****MINISTRY OF EDUCATION AND**

**TRAINING**

**FPT UNIVERSITY**

Capstone Project Briefcase Document

**Hospital Portal**

|  |  |
| --- | --- |
| **Group 4** | |
| **Group members** | Hồ Hoàng Long - SE61761  Trần Phương Linh - SE62029  Phạm Xuân Tùng - SE61590 (Not defense)  Bạch Minh Đức – SE61791(Drop out) |
| **Supervisor** | Phạm Công Thành |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** |  |

-Ho Chi Minh City, 20/4/2018-

**

**CAPSTONE PROJECT REGISTER**

Class: Duration time: from / /2018 To / /2018

(\*) Profession: <Software Engineer> Specialty: <ES> <IS>

x

(\*) Kinds of person make registers: Lecturer Students

x

1. Register information for supervisor (if have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Full name** | **Phone** | **E-Mail** | **Title** |
| Supervisor 1 | Phạm Công Thành |  | Thanhpc3@fe.edu.vn | Mr. |

2. Register information for students (if have)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Full name** | **Student code** | **Phone** | **E-mail** | **Role in Group** |
| Student 1 | Hồ Hoàng Long | SE61761 |  |  | Member |
| Student 2 | Trần Phương Linh | SE62029 |  |  | Member |
| Student 3 | Phạm Xuân Tùng | SE61590 |  |  | Member |
| Student 4 | Bạch Minh Đức | SE61791 |  |  | Leader |

3. Register content of Capstone Project

(\*) 3.1. Capstone Project name:

English: Hospital Portal

Vietnamese: Cổng thông tin bệnh viện

Abbreviation: HPS

Building the web application provides following services:

* Construct hospital information
* Build organization chart (department, board)
* Make work schedule
* Make night shift schedule (Doctor, nurse, laborer)
* Make hospital news/ announcements
* Show medical news
* Show scientific research
* Show provided services
* …

(\*) 3.2. Main proposal content (including result and product)

1. Theory and practice (document):

* Student should apply the software development process and the UML
* Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
* 3 tiers should be applied
* Server side technique:
  + Database design, OOA, OOD, OOP, MVC, Java technology, …
* Client side technique
  + HTML5, CSS, JavaScript, JQuery, Ajax, Androids …
* Communication technique
  + Exchange information and transfer data in effective in networks ...

1. Program:

* Nhân viên thường: “Normal/ Common” employee
  + Xem lịch trực, lịch công tác và thông tin trên cổng thông tin của bệnh viện: View night shift schedule, work schedule, and information on the hospital portal
* Nhân viên được cấp quyền: “Authorized” employee
  + Tạo lịch trực, lịch công tác, tin tức, thông báo, bài viết nghiên cứu khoa học…: Make night shift schedule, work schedule, news and announcements, research papers…
* Manager:
  + Duyệt lịch, duyệt tin, duyệt thông báo… thêm/xóa/sửa các nội dung thông tin trên cổng thông tin của bệnh viện: Approve schedule, news, announcements… add / delete / edit information on the hospital portal
* Report:
  + Lịch công tác ngày: Daily work schedule
  + Lịch công tác tuần: Weekly work schedule
  + Lịch trực ngày của khoa/bệnh viện: Daily night-shift schedule of department/ hospital
  + Lịch trực tuần của khoa/bệnh viện: Weekly night-shift schedule of department/ hospital
  + Chấm công lịch trực của khoa theo tháng: Monthly employee timesheet

1. Other products:

* Khi có sự thay đổi về lịch công tác, lịch trực đột xuất phát sinh trong ngày sẽ gửi tin nhắn SMS đến các số đt của các manager: If there is any change in the schedule or unexpected night-shift schedule, SMS will be sent to phone numbers of managers

4. Other comment (propose all relative thing if have)

N/A

|  |  |
| --- | --- |
| **Supervisor (If have)**  *(Sign and full name)* | Tp. HCM, date …… ………. /20 …..  **On behalf of Registers**  *(Sign and full name)* |

**Table of Contents**

[A.](#_3znysh7) Introduction 9

[1. Project Information 9](#_2et92p0)

[2. Introduction 9](#_tyjcwt)

[3. Current Situation 9](#_3dy6vkm)

[4. Problem Definition 11](#_1t3h5sf)

[5. Proposed Solution 11](#_4d34og8)

[5.1 Feature functions 11](#_2s8eyo1)

[5.2 Advantages and disadvantages 13](#_3rdcrjn)

[6. Functional Requirements 13](#_26in1rg)

[7. Role and Responsibility 14](#_lnxbz9)

[B.](#_1ksv4uv) Software Project Management Plan 15

[1. Problem Definition 15](#_44sinio)

[1.1 Name of this Capstone Project 15](#_2jxsxqh)

[1.2 Problem Abstract 15](#_z337ya)

[1.3 Project Overview 15](#_3j2qqm3)

[2. Project organization 18](#_2xcytpi)

[2.1 Software Process Model 18](#_1ci93xb)

[2.2 Roles and responsibilities 19](#_2bn6wsx)

[2.3 Tools and Techniques 19](#_3as4poj)

[3. Project Management Plan 20](#_1pxezwc)

[3.1 Software development life cycle 20](#_49x2ik5)

[3.2 Phases’ detail 2](#_147n2zr)1

[C.](#_2grqrue) Software Requirement Specification 22

[1. User Requirement Specification 2](#_vx1227)2

[2. System Requirement Specification 2](#_3tbugp1)4

[2.1 External Interface Requirement 2](#_28h4qwu)4

[2.2 System Overview Use Case 25](#_nmf14n)

[3. Software System Attribute](#_sqyw64) 26

[4. Conceptual Diagram](#_kgcv8k) 27

[D.](#_1jlao46) Software Design Description 29

[1. System Architectural Design](#_2iq8gzs) 29

[1.1 Web Application Architecture Description](#_xvir7l) 30

[2. Component Diagram](#_1x0gk37) 31

[3. Detailed Description](#_4h042r0) 33

[3.1 Class Diagram](#_1baon6m) 33

[3.2 Interaction Diagram](#_pkwqa1) 34

[4. Database Design 45](#_1tuee74)

[4.1 Entity relationship diagram (ERD) 4](#_4du1wux)5

[4.2 Data Dictionary](#_2szc72q) 46

4.3 Database Relationship Diagram 47

[E.](#_meukdy) Task sheet 49

**List of Tables**

[Table 1 : Roles and Responsibilities 1](#_35nkun2)4

[Table 2: Hardware Requirement for Server 17](#_1y810tw)

[Table 3: Hardware Requirement for Mobile 17](#_4i7ojhp)

[Table 4: Hardware Requirement for PC 16](#_4i7ojhp)

[Table 5: Roles and Responsibilities Details 19](#_qsh70q)

[Table 6: Tools and techniques 19](#_qsh70q)

[Table 7: Software Development Life Cycle Detail 21](#_2p2csry)

[Table 8: Conceptual Diagram Data Dictionary 28](#_34g0dwd)

Table 9: Component Dictionary 32

[Table 10: Class dictionary](#_34g0dwd) 34

Table 11: Entity Data Dictionary 46

Table 12: Database Relationship Diagram 48

**List of Figures**

[Figure 1: Usecase diagram 2](#_1fob9te)5

Figure 2: Conceptual Diagram 27

[Figure 3: System architecture design](#_1fob9te) 29

Figure 4: Component Diagram 31

Figure 5: Class Diagram 33

Figure 6: Entity relationship diagram 45

Figure 7: Database Relationship Diagram 47

**A. 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.

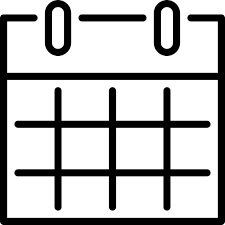
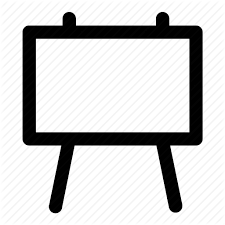
In addition, the system also supports many basic features such as post news, announcements, and research papers from the hospital to the community and schedule appointments for patients.

**3. Current Situation**

Current management process:

+ Medical booking: Guest have to call to hospital to booking an appointment. It is not suitable with large scale hospital because the number of patient coming to the hospital in a very crowded day. So that this method is only suitable for private clinics.

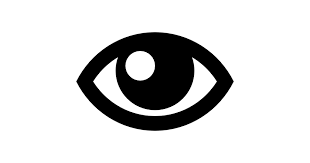
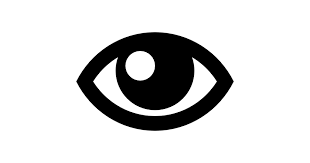
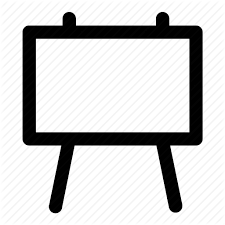
+ Create shift, working schedule: The person in charge (Department chief) create schedule on a form and sends it to the planning office to approval and storage. Then, the text is displayed on the notice board.



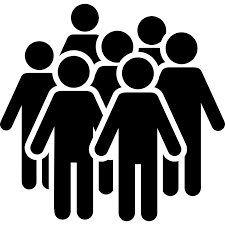
+ View shift, working schedule: When the doctor wants to know his schedule, each doctor must have a task scheduler, view the scheduler on the notice board, or ask the person in charge.



?!?



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

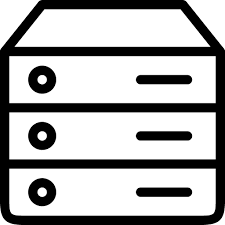


**4. Problem Definition**

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.

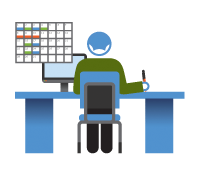
**5. Proposed Solution**

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, notification.

+ 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: Scheduler 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 will be approved and classified by manager and posted on the homepage. Guest can view public posts on the web.

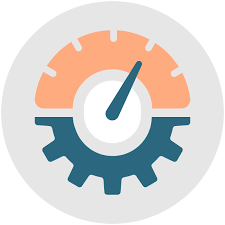
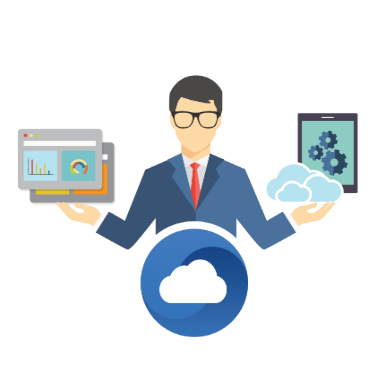


+Book medical examination: Each clinic will have a certain number of shifts, each of which will have a set number of appointments. Guest after selecting a clinic, entering personal information and set the appointment. When guests are successfully set, they will be notified of the specific time, sequence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

* Always have a team to maintain the operation of the system
* Have to train the hospital staff how to use the system

**6. Functional Requirements**

Function requirements of the system are listed as below:

+Guest: View article, booking medical examination, hospital information.

+User: Login, view schedule, view article.

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

+Poster: Login, view article, view schedule (shift, meeting, mission), post article.

+Manager: Login, view schedule (shift, meeting, mission), view article, check scheduler (shift, meeting, mission), create/update scheduler (shift, meeting, mission), update/approve article.

+Admin: Login, view schedule(shift, meeting, mission), view article, manage user info, department, room resource, category, service, view medical booking list and manage medical booking time.

**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. 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.

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, update, view and check mission schedule
* Create, update, view and check meeting schedule
* Create, update, view and check shift schedule
* Create, update, 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.
* 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**

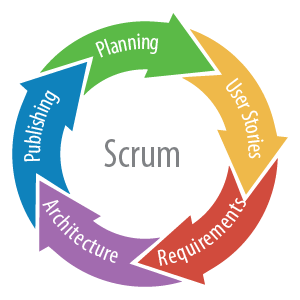
**2.1 Software Process Model**

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 |
| **4** | Phạm Xuân Tùng | 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, TungPX |
| ***2. Identify and clarify*** ***main functions.*** | Define which main functions system should provide. | LongHH, LinhTP, TungPX |
| ***3.Requirements specification*** | Defining the requirements in detail. | LongHH, LinhTP, TungPX |
| ***4. Implement code*** | - GUI Design  - Collect and validate information about hospital portal  - Connect to database  - Implement backend | LongHH, LinhTP, TungPX |

**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 category:
  + Create, update, Activate /deactivate category
* Manage service:
  + Create, update, Activate /deactivate service
* Manage Booking Medical:
  + Create, update, Activate /deactivate Booking Medical
* 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

- Manager article:

* Create, update, Activate /deactivate, approve article

***1.3 Scheduler Requirement:***

Scheduler guest who already login to the system by his/her account and uses service of system. The scheduler can uses 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 Poster Requirement:***

Poster is responsible for managing the hospital's information such as news, research, and contact. The scheduler can uses some following functions:

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

***1.5 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

***1.6 Guest Requirement:***

Guest is user out of the system.

* Booking medical examination, view booking history.
* View hospital information.

**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.
  1. **System Overview Use Case**

***Figure 1: Use case diagram***

**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.

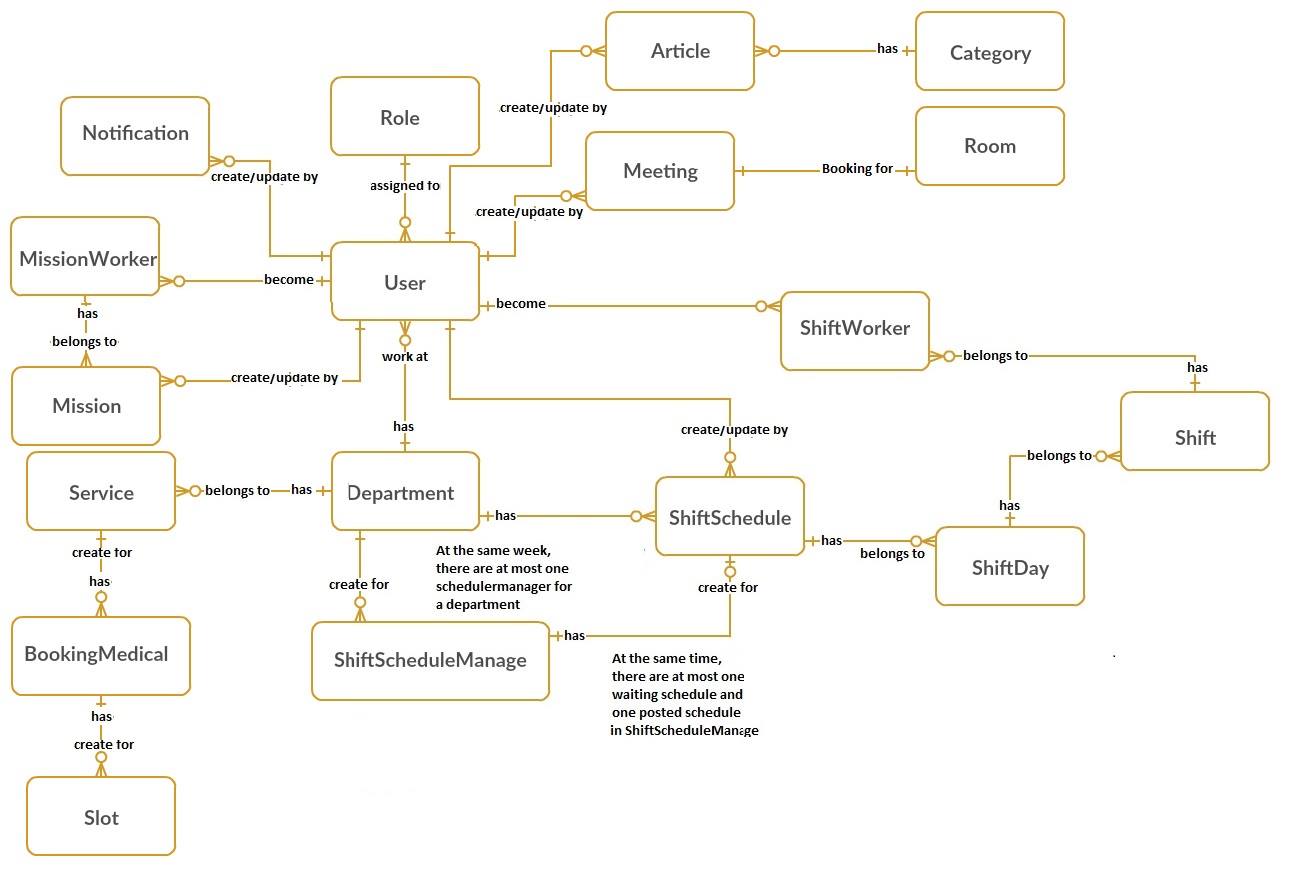
**3.6 Portability**

* Web application can be run on Chrome browser version 5.8 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**

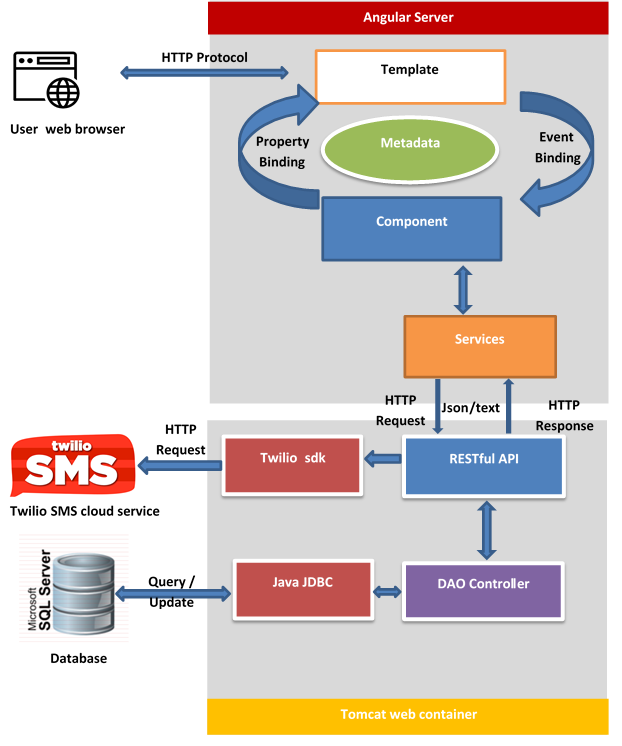
****

***Figure 2: 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.* |
| **Service** | *Contain the service information.* |
| **Category** | *Contain the category information.* |
| **Slot** | *Contain the slot information.* |
| **Booking medical** | *Contain the booking medical information.* |
| **Article** | *Contain the article information.* |

***Table 8: Conceptual Diagram Data Dictionary***

**D. Software Design Description**

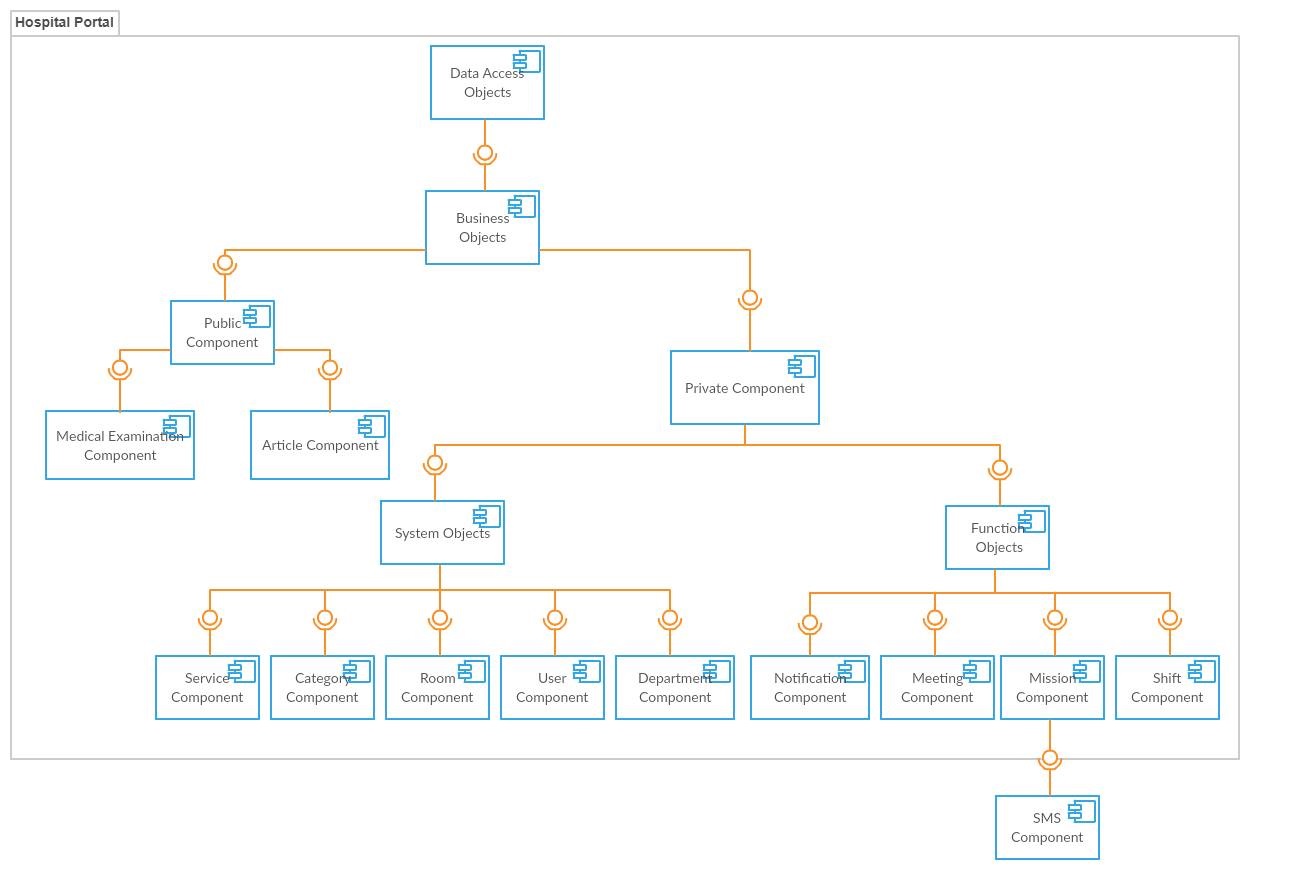
**1. System Architectural Design**

***Figure 3: System architecture design***

**1.1 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.

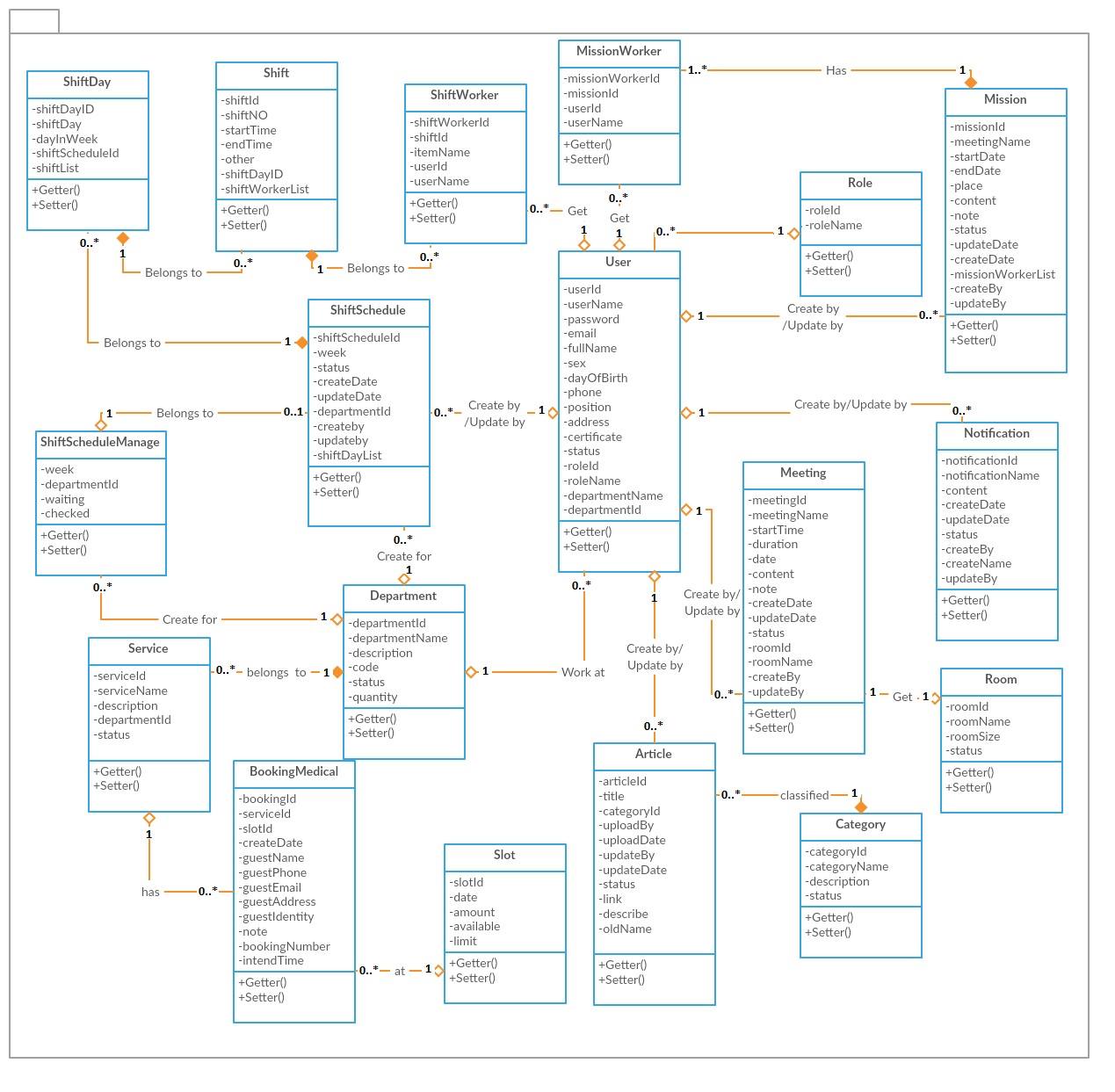
**2. Component Diagram**

***Figure 4: Component diagram***

|  |  |
| --- | --- |
| 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. |
| Private Component | Component to handle user activities in the system |
| Public Component | Component to handle guest activities in the system |
| Service Objects | Common objects to handle domain operations for each service component |
| Function Objects | Common objects to handle function for each system component. |
| Service Component | Component to handle service in the system |
| Category Component | Component to handle sategory in the system |
| 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 |
| Notification Component | Component to handle notification resource 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 |

**Table 9: Component Dictionary**

**3. Detailed Description**

**3.1 Class Diagram**

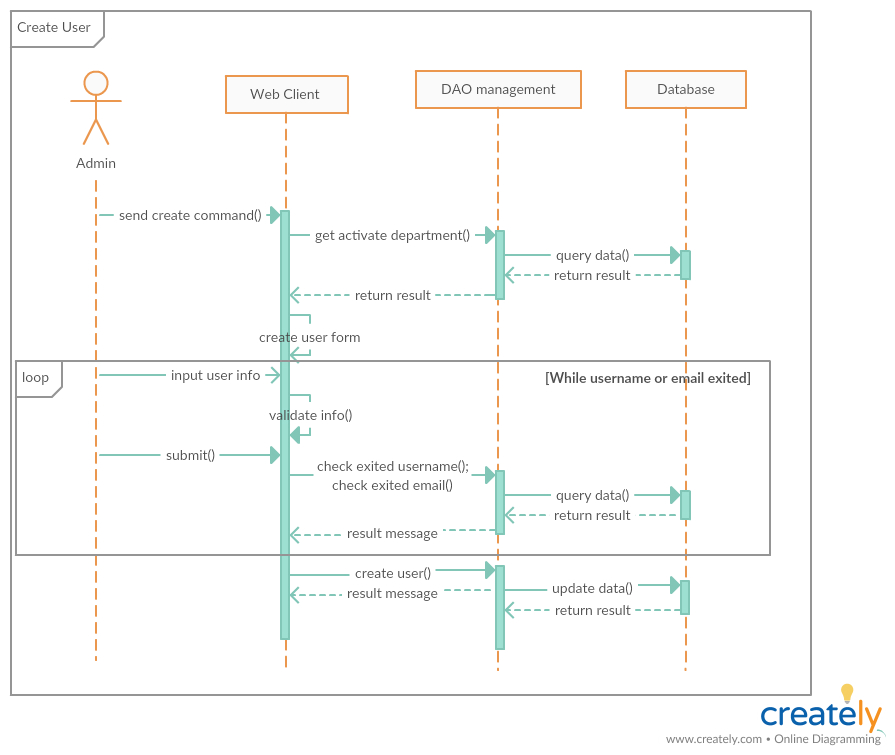
***Figure 5: Class diagram***

|  |  |
| --- | --- |
| *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 |
| MissionWorker | Contain the mission worker information |
| Shift worker | Contain the shift worker information |
| Role | Contain the role information |
| Shift | Contain the shift information |
| ShiftScheduleManage | Contain the shift schedule manage information |
| ShiftSchedule | Contain the shift schedule information |
| ShiftDay | Contain the shift day information |
| Category | Contain the shift category information |
| Article | Contain the shift article information |
| Slot | Contain the shift slot information |
| BookingMedical | Contain the shift booking medical information |
| Service | Contain the shift service information |

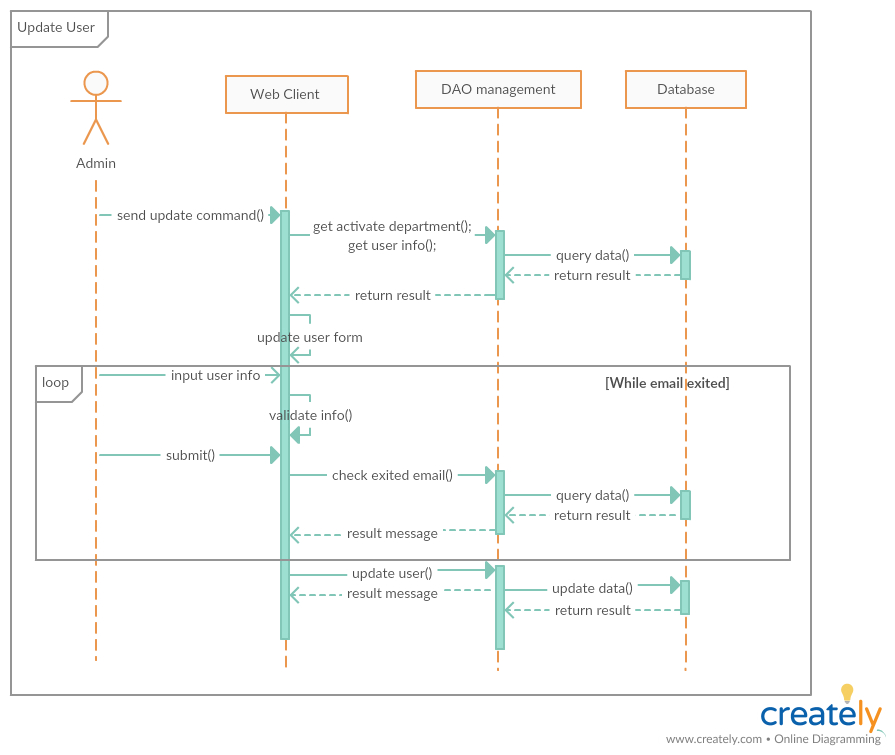
***Table 10: Class dictionary***

**3.2 Interaction Diagram**

***3.2.1 Create user***

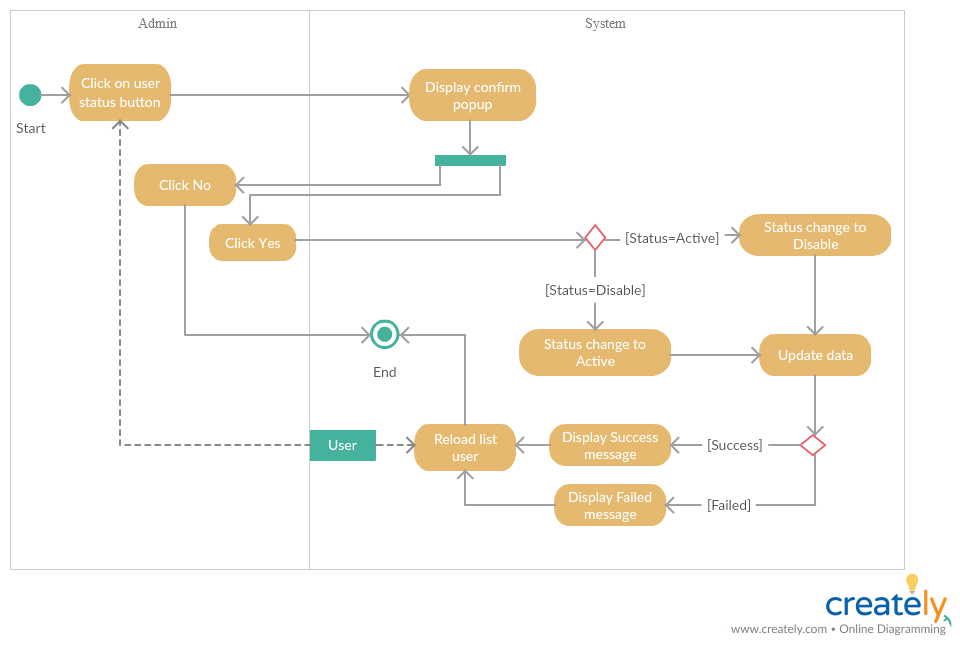
*****Summary: this diagram show process of admin creates new User.*

***3.2.2 Update user***

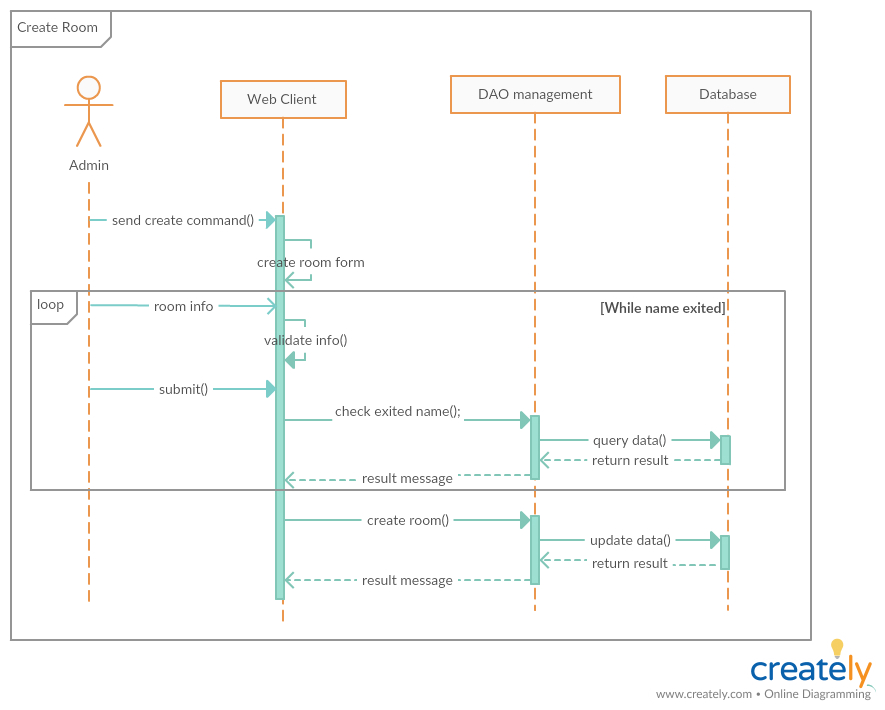
*****Summary: this diagram show process of admin update User*

***3.2.3 Activate/Deactivate user:***

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

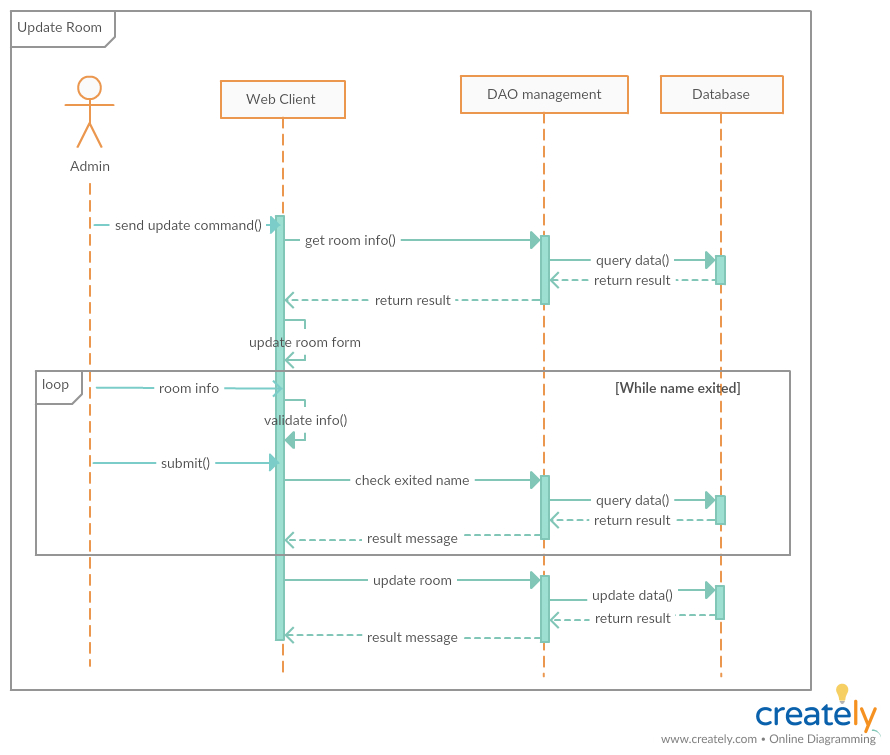
******

***3.2.4 Create room***

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

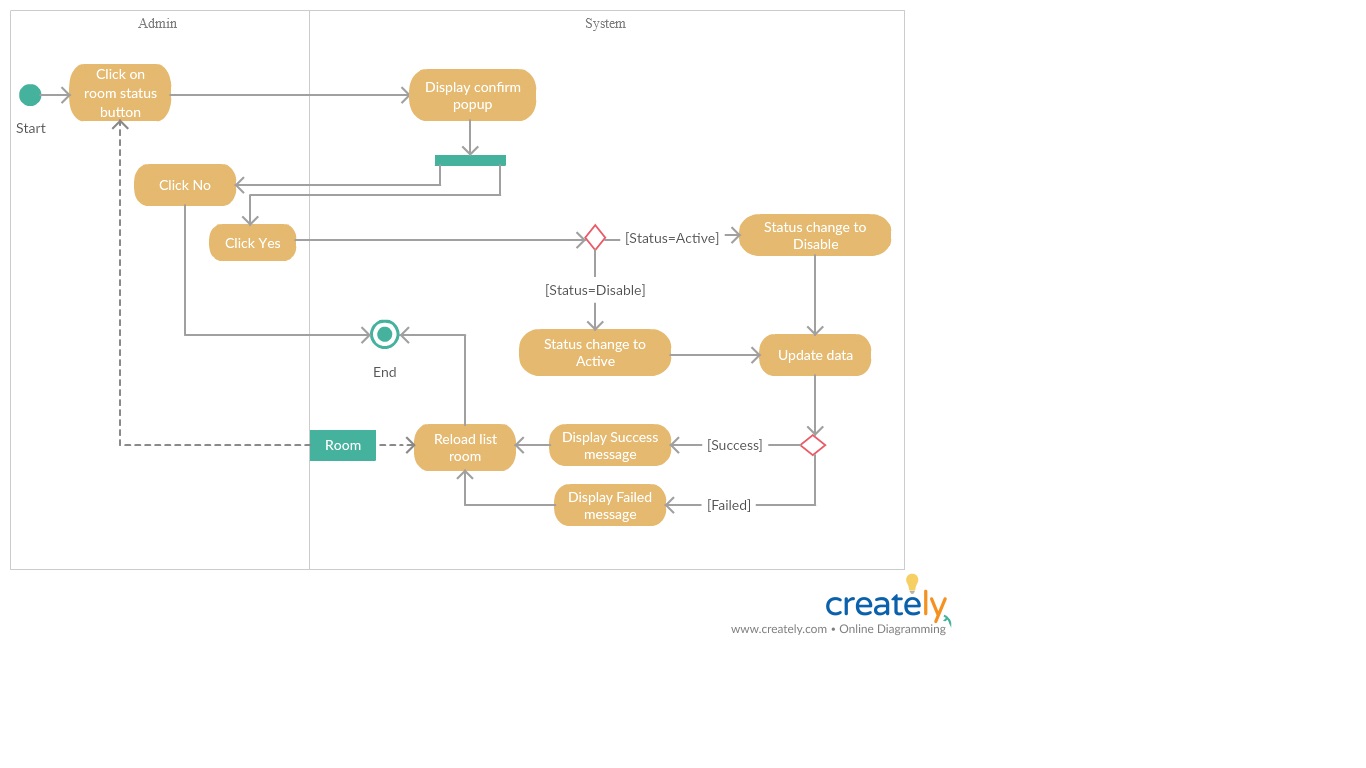
***3.2.5 Update room***

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

******

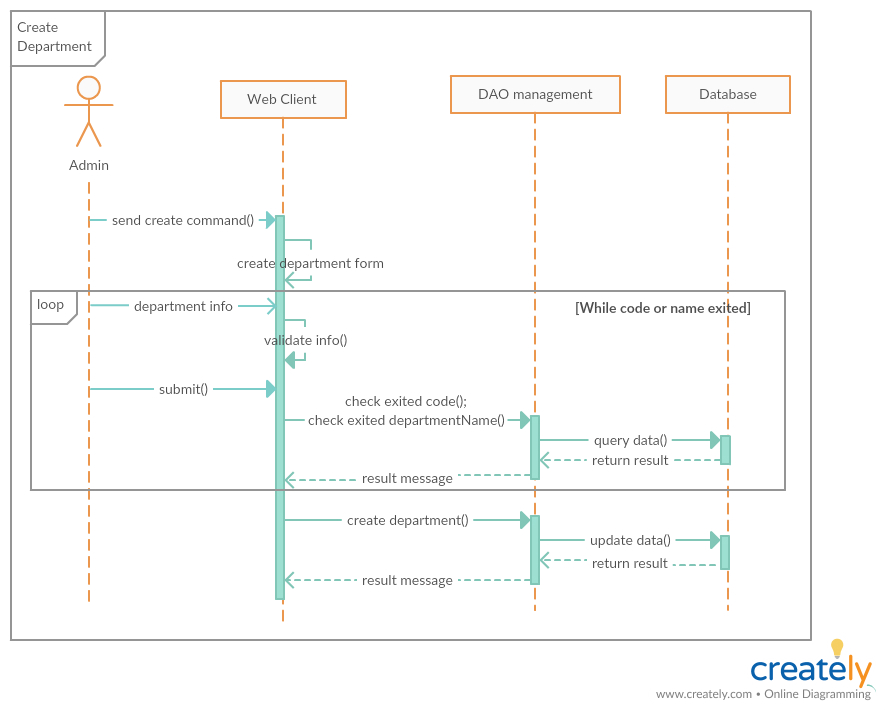
***3.2.6 Activate/Deactivate room:***

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

**

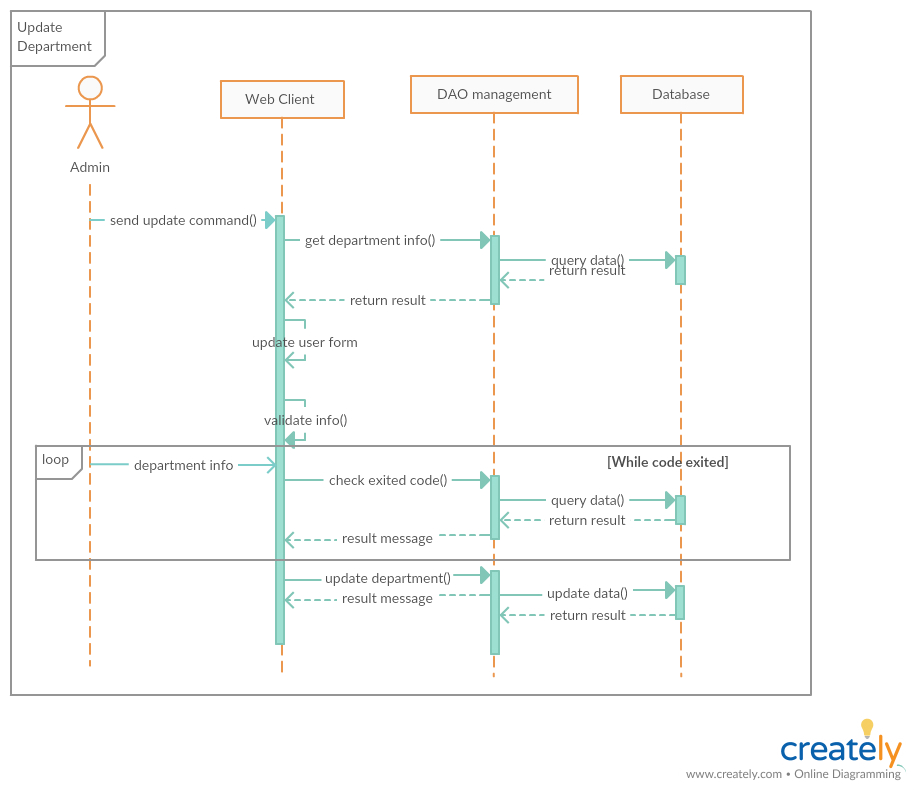
***3.2.7 Create department***

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

**

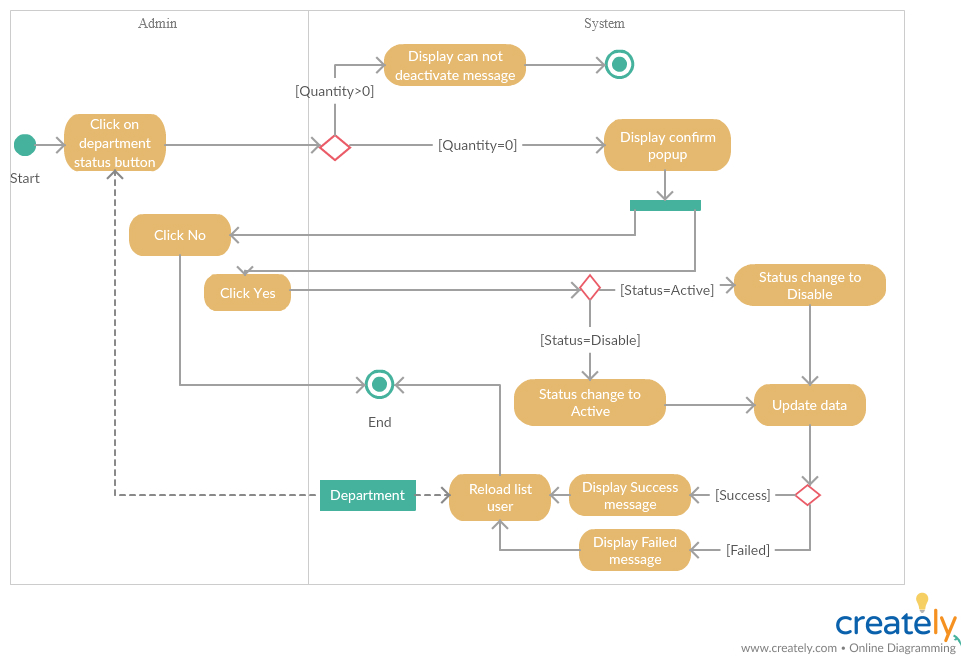
***3.2.8 Update department***

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

**

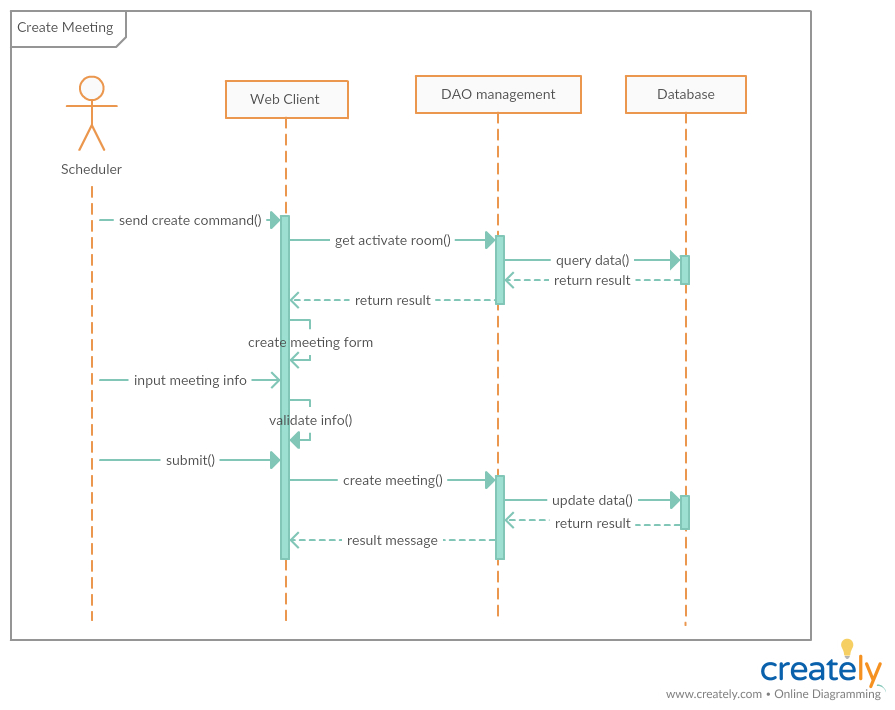
***3.2.9 Activate/Deactivate department***

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

****

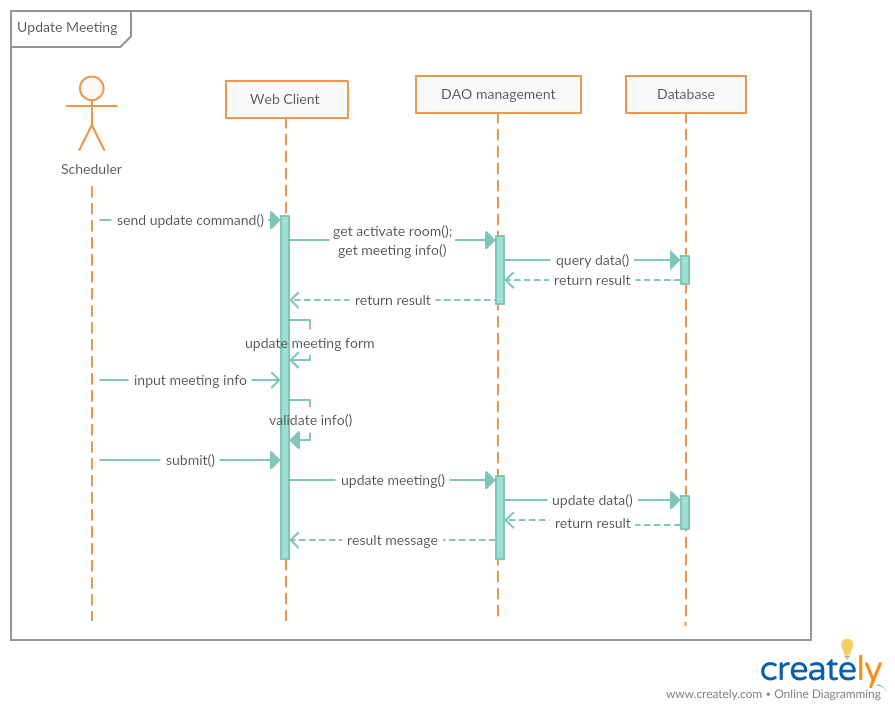
***3.2.10 Create meeting***

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

**

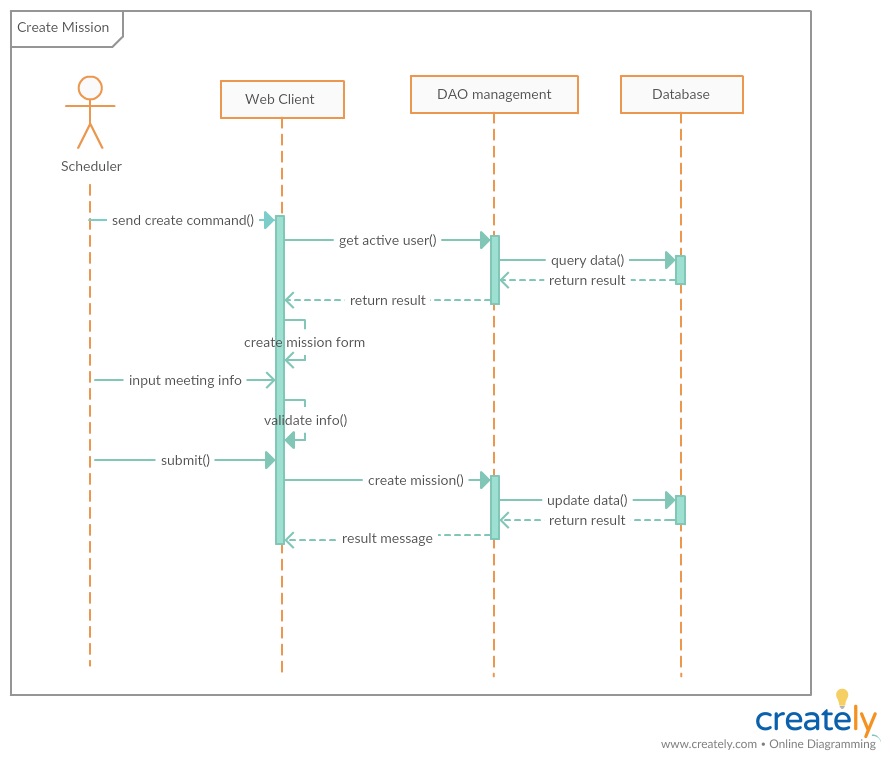
***3.2.11 Update meeting***

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

****

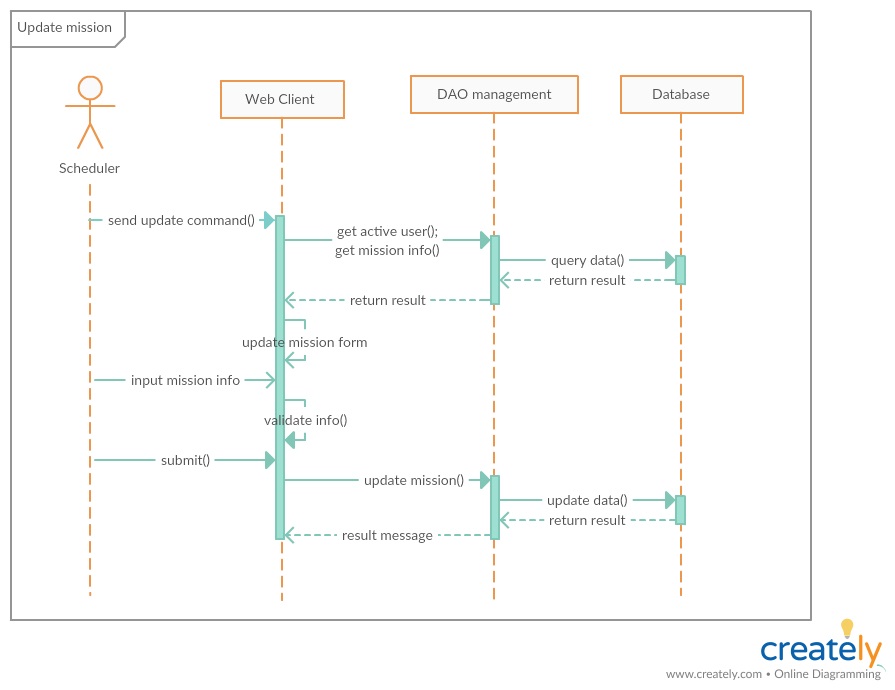
***3.2.12 Create mission***

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

****

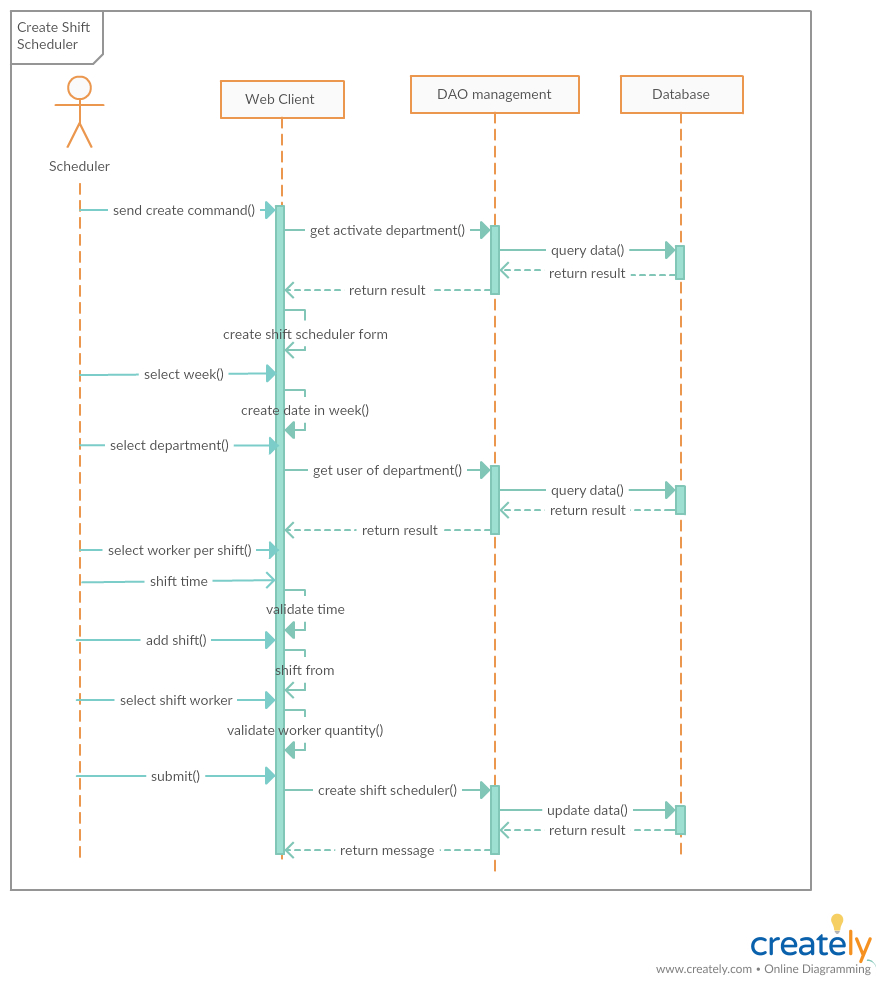
***3.2.13 Update mission***

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

****

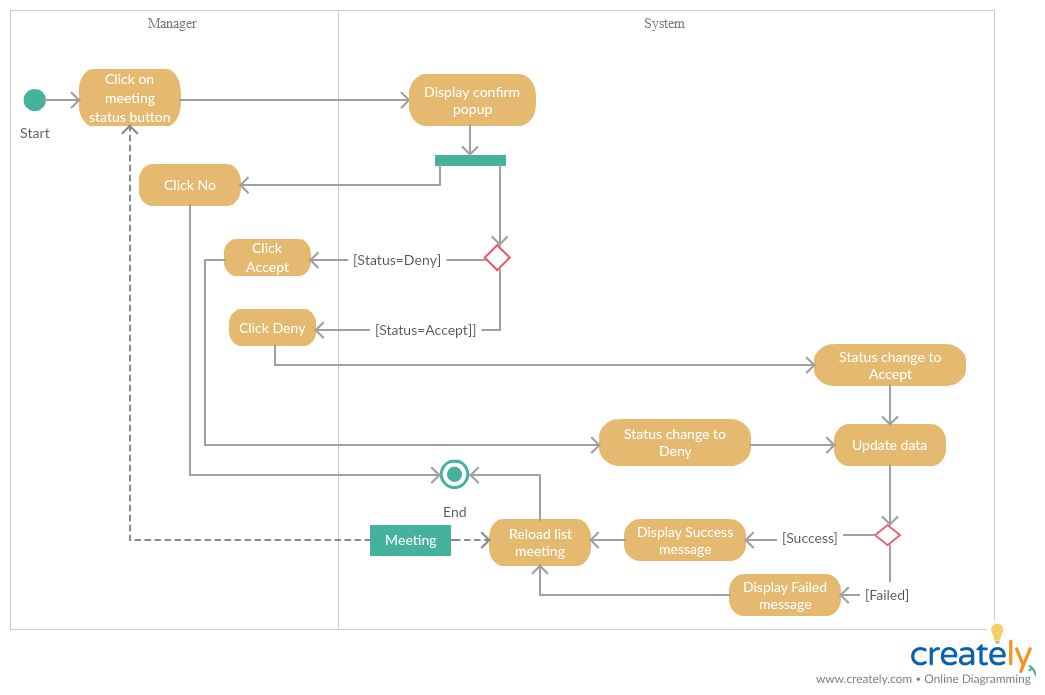
***3.2.14 Create shift schedule***

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

****

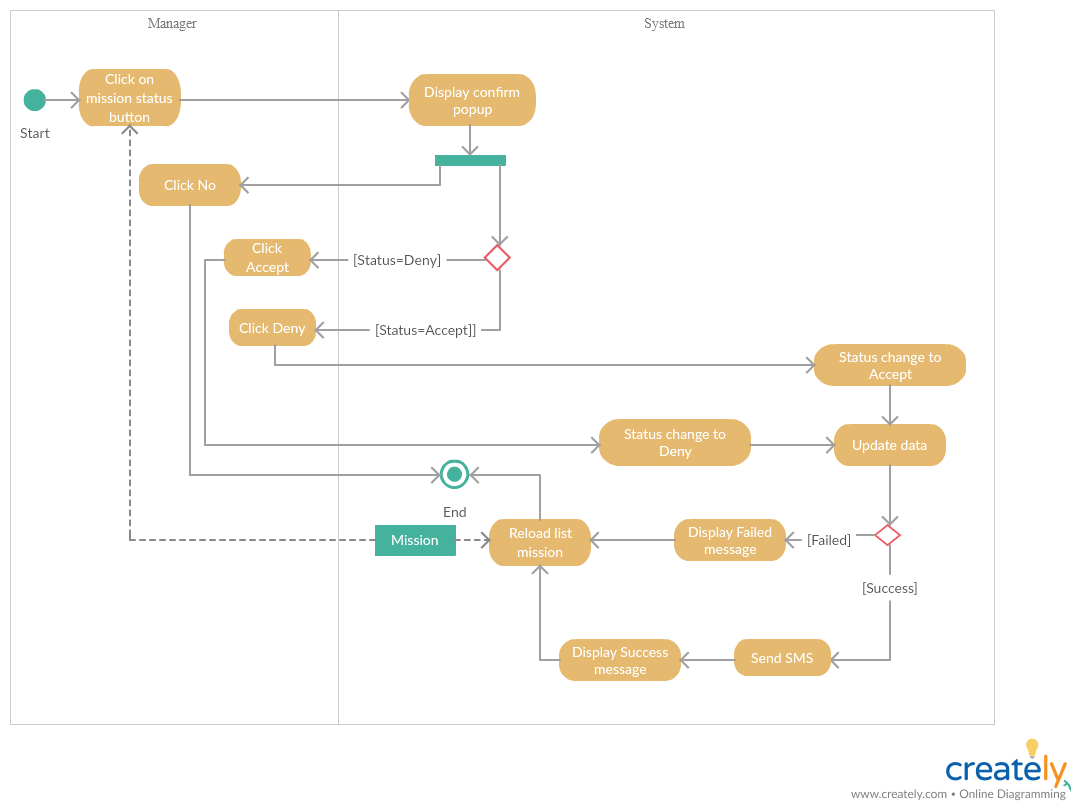
***3.2.15 update checked meeting status***

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

**

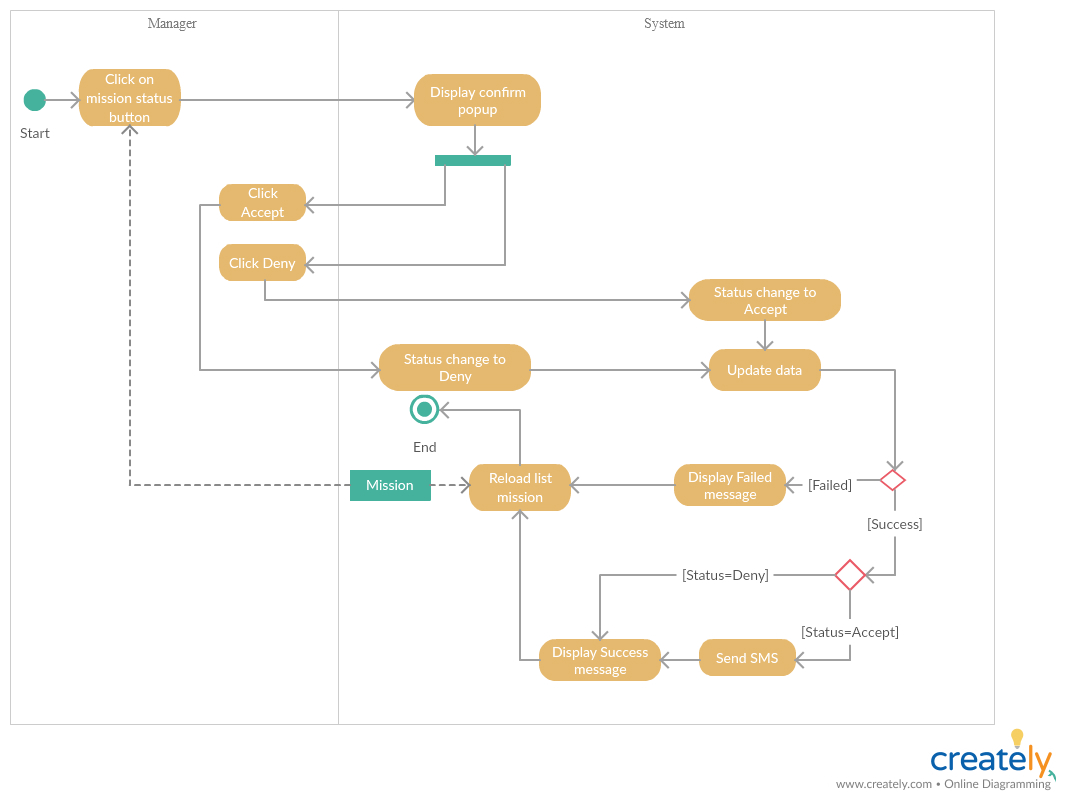
***3.2.16 update checked mission status***

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

****

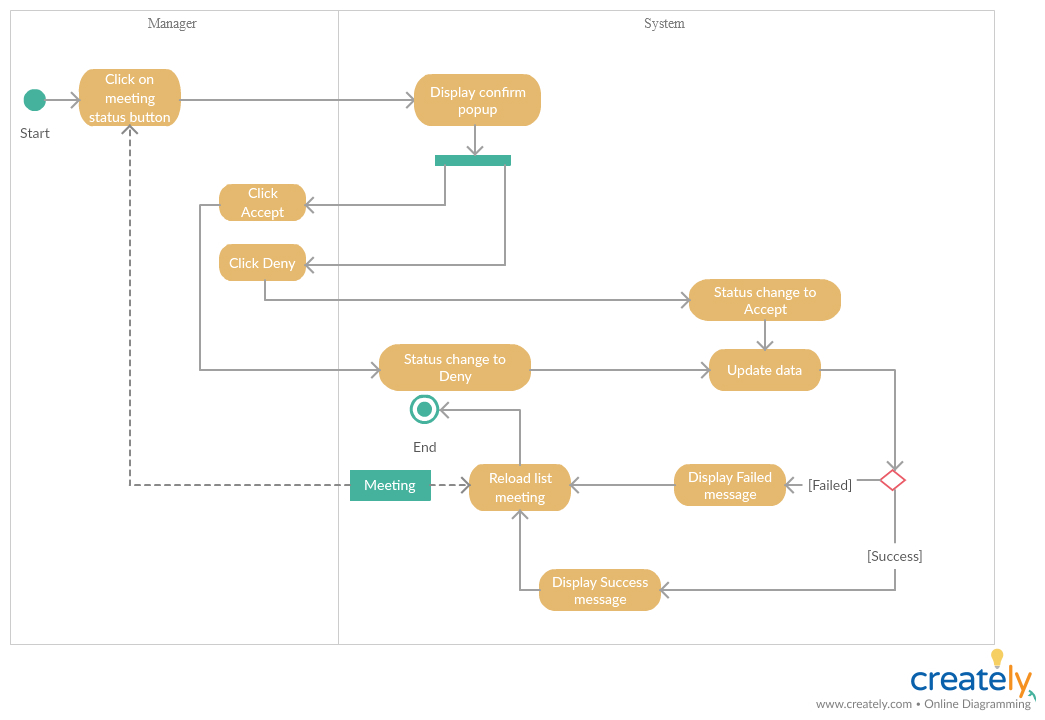
***3.2.17 check waiting mission status***

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

****

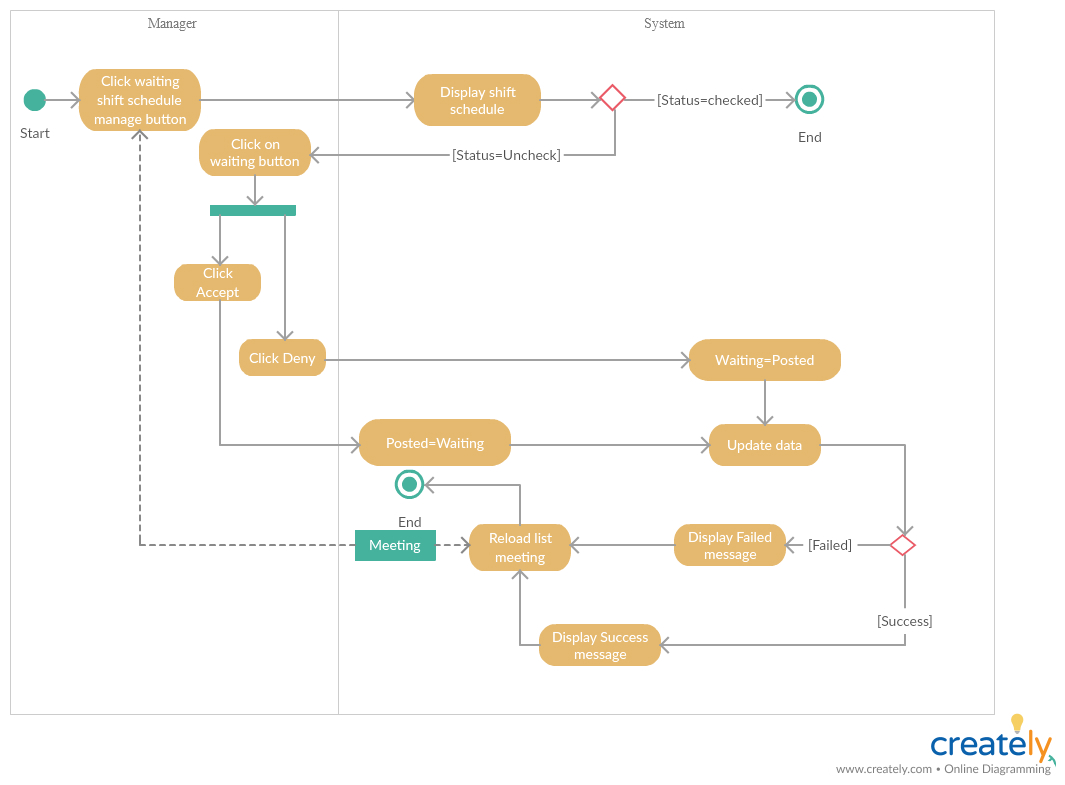
***3.2.18 check waiting meeting status***

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

****

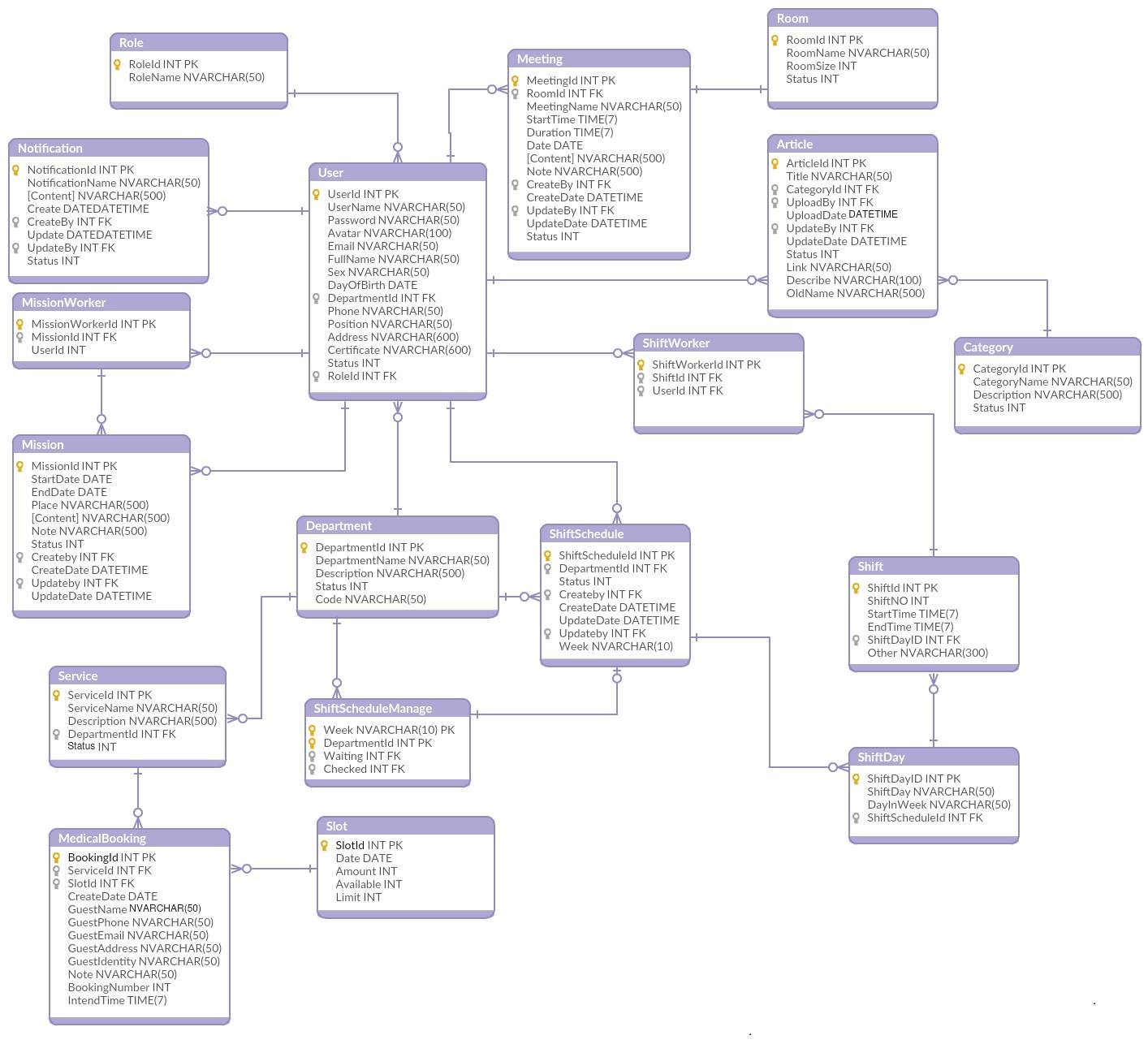
***3.2.19 check shift schedule***

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

**

**4. Database Design**

**4.1 Entity relationship diagram (ERD)**

****

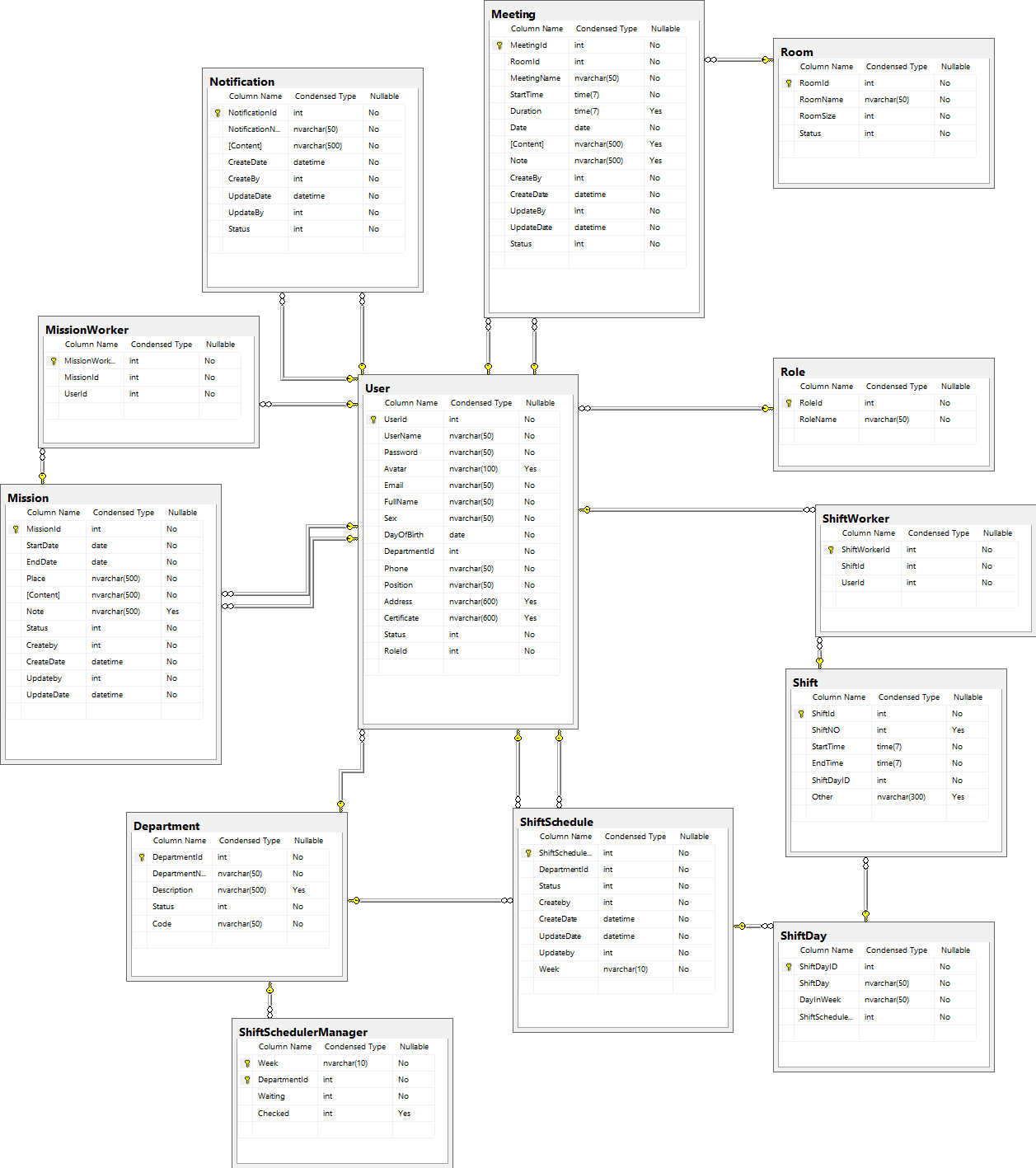
***Figure 6: Entity relationship diagram***

**4.2 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. |
| Category | Contain the user information. |
| Service | Contain the service information. |
| Slot | Contain the slot information. |
| BookingMedical | Contain the booking medical information. |
| Article | Contain the article information. |

***Table 11: Entity Data dictionary***

**4.3 Database Relationship Diagram**

***Figure 7: Database Relationship Diagram***

|  |  |
| --- | --- |
| **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. |

***Table 12: Database Relationship Diagram***

**E. TASK SHEET**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Product Deliverables** | **Task** | **LongHH** | **LinhTP** | **TungPX** | **Unit** | **Size** |
| 1 | Report1 - Introduction | **1.Project Information** | **O** |  |  |  | 1 |
| **2.Introduction** | **O** |  |  |  | 1 |
| **3.Current Situation** | **O** |  |  |  | 1 |
| **4.Problem Definition** | **O** |  |  |  | 1 |
| **5.Proposed Solution** | **O** |  |  |  | 1 |
| **6.Functional Requirements** | **O** | **O** |  |  | 1 |
| **7. Role and Responsibility** |  | **O** |  |  | 1 |
| 2 | Report2- Software Project Management Plan | **1. Problem Definition** |  | **O** |  |  | 1 |
| **2. Project organization** |  | **O** |  |  | 1 |
| **3. Project Management Plan** |  | **O** |  |  | 1 |
| **4. Coding Convention** |  | **O** |  |  | 1 |
| 3 | Report 3- Software Requirement Specification | **1. User Requirement Specification** |  | **O** |  |  | 1 |
| **2. System Requirement Specification** |  |  |  |  | 1 |
| **2.1 External Interface Requirement** |  | **O** |  |  | 1 |
| **2.2 System Overview Use Case** | **O** |  |  |  | 1 |
| **2.3 List of Use Case** |  |  |  |  | 1 |
| 2.3.1 <Admin>Overview Use Case |  | **O** |  |  | 1 |
| 2.3.2 <Manager>Overview Use Case | **O** | **O** |  |  | 1 |
| 2.3.3 <Scheduler>Overview Use Case | **O** |  |  |  | 1 |
| 2.3.4 <User>Overview Use Case |  | **O** |  |  | 1 |
| **3. Software System Attribute** |  | **O** |  |  | 1 |
| **4. Conceptual Diagram** | **O** |  |  |  | 1 |
| 4 | Report 4- Software Design Description | **1. Design Overview** | **O** |  |  |  | 1 |
| **2. System Architectural Design** | **O** |  |  |  | 1 |
| **3. Component Diagram** | **O** |  |  |  | 1 |
| **4. Detailed Description** |  |  |  |  | 1 |
| 4.1 Class Diagram | **O** |  |  |  | 1 |
| 4.2 Class Diagram Explanation |  | **O** |  |  | 1 |
| 4.3 Interaction Diagram | **O** |  |  |  | 5 |
| **5. Interface** |  |  |  |  | 1 |
| 5.1 Component interface |  | **O** |  |  | 1 |
| 5.2 User Interface Design |  | **O** |  |  | 1 |
| **6. Database Design** |  |  |  |  | 1 |
| 6.1 Entity relationship diagram (ERD) | **O** |  |  |  | 1 |
| 6.2 Data Dictionary |  | **O** |  |  | 1 |
| 5 | Report 5 - Software Implementation and Test Document | **1. Introduction** |  | **O** |  |  | 1 |
| **2. Database Relationship Diagram** |  |  |  |  | 1 |
| 2.1 Physical Diagram | **O** |  |  |  | 1 |
| 2.2 Data Dictionary |  | **O** |  |  | 1 |
| **3. Performance Measures** |  |  |  |  | 1 |
| **4. Test Plan** | **O** |  |  |  | 1 |
| **5. System Testing Test Case** |  |  |  |  | 1 |
| 5.1 Admin Test Case |  | **O** |  |  | 1 |
| 5.2 Manager Test Case |  | **O** |  |  | 1 |
| 5.3 User Test Case |  |  |  |  | 1 |
| 6 | Report 6 - Software User's Manual | **1. Installation Guide** |  |  |  |  | 1 |
| 1.1 Setting up environment at server side | **O** |  |  |  | 1 |
| 1.2 Deployment at server side | **O** |  |  |  | 1 |
| **2. User Guide** |  |  |  |  | 1 |
| 2.1 For User: |  | **O** |  |  | 1 |
| 2.2 For Scheduler: |  | **O** |  |  | 1 |
| 2.3 For Manager: |  | **O** |  |  | 1 |
| 2.4 For Admin: |  | **O** |  |  | 1 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Notes** |  |  |  |  |  |  |  |
| No. | Function Types | Function Point Size |  |  |  |  |  |
| 1 | Manage user (Create, update, activate/deactivate) | 3 |  |  |  |  |  |
| 2 | Manage department (Create, update, activate/deactivate) | 3 |  |  |  |  |  |
| 3 | Manage room (Create, update, activate/deactivate) | 3 |  |  |  |  |  |
| 4 | Manage notification (Create, update, activate/deactivate) | 3 |  |  |  |  |  |
| 5 | Manage mission schedule (Create, update, check, activate/deactivate) | 4 |  |  |  |  |  |
| 6 | Manage meeting schedule (Create, update, check, activate/deactivate) | 4 |  |  |  |  |  |
| 7 | Manage shift schedule (Create, update, check, activate/deactivate) | 4 |  |  |  |  |  |
| 8 | Manage article schedule (Create, update, check, activate/deactivate) | 3 |  |  |  |  |  |
| 9 | Manage booking medical examination (Create, update, check, activate/deactivate) | 3 |  |  |  |  |  |
| 10 | Manage category (Create, update, check, activate/deactivate) | 3 |  |  |  |  |  |
| 11 | Manage service (Create, update, check, activate/deactivate) | 3 |  |  |  |  |  |
| 12 | Manage notification (Create, update, check, activate/deactivate) | 2 |  |  |  |  |  |
| 13 | Login, Authentication | 2 |  |  |  |  |  |
| 14 | Mobile interface | 3 |  |  |  |  |  |