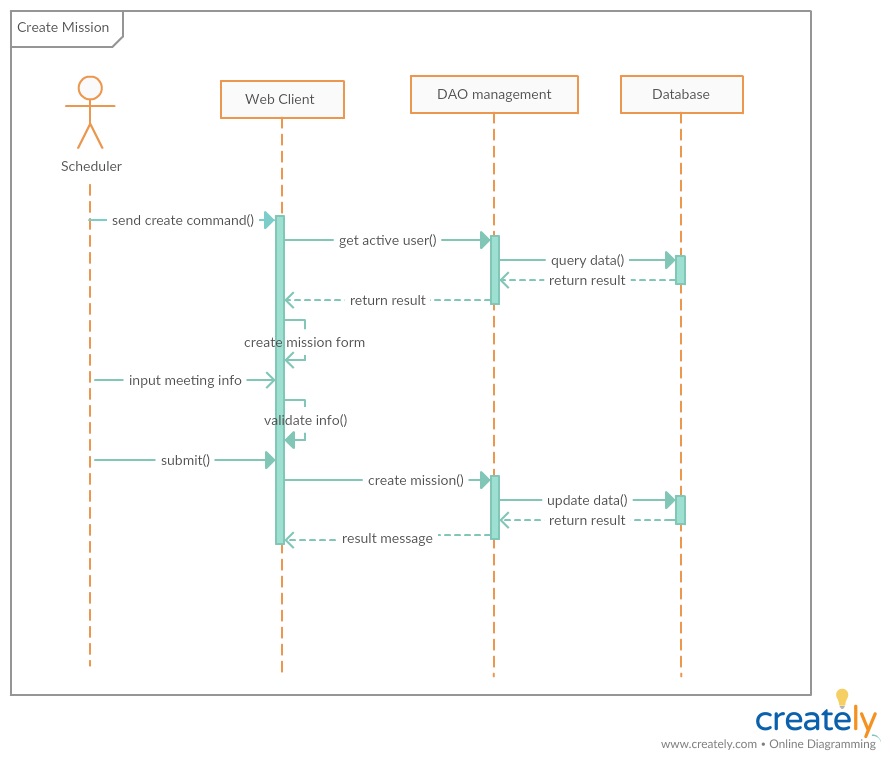
***4.3.11 Update meeting***

*Summary: this diagram show process of scheduler update meeting.*

****

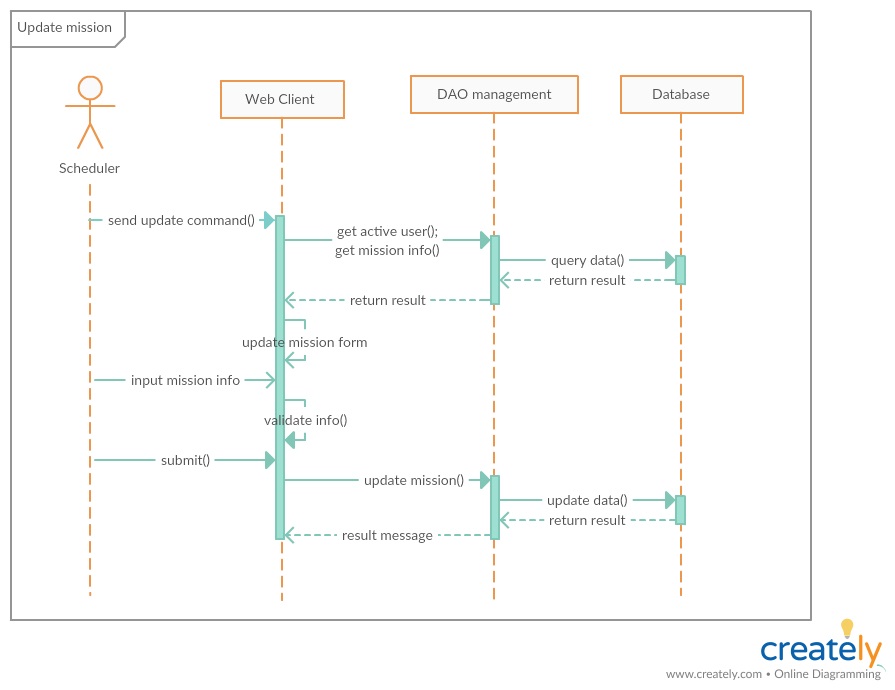
***4.3.12 Create mission***

*Summary: this diagram show process of scheduler creates new mission.*

****

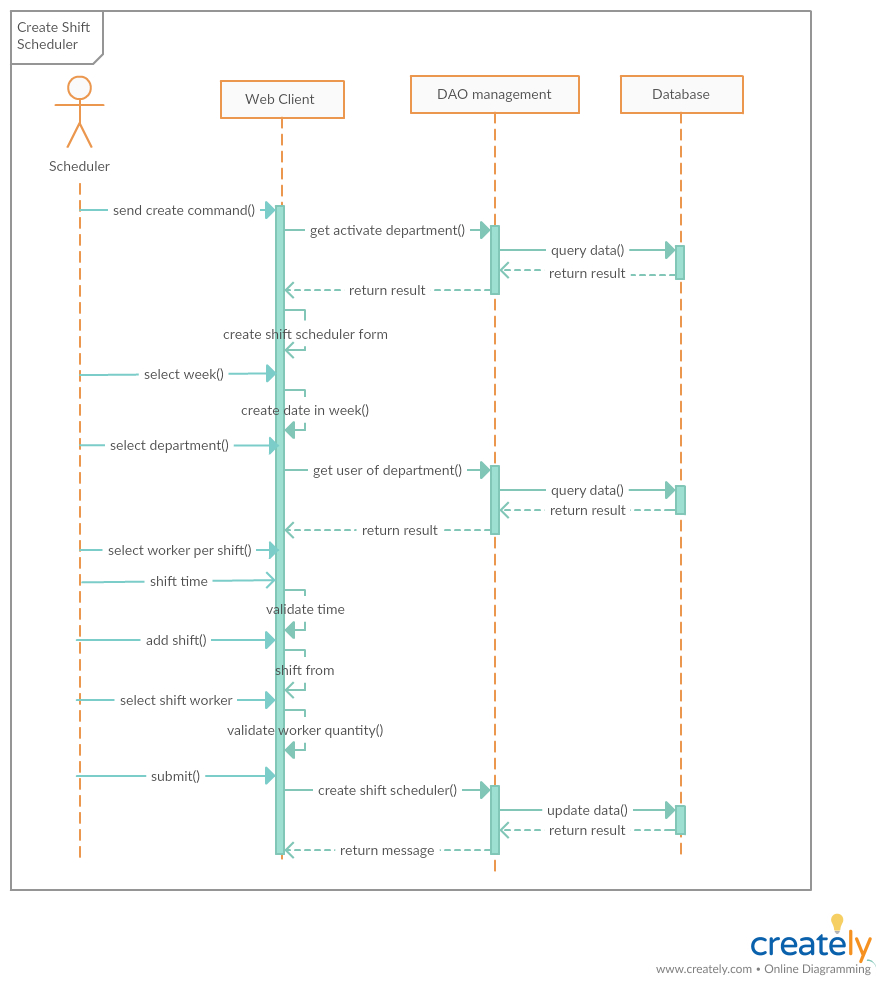
***4.3.13 Update mission***

*Summary: this diagram show process of scheduler update mission.*

****

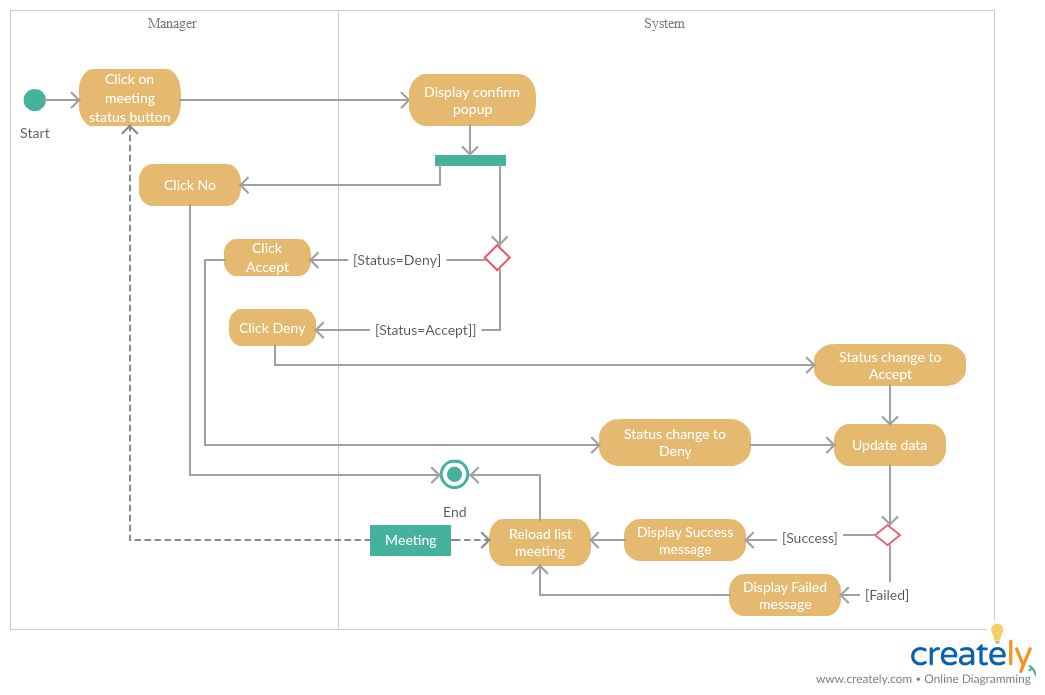
***4.3.14 Create shift schedule***

*Summary: this diagram show process of scheduler create shift schedule.*

****

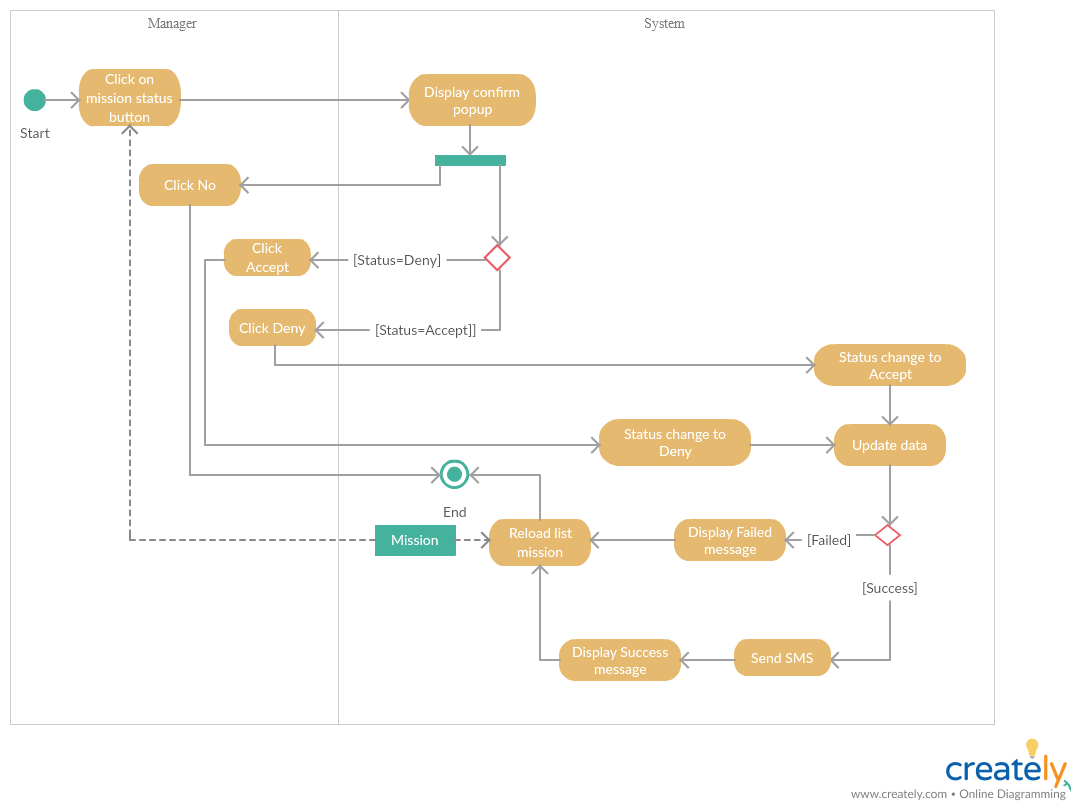
***4.3.15 update checked meeting status***

*Summary: this diagram show process of manager update checked meeting status*

**

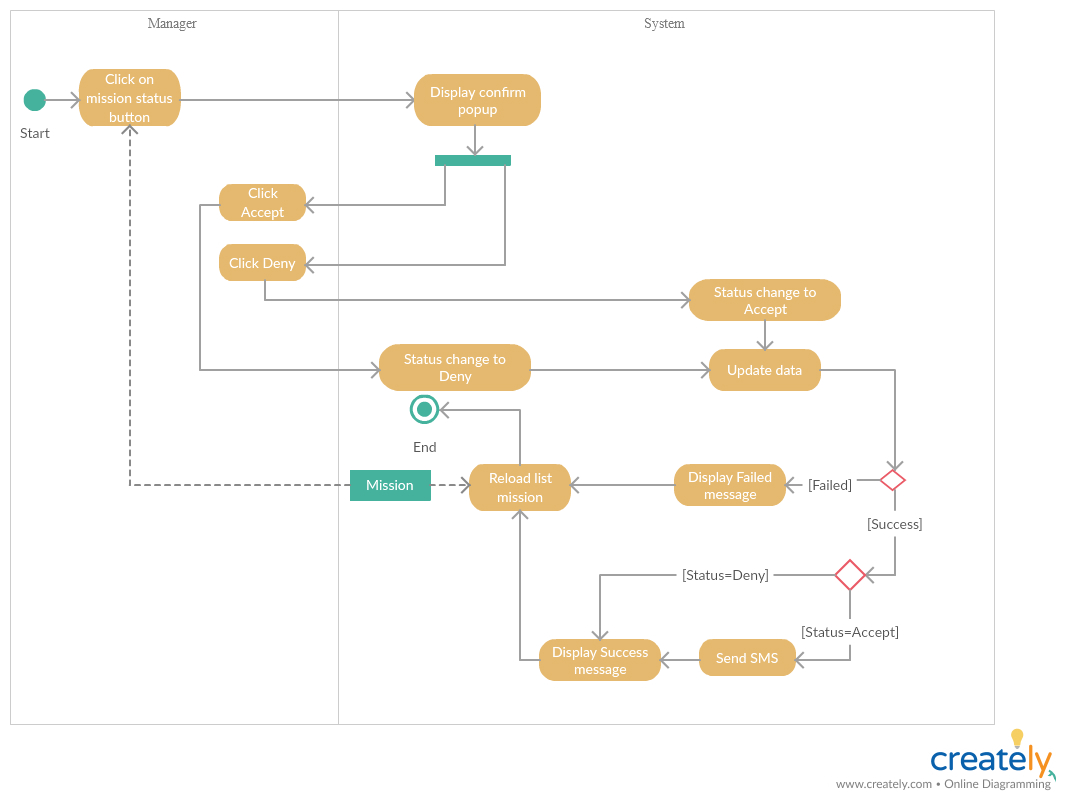
***4.3.16 update checked mission status***

*Summary: this diagram show process of manager update checked mission status*

****

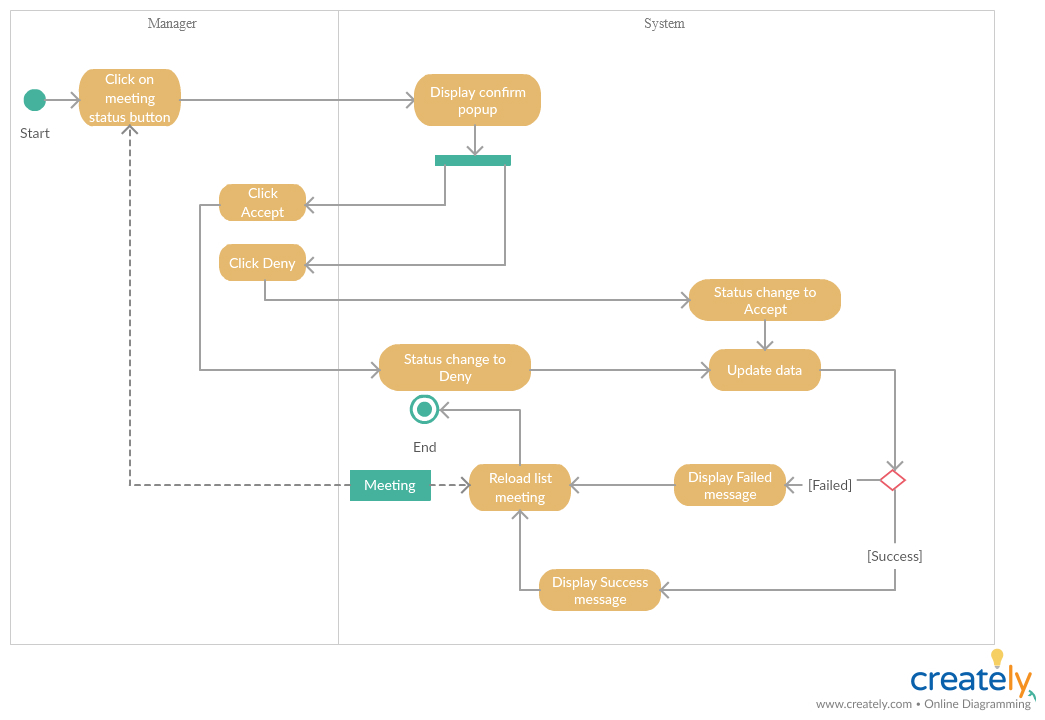
***4.3.17 check waiting mission status***

*Summary: this diagram show process of manager check waiting mission status*

****

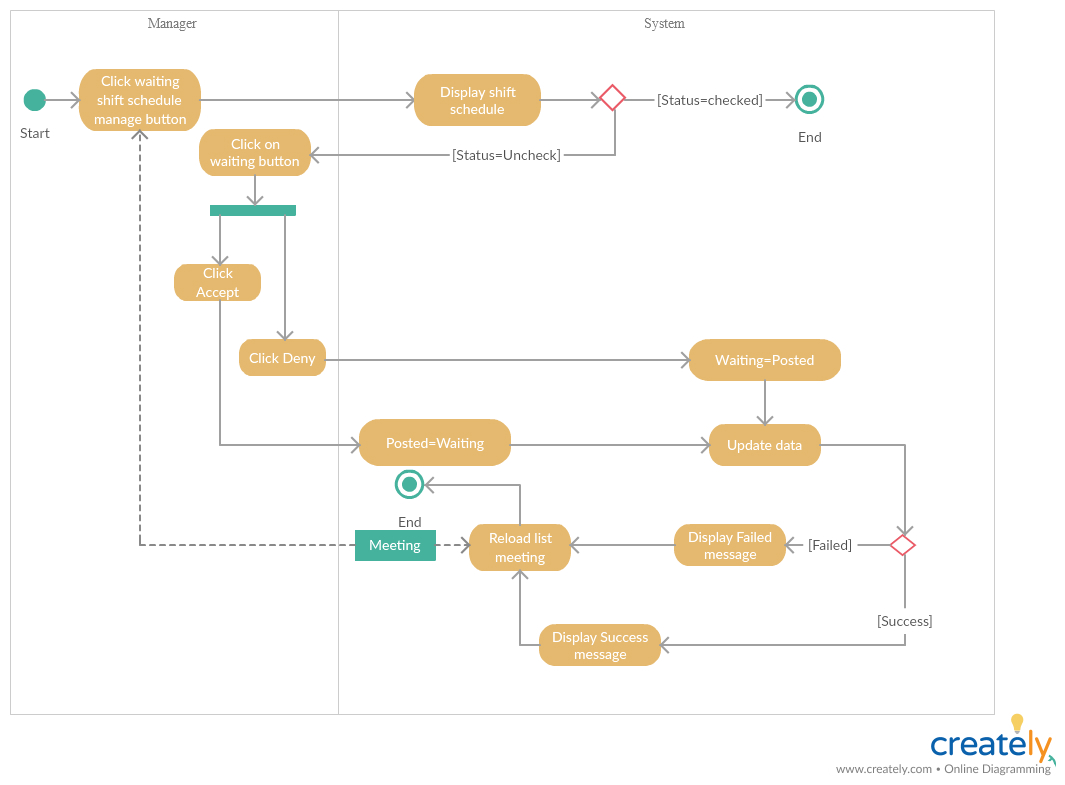
***4.3.18 check waiting meeting status***

*Summary: this diagram show process of manager check waiting meeting status*

****

***4.3.19 check shift schedule***

*Summary: this diagram show process of manager check shift schedule*

**

**5. Interface**  
**5.1 Component interface**

### **5.1.1 Module of User API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| checkLogin(String username, String password) | Login into Web Application | JSON of user | JsonException |
| getUserById (int userId) | Fine user by Id | JSON of user | JsonException |
| getListUser () | Get all user | JSON of list user | JsonException |
| createUser (User user) | Create new user information | String | N/A |
| updateUser (User user) | Update user information | String | N/A |
| deleteUser(int userId) | Delete one user | String | N/A |
| activeUser(int userId) | Active user status | String | N/A |
| getUserByDepartment(int depId) | Get list user filter by department | JSON of list user | JsonException |

Table 12: Module of User API

### **5.1.2 Module of Department API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| getListDepartment() | Get all department | JSON of department | JsonException |
| getDepartmentById(int depId) | Get department by Id | JSON of department | JsonException |
| createDepartment(Department department) | Create new department | String | N/A |
| updateDepartment(Department department) | Update one department | String | N/A |
| deleteDepartment(int depId) | Delete one department | String | N/A |

Table 13: Module of Department API

### **5.1.3 Module of Meeting API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| getListMeeting() | Get all meeting | JSON of meeting | JsonException |
| getMeetingById (int id) | Get meeting by Id | JSON of meeting | JsonException |
| createMeeting(Meeting meeting) | Create new Meeting | String | N/A |
| updateMeeting (Meeting meeting) | Update one Meeting | String | N/A |
| getAllMeetingByUser (int userId) | Get all meeting of one user | JSON of meeting | JsonException |

Table 14: Module of Meeting API

### **5.1.4 Module of Mission API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| getListMission() | Get all mission | JSON of mission | JsonException |
| getMissionById (int id) | Get mission by Id | JSON of mission | JsonException |
| createMission (Mission mission) | Create new mission | String | N/A |
| updateMission (Mission mission) | Update one mission | String | N/A |
| geAllMissionByUser (int userId) | Get all mission of one user | JSON of mission | JsonException |

Table 15: Module of Mission API

### **5.1.5 Module of Notification API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| getListNotification() | Get all notification | JSON of notification | JsonException |
| getActivateNotification (int id) | Get all notification by Id | JSON of notification | JsonException |
| createNotification (Notification notification) | Create new notification | String | N/A |
| updateNotification (Notification notification) | Update one notification | String | N/A |
| getNotificationById (int notificationId) | Get all notification of one user | JSON of notification | JsonException |

Table 16: Module of Notification API

### **5.1.5 Module of Role API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| getRoles() | Get all role | JSON of notification | JsonException |

Table 17: Module of Role API

### **5.1.6 Module of Room API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| getListRoom() | Get all room information | JSON of room list | JsonException |
| getRoomById(int roomId) | Get room by id | JSON of room | JsonException |
| createRoom(Room room) | Create new room resource | String | N/A |
| updateRoom(Room room) | Update room information | String | N/A |

Table 18: Module of Room API

### **5.1.7 Module of Shift Scheduler Resource API**

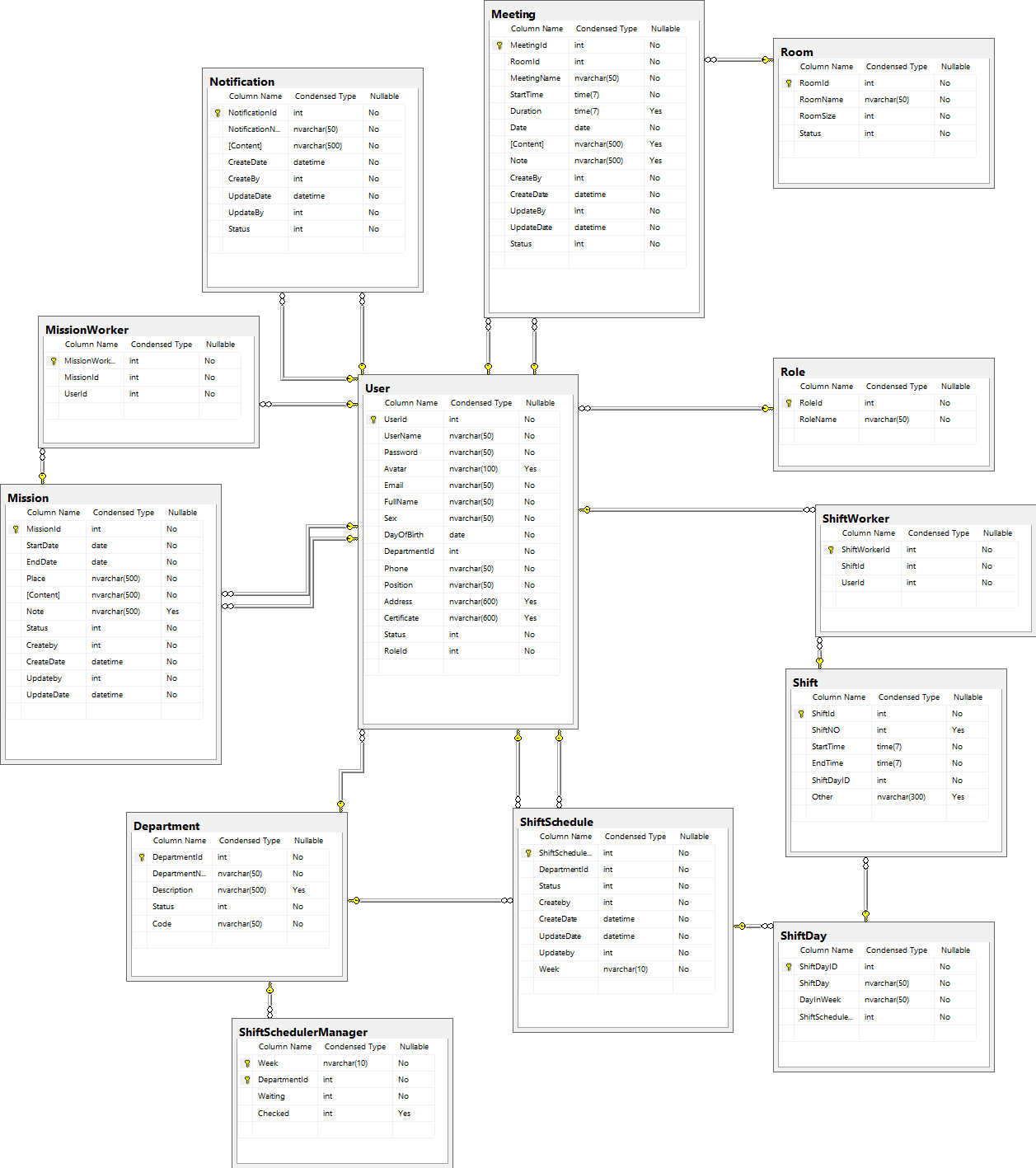
|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| getListShiftSchedule() | Get list shift schedule | JSON of shift schedule | JsonException |
| getWaitingShiftSchedules() | Get shift schedule which stauts is waiting by id | JSON of shift schedule | JsonException |
| createShiftScheduler (ShiftSchedule scheduler) | Create new shift schedule | String | N/A |
| getShiftSchedulesByWeek (String week, int depId) | Get shift schedule is filtered by department and week | JSON of shift schedule | N/A |

Table 19: Module of Shift schedule API

**5.2User Interface Design**

**5.2.1 Web Application - Login**

**6. Database Design**  
**6.1 Entity relationship diagram (ERD)**

****

**6.2 Data Dictionary**

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name Description** | **Entity Name Description** |
| Department | Contain the department information. |
| Meeting | Contain the meeting information. |
| Mission | Contain the mission information. |
| MissionWorker | Contain the mission worker status. |
| Notification | Contain the notification information. |
| Role | Contain the role information. |
| Room | Contain the room information. |
| Shift | Contain the shift information. |
| ShiftDay | Contain the shift day information. |
| ShiftSchedule | Contain the shift schedule information. |
| ShiftScheduleManager | Contain the shift schedule manager information. |
| ShiftWorker | Contain the shift worker information. |
| User | Contain the user information. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| Department | DepartmentId | Primary of department table | int | No |
| DepartmentName | Name of department | nvarchar(50) | No |
|  | Description | Description of department | nvarchar(500) | Yes |
|  | Status | Status of department | int | No |
|  | Code | Code of department | nvarchar(50) | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| Meeting | MeetingId | Primary of meeting table | int | No |
| RoomId | Room id of meeting | int | No |
|  | MeetingName | Name of meeting | nvarchar(50) | No |
|  | StartTime | Start time of meeting | Time(7) | No |
|  | Duration | Duration of meeting | Time(7) | Yes |
|  | Date | Date of meeting | date | No |
|  | Content | Content of meeting | nvarchar(500) | No |
|  | Note | Note of meeting | nvarchar(500) | Yes |
|  | CreateBy | Create by of meeting | int | No |
|  | CreateDate | Create date of meeting | date | No |
|  | UpdateBy | Update by of meeting | int | No |
|  | UpdateDate | Update date of meeting | date | No |
|  | Status | Status of meeting | int | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| Mission | MissionId | Primary of mission table | int | No |
| StartDate | Start date id of mission | date | No |
|  | EndDate | End date of mission | date | No |
|  | Place | Place of mission | nvarchar(500) | No |
|  | Content | Content of mission | nvarchar(500) | No |
|  | Note | Note of mission | nvarchar(500) | Yes |
|  | Status | Status of mission | int | No |
|  | Createby | Create by of mission | int | No |
|  | CreateDate | Create Date of mission | datetime | No |
|  | Updateby | Update by of mission | int | No |
|  | UpdateDate | Update Date of mission | datetime | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| MissionWorker | MissionWorkerId | Primary of mission worker table | int | No |
| MissionId | Primary of mission | int | No |
|  | UserId | Primary of user | int | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| Notification | NotificationId | Primary of notification table | int | No |
| NotificationName | Name of notification | nvarchar(50) | No |
|  | Content | Content of notification | nvarchar(500) | No |
|  | CreateDate | Create Date of notification | datetime | No |
|  | CreateBy | Create By of notification | int | No |
|  | UpdateDate | Update Date of notification | datetime | No |
|  | UpdateBy | Update By of notification | int | No |
|  | Status | Status of notification | int | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| Role | RoleId | Primary of role table | int | No |
| RoleName | Name of role | nvarchar(50) | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| Room | RoomId | Primary of room table | int | No |
| RoomName | Name of room | nvarchar(50) | No |
| RoomSize | Size of room | int | No |
| Status | Status of room | int | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| Shift | ShiftId | Primary of shift table | int | No |
| ShiftNo | No of shift | int | Yes |
| StartTime | Start Time of shift | Time(7) | No |
| EndTime | End Time of shift | Time(7) | No |
|  | ShiftDayId | Shift Day Id | int | No |
|  | Other | Note more about shift | nvarchar(300) | Yes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| ShiftDay | ShiftDayId | Primary of Shift day table | int | No |
| ShiftDay | Day of shift | nvarchar(50) | Yes |
| DayInWeek | Day In Week | nvarchar(50) | No |
| ShiftScheduleId | Shift Schedule Id | int | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| ShiftSchedule | ShiftScheduleId | Primary of Shift Schedule table | int | No |
| DepartmentId | Department Id | int | No |
| Status | Status of Shift Schedule | int | No |
| Createby | Create by of Shift Schedule | int | No |
| CreateDate | Create Date of Shift Schedule | datetime | No |
| UpdateDate | Update Date of Shift Schedule | datetime | No |
| Updateby | Update by of Shift Schedule | int | No |
| Week | Week of Shift Schedule | nvarchar(10) | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| ShiftScheduleManager | Week | Week of Shift Schedule | int | No |
| DepartmentId | Department Id | int | No |
| Waiting | Status of Shift Schedule | int | No |
| Checked | Status by of Shift Schedule | int | Yes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| ShiftWorker | ShiftWorkerId | Primary of Shift Worker table | int | No |
| ShiftId | Shift Id | int | No |
| UserId | User Id | int | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Domain** | **Null** |
| User | UserId | Primary of user table | int | No |
| UserName | User Name of user | nvarchar(50) | No |
| Password | Password of user | nvarchar(50) | No |
|  | Avatar | Avatar of user | nvarchar(100) | Yes |
|  | Email | Email of user | nvarchar(50) |  |
|  | FullName | Full Name of user | nvarchar(50) |  |
|  | Sex | Sex of user | nvarchar(50) |  |
|  | DayOfBirth | Day Of Birth of user | date |  |
|  | DepartmentId | Department Id of user | int |  |
|  | Phone | Phone of user | nvarchar(50) |  |
|  | Position | Position of user | nvarchar(50) |  |
|  | Address | Address of user | nvarchar(600) | Yes |
|  | Certificate | Certificate of user | nvarchar(600) | Yes |
|  | Status | Status of user | int |  |
|  | RoleId | Role Id of user | int |  |

**7. Algorithms**

***7.1Document Breakdown***  
***7.1.1 Definition***

***7.1.2 Define Problem***

***7.1.3 Solution***

***7.1.4 Complexity***

***7.1.5 Flowchart***

***7.2 String Comparison***  
***7.2.1 Define Problem***

***7.2.2 Requirement***

***7.2.3 Solution***

***7.2.4 Example***

**E. System Implementation & Test**  
**1. Introduction**

**1.1 Overview**

This section provides in detail all necessary information about implementation information and testing procedure of Hospital Portal includes test plans, test cases, test result, test environments, pass, fail, criteria and risks estimations as well as a checklist to cover all possible cases.

**1.2 Test Approach**

* Goal: Test all features in the whole system based on the core flow.
* Method: black-box testing
* Technique: check list

The testing for this project will consists of unit test. Testing the program as a complete system to ensure that the software requirements have been met.

* Testing would be performed by all member of team and approved by team leader.
* System testing is focused on assessing the system’s reliability. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

**2. Database Relationship Diagram**  
**2.1 Physical Diagram**

**2.2 Data Dictionary**

**3. Performance Measures**

***3.1 Clustering Performance***

**4. Test Plan  
4.1 Features to be tested**

- Admin: Login, create user, update user, active/deactivate user, create department, update department, active/deactivate department, create room, update room, active/deactivate room

- Manager: create mission schedule, update mission schedule, active/deactivate mission schedule, create meeting schedule, update meeting schedule, active/deactivate meeting schedule, create notification, update notification, active/deactivate notification

- User: Search meeting schedule

**4.2 Features not to be tested**

- Manager: Send SMS to employee

- User: View mission schedule, view meeting schedule, view shift schedule, view notification

**5. System Testing Test Case**

***5.1 Admin Test Case***

***5.1.1 Login event***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-1 | Login with empty input | 1. Go to login page  2. Input:  Case 1:   * Username: empty * Password: empty   Case 2:   * Username: ‘admin’ * Password: empty   Case 3:   * Username: empty * Password: ‘\*\*\*\*\*’   3. Click button “Login” | System show error message “Invalid username or password” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-2 | Login with account not exist | 1. Go to login page  2. Input:   * Username: ‘test’ * Password: ‘\*\*\*\*\*\*\*’   3. Click button “Login” | System show error message “Invalid username or password” |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.1.2 Create/ Update user event***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-3 | Create/ Update user with empty username, email, full name, day of birth, phone | 1. Login at admin role successfully.  2. Go to manage user page.  3. Click add new user.  4. Input:   * Username: Empty * Email: Empty * Full name: Empty * Sex: Male * Day of birth: Empty * Department: “BOARD OF DIRECTOR” * Phone: Empty * Position: “Doctor” * Address: “1b Nha Chung, Da Lat, Lam Dong” * Certificate: “FPT University” * Role: “Manager” | System show error message “Username is required”  “Email is required”  “Full name is required”  “Day of birth is required”  “Phone is required” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-4 | Create/ Update user with invalid email, invalid day of birth, invalid phone | 1. Login at admin role successfully.  2. Go to manage user page.  3. Click add new user.  4. Input:   * Username: “LinhTP” * Email: “AeonCity” * Full name: “Tran Nguyen Quang” * Sex: Male * Day of birth: “12/05/2018” * Department: “BOARD OF DIRECTOR” * Phone: “9723589884” * Position: “Doctor” * Address: “1b Nha Chung, Da Lat, Lam Dong” * Certificate: “FPT University” * Role: “Manager” | System show error message “Email is invalid”.  “Day of birth is invalid”  “Phone is invalid” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-5 | Create/ Update user doesn’t exists in system | 1. Login at admin role successfully.  2. Go to manage user page.  3. Click add new user.  4. Input:   * Username: “LinhTP” * Email: “AeonCity@gmail.com” * Full name: “Tran Nguyen Quang” * Sex: Male * Day of birth: “02-02-1996” * Department: “BOARD OF DIRECTOR” * Phone: “01245497985” * Position: “Doctor” * Address: “1b Nha Chung, Da Lat, Lam Dong” * Certificate: “FPT University” * Role: “Manager” | System show message “Success”. |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.1.3 Active/ Deactivate user event***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-6 | Active/ Deactivate user | 1. Login at admin role successfully.  2. Go to manage user page.  3. Click Active/ deactivate button  4. Click yes button | System show message “Success”. |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.1.4 Create/ Update department event***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-7 | Create/ Update department with empty department name, empty department code | 1. Login at admin role successfully.  2. Go to manage department page.  3. Click Add button  4. Input:   * Department name: Empty * Department code: Empty * Description: “Test” | System show error message “Name is required”.  “Code is required” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-8 | Create/ Update department doesn’t exists in system | 1. Login at admin role successfully.  2. Go to manage department page.  3. Click Add button  4. Input:   * Department name: “BOARD OF DIRECTOR” * Department code: “D1” * Description: “Test” | System show message “Success” |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.1.5 Active/ Deactivate department event***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-9 | Active/ Deactivate department | 1. Login at admin role successfully.  2. Go to manage department page.  3. Click Active/ deactivate button.  4. Click yes button. | System show message “Success”. |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.1.6 Create/ Update room event***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-10 | Create/ Update room with empty | 1. Login at admin role successfully.  2. Go to manage room page.  3. Click Add button  4. Input:   * Room name: Empty * Room code: Empty | System show error message “Room name is required” “Room size is required” |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.1.7 Active/Deactivate room event***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-11 | Active/ Deactivate room | 1. Login at admin role successfully.  2. Go to manage room page.  3. Click Active/ deactivate button.  4. Click yes button. | System show message “Success”. |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.2 Manager Test Case***

***5.2.1 Create/ Update mission schedule***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-12 | Create/ Update mission schedule with empty member , empty start date, end date, place | 1. Login at manager role successfully.  2. Go to manage mission page.  3. Click add button.  4. Input:   * Member: Empty * Start date: Empty * End date: Empty * Place: Empty | System show error message “Member is required”.  “Start date is required”  “End date is required”  “Place is required” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-13 | Create/ Update mission schedule with invalid end date | 1. Login at manager role successfully.  2. Go to manage mission page.  3. Click add button.  4. Input:   * Member: LinhTP, LongHH * Start date: “6/4/2018” * End date: “5/4/2018” * Place: “Ha Noi” | System show error message  “End date must after start date” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-14 | Create/ Update valid mission schedule valid | 1. Login at manager role successfully.  2. Go to manage mission page.  3. Click add button.  4. Input:   * Member: LinhTP, LongHH * Start date: “6/4/2018” * End date: “10/4/2018” * Place: “Ha Noi” | System show message “Success”. |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.2.2 Active/Deactivate mission schedule***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-15 | Active/deactivate mission schedule | 1. Login at manager role successfully.  2. Go to manage mission page.  3. Click Active/Deactivate button  4. Click yes button. | System show message “Success”. |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.2.3*** ***Create/ Update meeting schedule***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-16 | Create/ Update meeting schedule with empty meeting name, start time, end time, date, content | 1. Login at manager role successfully.  2. Go to manage meeting page.  3. Click Add button  4. Input   * Meeting name: Empty * Room: “201” * Start time: Empty * End time: Empty * Date: Empty * Content: Empty | System show error message “Name is required”.  “Start time is required”  “End time is required”  “Date is required”  “Content is required” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-17 | Create/ Update valid meeting schedule | 1. Login at manager role successfully.  2. Go to manage meeting page.  3. Click Add button  4. Input   * Meeting name: “Training skill” * Room: “201” * Start time: “10:00 AM” * End time: “12:00 AM” * Date: “6/4/2018” * Content: “Carry pen and notebook” | System show message “Success” |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.2.4 Active/Deactivate meeting schedule***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-18 | Active/ Deactivate meeting schedule | 1. Login at manager role successfully.  2. Go to manage meeting page.  3. Click Active/ Deactivate button  4. Click yes button | System show message “Success” |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.2.5 Create/Update notification***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-19 | Create/ Update notification with empty name, content | 1. Login at manager role successfully.  2. Go to manage notification page.  3. Click Add button  4. Input:   * Notification name: Empty * Content: Empty | System show error message “Name is required”  “Content is required” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-20 | Create/ Update valid notification | 1. Login at manager role successfully.  2. Go to manage notification page.  3. Click Add button  4. Input:   * Notification name: “Unexpected change” * Content: “There is a fire at G floor. Required everyone evacuation” | System show message “Success” |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.2.6 Active/Deactivate notification***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-21 | Active/ Deactivate notification | 1. Login at manager role successfully.  2. Go to manage notification page.  3. Click Active/Deactivate button  4. Click yes button | System show message “Success” |  | Pass | From 1/3/2017 To  5/04/2017 |

***5.3 User Test Case***

***5.3.1 Search meeting schedule***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Inter-test case dependence** | **Result** | **Test Date** |
| TH-22 | Search meeting schedule with invalid to date | 1. Login at user role successfully.  2. Go to meeting schedule page.  3. Click quick search tab  4. Input   * From: “ 6/4/2018” * To: “5/4/2018”   5. Click search button | System show error message “To date must be after from date” |  | Pass | From 1/3/2017 To  5/04/2017 |
| TH-23 | Search valid meeting schedule | 1. Login at user role successfully.  2. Go to meeting schedule page.  3. Click quick search tab  4. Input   * From: “ 6/4/2018” * To: “10/4/2018”   5. Click search button | System show meeting list in range |  | Pass | From 1/3/2017 To  5/04/2017 |