****MINISTRY OF EDUCATION AND**

**TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Hospital Portal**

|  |  |
| --- | --- |
| **Group 4** | |
| **Group members** | Hồ Hoàng Long - SE61761  Trần Phương Linh - SE62029  Phạm Xuân Tùng - SE61590 |
| **Supervisor** | Phạm Công Thành |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** |  |

-Ho Chi Minh City, 8/1/2018-

**Table of Contents**

**List of Tables**

**List of Figures**

**Definitions, Acronyms, and Abbreviations**

**A.Report No. 1 Introduction**

**1. Project Information**

Project name: **Hospital Portal**

Project Code:

Product Type: **web app**

Start Date: **8/1/2018**

End Date: **27/4/2018**

**2. Introduction**



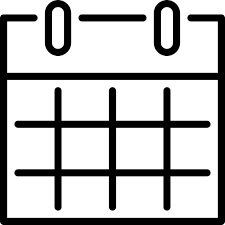
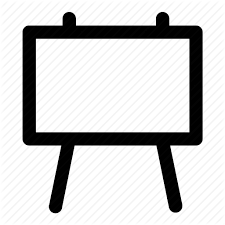
We built a system that helps hospitals optimize the management process in the present. During the analysis, we believe that booking and storing meeting schedules, mission schedules, shift schedules using web and mobile applications is more effective than traditional methods in present hospitals, on the aspects of accessible, accuracy, instantaneous.

In addition, the system also supports many basic features such as post news, announcements, research papers from the hospital to the community and schedule appointments for patients.

**3. Current Situation**

Current management process:

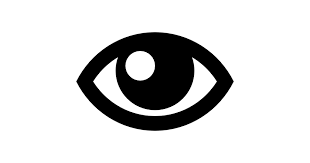
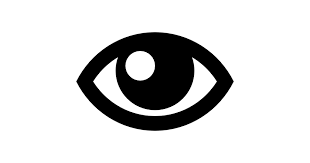
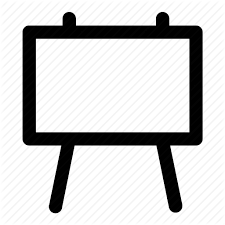
+ Create shift, working schedule: The person in charge (Department chief) create schedule on a form and sends it to the planning office to approval and storage. Then, the text is displayed on the notice board.



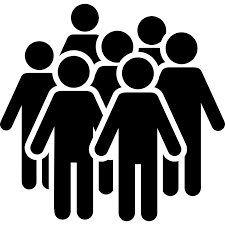
+ View shift, working schedule: When the doctor wants to know his schedule, each doctor must have a task scheduler, view the scheduler on the notice board, or ask the person in charge.



?!?



+ In case of sudden change or meeting incurred: There must be a department that informs each employee.

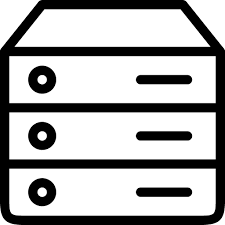


**4. Problem Definition**

Below are disadvantages of current situation:

+ Confusion when scheduling shift: doctor’s scheduler today are mainly handmade, preserve by document, so that duplicate and loss of information is unavoidable. It is difficult to accurately scheduling and only few people who have experience about scheduling can do it well.

+ Hard to search, arrange and preserve: Finding information in the past is difficult because it requires a large repository of data, take time and effort, low accuracy. Even if the data is stored in a computer, it must be through some departments.



+ In the event of an unexpected change or meeting schedule: the immediate notification is waste human resources and possibility of deficiency is high.

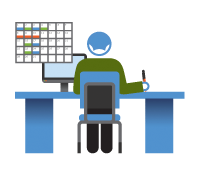
**5. Proposed Solution**

Our proposed solution is to build a management system named “Hospital portal” to resolve the current situations and compatible with current hospital, we also design the system to be scalable so we can deploy this system to any hospital in future plan. Hospital portal includes a web application and mobile interface with following function.

1. **Feature functions:**

**+** Manage data resource: The system supports the management of information such as user, department, room…

+ Manage shift schedule: “Scheduler” (the person in charge on the calendar) can create shift schedules for each department weekly, then the schedule will be sent to “Manager” to check and approved. User will access the system to view the calendar:



- Scheduling: Manager can adjust the time, select employee for the shift (employee list is displayed by department).

+ Manage meeting/mission schedule: For meeting, scheduler input time, content and conference room selection. For the mission, the scheduler will input time, location, content and employee selection. Each pharmacy employee will receive a message when the mission is approved or changed. The process is the same as the shift scheduler.

+ Post article: Poster upload articles to the system. Articles approved and classified by manager will be posted on the homepage. Guest can view public posts on the web.

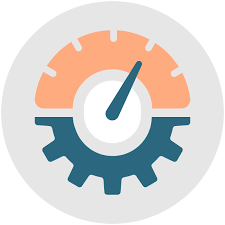
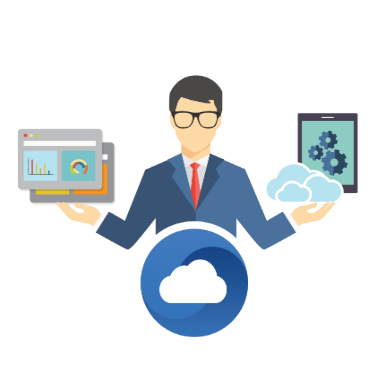


+Book medical examination: Each clinic will have a certain number of shifts, each of which will have a set number of appointments. Guest after selecting a clinic, entering personal information and taking a certain fee to set the appointment. When guests are successfully set, they will be notified of the specific time, sequence number and a confirmation code.

1. **Advantages and disadvantages:**

**Advantages**

* Incorporation of hospital management procedures and portals. The application can be accessed via the internet
* Accuracy and instantaneous.
* Synchronization and automation, reducing administrative procedures and human resources.



**Disadvantages**

* The hospital costs a certain amount of money to maintain the system (server, host).

**6. Functional Requirements**

Function requirements of the system are listed as below:

+ Guest: View article, hospital information; set examination schedule and cancel calendar set.

+User: Login; view article, hospital information, schedule.

+Scheduler: Login; view article, hospital information, schedule; create, update scheduler (shift, meeting, mission).

+Poster: Login, view article, hospital information, schedule, post article.

+Manager: Login; view article, hospital information, schedule; check scheduler, article;

+Admin: Login; view article, hospital information, schedule; Manage User info, department, service, category, room resource; view activity history.

**7. Role and Responsibility**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***No*** | ***Full Name*** | ***Role*** | ***Position*** | ***Contact*** |
| 1 | Phạm Công Thành | *Project Manager* | *Supervisor* | thanhpc3@fe.edu.vn |
| 3 | Trần Phương Linh | *Developer* | *Member* | linhtpse62029@fpt.edu.vn |
| 4 | Hồ Hoàng Long | *Leader* | *Member* | longhhse61761@fpt.edu.vn |
| 5 | Phạm Xuân Tùng | *Developer* | *Member* | tungpxse61590@fpt.edu.vn |

*Table 1: Roles and Responsibilities*

**B. Report No.2 Software Project Management Plan**  
**1. Problem Definition**  
**1.1 Name of this Capstone Project**

*Official name: Hospital portal*  
*Vietnamese name: Hệ thống cổng thông tin bênh viện*

**1.2 Problem Abstract**

This project is the solution for the problem of setting work schedule in hospital, staff management. The scope of this problem seems to be old and boring, but we realize that we still exploit the missing points of other systems and then development of the hospital portal has the strengths to support the management of hospitals more effectively.

Hospital portal is a website system that let user access from everywhere, the system also support on computer and mobile phone.

Nowadays, almost people have cell phones. At that point, the system sends a message to the user if there is any announce from the hospital.

We are also limited in time and human resources to develop a complete system but we build hospital portal by technology that makes the system easier to upgrade.

**1.3 Project Overview**  
**1.3.1 Current Situation**

*Below are the problems encountered in this project:*

* ***Skill:*** Lack of knowledge of common terminology used in hospitals, no one in team have been in contact directly with the hospital workflow.
* ***Absence of team members:*** Lack of human resources and time to build the project. The system need more than 13 weeks to complete all function that clarification and full details.
* ***Security***: There are some possible problem could happen with system such as: attacked during data transmission caused data loss, data corruption.

#### 1.3.2 The Proposed System

According to the immediate demand for hospital information. We build the website system that support access data even by PC, laptop, cell phone or tablet. We assign task for each member in team in vertical to make sure if any member cannot continue to work in our team there will be not profound influence to the project processes.

**Website for admin:**

This website is used by employees of the hospital include: “Admin, Manager, Scheduler and User” to interactive with functions of this system.

* Manage user.
* Manage department.
* Manage room resources.
* Manage meeting schedule.
* Manage mission schedule.
* Manage shift schedule.
* Manage notification.

**1.3.3 Boundaries of the System**

**The system can:**

* Management of personal information of staff in the hospital.
* Management of the hospital department
* Management of the hospital meeting room resource
* Create,view and check mission schedule
* Create,view and check meeting schedule
* Create,view and check shift schedule
* Create,view and check notification
* Send SMS to employee if they join in any mission.

**1.3.4 Future plans**

Now, the system is user for one hospital only, so that it is not diversity about coverage and fluent. Our aspiration is that the system can be used in many hospitals so that the architecture of the system is very easy to maintain and expand.

* Customer don’t have to confuse when they want to find a hospital that provide what kind of service they need and could evaluate quantity thought feedback field.
* Allow customer to book medical appointment.
* Suggest to scheduler the best suitable schedule to reduce time.

**1.3.5 Development Environment**

**1.3.5.1 Hardware requirements**

**For server**

|  |  |  |
| --- | --- | --- |
| **Server** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operating System** | Window Server 2008 | Window Server 2008 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| **Computer Memory** | 1GB RAM | 2GB or more |

***Table 2: Hardware Requirement for Server***

**For Mobile**

|  |  |  |
| --- | --- | --- |
| **Mobile** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | 2 Mbps | 4 Mbps |
| **Operating System** | Android 4.4.2 | Android 6.0 |
| **Hardware** | Bluetooth 4.0 supported | Bluetooth 4.0 supported |
| **Memory** | 1GB | 2GB or more |

Table 3: Hardware Requirement for Mobile

**For PC**

|  |  |  |
| --- | --- | --- |
| **PC** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operating System** | Window 7 | Window 7 or more. |
| **Computer Memory** | 2GB RAM | 4GB RAM or more |

Table 4: Hardware Requirement for PC.

* + - 1. **Software requirements**

|  |  |  |
| --- | --- | --- |
| **Software** | **Name / Version** | **Description** |
| Operating system | Window 7 or above | Operating system and platform for development |
| Environment | Java EE 5 or above | Specification for developing web application |
| IDE | Netbeans 8.2, Visual studio code | Used to implement web application. |
| DBMS | SQLServer | Used to create and manage the database for system. |
| Source control | Source tree | Used for source control. |
| Web browser | Chrome 42 or above | Testing browser |

**2. Project organization**

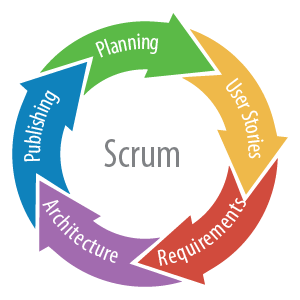
**2.1 Software Process Model**

This project is developed under scrum model. We apply customized scrum model to capable with current situation in our team. We choose this model because the following reasons:

- Variability: This project is not new, but because members of the team still do not have a complete view of hospital operating procedures, the continuous reception of new information is necessary. We collect software requirements from customers, thereby gradually improving the system structure. Therefore, the system will be constantly improved and changed to suit customer requirements.

- Humans resource and time limitation: Scrum has proven to deliver products to end-users 30% -40% faster than traditional methods. Because the Scrum model works with the principle of breaking down the software that needs to be produced into small parts to grow. So with such a rush, the use of the scum model is extremely suitable for our small group of people.

- User involvement: This project must be regularly reviewed and reviewed by the user so the requirement of the scrum process is necessary. We can minimize the risk of complete failure when a member withdraws from the group. As Scrum works in stages, the project team can take steps, then draw on the experience or continue to build on the advantages to improve the product.



Reference: <https://www.nomagic.com/news/new-noteworthy/magicdraw-noteworthy/magicdraw-18-2-fr/magicdraw-18-2-fr-all#scrum>

**2.2 Roles and responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Phạm Công Thành | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Hồ Hoàng Long | Team Leader,  BA, DEV, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **3** | Trần Phương Linh | Team member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |

Table 5: Roles and Responsibilities Details

**2.3 Tools and Techniques**

|  |  |
| --- | --- |
|  | **Techniques** |
| **Front-end** | HTML5  CSS3  Javascript  JQuery  Bootstrap  Angular 5 |
| **Back-end** | JavaEE  Restful API |

*Table 6: Tools and techniques*

**3. Project Management Plan**  
**3.1 Software development life cycle**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource Needed** | **Dependencies**  **and Constrains** | **Risks** |
| Planning | - Analyze the current situation  - Plan whole project | - Introduction of proposed system  - Project task plan | 9 man-days | N/A | N/A |
| Requirement analysis | - Identify and  clarify project  requirements | - Software Requirement Specification  - Prototypes | 21 man-days | N/A | - Missing requirement  - Unclear scope of  project  - Lack of member share of understand |
| Design | - Architecture Design for the System  - Detail design using top-down breakdown  - Choose the Architecture Style | - Software design document  - Technology  notes | 21 man-days | Depend on  “Requirement  Analysis” | - Lack of  experience  - Not fulfil  requirement |
| Implementation | - Implement the all functions for the System | - Runnable System | 12 man-days | Depend on  “Design” | N/A |
| Deployment | - Deploy the system to server | - Deployment Result | 9 man-days | Depend on  “Implementation” | N/A |
| Testing | - Unit testing  - Component testing  - Intergration testing  - Acceptance testing  - Fix error | - Test Case List  - Test Report | 21 man-days | Depend on  “Deployment” | - Lack of  error |
| Evaluation | - Demo for user | - Feedback from user | 21 man-days | Depend on  “Testing” | N/A |

Table 7: Software Development Life Cycle Detail

**3.2 Phase Detail**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| ***1. Collect requirements*** | Find which systems currently provide similar service, their strengths and weakness. | LongHH, LinhTP |
| ***2. Identify and clarify*** ***main functions.*** | Define which main functions system should provide. | LongHH, LinhTP |
| ***3.Requirements specification*** | Defining the requirements in detail. | LongHH, LinhTP |
| ***4. Implement code*** | - GUI Design  - Collect and validate information about hospital portal  - Connect to database  - Implement backend | LongHH, LinhTP |

**4. Coding Convention**

Java: Using to develop web service.

* Naming convention:
* Variable and method names are in mixed case, with first letter of each internal word capitalized except first word.
* Method names should be verbs.
* Class names should be nouns, in mixed case with first letter of each internal word capitalized.
* Constant names should be all uppercase with words separated by underscore.
* Comment:
* Using /\* \*/ for block comments.
* Using // for line comments.

Using Java coding convention from:

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

Angular: Using to develop mobile application.

* Naming Convention:
  + - * + Naming conventions are hugely important to maintainability and readability. This guide recommends naming conventions for the file name and the symbol name.
* Symbols and file names:
* Do use consistent names for all assets named after what they represent.
* Do use upper camel case for class names.
* Do match the name of the symbol to the name of the file.

Using Angular coding convention from:

https://angular.io/guide/styleguide **C. Report No. 3 Software Requirement Specification**  
**1. User Requirement Specification**

***1.1 Admin Requirement:***

Admin is a person who has permission to manage some action of the project. To use all functions, user must login. There are some functions admin can use:

* Login
* Manage user:
  + Create, update, Activate /deactivate user
* Manage department:
  + Create, update, Activate /deactivate department
* Manage room:
  + Create, update, Activate /deactivate room
* View schedule
  + View mission schedule
  + View meeting schedule
  + View shift schedule

***1.2 Manager Requirement:***

Manager who has higher privileges than the scheduler. They have permission to approve and change status of one mission. The manager can use some following functions:

* Login
* Manage schedule:
  + Create, update, Activate /deactivate, approve mission schedule
  + Create, update, Activate /deactivate, approve meeting schedule
  + Create, update, Activate /deactivate, approve shift schedule
* Manage notification:
  + Create, update, Activate /deactivate, approve notification
* View schedule
  + View mission schedule
  + View meeting schedule
  + View shift schedule

***1.3 Scheduler Requirement:***

Scheduler guest who already login to the system by his/her account and uses service of system. The scheduler can use some following functions:

* Login
* Manage schedule:
  + Create, update mission schedule
  + Create, update meeting schedule
  + Create, update shift schedule
* View schedule
  + View mission schedule
  + View meeting schedule
  + View shift schedule

***1.4 User Requirement:***

User is normal staff of hospital they are only able to view the calendar. The user can use some following functions:

* Login
* View schedule
  + View mission schedule
  + View meeting schedule
  + View shift schedule

**2. System Requirement Specification**  
**2.1 External Interface Requirement**  
**2.1.1 User Interface**

* General requirement for graphics user interface is the GUI should be simple, clear, intuitive, and reminiscent.
* User interface is display all by English.
* The user interface uses consistent palette of colors between the text and the background.
* The user interface displays best on 1600x900-screen size.

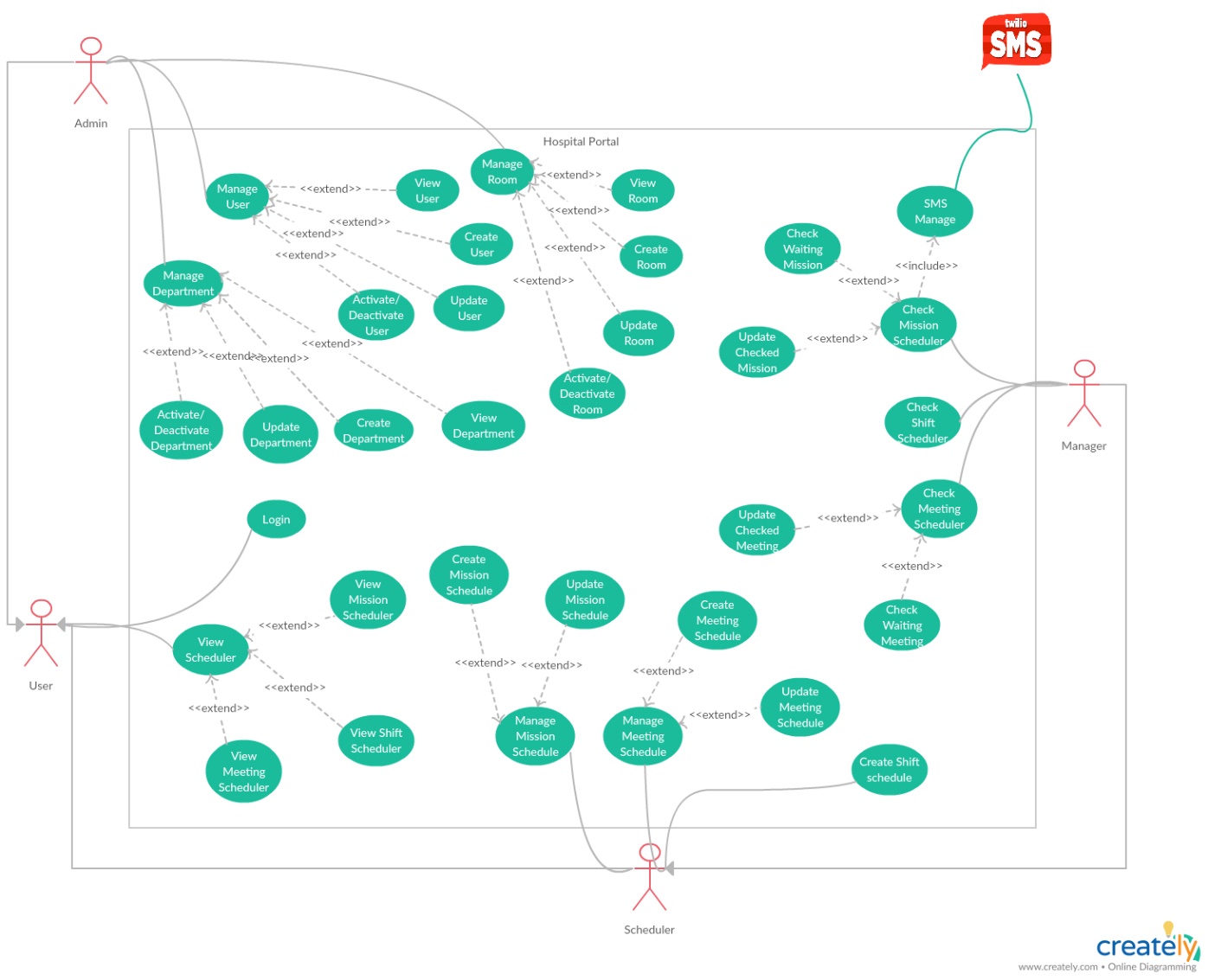
**2.1.2 Hardware Interface**

* RAM: 2GB
* CPU: 1.2GHz
  + 1. **Software Interface**
* Web application: work with Firefox (v30 or above), Chromes (v14 or above), Internet Explorer (v10 or above) browse

**2.1.4 Communication Protocol**

* Use HTTP protocol 1.1 for communication between the web browser and the web server.

**2.2 System Overview Use Case**

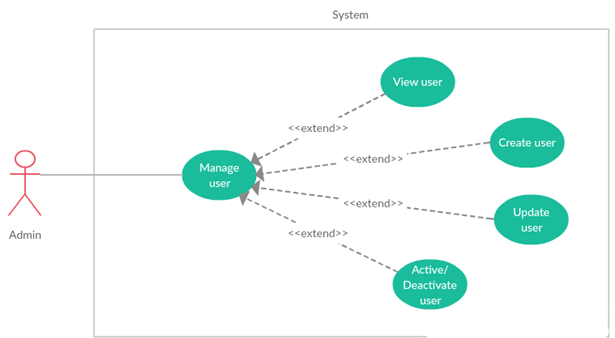


* 1. **List of Use Case**

***2.3.1 <Admin>Overview Use Case***

******

***2.3.1.1 <Admin> Manage user***

****

***2.3.1.1.1 <Admin> View user***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP01 | | | |
| **Use Case No.** | UC\_HP01 | **Use Case Version** | 2.0 |
| **Use Case Name** | View user by Admin | | |
| **Author** | LinhTP | | |
| **Date** | 17/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows Admin to view list user and detail.   **Goal:**   * Admin can view all user and user information.   **Triggers:**   * Admin click tag [Manage User] on menu bar.   **Preconditions:**   * System had connected to database. * User login with role Admin.   **Post Conditions:**   * **Success:** System show all user and detail. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage User] on menu bar. | System move to interface, where show all users in list.  On each row there are:  -Username  -Full name  -Department  -Position  -Role  -Phone  -Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [View] icon on selected user row | System show user detail page.  On page is all user info load from database:  -Username  -Email  -Full name  -Sex  -Day of birth  -Department  -Phone  -Position  -Address  -Certificate  -Role |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no user in the data | Show label massager “There is no user in the data”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**   * Paging list user. * Have search box, and filter by department. * The displayed information must be match the employee information. | | | |

***2.3.1.1.2 <Admin> Update user***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP02 | | | |
| **Use Case No.** | UC\_HP02 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update user | | |
| **Author** | LinhTP | | |
| **Date** | 17/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows Admin to update user detail.   **Goal:**   * Admin can update user information.   **Triggers:**   * Admin click [Update] icon.   **Preconditions:**   * User login with role Admin. * System had connected to database. * Selected User exited in database.   **Post Conditions:**   * **Success:** User information updated in data. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage User] on menu bar. | System move to interface, where show all users in list.  On each row there are:  -Username  -Full name  -Department  -Position  -Role  -Phone  -Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [Update] icon on selected user row | System show update user page.  [**Alternative 2]**  On popup is all user info load from database.  -Username (cannot change): textbox up to 50 characters long.  -Email (require): Text box with email validation.  -Full name (require): textbox, Does not contain numbers or special characters, up to 128 characters long.  -Sex (require): Combo box.  -Day of birth (require): date picker.  -Department (require): Select from Department dropdown list.  -Phone (require): Text box, only up to 20 numbers.  -Position (require): multiple choice list.  -Address (require): Text area.  -Certificate: Text area.  -Role (required): Select from role dropdown list. | | 3 | Admin input new information in update form. | System validation after information inputted. | | 4 | Admin click [save] button | Appears notification dialog  “Success” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no user in the data | Show label massager “There is no user in the data”. | | 2 | Admin click “Back” button | Back to user list page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** Activity history  **Business Rules:**   * Information validation by hospital requirement template. | | | |

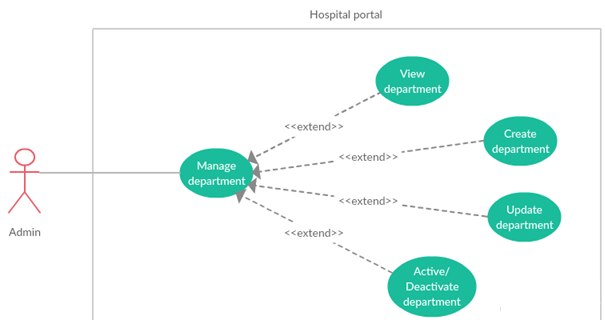
***2.3.1.1.3 <Admin> Create user***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP03 | | | |
| **Use Case No.** | UC\_HP03 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create user | | |
| **Author** | LinhTP | | |
| **Date** | 17/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows Admin to create new user.   **Goal:**   * Admin can create new user.   **Triggers:**   * Admin click [Create new user] button.   **Preconditions:**   * System had connected to database. * User login with role Admin.   **Post Conditions:**   * **Success:** New user create in data. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage User] on menu bar. | System move to interface, where show all users in list.  On each row there are:  -Username  -Full name  -Department  -Position  -Role  -Phone  -Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [Add] button. | System show Create user page.  [**Alternative 2]**  On popup is user info input form:  -Username (cannot change): textbox.  -Email (require): Text box with email validation.  -Full name (require): textbox, Does not contain numbers or special characters, up to 128 characters long.  -Sex (require): Combo box.  -Day of birth (require): date picker.  -Department (require): Select from Department dropdown list.  -Phone (require): Text box, only up to 20 numbers.  -Position (require): multiple choice list.  -Address (require): Text area.  -Certificate: Text area.  -Role (required): Select from role dropdown list. | | 3 | Admin input user information in create form. | System validation after information inputted.  Check duplicate username in database.  [**Alternative 3]** | | 4 | Admin click [Save] button | Appears notification dialog  “Success” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no user in the data | Show label massager “There is no user in the data”. | | 2 | Admin “Back” button | Back to user list page. | | 3 | User name exited in database. | Show message:” This username exit could not complete the transaction”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** Activity history  **Business Rules:**   * Information validation by hospital requirement template. | | | |

***2.3.1.1.4 <Admin> Activate/Deactivate user***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP04 | | | |
| **Use Case No.** | UC\_HP04 | **Use Case Version** | 2.0 |
| **Use Case Name** | Activate /Deactivate user | | |
| **Author** | LinhTP | | |
| **Date** | 17/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows Admin to Activate / deactivate user.   **Goal:**   * User account are Activate /Deactivate.   **Triggers:**   * Admin click [Activate /Deactivate] button.   **Preconditions:**   * User login with role Admin. * System had connected to database. * User account is Activate.   **Post Conditions:**   * **Success:** User account was Activate /deactivate. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage User] on menu bar. | System move to interface, where show all users in list.  On each row there are:  -Username  -Full name  -Department  -Position  -Role  -Phone  -Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [Deactivate] icon on selected user row | Appears confirmation dialog  “Are you sure to Activate /deactivate this user?”  [**Alternative 2]** | | 3 | Admin click [yes] button | User account status change to Activate /deactivate. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no user in the data | Show label massager “There is no user in the data”. | | 2 | Admin click no on confirmation dialog | Close confirmation dialog |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** Activity history  **Business Rules:**   * N/A | | | |

***2.3.1.2<Admin> Manage department***



***2.3.1.2.1 <Admin> View department***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP05 | | | |
| **Use Case No.** | UC\_HP05 | **Use Case Version** | 2.0 |
| **Use Case Name** | View Department | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin to view Department and username of user in Department.   **Goal:**   * Admin can view Department and user in Department.   **Triggers:**   * Admin click “View” Icon.   **Preconditions:**   * System had connected to database. * User login with role Admin.   **Post Conditions:**   * **Success:** System show all user in the selected specially * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage Department] on menu bar. | System move to interface, where show all Department in list.  On each row there are:  - Name  - Code.  - Quantity  - Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [View] icon on selected user row | System show user detail page.  On Department information page:   * Department name: * Department code: * Description |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no Department in the data | Show label massager “There is no Department in the data”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The Department must show correctly  -The page must have good design | | | |

***2.3.1.2.2 <Admin> Create department***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP06 | | | |
| **Use Case No.** | UC\_HP06 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create Department | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin to create new Department. * **Goal:** * Admin can create a Department.   **Triggers:**   * Admin click “Create new Department” button.   **Preconditions:**   * System had connected to database. * User login with role Admin.   **Post Conditions:**   * **Success:** A new Department created. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage Department] on menu bar. | System move to interface, where show all Department in list.  On each row there are:  - Name  - Code.  - Quantity  - Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [Add] button. | System show create Department page with category info input form:  Department name (require): textbox, up to 512 characters long.  Department code (required): textbox, up to 10 character long.  Description: text area. | | 3 | Admin input Department information in create form.  Admin click [Save] button | [**Alternative 2]**  [**Alternative 3]**  Department will be create on data. Appears “Success” message.  Back to Department list |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no Department in the data | Show label massager “There is no Department in the data”. | | 2 | This Department name is existed | Appears “Name existed.” Message. | | 3 | This Department code is existed | Appears “Code existed.” Message. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The Department must show correctly  -The page must have good design | | | |

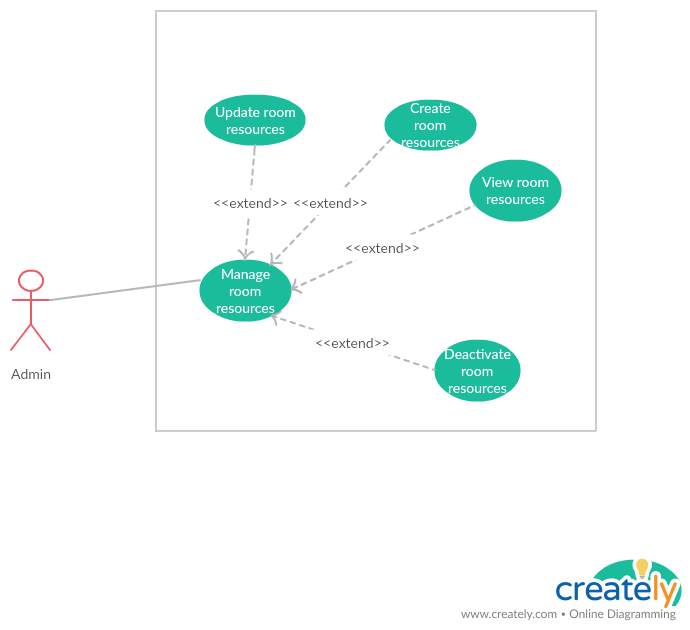
***2.3.1.2.3 <Admin> Update department***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP07 | | | |
| **Use Case No.** | UC\_HP07 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update Department | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin to update a Department.   **Goal:**   * Admin can update a Department.   **Triggers:**   * Admin click [update] icon.   **Preconditions:**   * System had connected to database. * User login with role Admin.   **Post Conditions:**   * **Success:** A Department updated. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage Department] on menu bar. | System move to interface, where show all Department in list.  On each row there are:  - Name  - Code.  - Quantity  - Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [View] icon | System show update Department page with category info load from database:  Department name (require): textbox, up to 512 characters long.  Department code (required): textbox, up to 10 character long.  Description: text area.  [**Alternative 2]** | | 3 | Admin input Department information in update form.  Admin click [save] button | [**Alternative 3]**  Department will be create on data. Appears “Success” message.  Back to Department list  [**Alternative 4]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no Department in the data | Show label massager “There is no Department in the data”. | | 2 | Admin click [Back] button | Back to department list page. | | 3 | This Department name is existed | Appears “Name exited.” Message. | | 4 | This Department code is existed | Appears “Code exited.” Message. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The Department must show correctly  -The page must have good design | | | |

***2.3.1.2.4 <Admin> Activate /Deactivate department***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP08 | | | |
| **Use Case No.** | UC\_HP08 | **Use Case Version** | 2.0 |
| **Use Case Name** | Activate /Deactivate Department | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin to Activate /deactivate Department.   **Goal:**   * Admin can Activate /deactivate Department.   **Triggers:**   * Admin click “Activate /Deactivate” Icon.   **Preconditions:**   * User login with role Admin. * System had connected to database. * This Department is Activate /deactivate.   **Post Conditions:**   * **Success:** Department is Deactivate * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage Department] on menu bar. | System move to interface, where show all Department in list.  On each row there are:  - Name  - Code.  - Quantity  - Status  -View icon.  [**Alternative 1]**  [**Alternative 2]** | | 2 | Admin click [Activate /Deactivate] button on selected Department row | Category is Activate /deactivate on data. Appears “Success” message.  Back to category list  [**Alternative 3]** | | 3 | Admin click [yes] button |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no Department in the data | Show label massager “There is no Department in the data”. | | 2 | This Department has some activate user | Show label massager “Can’t disable this department. There are Activate account in this department”.  Department no change. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The Department must show correctly  -The page must have good design | | | |

***2.3.1.3 <Admin> Manage room resource***

****

***2.3.1.3.1 <Admin> View room resource***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP09 | | | |
| **Use Case No.** | UC\_HP09 | **Use Case Version** | 2.0 |
| **Use Case Name** | View room resources | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin to view room resources information.   **Goal:**   * Admin can view room resources information.   **Triggers:**   * Admin click “View” Icon.   **Preconditions:**   * User login with role Admin. * System had connected to database.   **Post Conditions:**   * **Success:** System show resources information * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage room] on menu bar. | System move to Manage room resources interface, where show all meeting room in list.  On each row there are:  -Room name  -Room size (max people).  -Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [View] icon on selected user row | System show room detail popup.  On popup is room information:  Room name  Room size |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no room in the data | Show label massager “There is no room in the data”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The room must show correctly  -The page must have good design | | | |

***2.3.1.3.2 <Admin> Create room resource***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP10 | | | |
| **Use Case No.** | UC\_HP10 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create room resources | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin to create new room resources.   **Goal:**   * Admin can create a room resources.   **Triggers:**   * Admin click “Add” button.   **Preconditions:**   * User login with role Admin. * System had connected to database.   **Post Conditions:**   * **Success:** A new room created. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage room] on menu bar. | System move to Manage room resources interface, where show all meeting room in list.  On each row there are:  -Room name  -Room size (max people).  -Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [Add] button. | System show create room page with room info input form:  Room name (require): textbox, up to 512 characters long.  Room size (max people). Input number. | | 3 | Admin input room information in create form.  Admin click [Save] button | New room will be create on data. Appears “Success” message.  Back to room list  [**Alternative 2]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no room in the data | Show label massager “There is no room in the data”. | | 2 | Admin click “Back” button | Back to room list page |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The room must show correctly  -The page must have good design | | | |

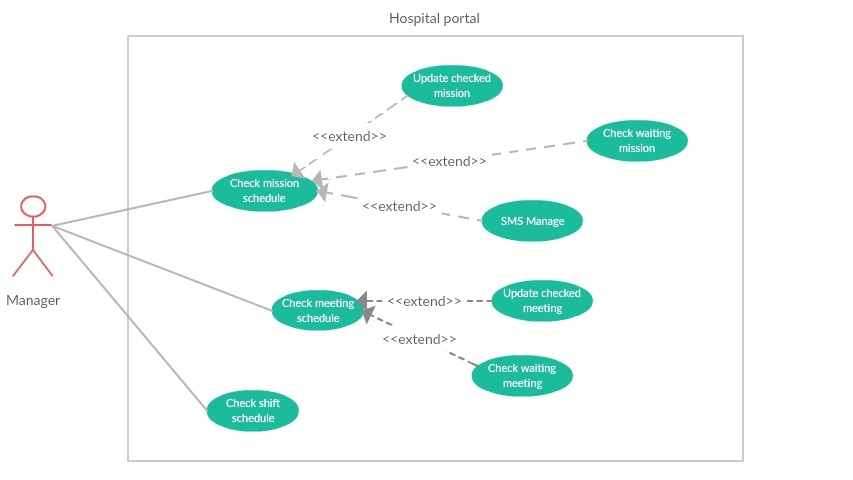
***2.3.1.3.3 <Admin> Update room resource***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP11 | | | |
| **Use Case No.** | UC\_HP11 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update room resources | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin to update a room resources information.   **Goal:**   * Admin can create a room resources.   **Triggers:**   * Admin click [View] icon   **Preconditions:**   * User login with role Admin. * System had connected to database.   **Post Conditions:**   * **Success:** Room updated. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage room] on menu bar. | System move to Manage room resources interface, where show all meeting room in list.  On each row there are:  -Room name  -Room size (max people).  -Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [update] icon on selected room row | System show update room popup with room information load from database:  Room name (require): textbox, up to 512 characters long.  Room size (max people). Input number. | | 3 | Admin input room information in create form.  Admin click [Save] button | Room will be create on data. Appears “Success” message.  Back to room list  [**Alternative 2]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no room in the data | Show label massager “There is no room in the data”. | | 2 | Admin click “Back” button | Back to room list page |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The room must show correctly  -The page must have good design | | | |

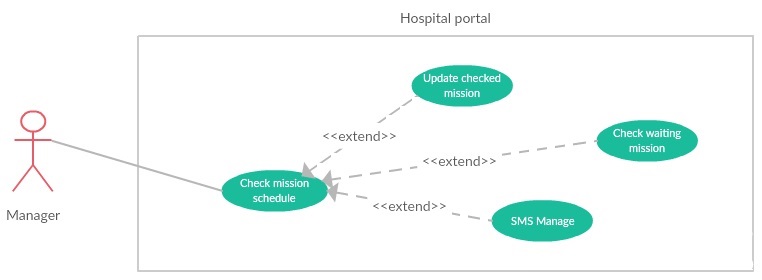
***2.3.1.3.1 <Admin> Activate /Deactivate room resource***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP12 | | | |
| **Use Case No.** | UC\_HP12 | **Use Case Version** | 2.0 |
| **Use Case Name** | Activate /Deactivate room resources | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * This use case allows admin to Activate /deactivate room.   **Goal:**   * Admin can Activate /deactivate room.   **Triggers:**   * Admin click “Activate /Deactivate” button.   **Preconditions:**   * User login with role Admin. * System had connected to database. * This room is Activate /deactivate.   **Post Conditions:**   * **Success:** room is Activate /Deactivate * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click tag [Manage room] on menu bar. | System move to Manage room resources interface, where show all meeting room in list.  On each row there are:  -Room name  -Room size (max people).  -Status  -View icon.  [**Alternative 1]** | | 2 | Admin click [Activate /Deactivate] button on selected room row | Category is deactivate on data. Appears “Success” message.  Back to category list  [**Alternative 2]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no room in the data | Show label massager “There is no room in the data”. | | 2 | Admin click “Back” button | Back to room list page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The room must show correctly  -The page must have good design | | | |

***2.3.2 <Manager>Overview Use Case***



***2.3.2.1 <Manager> Check mission schedule***

******

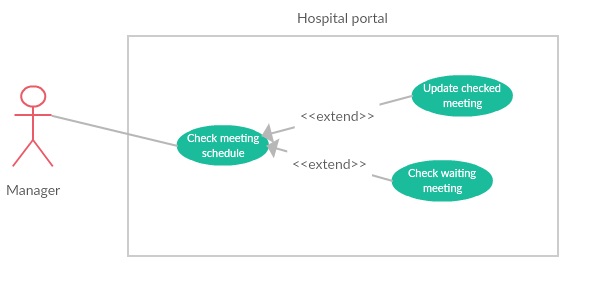
***2.3.2.1.1 <Manager> Check waiting mission***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP13 | | | |
| **Use Case No.** | UC\_HP13 | **Use Case Version** | 2.0 |
| **Use Case Name** | Check waiting mission | | |
| **Author** | LongHH | | |
| **Date** | 13/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager.   **Summary:**   * This use case allows manager to check mission schedule.   **Goal:**   * Manager can check mission schedule.   **Triggers:**   * Manager sends command to check mission schedule.   **Preconditions:**   * User must login into the system with role Manager. * Mission status is waiting.   **Post Conditions:**   * **Success:** Manager could check mission schedule successfully. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager goes to manage mission schedule page. | Display 2 tabs (waiting and checked), in each tab:  Display mission list in table if system has any mission schedule to manage.  On each row there are:   * Start date * End date * Place * Worker * Status * Update button | | 2 | Manager click on “Waiting Mission” tab to view uncheck missions. | Display table of uncheck mission | | 3 | Manager click on “Waiting” button and select “Accept” if the mission schedule is appropriate.  [**Alternative 1]** | Mission status change to Accept, Mission upload to portal  System send SMS to all user in worker list to notify. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Manager click on “Waiting” button and select “Deny” if the mission schedule is inappropriate. | The status of mission change to “Denied”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server. | System show message the “Connection to the server has been lost. Please check your internet connection or try again later.” when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * Worker in list will be received SMS by the phone number in account. | | | |

***2.3.2.1.2 <Manager> Update checked mission***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP14 | | | |
| **Use Case No.** | UC\_HP14 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update checked mission | | |
| **Author** | LongHH | | |
| **Date** | 13/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager.   **Summary:**   * This use case allows manager to update checked mission schedule status.   **Goal:**   * Mission schedule status was updated.   **Triggers:**   * Manager sends command to update mission schedule status.   **Preconditions:**   * User must login into the system with role Manager. * Mission was checked (status is Accept or Deny)   **Post Conditions:**   * **Success:** Mission schedule status updated successfully. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager goes to manage mission schedule page. | Display 2 tabs (waiting and checked), in each tab:  Display mission list in table if system has any mission schedule to manage.  On each row there are:   * Start date * End date * Place * Worker * Status * Update button | | 2 | Manager click on “Checked Mission” tab to view uncheck missions. | Display table of checked mission | | 3 | Manager click on “Deny” button and select “Accept” if the mission schedule is appropriate.  [**Alternative 1]**  [**Alternative 2]** | Mission status change to Accept, Mission upload to portal  System send SMS to all user in worker list to notify. | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Manager click on “Accept” button and select “Deny” if the mission schedule is inappropriate. | The status of mission change to “Denied”.  Mission hidden from the portal.  System send SMS to all user in worker list to notify. | | 2 | Manager click on “No” button | The status not change. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server. | System show message the “Connection to the server has been lost. Please check your internet connection or try again later.” when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * Worker in list will be received SMS by the phone number in account. | | | |

***2.3.2.2 <Manager> Check meeting schedule***

******

***2.3.2.2.1 <Manager> Check waiting meeting***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP15 | | | |
| **Use Case No.** | UC\_HP15 | **Use Case Version** | 2.0 |
| **Use Case Name** | Check waiting meeting | | |
| **Author** | LongHH | | |
| **Date** | 13/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager.   **Summary:**   * This use case allows manager to check meeting schedule.   **Goal:**   * Manager can check meeting schedule.   **Triggers:**   * Manager sends command to check meeting schedule.   **Preconditions:**   * User must login into the system with role Manager. * Meeting status is waiting.   **Post Conditions:**   * **Success:** Manager could check meeting schedule successfully. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager goes to manage meeting schedule page. | Display 2 tabs (waiting and checked), in each tab:  Display mission list in table if system has any meeting schedule to manage.  On each row there are:   * Meeting name * Room * Date * Start time * Duration * Status * Update button | | 2 | Manager click on “Waiting Meeting” tab to view uncheck meeting. | Display table of uncheck meeting | | 3 | Manager click on “Waiting” button and select “Accept” if the meeting schedule is appropriate.  [**Alternative 1]** | Meeting status change to Accept, Meeting upload to portal. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Manager click on “Waiting” button and select “Deny” if the meeting schedule is inappropriate. | The status of meeting change to “Denied”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server. | System show message the “Connection to the server has been lost. Please check your internet connection or try again later.” when the internet is lost. |   **Relationships:** N/A  **Business Rules:** | | | |

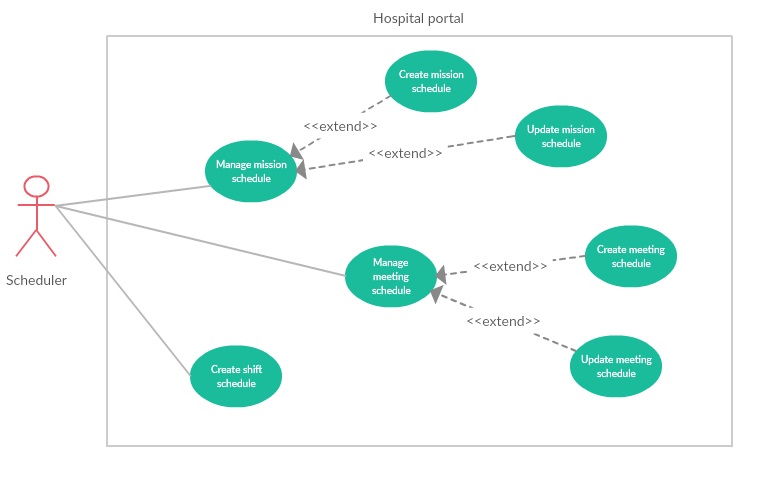
***2.3.2.2.2 <Manager> Update checked meeting***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP16 | | | |
| **Use Case No.** | UC\_HP16 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update checked meeting | | |
| **Author** | LongHH | | |
| **Date** | 13/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager.   **Summary:**   * This use case allows manager to update checked meeting schedule status.   **Goal:**   * Meeting schedule status was updated.   **Triggers:**   * Manager sends command to check meeting schedule.   **Preconditions:**   * User must login into the system with role Manager. * Meeting is checked (status is Accept or Deny).   **Post Conditions:**   * **Success:** Meeting schedule status updated successfully**.** * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager goes to manage meeting schedule page. | Display 2 tabs (waiting and checked), in each tab:  Display mission list in table if system has any meeting schedule to manage.  On each row there are:   * Meeting name * Room * Date * Start time * Duration * Status * Update button | | 2 | Manager click on “Checked Meeting” tab to view uncheck meeting. | Display table of uncheck meeting | | 3 | Manager click on “Deny” button and select “Accept” if the meeting schedule is appropriate.  [**Alternative 1]**  [**Alternative 2]** | Meeting status change to Accept, Meeting upload to portal. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Manager click on “Accept” button and select “Deny” if the meeting schedule is inappropriate. | The status of meeting change to “Denied”.  Meeting hidden from the portal. | | 2 | Manager click on “No” button | The status not change. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server. | System show message the “Connection to the server has been lost. Please check your internet connection or try again later.” when the internet is lost. |   **Relationships:** N/A  **Business Rules:** | | | |

***2.3.2.3 <Manager> Check shift schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP17 | | | |
| **Use Case No.** | UC\_HP17 | **Use Case Version** | 2.0 |
| **Use Case Name** | Check shift schedule | | |
| **Author** | LongHH | | |
| **Date** | 13/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager.   **Summary:**   * This use case allows manager to check shift schedule.   **Goal:**   * Manager can check shift schedule.   **Triggers:**   * Manager sends command to check shift schedule.   **Preconditions:**   * User must login into the system with role Manager. * The waiting shift schedule status is uncheck.   **Post Conditions:**   * **Success:** Manager could shift meeting schedule successfully. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager goes to manage shift schedule page and select Department. | Display 3 tabs (“In coming”, “Quick search” and “Pass”), in each tab:  Display shift schedule manage panel of this department in timeline list.  On each panel there are:   * Start day –End day. * Waiting schedule and status button. * Posted schedule and status button. | | 2 | Manager select the “Waiting : uncheck” scheduler button | Display uncheck shift scheduler | | 3 | Manager click on “Waiting” button and select “Accept” if the shift schedule is appropriate.  [**Alternative 1]** | Posted Shift schedule = Waiting Shift schedule  Waiting and posted status change to “posted”. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Manager click on “Waiting” button and select “Deny” if the meeting schedule is inappropriate. | Waiting Shift schedule = Posted Shift schedule  Waiting Shift schedule status = Posted Shift schedule status |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server. | System show message the “Connection to the server has been lost. Please check your internet connection or try again later.” when the internet is lost. |   **Relationships:** N/A  **Business Rules:** | | | |

***2.3.3 <Scheduler>Overview Use Case***



***2.3.3.1 < Scheduler > Create shift schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP18 | | | |
| **Use Case No.** | UC\_HP18 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create Shift Schedule | | |
| **Author** | LongHH | | |
| **Date** | 13/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager/Scheduler.   **Summary:**   * This use case allows Manager/Scheduler to create Shift Scheduler.   **Goal:**   * Manager/Scheduler can create a Shift Scheduler.   **Triggers:**   * Manager/Scheduler sends command to create a Shift Scheduler.   **Preconditions:**   * User must login into the system with role Manager or Scheduler.   **Post Conditions:**   * **Success:** Manager/Scheduler could create Shift Scheduler successfully. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager/Scheduler goes to manage shift schedule page. | Display 3 tabs (“In coming”, “Quick search” and “Pass”), in each tab:  Display shift schedule manage panel of this department in timeline list.  On each panel there are:   * Start day –End day. * Waiting schedule and status button. * Posted schedule and status button. | | 2 | Manager/Scheduler select the “Create new scheduler” button | Display create new shift scheduler page | | 3 | Manager/Scheduler select Department, Employee per shift number and week | System validate and create the schedule | | 4 | Manager/Scheduler select Start time and End time and click “+” button to add shift. | System validate and add more shift to every date in scheduler. | | 5 | Manager/Scheduler select employees in every shift | System validate and activate “Finish” button. | | 6 | Manager/Scheduler click “Finish” button. | Shift schedule save to schedule manage waiting by status uncheck. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server. | System show message the “Connection to the server has been lost. Please check your internet connection or try again later.” when the internet is lost. | | 2 | Department not selected | Department label background change to red, deactivate finish button. | | 3 | Week not selected | Week label background change to red, deactivate finish button. | | 4 | Employee per shift invalid | Employee per shift label background change to red, deactivate finish button. | |  |  |  |   **Relationships:** N/A  **Business Rules:**   * Worker in list will be received SMS by the phone number in account. | | | |

***2.3.3.2 < Scheduler > Manage mission schedule***

***2.3.3.2.1 <Scheduler> Create mission schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP19 | | | |
| **Use Case No.** | UC\_HP19 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create mission schedule | | |
| **Author** | LongHH | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager, scheduler.   **Summary:**   * This use case allows manager/scheduler to create new mission schedule.   **Goal:**   * Manager/scheduler can create new mission schedule.   **Triggers:**   * Manager/scheduler click “Add” button.   **Preconditions:**   * User login with role Manager/Scheduler. * System had connected to database.   **Post Conditions:**   * **Success:** Mission schedule is created completely. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager/scheduler click tag [Manage mission schedule] on menu bar. | System move to Manage mission schedule interface, where show all mission schedule in list.  On each row there are:  -Start date  -End date  -Place  -Worker  -Status  -View icon.  [**Alternative 1]** | | 2 | Manager/scheduler click [Add] button. | System show create mission schedule page with mission schedule info input form:   * Member (required): Multiple choice dropdown list. * Start date (required): Date picker. * End date (required): Date picker * Place (required): input text * Content (required): input text (up to 512 characters long) * Note: text area (up to 512 characters long) | | 3 | Manager/scheduler input mission information in create form.  Manager/scheduler click [Save] button | Mission will be create on data. Appears “Success” message.  Back to mission list  [**Alternative 2]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no mission schedule in the data | Show label massager “There is no mission schedule in the data”. | | 2 | Manager/scheduler click “Back” button | Back to mission schedule list page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The mission schedule must show correctly  -The page must have good design | | | |

***2.3.3.2.2 <Scheduler> Update mission schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP20 | | | |
| **Use Case No.** | UC\_HP20 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update mission schedule | | |
| **Author** | LongHH | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager, scheduler.   **Summary:**   * This use case allows manager/scheduler to update mission schedule.   **Goal:**   * Manager/scheduler can update mission schedule.   **Triggers:**   * Manager/scheduler click “View” icon.   **Preconditions:**   * User login with role Manager/Scheduler. * System had connected to database.   **Post Conditions:**   * **Success:** Mission schedule is updated completely. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager/scheduler click tag [Manage mission schedule] on menu bar. | System move to Manage mission schedule interface, where show all mission schedule in list.  On each row there are:  -Start date  -End date  -Place  -Worker  -Status  -View icon.  [**Alternative 1]** | | 2 | Manager/scheduler click [View] icon. | System show update mission schedule page with mission schedule info input form:   * Member (required): Multiple choice dropdown list. * Start date (required): Date picker. * End date (required): Date picker * Place (required): input text * Content (required): input text (up to 512 characters long) * Note: text area (up to 512 characters long) | | 3 | Manager/scheduler input mission information in create form.  Manager/scheduler click [Save] button | Mission will be create on data. Appears “Success” message.  Back to mission list  [**Alternative 2]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no mission schedule in the data | Show label massager “There is no mission schedule in the data”. | | 2 | Manager/scheduler click “Back” button | Back to mission schedule list page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The mission schedule must show correctly  -The page must have good design | | | |

***2.3.3.3 < Scheduler > Manage meeting schedule***

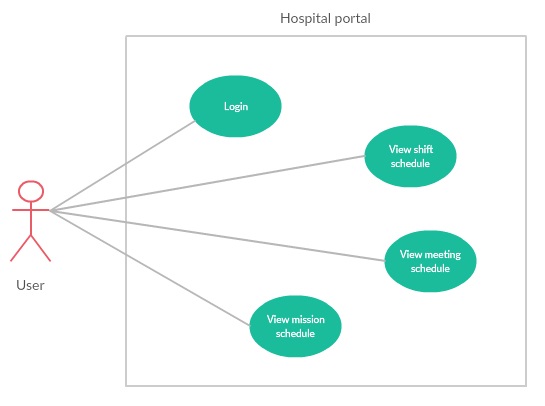
***2.3.3.3.1 <Scheduler> Create meeting schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP21 | | | |
| **Use Case No.** | UC\_HP21 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create meeting schedule | | |
| **Author** | LongHH | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager, scheduler.   **Summary:**   * This use case allows manager/scheduler to create new meeting schedule.   **Goal:**   * Manager/scheduler can create new meeting schedule.   **Triggers:**   * Manager/scheduler click “Add” button.   **Preconditions:**   * User login with role Manager/Scheduler. * System had connected to database.   **Post Conditions:**   * **Success:** Meeting schedule is created completely. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager/scheduler click tag [Manage meeting schedule] on menu bar. | System move to Manage meeting schedule interface, where show all meeting schedule in list.  On each row there are:  -Meeting name  -Room  -Date  -Start time  -End time  -Status  -View icon.  [**Alternative 1]** | | 2 | Manager/scheduler click [Add] button. | System show create meeting schedule page with meeting schedule info input form:   * Meeting name(required): input text (up to 256 characters long) * Room(required): Dropdown list * Start time (required): Time picker. * End time (required): Time picker * Date (required): Date picker * Content (required): input text (up to 512 characters long) * Note: text area (up to 512 characters long) | | 3 | Manager/scheduler input meeting information in create form.  Manager/scheduler click [Save] button | Meeting will be create on data. Appears “Success” message.  Back to meeting list  [**Alternative 2]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no meeting schedule in the data | Show label massager “There is no meeting schedule in the data”. | | 2 | Manager/scheduler click “Back” button | Back to meeting schedule list page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The meeting schedule must show correctly  -The page must have good design | | | |

***2.3.3.3.2 <Scheduler> Update meeting schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP22 | | | |
| **Use Case No.** | UC\_HP22 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update meeting schedule | | |
| **Author** | LongHH | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * Manager, scheduler.   **Summary:**   * This use case allows manager/scheduler to update meeting schedule.   **Goal:**   * Manager/scheduler can update meeting schedule.   **Triggers:**   * Manager/scheduler click “Add” button.   **Preconditions:**   * User login with role Manager/Scheduler. * System had connected to database.   **Post Conditions:**   * **Success:** Meeting schedule is updated completely. * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Manager/scheduler click tag [Manage meeting schedule] on menu bar. | System move to Manage meeting schedule interface, where show all meeting schedule in list.  On each row there are:  -Meeting name  -Room  -Date  -Start time  -End time  -Status  -View icon.  [**Alternative 1]** | | 2 | Manager/scheduler click [View] button. | System show create meeting schedule page with meeting schedule info input form:   * Meeting name(required): input text (up to 256 characters long) * Room(required): Dropdown list * Start time (required): Time picker. * End time (required): Time picker * Date (required): Date picker * Content (required): input text (up to 512 characters long) * Note: text area (up to 512 characters long) | | 3 | Manager/scheduler input meeting information in update form.  Manager/scheduler click [Save] button | Meeting will be update on data. Appears “Success” message.  Back to meeting list  [**Alternative 2]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no meeting schedule in the data | Show label massager “There is no meeting schedule in the data”. | | 2 | Manager/scheduler click “Back” button | Back to meeting schedule list page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The meeting schedule must show correctly  -The page must have good design | | | |

***2.3.4 <User>Overview Use Case***



***2.3.4.1 < User> View shift schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP23 | | | |
| **Use Case No.** | UC\_HP23 | **Use Case Version** | 2.0 |
| **Use Case Name** | View shift schedule | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * User.   **Summary:**   * This use case allows user to view shift schedule.   **Goal:**   * User can view shift schedule.   **Triggers:**   * User click “Shift schedule” tag.   **Preconditions:**   * System had connected to database. * User login with role User.   **Post Conditions:**   * **Success:** System show shift schedule * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click tag [Shift schedule] on menu bar. | System move to interface, where shift schedule is showed | | 2 | User choice department on dropdown list.  User choice week on dropdown list. | System show shift schedule detail of one week being filtered by department.  [**Alternative 1]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no shift schedule in the data | Show label massager “This schedule is empty please select other schedule”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The shift schedule must show correctly  -The page must have good design | | | |

***2.3.4.2 < User> View meeting schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP24 | | | |
| **Use Case No.** | UC\_HP24 | **Use Case Version** | 2.0 |
| **Use Case Name** | View meeting schedule | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * User.   **Summary:**   * This use case allows user to view meeting schedule.   **Goal:**   * User can view meeting schedule.   **Triggers:**   * User click “Meeting schedule” tag.   **Preconditions:**   * System had connected to database. * User login with role User.   **Post Conditions:**   * **Success:** System show meeting schedule * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click tag [Meeting schedule] on menu bar. | System move to interface, where meeting schedule is showed  [**Alternative 1]** | | 2 | User click “In coming” tab  [**Alternative 2]**  [**Alternative 3]** | System show meeting schedule in the future. | | 3 | User click “View” button | System show detail of the meeting schedule. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no meeting schedule in the data | Show label massager “This schedule is empty please select other schedule”. | | 2 | User click “Quick search” tab | System show from date picker and to date picker to search meeting in range. | | 3 | User click “Pass” tab | System show meeting schedule in the pass. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The meeting must show correctly  -The page must have good design | | | |

***2.3.4.3 < User> View mission schedule***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP25 | | | |
| **Use Case No.** | UC\_HP25 | **Use Case Version** | 2.0 |
| **Use Case Name** | View mission schedule | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * User.   **Summary:**   * This use case allows user to view mission schedule.   **Goal:**   * User can view mission schedule.   **Triggers:**   * User click “Mission schedule” tag.   **Preconditions:**   * System had connected to database. * User login with role User.   **Post Conditions:**   * **Success:** System show mission schedule * **Fail:** System display “Error” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click tag [Mission schedule] on menu bar. | System move to interface, where mission schedule is showed  [**Alternative 1]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no mission schedule in the data | Show label massager “You don't join in any mission”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:**  **-**The mission must show correctly  -The page must have good design | | | |

***2.3.4.4 < User> Login***

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -** UC\_HP26 | | | |
| **Use Case No.** | UC\_HP26 | **Use Case Version** | 2.0 |
| **Use Case Name** | Login | | |
| **Author** | LinhTP | | |
| **Date** | 20/1/2018 | **Priority** | Normal |
| **Actor:**   * User.   **Summary:**   * This use case allows user to login to the system.   **Goal:**   * User can login to the system.   **Triggers:**   * User click “Login” tag.   **Preconditions:**   * System had connected to database. * User already has account in system.   **Post Conditions:**   * **Success:** User could login to the system successfully * **Fail:** System display Error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User access to the system. | System show login page and display:   * Username: textbox * Password: textbox * Login: button | | 2 | User input information and click “Login” button | System check information and lead to “Hospital-portal” page  [**Alternative 1]** |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Account is not exist in system | Show label massager “Invalid username or password”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Disconnect to server | System show message the “Connection to the server has been lost. Please check your internet connection or try again later. ” when the internet is lost |   **Relationships:** N/A  **Business Rules:** N/A | | | |

**3. Software System Attribute**

**3.1 Usability**

* Ease-of-use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its General-users.
* 95% of all users will be satisfied with the usability of the product.
* User: must be able to access the change in progress without prior knowledge of the application.
* Time:
* 100% of the users will be able to view schedule less than 10 minutes without requiring assistance.
* 100% of the users will be able to search information and accept information easily in less than 3 minutes.
* Conformity: this system must conform to friendly interface.

**3.2 Reliability**

* Accuracy:
  + Process: All data before insert to database will be validation.

**3.3 Availability**

* The system shall be available 24 hours per day, 360 days per year.
* The system shall not lose any transaction data.
* The system shall log in a user within 5 seconds.

**3.4 Security**

* Each role of user has a specific permission to interact with system.
* System always checks authorization and authentication before doing anything.

**3.5 Maintainability**

* System is design by two distinct part (front–end: angular, back-end: java) so it is very easy to maintain and extend.

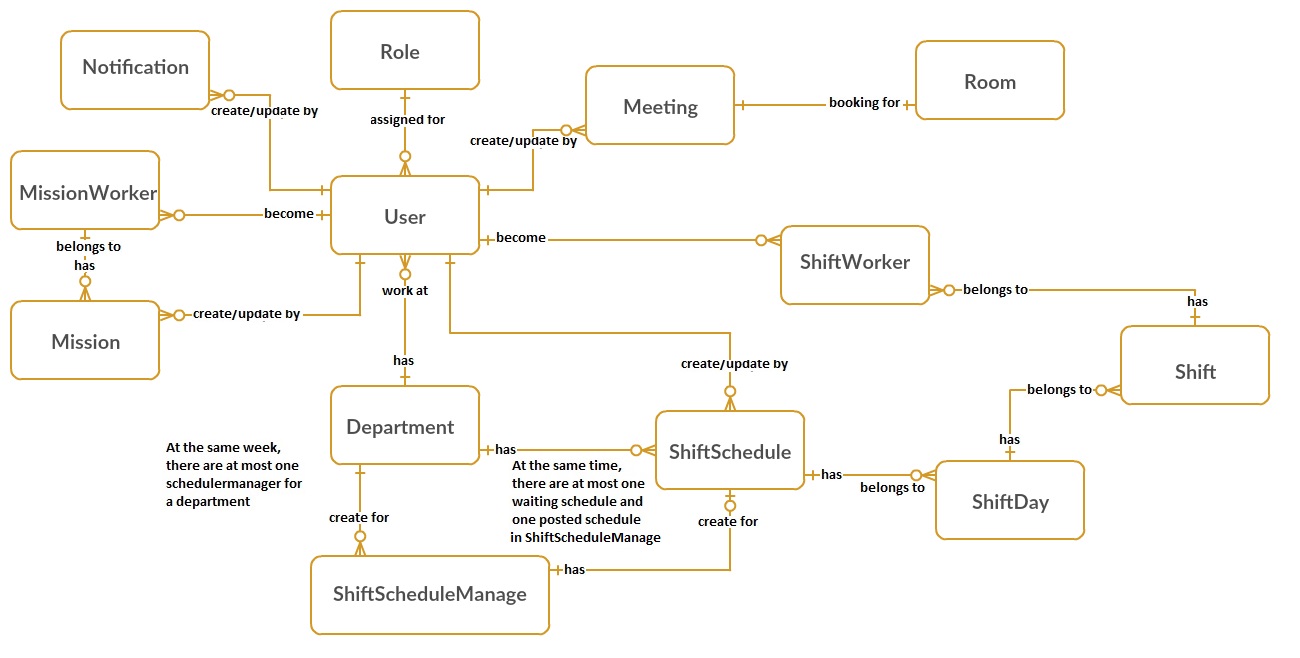
**3.6 Portability**

* Web application can be run on Chrome browser version 58 or later.

**3.7 Performance**

* Respond time for a transaction: average 1 seconds
  + System must handle 100 transactions/second”
* System must have less than 1hr downtime/3 months” supportability
* This software should run well on a PC or similar hardware (laptop, smart phone, tablet).
* It can work well with both Windows and Mac OS.
* The interface will be implemented as website and can be opened by most popular web browsers like IE>6.0, Firefox 4.0, Chrome 5.0.
* In standard workload, the CPU usage shall be less than 50%, leaving 50% for background jobs.
* Production of a simple schedule shall take less than 20 seconds for 95% of the cases.

**4. Conceptual Diagram**

****

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

***Table 11: Conceptual Diagram Data Dictionary***

**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.*

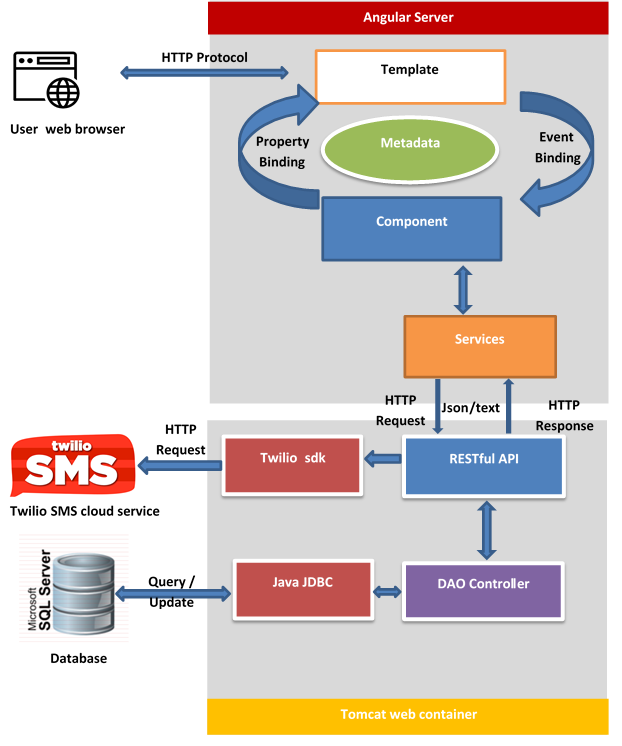
*- The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.*

*- 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**

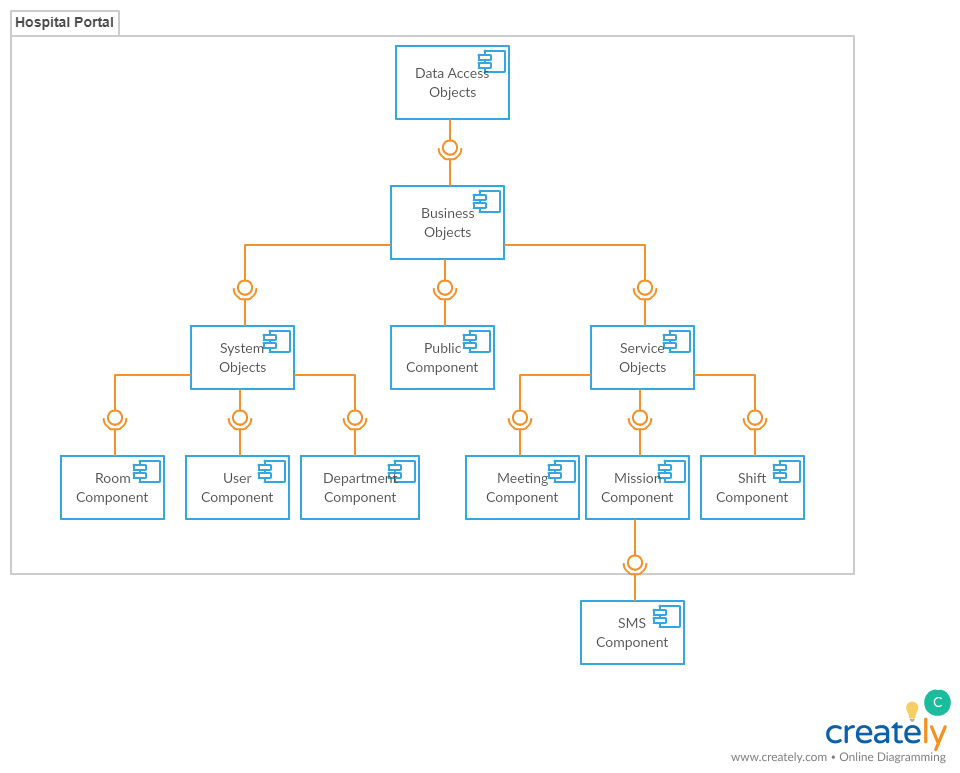
****

**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.

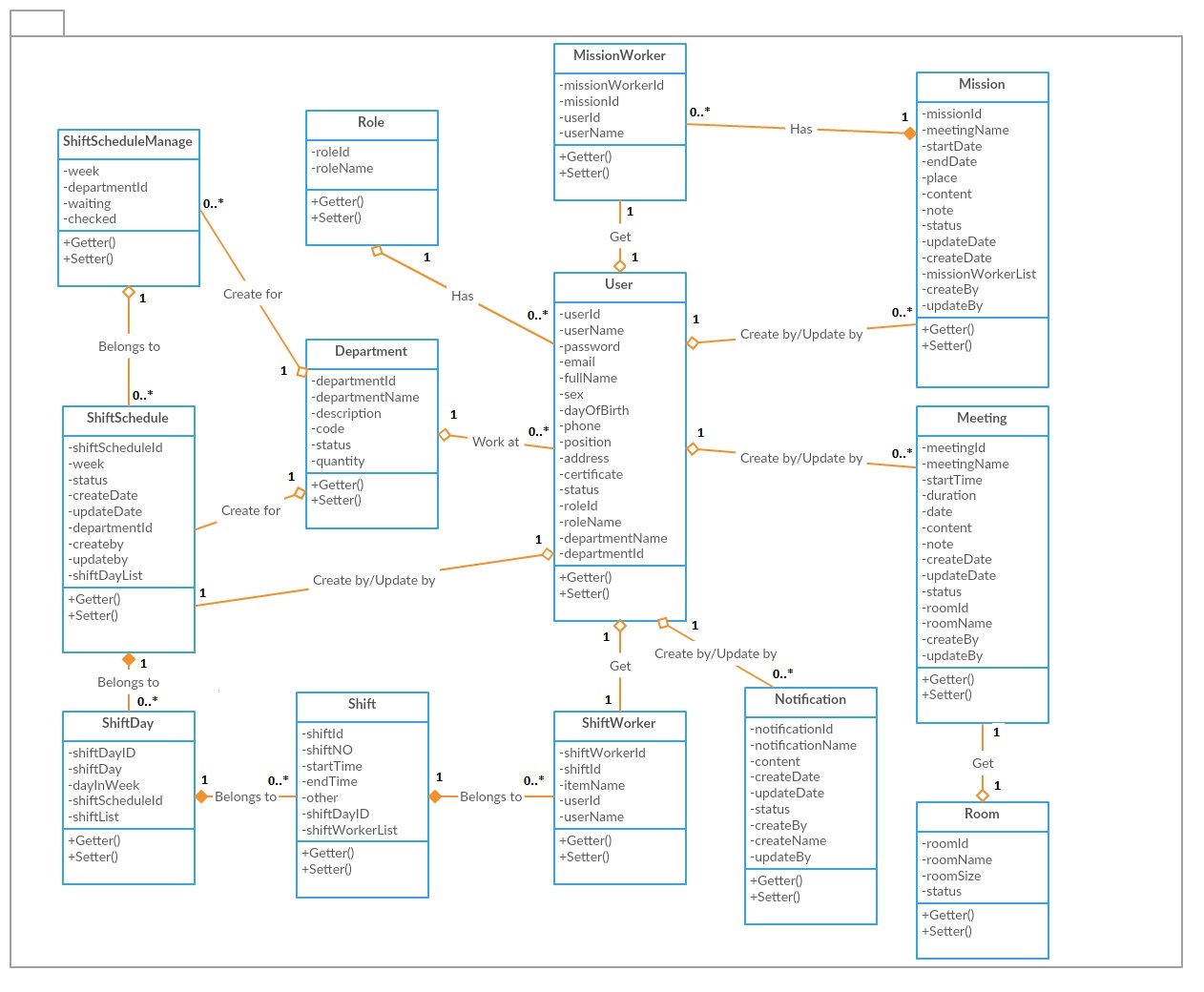
**3. Component Diagram**

**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*

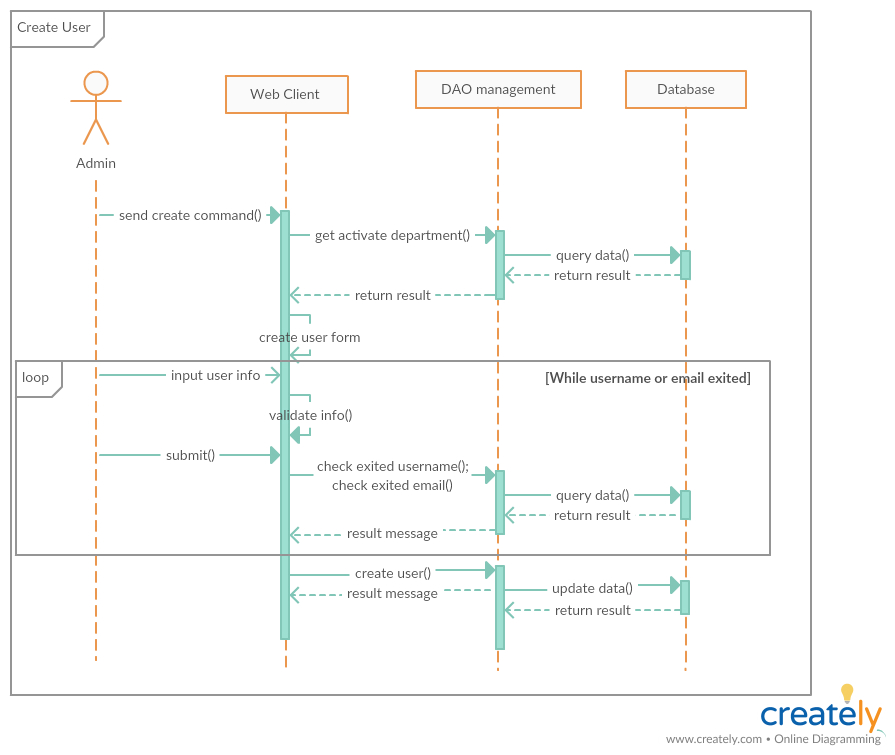
|  |  |  |  |
| --- | --- | --- | --- |
| ***Attribute*** | ***Type*** | ***Visibility*** | ***Description*** |
| RoomId | int | Private | Unique identifier of room |
| RoomName | string | Private | Room name |
| RoomSize | int | Private | Number of seats in room |
| Status | int | Private | Status of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

*Method*

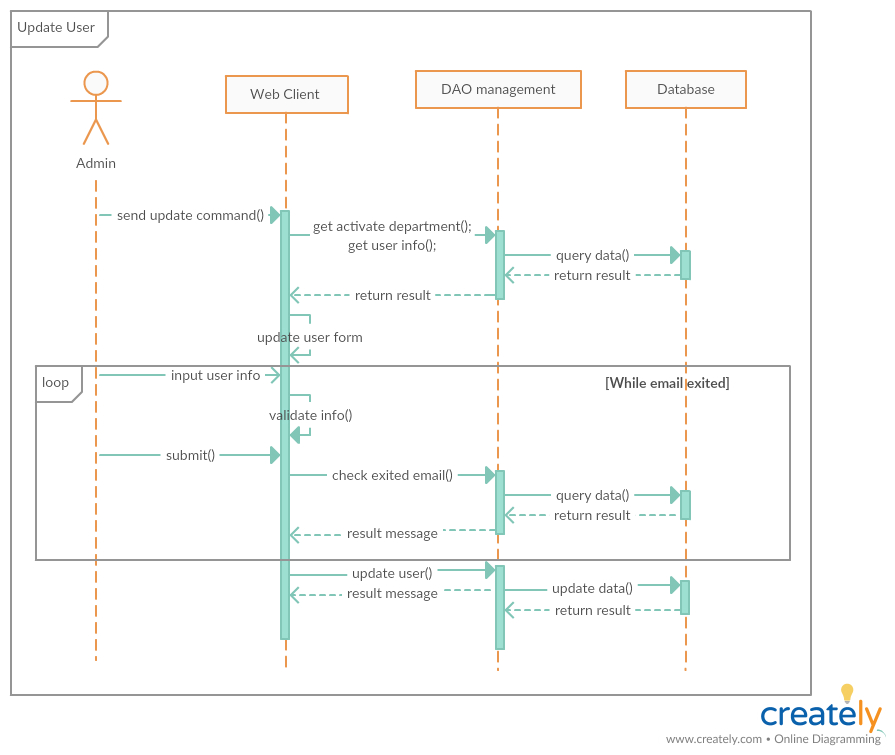
|  |  |  |  |
| --- | --- | --- | --- |
| ***Method*** | ***Return type*** | ***Visibility*** | ***Description*** |
| *Getter* | *Attribute type* | *Public* | *Get attribute value* |
| *Setter* | *Void* | *Public* | *Set value of attribute* |

**4.3 Interaction Diagram**

***4.3.1 Create user***

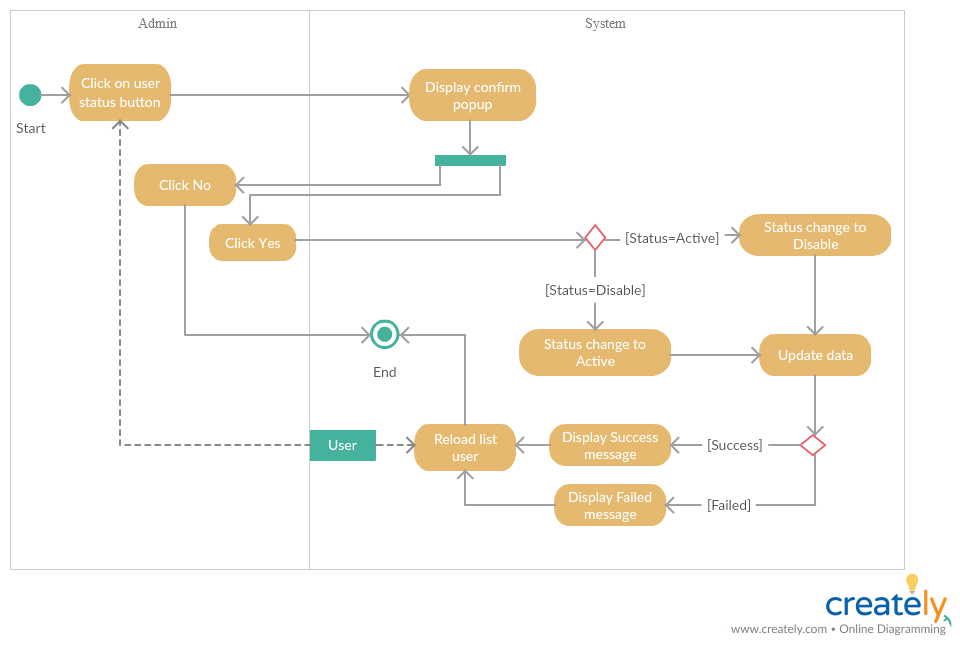
*****Summary: this diagram show process of admin creates new User.*

***4.3.2 Update user***

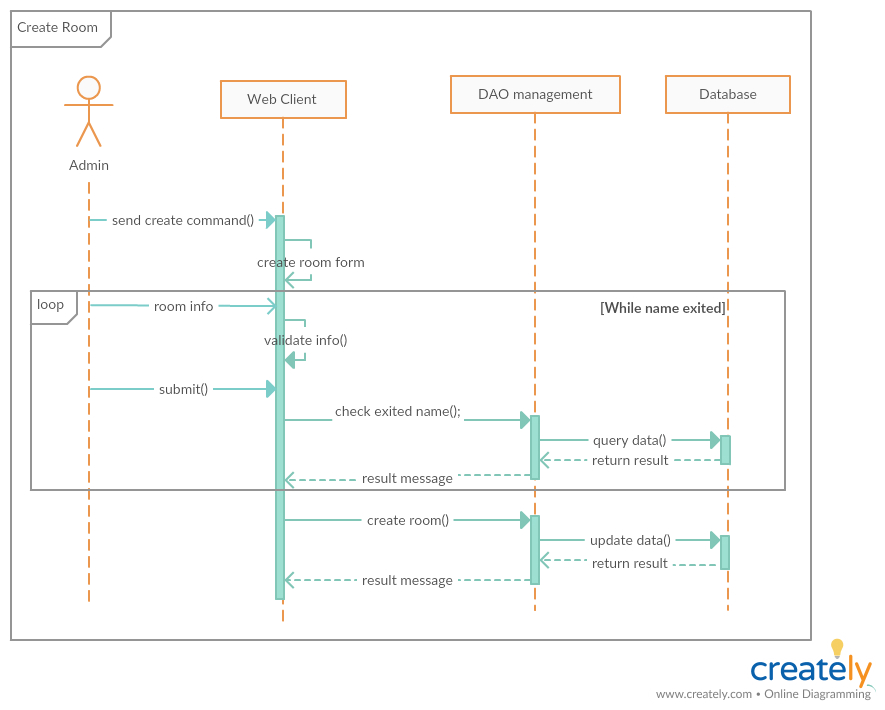
*****Summary: this diagram show process of admin update User*

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

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

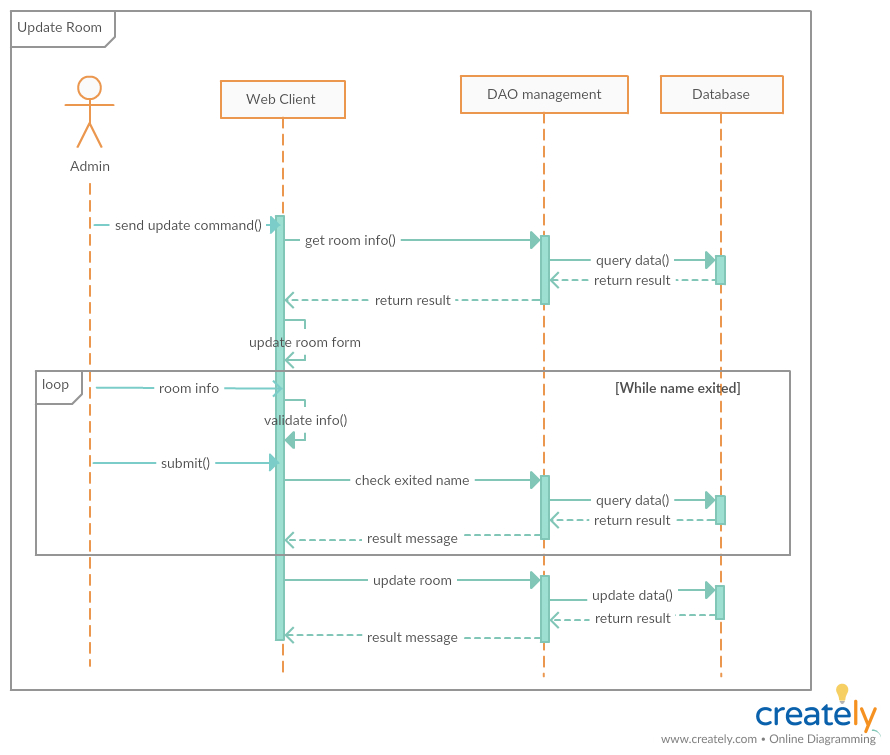
******

***4.3.4 Create room***

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

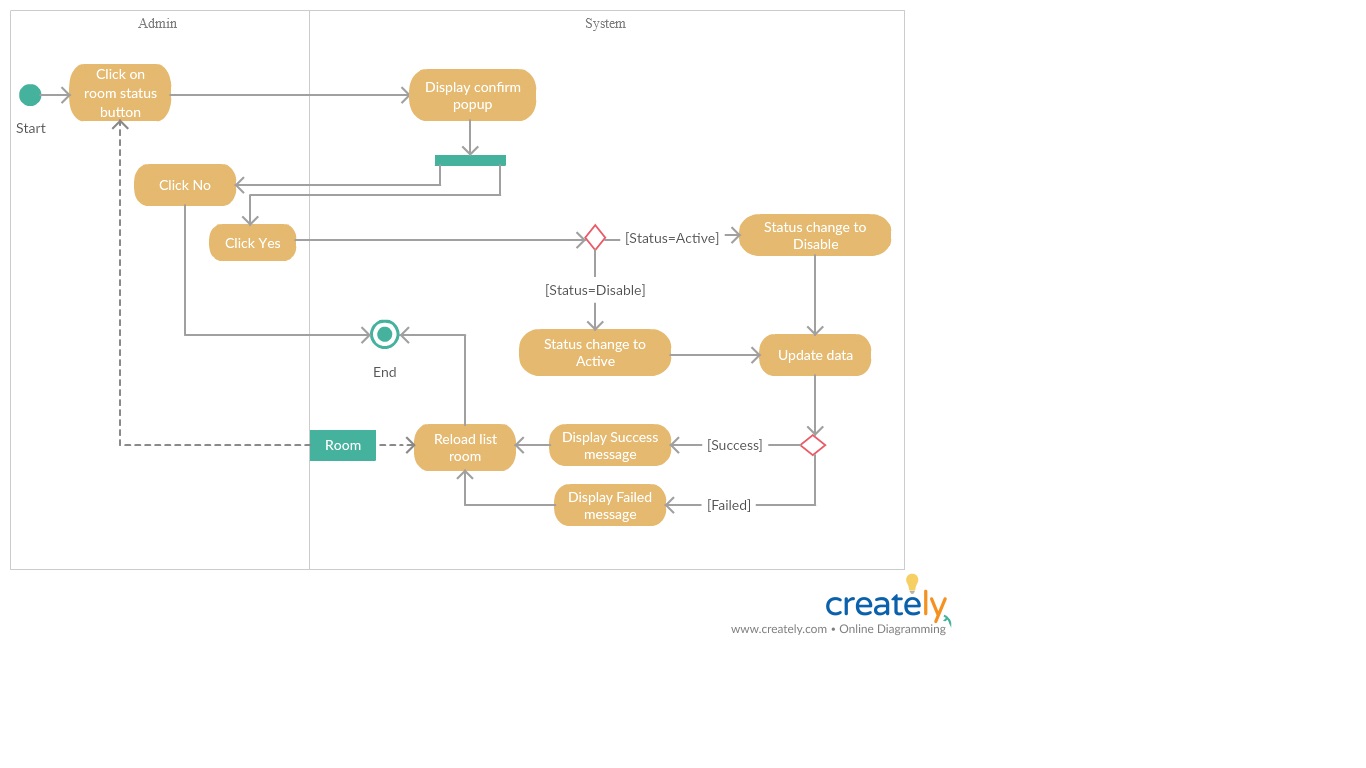
***4.3.5 Update room***

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

******

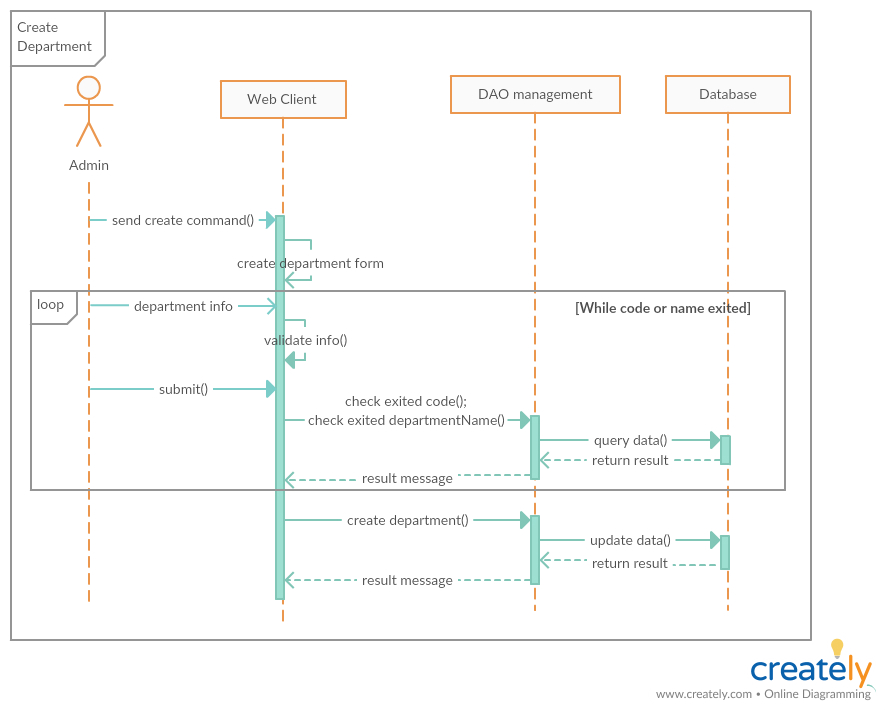
***4.3.6 Activate/Deactivate room:***

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

**

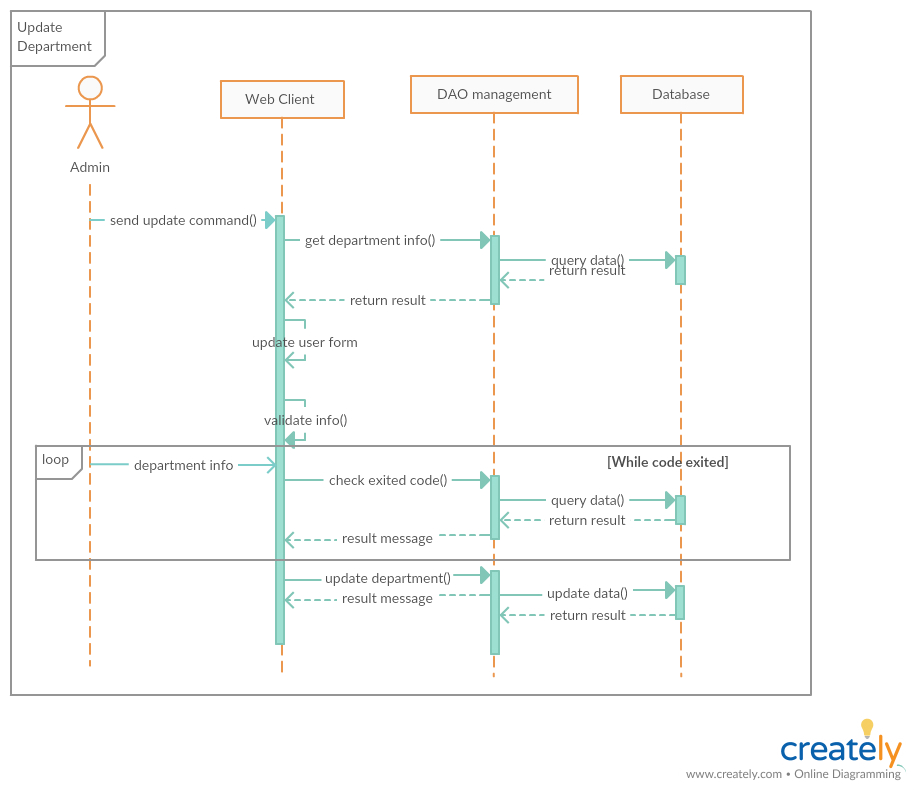
***4.3.7 Create department***

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

**

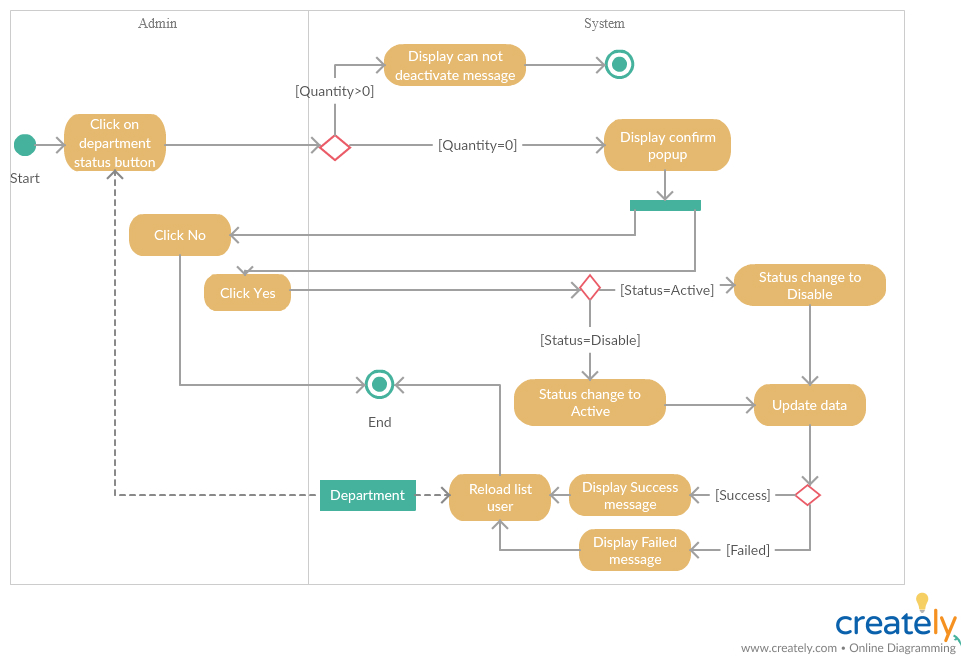
***4.3.8 Update department***

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

**

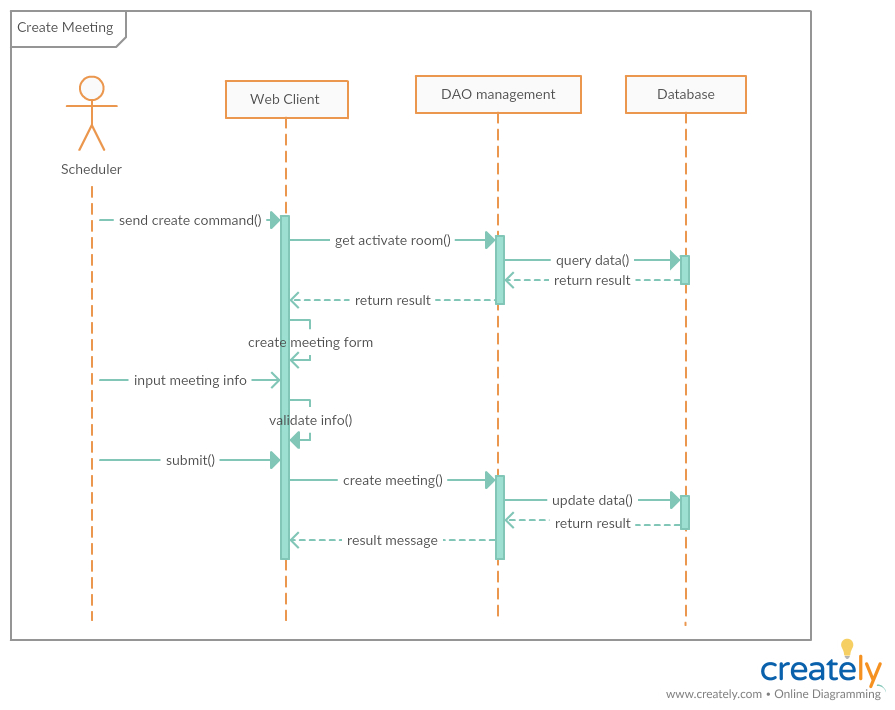
***4.3.9 Activate/Deactivate department***

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

****

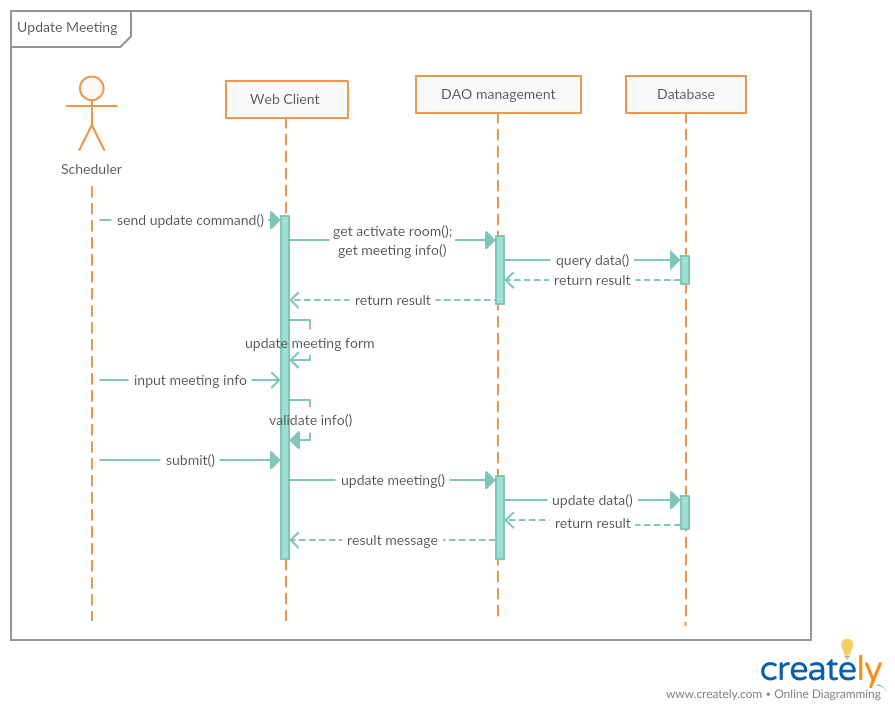
***4.3.10 Create meeting***

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

**

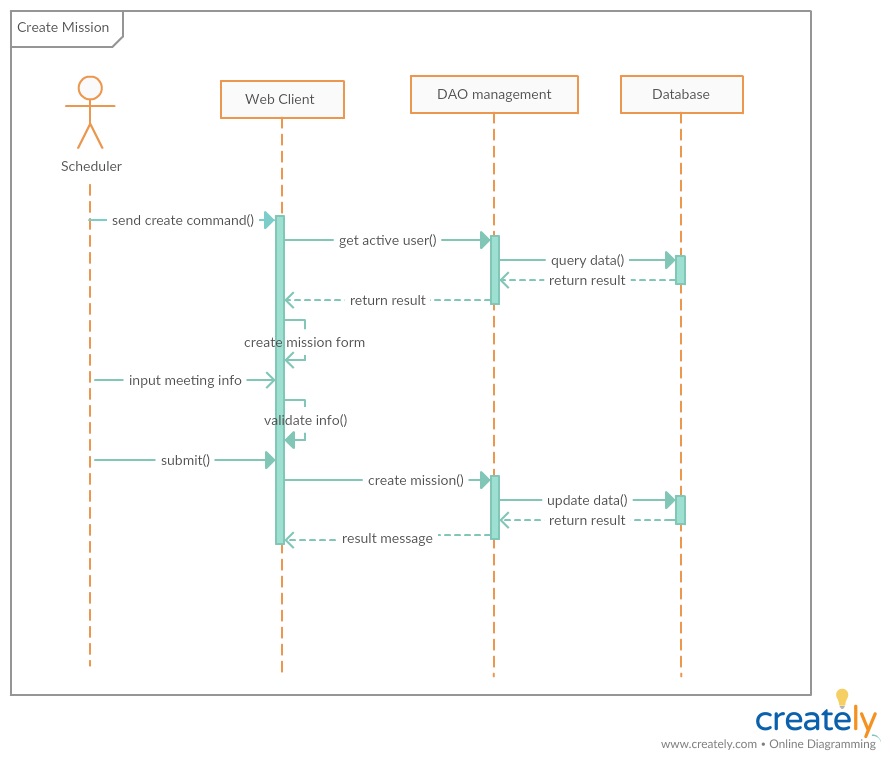
***4.3.11 Update meeting***

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

****

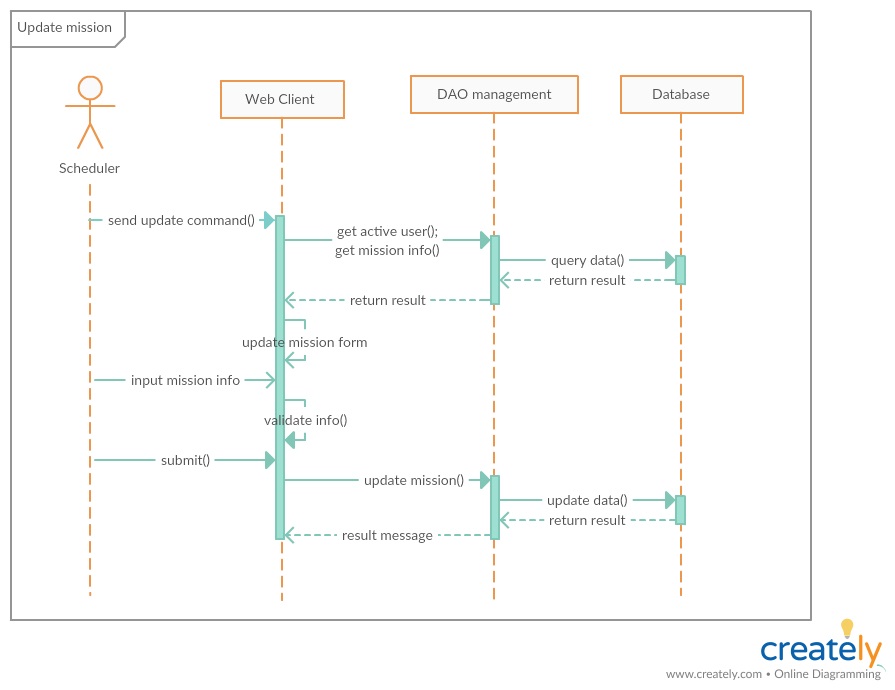
***4.3.12 Create mission***

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

****

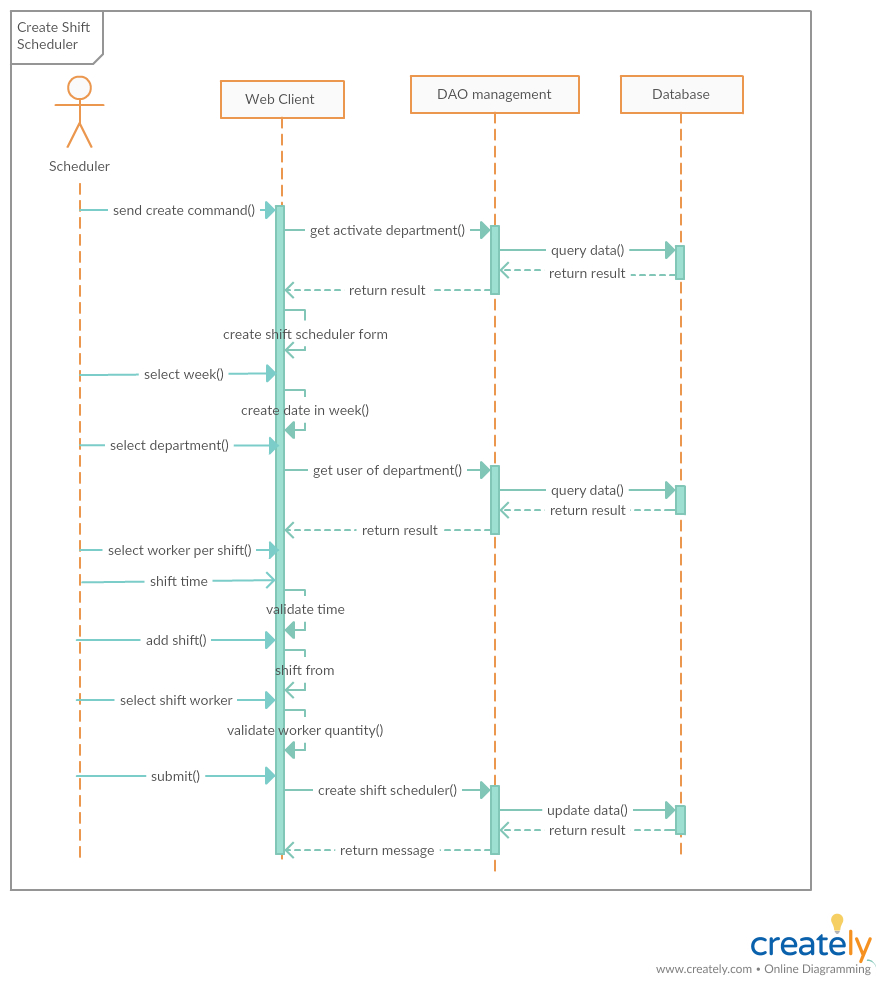
***4.3.13 Update mission***

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

****

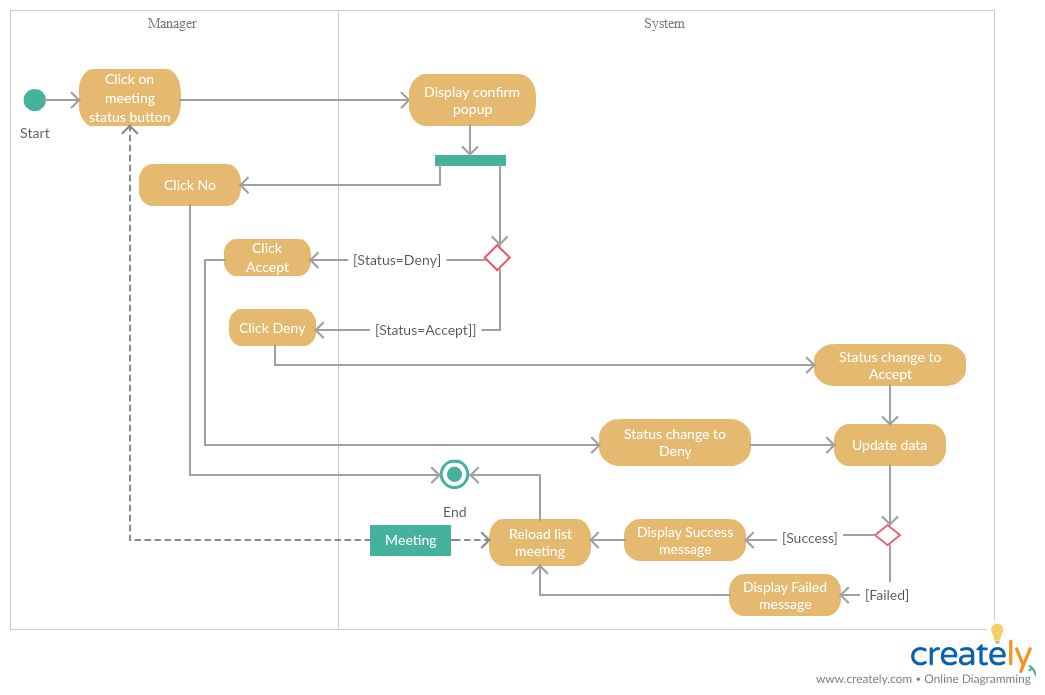
***4.3.14 Create shift schedule***

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

****

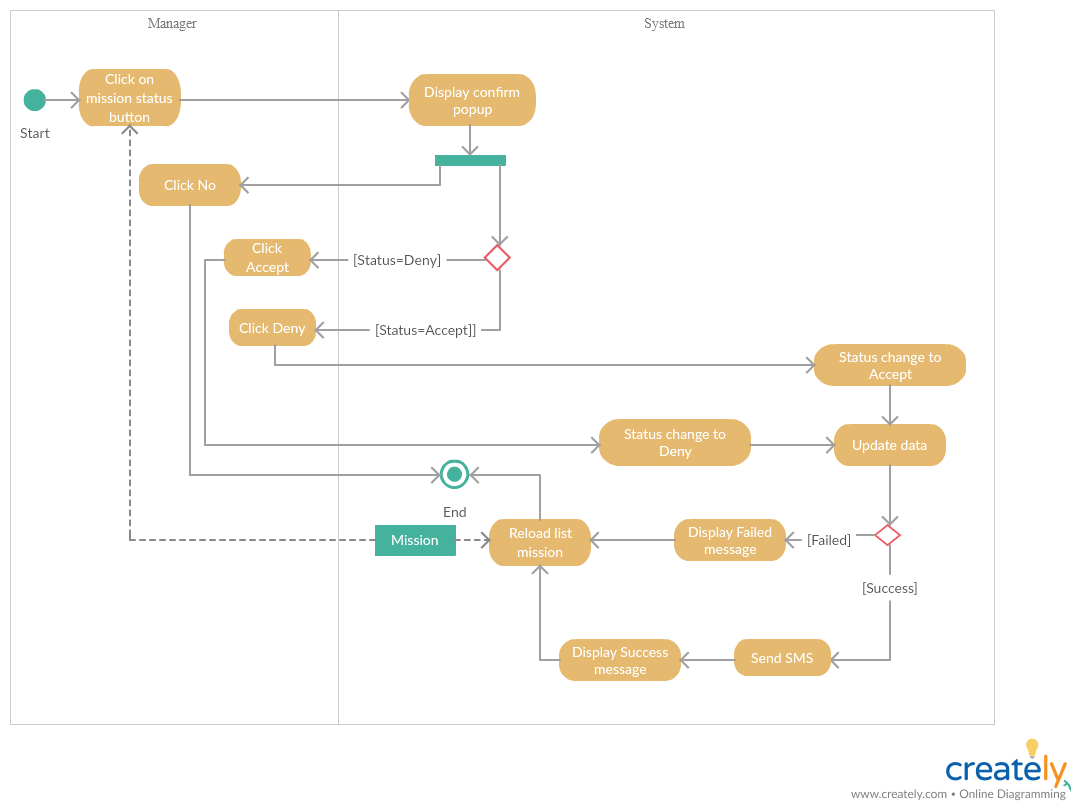
***4.3.15 update checked meeting status***

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

**

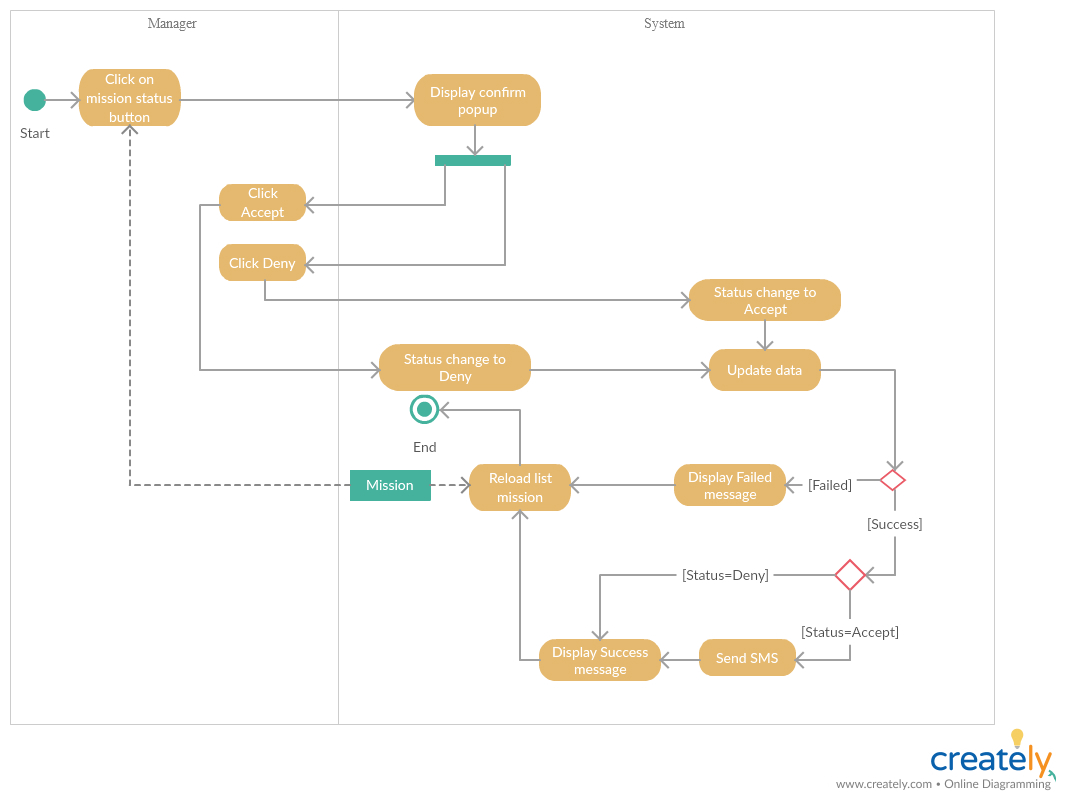
***4.3.16 update checked mission status***

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

****

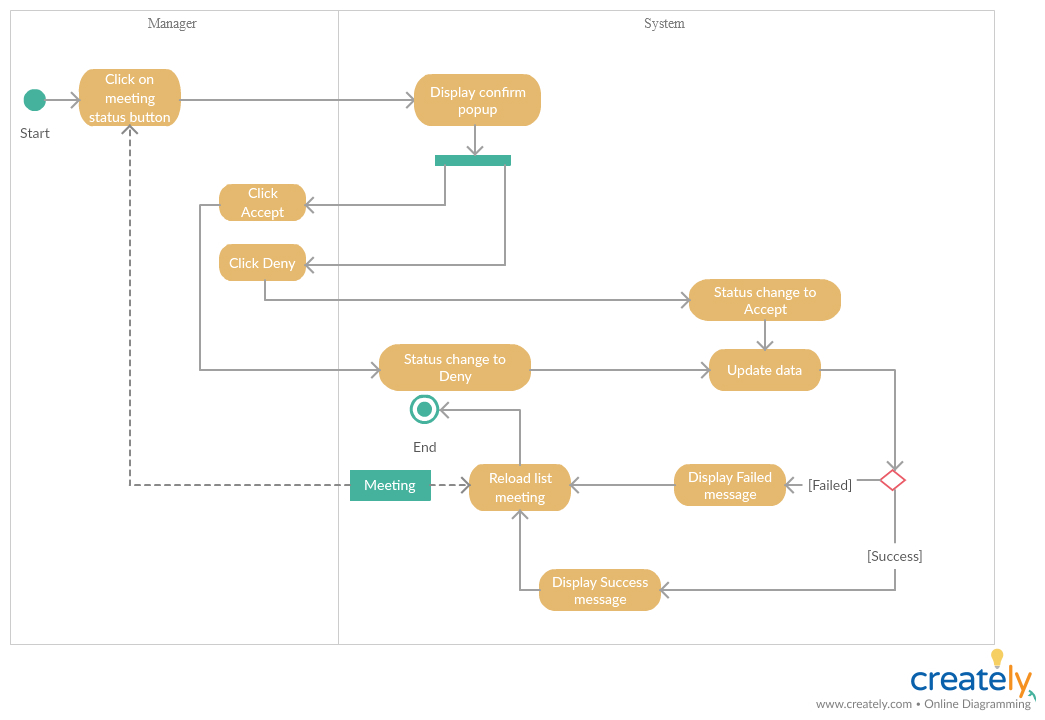
***4.3.17 check waiting mission status***

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

****

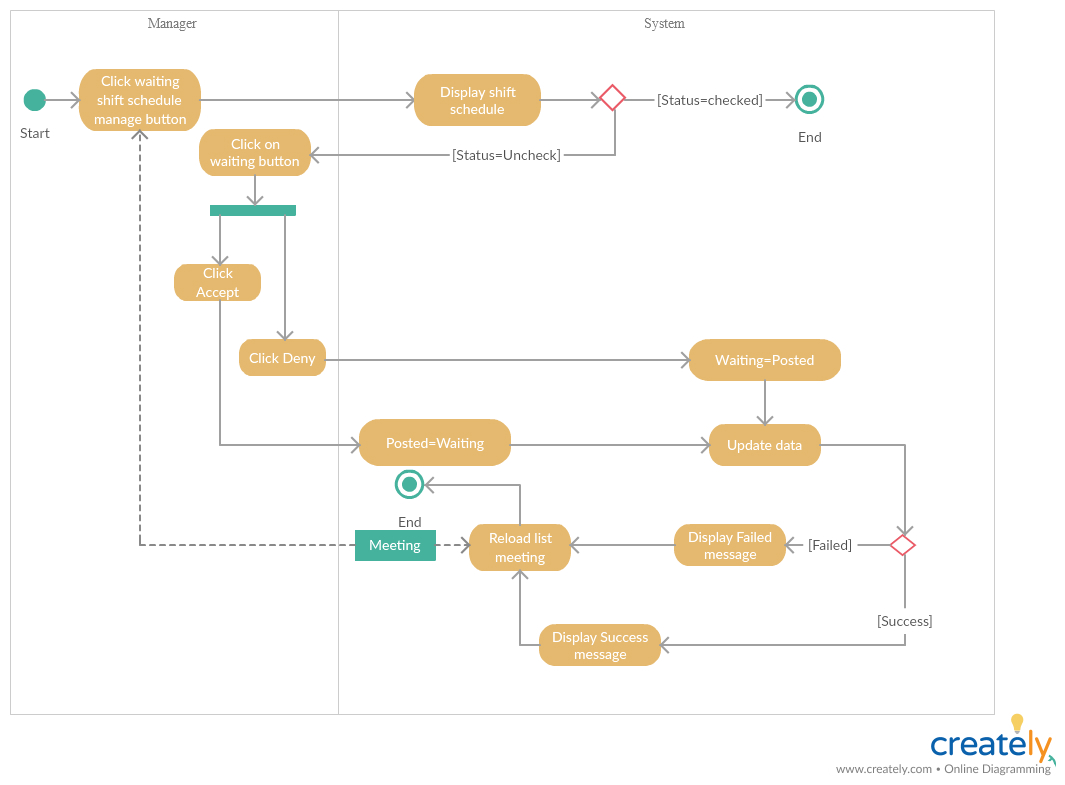
***4.3.18 check waiting meeting status***

*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**

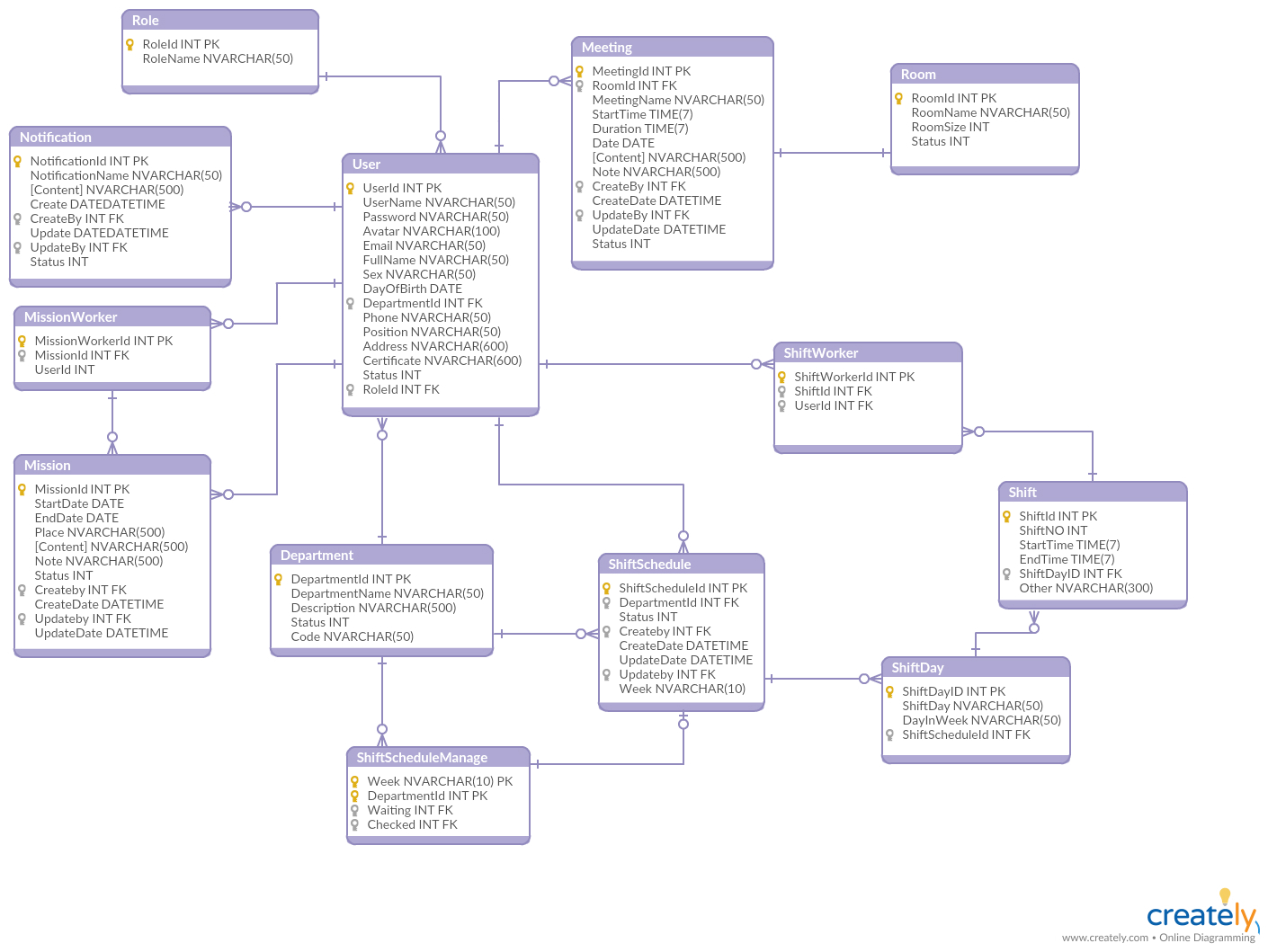
|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Description** | **Output** | **Exception** |
| getListShiftSchedule() | Get list shift schedule | JSON of shift schedule | JsonException |
| getWaitingShiftSchedules() | Get shift schedule which stauts is waiting by id | JSON of shift schedule | 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** | **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.

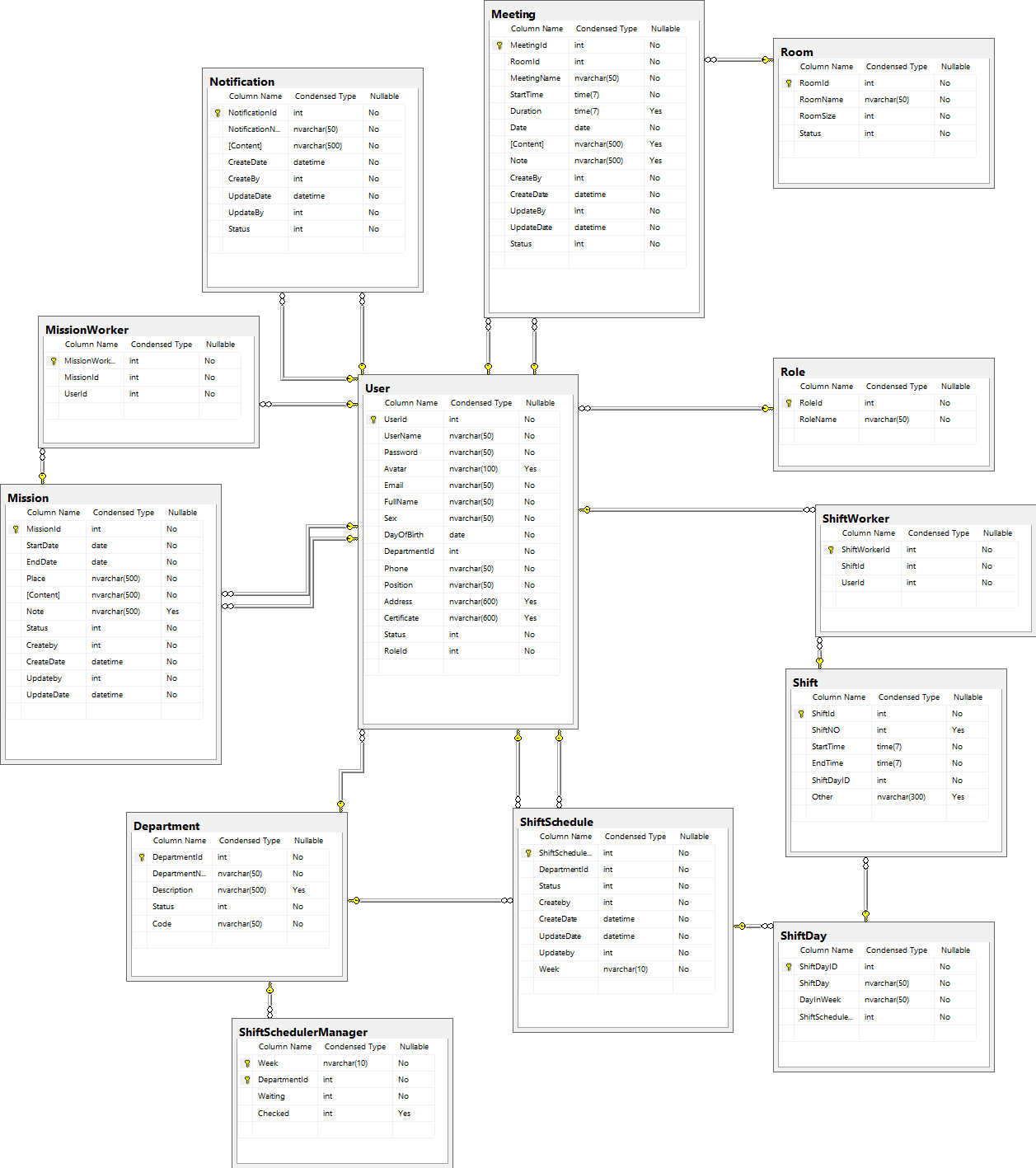
**1.2 Test Approach**

* 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**

|  |  |
| --- | --- |
| **Database Relationship Diagram: describe content of all table** | |
| **Table Name** | **Table 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. |

**3. Performance Measures**

***3.1 Clustering Performance***

***N/A***

**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 |

**F. Software User’s Manual**  
**1. Installation Guide**  
**1.1 Setting up environment at server side**

The following software must be installed into the server machine for user to run website smoothly as well as be able to check product effectively:

* + 1. **Hardware requirements**
* Here is our minimum hardware recommendation:
  + CPU: Quad core 2GHz+ CPU
  + RAM: 2GB or more of RAM
  + Minimum database space: 10GB
    1. **Software requirements**

The server or its components (core server, language servers) require a computer with a supported operating system. The server requires a compatible web browser and the Java Development Kit (JDK). You can either use Oracle Java SE 8 JDK x64 or IBM Java 8 SDK 64-bit.

**1.2 Deployment at server side**

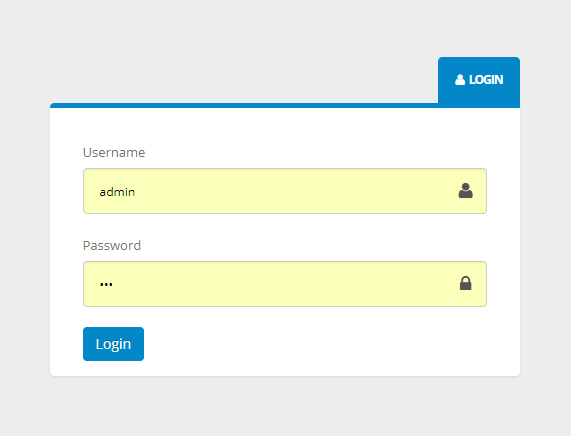
|  |
| --- |
| **Con trong** |

**1.2.1 Prepare deployment package**  
**1.2.2 Configure Server before deploy**  
**1.2.3 Deploy web application on server**  
**1.3 Setting up the environment at client side**  
**1.3.1 Setting up for computer**

**2. User Guide**

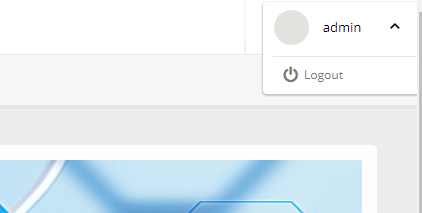
**2.1 For User:**

**2.1.2 Login**

****

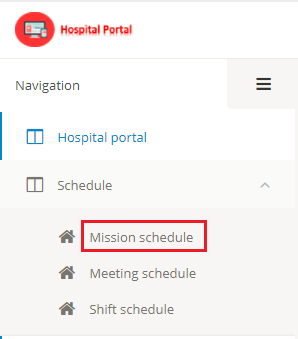
When you access to our website, you will see this login form. Please type your username and password then click on “Login” button.

**2.1.2 Logout**

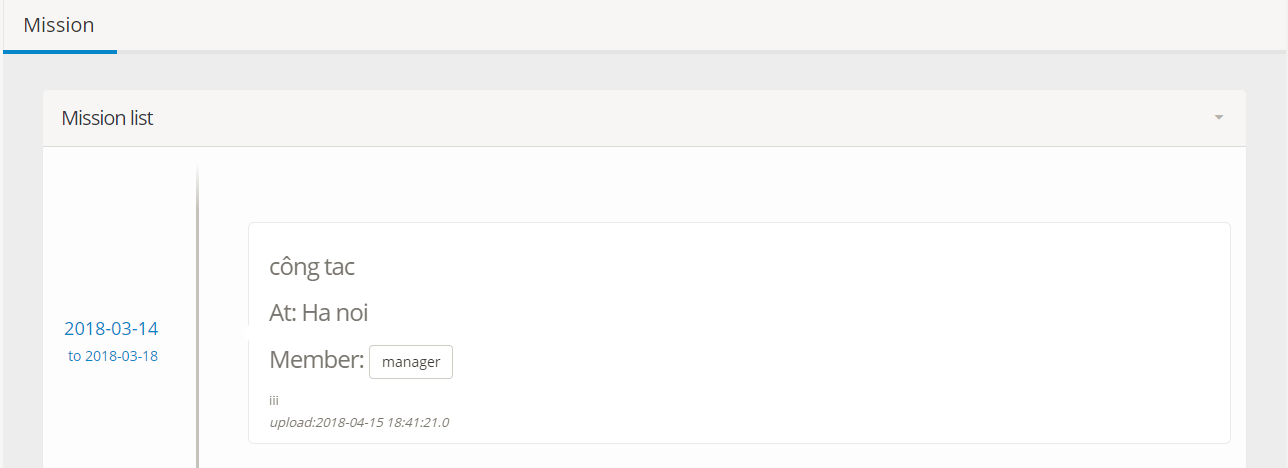


If you don’t want to be use this site anymore, you can logout of our system by clicking in “${username}” pop up button, choosing “Logout option”

**2.1.3 View mission schedule**

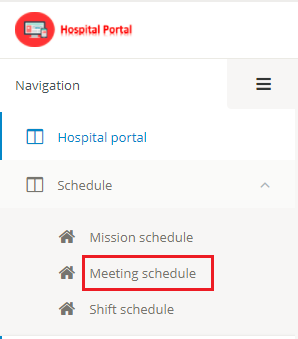


If you want to view mission schedule, please click “Schedule” on menu bar and then choose “Mission schedule” option.

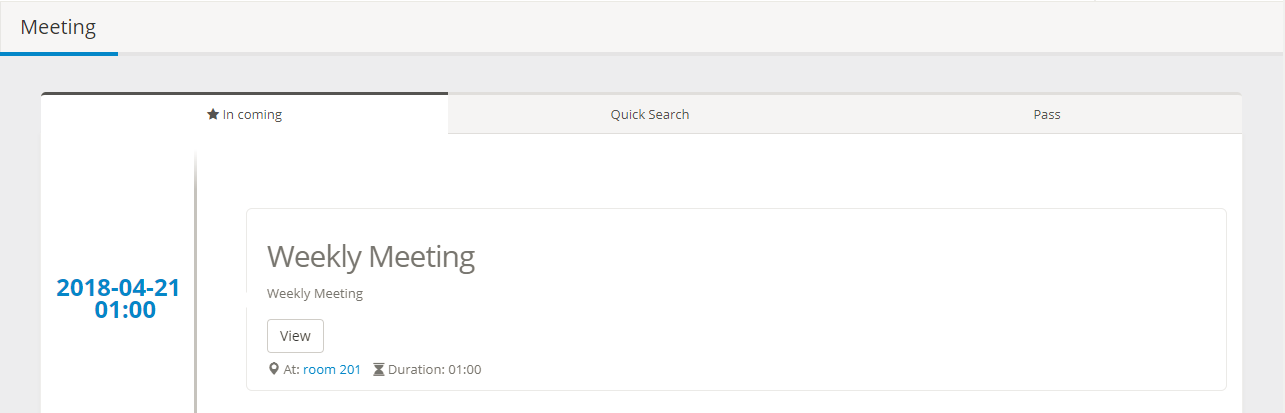


Mission list will show nearby.

**2.1.4 View meeting schedule**

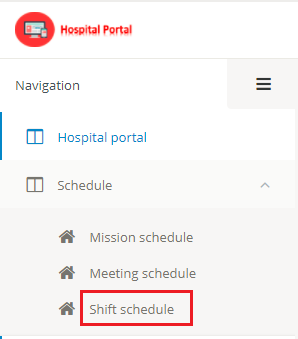


If you want to view meeting schedule, please click “Schedule” on menu bar and then choose “Meeting schedule” option.

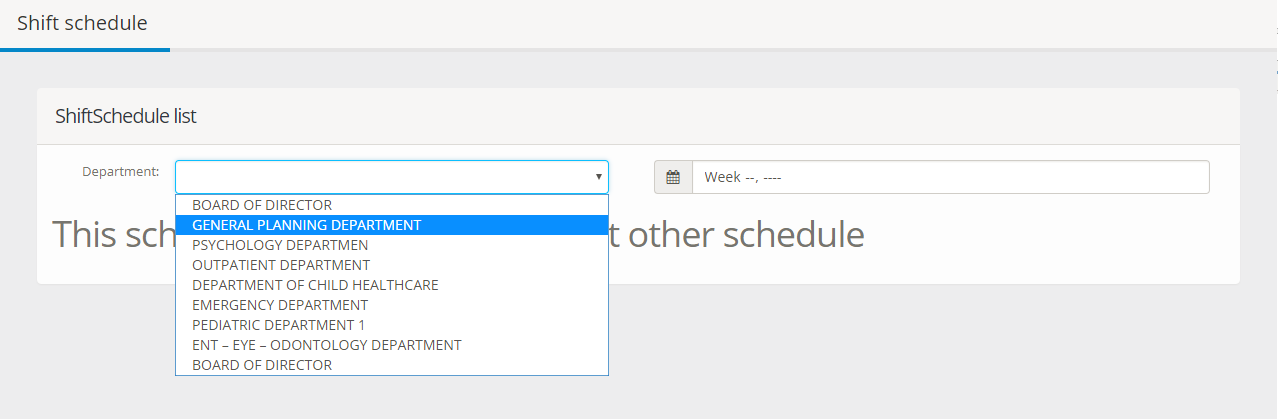


Meeting list will show nearby.

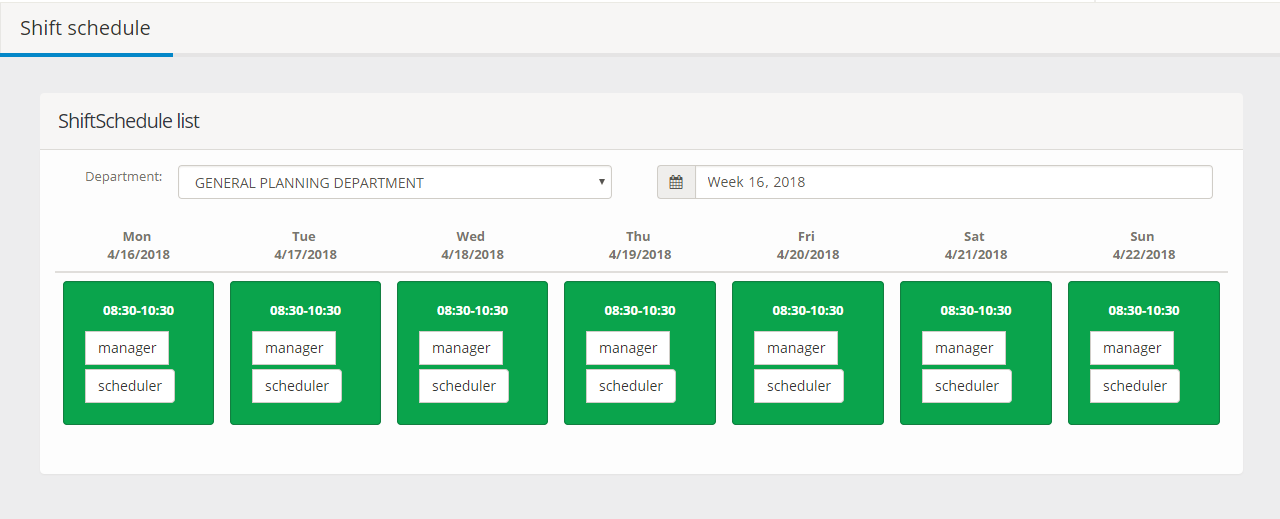
**2.1.5 View shift schedule**

****

If you want to view shift schedule, please click “Schedule” on menu bar and then choose “Shift schedule” option.



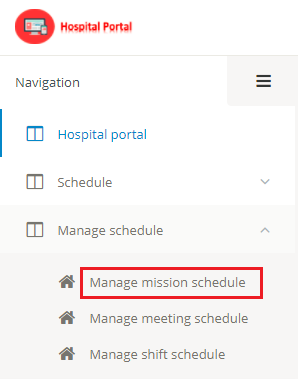
Second choose department to view shift schedule.



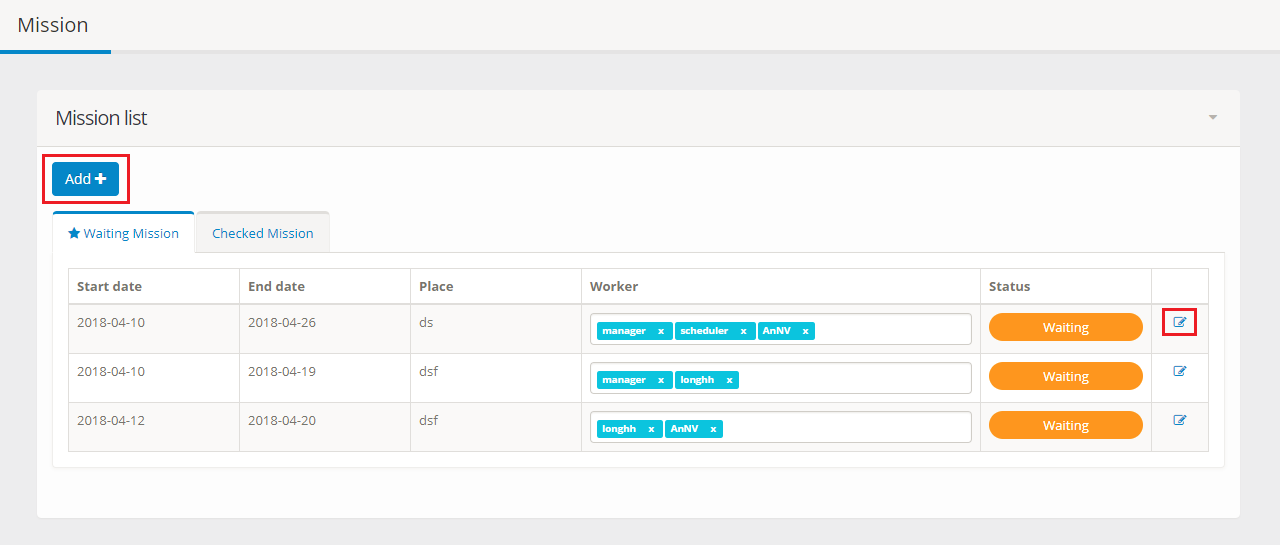
Then choose week, the shift schedule will show below

**2.2 For Scheduler:**

**2.2.1 Create/Update mission schedule**

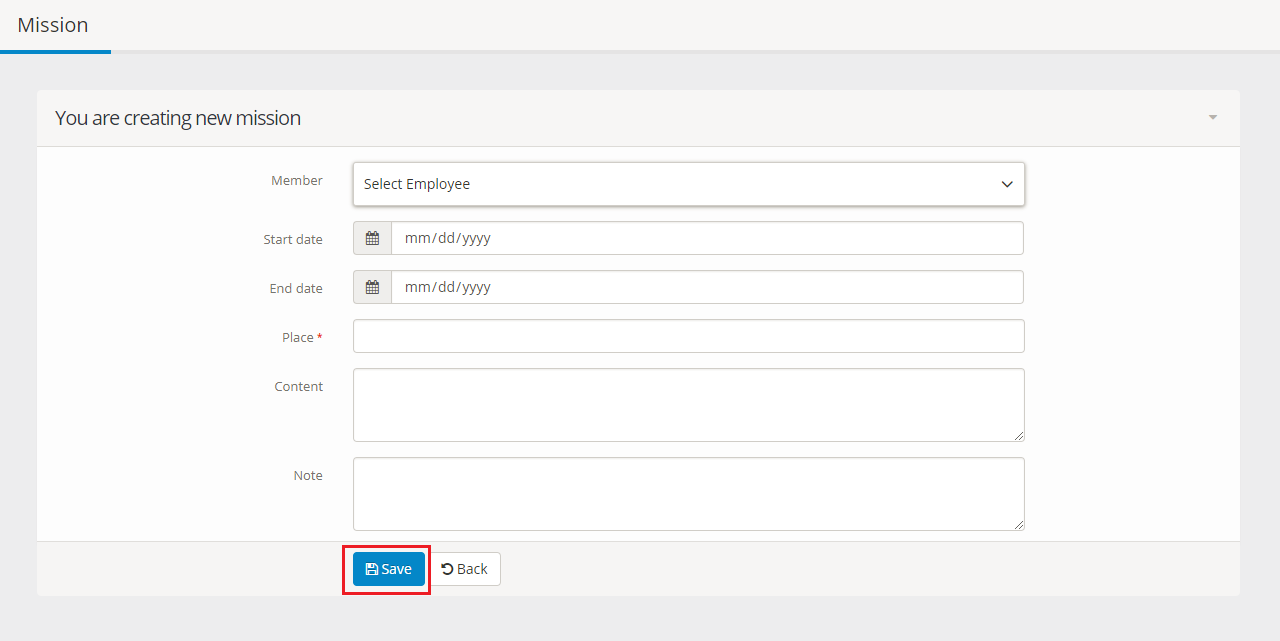
****

If you want to manage mission schedule, please click “Manage schedule” on menu bar and then choose “Manage mission schedule” option.

****

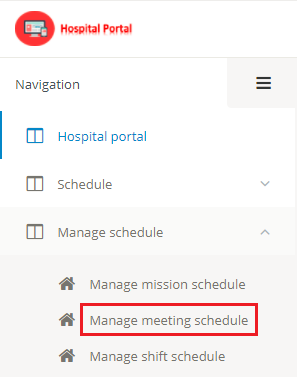
Click “Add” button on upper left corner of manage mission page to create new mission.

Click “View” icon on the right to update mission.

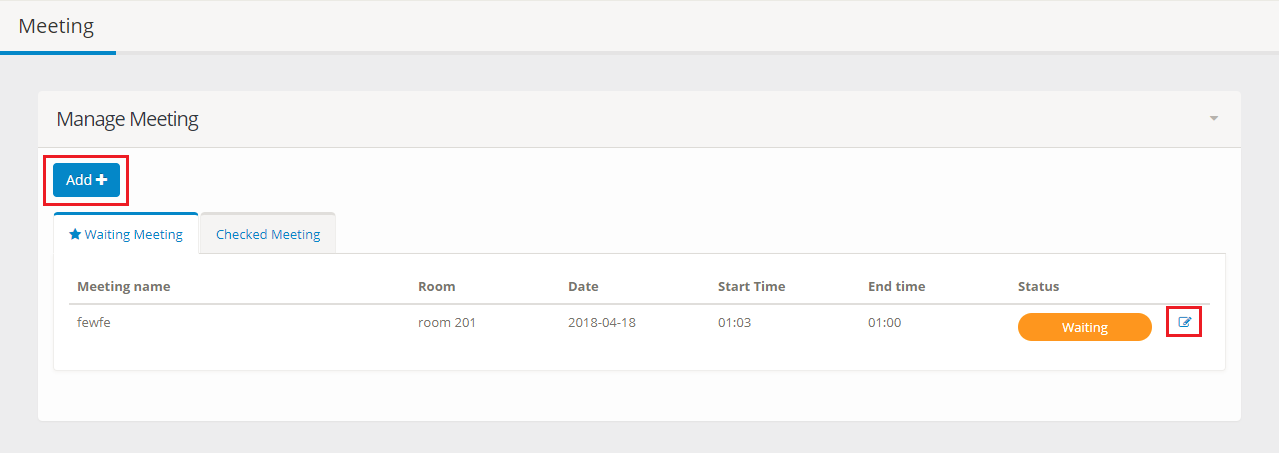
****

Please fill in all text box and click “Save” button

**2.2.2 Create/Update meeting schedule**

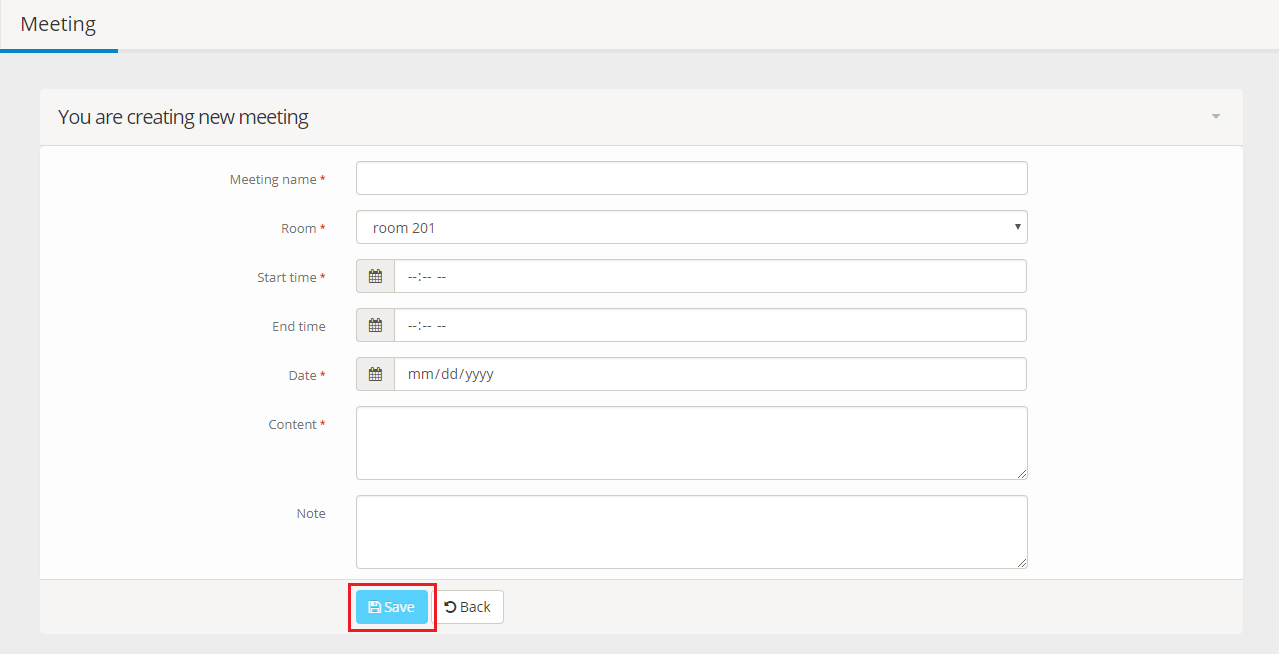
****

If you want to manage meeting schedule, please click “Manage schedule” on menu bar and then choose “Manage meeting schedule” option.



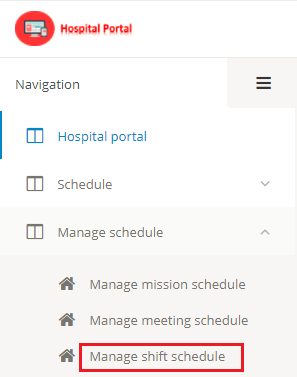
Click “Add” button on upper left corner of manage mission page to create new meeting.

Click “View” icon on the right to update meeting.

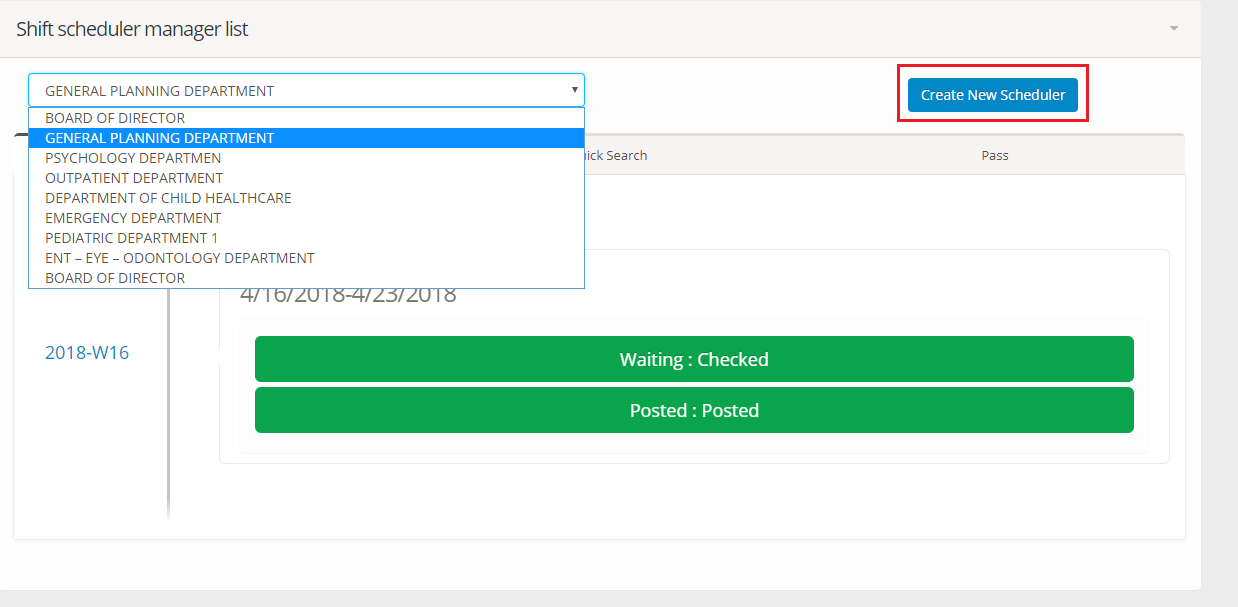


Please fill in all text box and click “Save” button

**2.2.3 Create shift schedule**

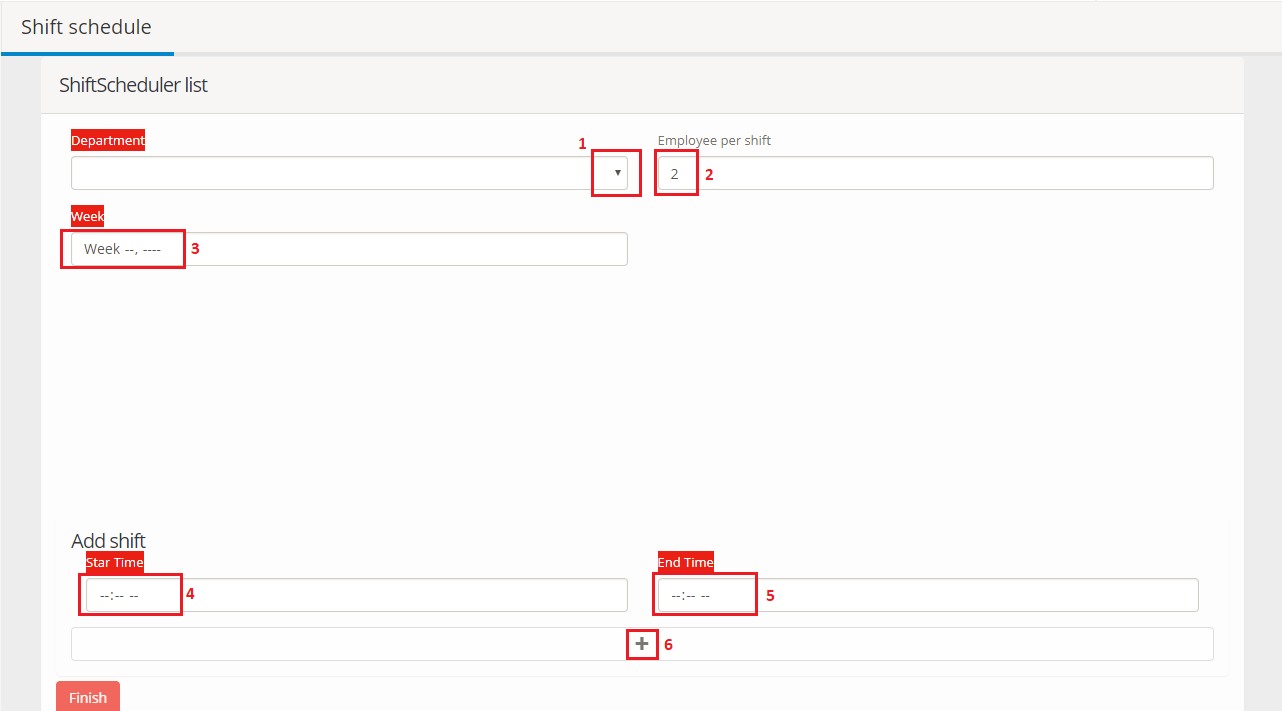
****

If you want to manage shift schedule, please click “Manage schedule” on menu bar and then choose “Manage shift schedule” option.

****

Please choose one department to view in coming shift schedule filter by department.

Click “Create new scheduler” to create new shift schedule.



Step1: Choose department

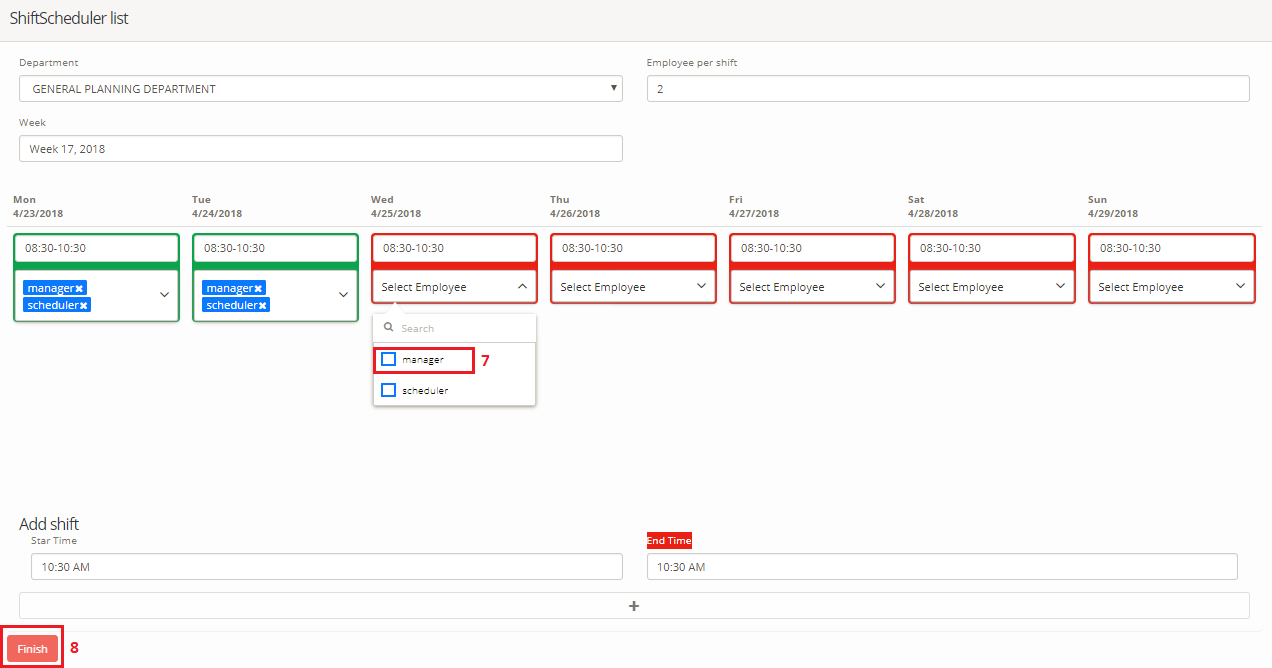
Step2: Input number of employees in one shift

Step3: Choose week

Step4: Input start time of one shift

Step5: Input end time of one shift

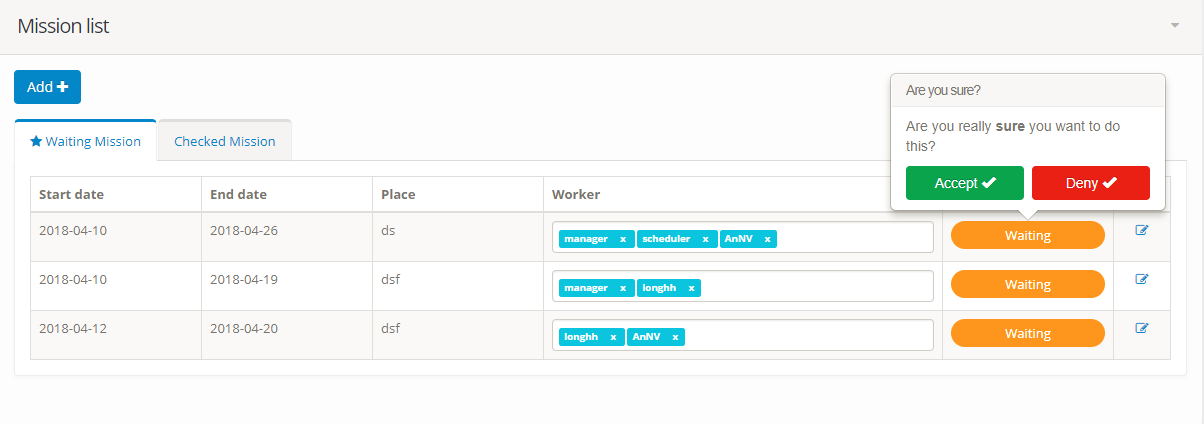
Step6: Click “+” icon to create one shift



Step7: Choose exactly number in “Employee per shift”, system will change to green when it is valid.

Step8: Click “Finish” button to create new department.

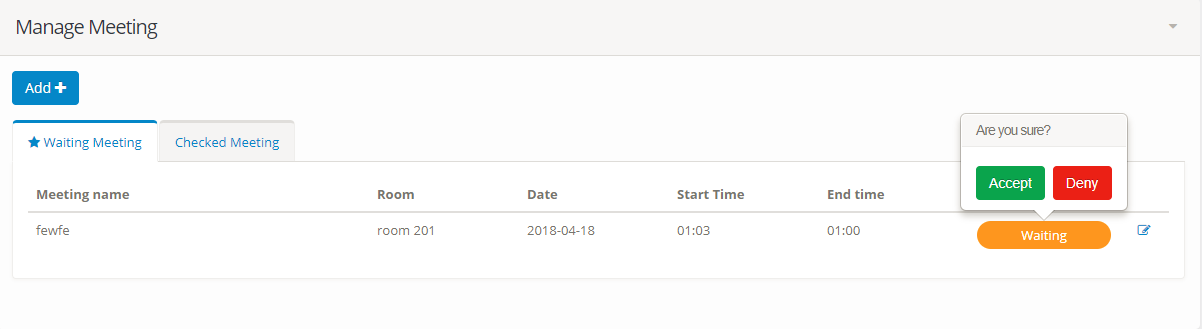
**2.2.4 Approved mission schedule**



In show mission list screen, click “Waiting” button.

Click “Accept” or “Deny” to approve mission.

**2.2.5 Approved meeting schedule**



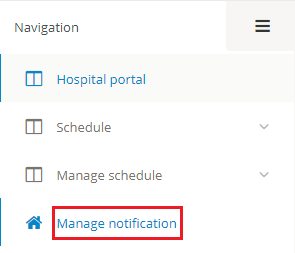
In show mission list screen, click “Waiting” button.

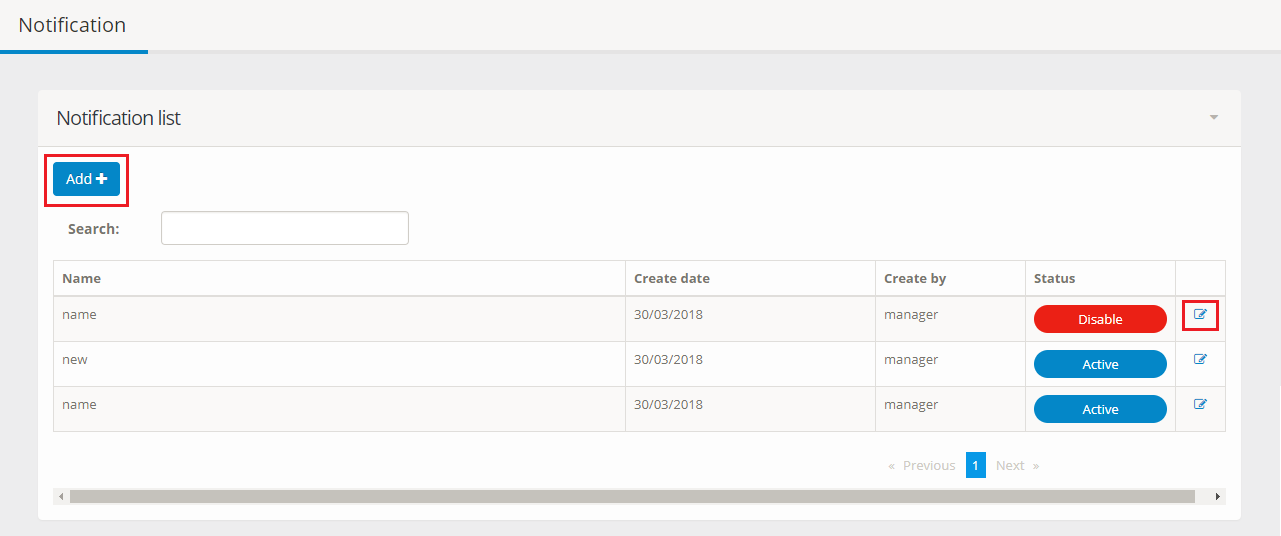
Click “Accept” or “Deny” to approve mission.

**2.3 For Manager:**

**2.3.1 Create/Update notification**

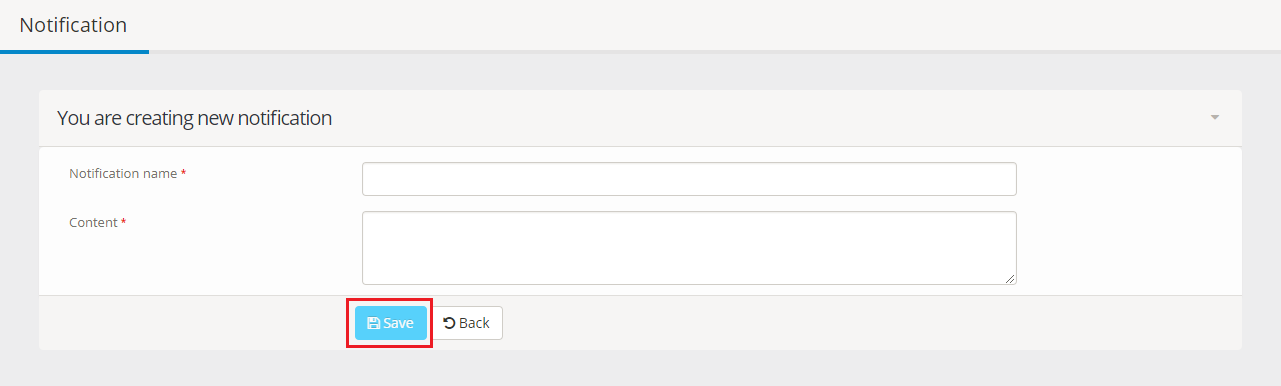
If you want to manager notification, please click “Notification” on menu bar



****

Click “Add” button on upper left corner of notification page to create new notification.

Click “View” icon on the right to update notification.

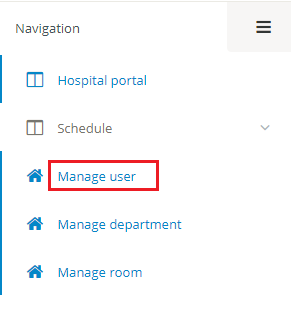


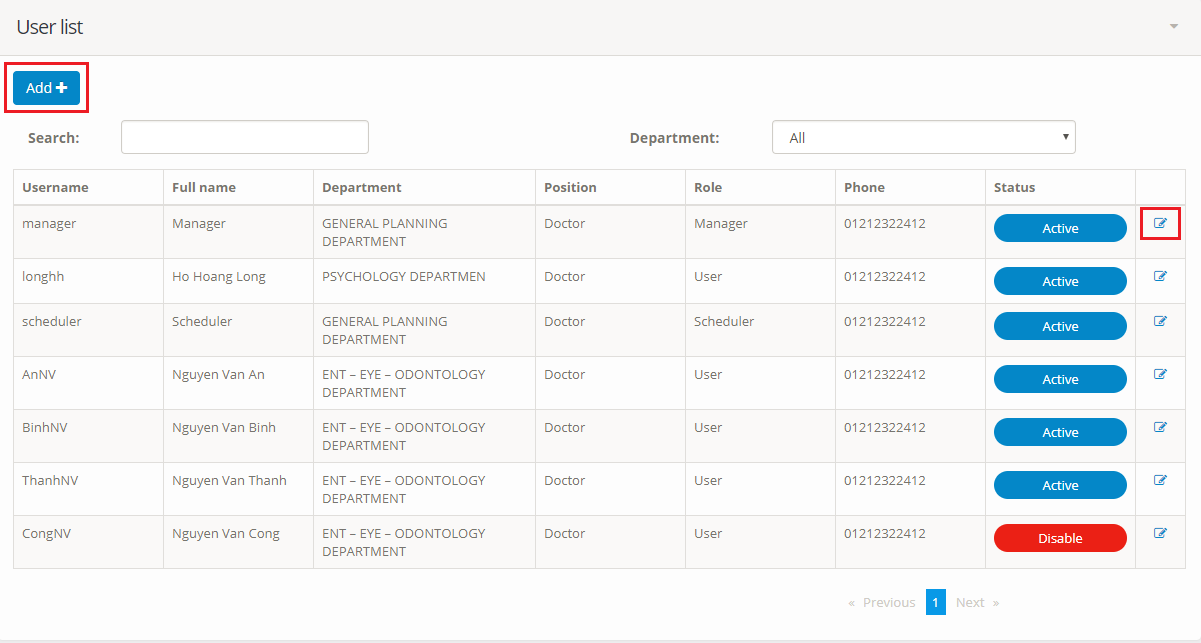
Please fill in all text box and click “Save” button

**2.4 For Admin:**

**2.4.1 Create/Update user**

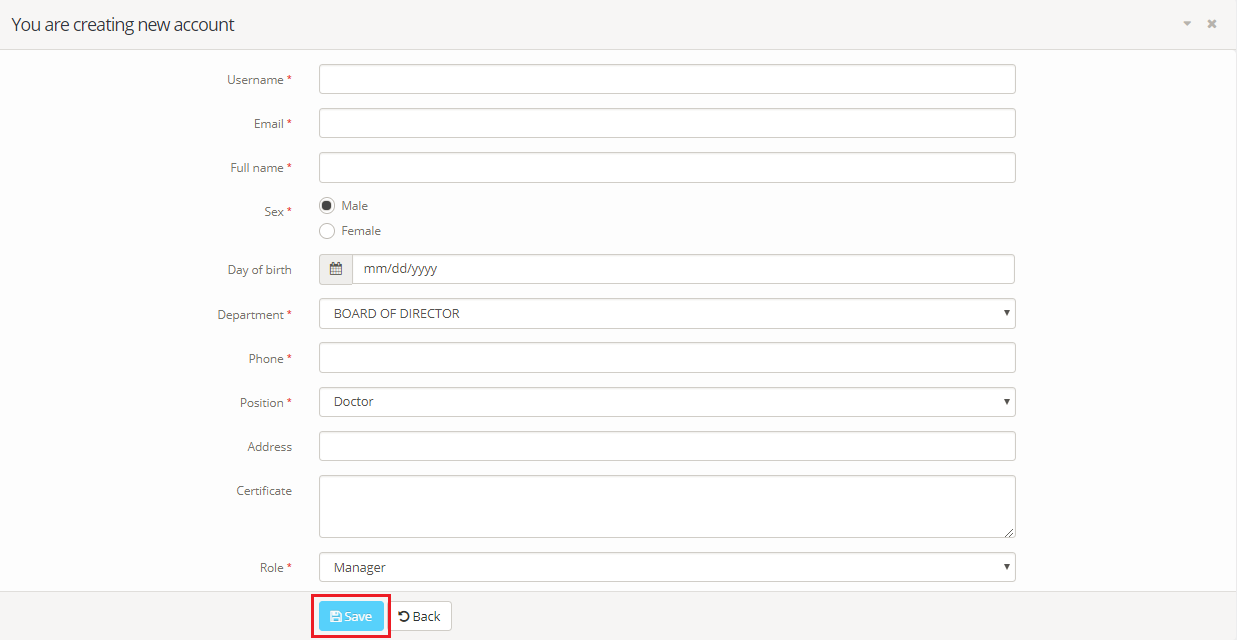
If you want to manager user, please click “Manage user” on menu bar





Click “Add” button on upper left corner of user page to create new user.

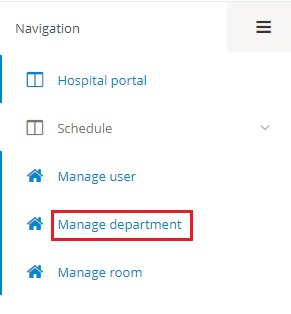
Click “View” icon on the right to update user.

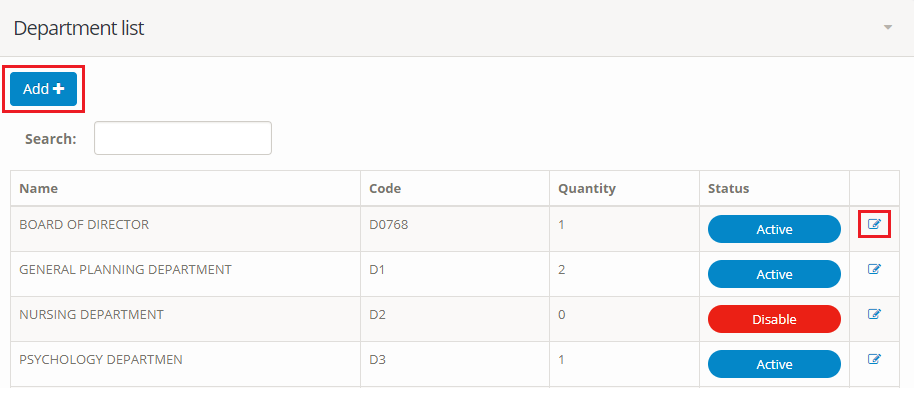


Please fill in all text box and click “Save” button

**2.4.2 Create/Update department**

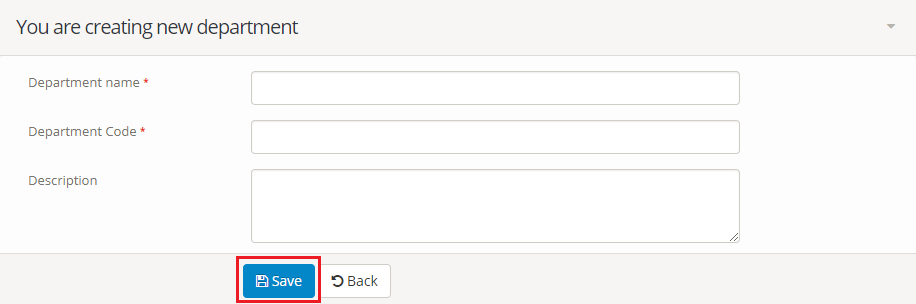
If you want to manager department, please click “Manage department” on menu bar

****

****

Click “Add” button on upper left corner of department page to create new department.

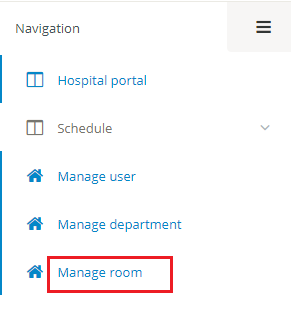
Click “View” icon on the right to update department.

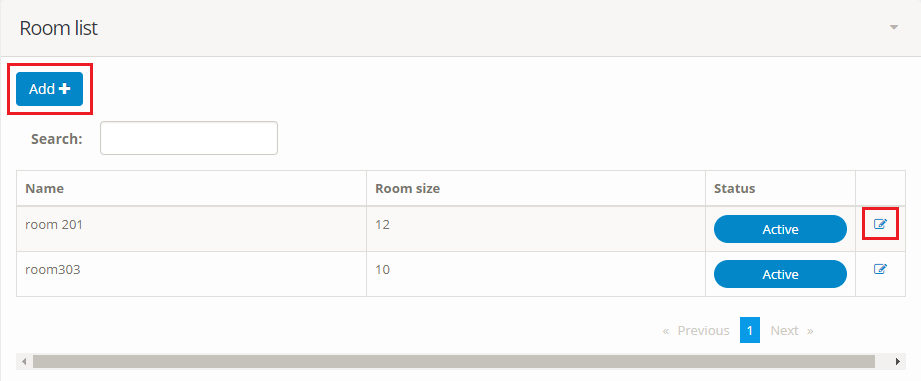
****

Please fill in all text box and click “Save” button

**2.4.3 Create/Update room**

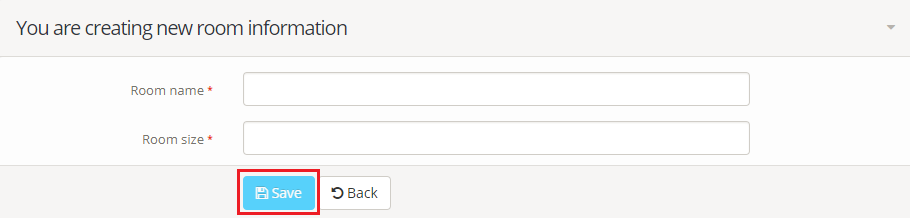
If you want to manager room, please click “Manage room” on menu bar





Click “Add” button on upper left corner of room page to create new room.

Click “View” icon on the right to update room.



Please fill in all text box and click “Save” button

**G. Appendix**