|  |  |
| --- | --- |
| logo ngan.png | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Office Rental Service |
|  |
| |  |  | | --- | --- | | **Group 6** | | | **Group Members** | Lê Xuân Tiến – Team Leader – SE60897  Nguyễn Vũ Hoàng Quốc – Team Member – SE61112  Trương Tiến Thành – Team Member – SE61052  Trần Lê Tuấn – Team Member – 60350 | | **Supervisor** | Mr. Nguyễn Trọng Tài | | **Ext Supervisor** | N/A | | **Capstone Project Code** | ORS | |
| - Hồ Chí Minh City, May 2015 - |

*This page is intentionally left blank*

Table of Contents

[A. Report No 1 - Introduction 2](#_Toc426363829)

[1. Project Information 2](#_Toc426363830)

[2. Introduction 2](#_Toc426363831)

[3. Current Situation 2](#_Toc426363832)

[4. Problem Definition 2](#_Toc426363833)

[5. Proposed Solution 2](#_Toc426363834)

[5.1 Feature functions 2](#_Toc426363835)

[5.2 Advantage and disadvantage 2](#_Toc426363836)

[6. Functional Requirements 2](#_Toc426363837)

[7. Role and Responsibility 2](#_Toc426363838)

[B. Project Management Plan (PMP) 2](#_Toc426363839)

[1. Problem Definition 2](#_Toc426363842)

[1.1 Name of this Capstone Project 2](#_Toc426363846)

[1.2 Problem Abstract 2](#_Toc426363847)

[1.3 Project Overview 2](#_Toc426363848)

[2. Project organization 2](#_Toc426363849)

[2.1 Software Process Model 2](#_Toc426363852)

[2.2 Roles and Responsibilities 2](#_Toc426363853)

[2.3 Tools and Techniques 2](#_Toc426363854)

[3. Project management plan 2](#_Toc426363855)

[3.1 Software development life cycle 2](#_Toc426363857)

[3.2 Phase Detail 2](#_Toc426363858)

[3.3 All Meeting Minutes 2](#_Toc426363859)

[4. Coding Convention 2](#_Toc426363860)

[C. Software Requirements Specifications (SRS) 2](#_Toc426363861)

[1. User Requirement Specification 2](#_Toc426363862)

[1.1 Guest requirement 2](#_Toc426363863)

[1.2 Customer requirement 2](#_Toc426363864)

[1.3 Admin requirement 2](#_Toc426363865)

[1.4 Manager requirement 2](#_Toc426363866)

[1.5 Staff requirement 2](#_Toc426363867)

[1.6 System requirement 2](#_Toc426363868)

[2. System Requirement Specification 2](#_Toc426363869)

[2.1 External Interface Requirements 2](#_Toc426363870)

[2.2 System Overview Use Case: 2](#_Toc426363871)

[2.3 List of Use Case: 2](#_Toc426363872)

[3. Software System Attribute 2](#_Toc426363873)

[3.1 Usability 2](#_Toc426363874)

[3.2 Reliability 2](#_Toc426363875)

[3.3 Availability 2](#_Toc426363876)

[3.4 Security 2](#_Toc426363877)

[3.5 Maintainability 2](#_Toc426363878)

[3.6 Portability 2](#_Toc426363879)

[3.7 Performance 2](#_Toc426363880)

[4. Conceptual Diagram 2](#_Toc426363881)

[D. Report 4: Software Design Description 2](#_Toc426363882)

[1. Design Overview 2](#_Toc426363883)

[2. System Architectural Design 2](#_Toc426363884)

[2.1 Choice of System Architecture 2](#_Toc426363885)

[3. Component Diagram 2](#_Toc426363886)

[4. Detailed Description Explanation 2](#_Toc426363887)

[4.1 Class Diagram 2](#_Toc426363888)

[4.2 Interaction Diagram 2](#_Toc426363889)

[5. Database Design 2](#_Toc426363890)

[5.1 Logical Diagram 2](#_Toc426363891)

[5.2 Data Dictionary 2](#_Toc426363892)

[6. Algorithms 2](#_Toc426363893)

[6.1 Clustering data 2](#_Toc426363894)

[6.2 Matching data 2](#_Toc426363895)

**List of Figures**

[Figure 1: Software process model 2](#_Toc426363896)

[Figure 2: Use Case Overview 2](#_Toc426363897)

[Figure 3: <Guest>Overview Use Case 2](#_Toc426363898)

[Figure 4: <Guest> Register 2](#_Toc426363899)

[Figure 5: <Guest> Login 2](#_Toc426363900)

[Figure 5: <Guest, Customer> Search office 2](#_Toc426363901)

[Figure 6: <Guest, Customer> View office detail 2](#_Toc426363902)

[Figure 7: <Customer>Overview Use Case 2](#_Toc426363903)

[Figure 4: <Customer> View customer info 2](#_Toc426363904)

[Figure 10: <Customer> Edit customer info 2](#_Toc426363905)

[Figure 10: <Customer> Change password 2](#_Toc426363906)

[Figure 4: <Customer> Create appointment request 2](#_Toc426363907)

[Figure 4: <Customer> View rental request list 2](#_Toc426363908)

[Figure 4: <Customer> View repair request list 2](#_Toc426363909)

[Figure 4: <Customer> View list repair history 2](#_Toc426363910)

[Figure 4: <Customer> Request extend contract 2](#_Toc426363911)

[Figure 4: <Customer> Request return contract 2](#_Toc426363912)

[Figure 4: <Customer> View list contract 2](#_Toc426363913)

[Figure 4: <Customer> View contract 2](#_Toc426363914)

[Figure 4: <Customer> Request repair 2](#_Toc426363915)

[Figure 4: <Customer> Request repair 2](#_Toc426363916)

[Figure 4: <Customer> Create office request 2](#_Toc426363917)

[Figure 12: <Customer> View Result Request Office 2](#_Toc426363918)

[Figure 6: Component Diagram 2](#_Toc426363919)

[Figure 7: Class Diagram 2](#_Toc426363920)

[Figure 8: Class Diagram 2](#_Toc426363921)

[Figure 9: Service Package Class Diagram 2](#_Toc426363922)

[Figure 10: Class Diagram 2](#_Toc426363923)

[Figure 11: Class Diagram 2](#_Toc426363924)

[Figure 12: Class Diagram 2](#_Toc426363925)

[Figure 13: Mobile Package Class Diagram 2](#_Toc426363926)

[Figure 14: View Appointment Request List 2](#_Toc426363927)

[Figure 15: Assign Appointment 2](#_Toc426363928)

[Figure 16: Cancel Appointment 2](#_Toc426363929)

[Figure 17: Create Contract 2](#_Toc426363930)

[Figure 18: View Contract List 2](#_Toc426363931)

[Figure 19: View Contract Detail 2](#_Toc426363932)

[Figure 20: Edit Contract 2](#_Toc426363933)

[Figure 21: View Repair Request 2](#_Toc426363934)

[Figure 22: Assign Repair Request 2](#_Toc426363935)

[Figure 23: Cancel Repair Request 2](#_Toc426363936)

[Figure 24: View Rental Request 2](#_Toc426363937)

[Figure 25: Assign Rental request 2](#_Toc426363938)

[Figure 26: Cancel Rental Request 2](#_Toc426363939)

[Figure 27: View Result Request Office 2](#_Toc426363940)

[Figure 28: Check Contract Due Date 2](#_Toc426363941)

[Figure 29: Suggest Request Office 2](#_Toc426363942)

[**Figure 30: Logical Database Diagram** 2](#_Toc426363943)

**List of Tables**

[Table 1: Roles and Responsibilities 2](#_Toc426363944)

[Table 2: Hardware Requirement for Server 2](#_Toc426363945)

[Table 3: Hardware Requirement for Web User 2](#_Toc426363946)

[Table 4: Requirement for mobile app 2](#_Toc426363947)

[Table 5: Role and Responsibilities 2](#_Toc426363948)

[Table 6: Software development lifecycle 2](#_Toc426363949)

[Table 7: Phase 1: Requirement Analysis 2](#_Toc426363950)

[Table 8: Phase 2: Design 2](#_Toc426363951)

[Table 9: Phase 3: Implement 2](#_Toc426363952)

[Table 10: Phase 4: Testing 2](#_Toc426363953)

[Table 11: Phase 5: Operation and maintenance 2](#_Toc426363954)

[Table 12: Overall user requirement specification 2](#_Toc426363955)

[Table 13: <Guest> Register 2](#_Toc426363956)

[Table 13: <Guest> Login 2](#_Toc426363957)

[Table 4: Search office 2](#_Toc426363958)

[Table 5: <Guest, Customer> View office detail 2](#_Toc426363959)

[Table 5: <Customer> View customer info 2](#_Toc426363960)

[Table 5: <Customer> Edit customer info 2](#_Toc426363961)

[Table 5: <Customer> Change password 2](#_Toc426363962)

[Table 5: <Customer> Create appointment request 2](#_Toc426363963)

[Table 5: <Customer> View rental request list 2](#_Toc426363964)

[Table 5: <Customer> View repair request list 2](#_Toc426363965)

[Table 5: <Customer> View list repair history 2](#_Toc426363966)

[Table 5: <Customer> Request extend contract 2](#_Toc426363967)

[Table 5: <Customer> Request return contract 2](#_Toc426363968)

[Table 5: <Customer> View rental request list 2](#_Toc426363969)

[Table 5: <Customer> View rental request list 2](#_Toc426363970)

[Table 5: <Customer> Request repair 2](#_Toc426363971)

[Table 5: <Customer> Request rental 2](#_Toc426363972)

[Table 5: <Customer> Create office request 2](#_Toc426363973)

[Table 5: <Customer> Create office request 2](#_Toc426363974)

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| ORS | Office Rental Service |
| SRS | Software Requirement Specification |
| Admin | Administrator |
| SMS | Short message service |
| Info | Information |

# Report No 1 - Introduction

## Project Information

* Project name: **Office Rental Service**
* Project Code: **ORS**
* Product Type: **Website Application**
* Start Date: **May 11th, 2015**
* End Date: **August 22nd, 2015**

## Introduction

Nowadays, a lot of companies are established every day. When new companies are established or some companies want to expand their companies, they need to find where they put their offices. To find a suitable office for them, they need you to add some equipment they need, repair some equipment… All of that thing will get you a lot of time to do it and it is very complex to make contract with those companies. With this system, you can easily to manage your offices, equipment and contracts and it takes you less time than now.

## Current Situation

In Vietnam, to find a suitable office, company would rather consider the following actions:

* If company has large resource, and require a specific requirement and equipment, they would build a new office. This will takes lots of time and money, only adapted for foreign companies.
* Go to a broker company, fill in the form and ask for help to find a suitable office. This usually takes some days to find and cost fee for broker companies.
* Search the internet for office. There are some website in Vietnam has information about the office for rent. But the information is usually not richly or sometimes outdated.

## Problem Definition

To build a new office requires a lot of time, effort, and spend a lot of costs. Besides, there are many new office building has been built recently, but had not been rented, and many company has unused office space. The matching between the demand and the supply is critical required. But using tradition approach for searching office, we have some limitations:

* There are some classifieds website that put the office information. However, the office information is lack, the search criteria is not very effective, and the information about the amenities is hardly found.
* Every company choose for themselves the best office suite with their company: location, area, office facilities, finding a best match will be take time, costs. This work usually made by a third party broker company, and it will take days to finish and we would pay a fee to that company.
* When company want to rent the office or have a rented office but conditions where infrastructure does not guarantee, the company business will be very difficult to contact the lessors to consider repairing office suite company conditions, causing company to consume resources.

## Proposed Solution

The website support customer to select appropriate and process office rental online. The website also support manager to track their business.

### Feature functions

* Our website offers many methods for customer to search office (by size, price, place, amenities…). When they find approtiate office, our staff will arrange to meet them at that office. After check around, we can make a contract right away. The contract will be tracked by our website for further support.
* If customers request repair something in their office, the website will notify our staff to make sure they will fix that problem as soon as posible.
* Our system also supports the addition of devices such as power sockets, fax machines… at the request of customers.

### Advantage and disadvantage

The advantages and disadvantages of the proposed solution:

* Advantages:
  + User friendly and specialized interface.
  + Providing search engines criteria: price, location, amenities...
  + Providing full information about the office. Support map for customers
  + Easy to make an appointment
  + When a deal is made, the contract is tracked by website, so the office state is update in real time for further search.
  + Can request for repair, or add new equipment based on the contract.
  + The repair request status will be update by notification system
  + The repair request is easily manage by automatic assign system
  + The contract making and repair status update is manage by mobile system so the staff will easy to update when go to the offices.
* Disadvantages:
  + The partner who own the office must contact us to post their information.
  + The mobile system require internet connection.
  + The function of notification system is limited.
  + The matching solution based on location and needs will take more time than search by keyword in traditional approach

## Functional Requirements

The functional requirements of the system are based on four main actor as below:

* Guess
  + Search for office base on places, category, amenities
  + View office detail
  + Register information to make an appointment
  + Login
* Customer:
  + Review office
  + Request rental appointment
  + Request repair
* Admin:
  + View list/Create/Update/Delete Manager.
  + View list/Create/Update/Delete Staff.
* Manager:
  + Approve/Cancel/Delete contract
  + Assign repair
  + View Statistic
* Staff:
  + Create contract
  + View list/Create/Update/Delete office
  + Approve/Delete rental
  + Update status request repair.
  + Approve/Delete review

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Nguyễn Trọng Tài | Project Manager | Instructor | [taint@fpt.edu.vn](mailto:taint@fpt.edu.vn) |
| 2 | Lê Xuân Tiến | Developer | Team Leader | [tienlxse60897@fpt.edu.vn](mailto:tienlxse60897@fpt.edu.vn) |
| 3 | Nguyễn Vũ Hoàng Quốc | Developer | Team Member | [quocnvhse61112@fpt.edu.vn](mailto:quocnvhse61112@fpt.edu.vn) |
| 4 | Trương Tiến Thành | Developer | Team Member | [thanhttse61052@fpt.edu.vn](mailto:thanhttse61052@fpt.edu.vn) |
| 5 | Trần Lê Tuấn | Developer | Team Member | [tuantl60350@fpt.edu.vn](mailto:tuantl60350@fpt.edu.vn) |

Table 1: Roles and Responsibilities

# Project Management Plan (PMP)



## Problem Definition



### Name of this Capstone Project

* **Official name**: Office Rental Service
* **Vietnamese name:** Dịch vụ cho thuê văn phòng
* **Abbreviation:** ORS

### Problem Abstract

Almost medium and small companies don’t build a new office. They usually look up on internet to rent office. But they can’t find a suitable office for them. Or the price to hire a third party broker company is too high. Besides, the office they rent doesn’t provide the necessary equipment and the time to repair is too long. So, our system will solve that problem. It not only helps you find a suitable office but also has other useful services.

### Project Overview

#### Current Situation

There are some current websites such as vanphongthue.com.vn, rongbay.com, timvanphong.vn, etc. All of them have some advantages and disadvantages.

* Advantages:
  + Friendly and specialized interface.
  + Providing search engines criteria.
  + Direct consultation with the manager.
* Disadvantages:
  + Only supply direct meeting, not arrange for a meeting.
  + Not regularly updated information.
  + Not support map.
  + Few pictures describing office.
  + Not provide equipment.

#### The Proposed System

The system will help customers find suitable office. If you don’t find a suitable office, you can make a request. When our system find some office that meet your requirement, it will send email for you. If you need repair office, our system will send staff to fix that problem as soon as possible. Moreover, our system also provide rental equipment for you.

In more detail, the system would contain following features:

##### Website

* Guess can search office and view detail. If they want to make appointment, they need to register.
* Customer can review office, request appointment, request office (if they can’t find ones), request rental equipment and request repair.
* Admin can manage staff and manager.
* Manager can create and delete contract. They also can assign staff to repair for customer and view statistic of system.
* Staff can view list, create, update and delete office. When staff finish repair for customer, they can report to manager. They also can approve or delete rental equipment.

##### Mobile Application

* Staff can check appointment task and update status request repair.

#### Boundaries of the System

* The system is intended for customers who want to rent office (such as businessman, medium or small company,…).
* The language of the system is Vietnamese
* The complete product includes:
  + The website, for admin, staff, customers and guest to interactive with the system.
  + Mobile application for staff to check task and update status request repair.
  + All the process involved document.

#### Development Environment

* **Hardware requirement for server computer**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (2Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Window server 2008 | Window server 2012 |
| Computer Processor | Intel® Celeron® Processor 2957U (1.40 GHz) | Intel® Core™ i3-3210 Processor |
| Computer Memory | 1GB RAM | 4GB or more |

Table 2: Hardware Requirement for Server

* **Hardware Requirement for Web User**

|  |  |  |
| --- | --- | --- |
| **Web** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | 2Mbps | 4Mbps |
| **Web Browser** | Firefox, Chrome, IE 8 | Firefox, Chrome, IE 9 |

Table 3: Hardware Requirement for Web User

All computers must be connected to the Internet.

* **Hardware requirement for mobile app**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | 2Mbps | 4 Mbps |
| Operating System | Android 4.0 | Android 4.4 |
| Hardware |  |  |
| Memory | 10MB | 10MB |

Table 4: Requirement for mobile app

* **Software requirements**
  + Operating system: Windows 7, or above;
  + Framework: Hibernate with Java Persistence;
  + Modeling Tool: Visual Paradigm Community;
  + IDE: Intellji Idea;
  + DBMS: Microsoft SQL Server;
  + Source Control: GitHub, Git for windows, Tortoise Git.

## Project organization



### Software Process Model

We choose waterfall model.

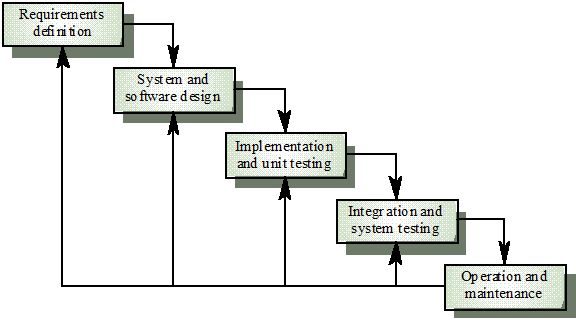


Figure 1: Software process model

This model is easy to manage and understand. For our project, we don’t have a lot of time so we use this model to help us release our project on time. This model uses for short project and it suitable for our project, which is small with 4 months and requirements are easy to clear.

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Nguyễn Trọng Tài | Supervisor/ Project manager | * Specify user requirement. * Control the development process. * Support technical and business analysis. * Review document and product application. |
| **2** | Lê Xuân Tiến | Team Leader, Developer, Tester | * Monitor process * Create project plan and distribute tasks * Clarify requirements. * Design database. * Prepare documents. * GUI Design. * Create code guide and form. * Coding. * Testing. * Deploy final product. |
| **3** | Nguyễn Vũ Hoàng Quốc | Developer, Tester | * Clarify requirements. * Prepare documents. * Review Database. * GUI Design. * Create test cases. * Coding. * Testing. |
| **4** | Trương Tiến Thành | Developer, Tester | * Clarify requirements. * Prepare documents. * Review Database. * GUI Design. * Create test cases. * Coding. * Testing. |
| **5** | Trần Lê Tuấn | Developer, Tester | * Clarify requirements. * Prepare documents. * Review Database. * GUI Design. * Create test cases. * Coding. * Testing. |

Table 5: Role and Responsibilities

### Tools and Techniques

* *Front-end*: CSS3, HTML5, JavaScript, JQuery, Ajax, AngularJS, Java Android, PhoneGap.
* *Back-end*: Java 1.7, Hibernate, Java Persistence.
* *Web Server*: Tomcat 7.
* *Developing Tool*: JetBrains IntelliJ IDEA 14, Eclipse.
* *Database Management System:* Microsoft SQL Server 2008.
* *Source Control:* Git-1.9.5-preview, TortoiseGit-1.8.14.0.
* *Modeling Tool*: Visual Paradigm 12.0 Commnunity.
* *Document Tool*: Microsoft Office 2013.

## Project management plan



### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource**  **needed** | **Dependencies**  **and Constrains** | **Risks** |
| **Requirement**  **Analysis** | - Collect requirements.  - Analyze requirements  -Identify and clarify requirements for the system in general. | -Introduction of proposed system. (report 1)  -Project Task Plan. (report 2)  -Software requirement specification. (report 3) | 40 man-days | N/A | - Missing requirement  - Unclear scope of project  - Lack of member share  - Time resource not well managed |
| **Design** | - Design database  - Architecture design for the system  - Detail design using top-down break down  - Choose Architecture style | - Software Design Document  - Base code structure  - Architecture notes  (report 4) | 74 man-days | Depend on “Requirement Analysis” | - Lack of experience.  - Not fulfil requirement. |
| **Implement** | - Implement all system functions | - Source code  - Final deploy packet | 120 man-days | - Software requirement specification.  - Software Design Document | - Lack of member experience.  - Different of member skill |
| **Testing** | - Create test plan  - Create test cases  - Perform unit test cases and system integration test | - Test plan  - Test cases  - Test report  (report 5) | 80 man-days | - Coding is finished  - Based on SRS | - Lack of experience leads to lack of test cases  - Time schedule |
| **Operation and maintenance** | - Create user’s manual, includes server deployment and end-user manual | - Report 6: System User’s Manual | 20 man-days | - Testing is finished  - Based on SRS | - Lack of experience can lead to hard-to-understand document for user  - Time deadline |

Table 6: Software development lifecycle

### Phase Detail

Below are all the major tasks that need to be performed sequentially during the development of the system.

#### Phase 1: Requirement Analysis

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Collect and analyze requirements** | Find the traditional approach for the service, and current IS system for office rental services. Analyze their strengths and weaknesses | TienLX, QuocNVH, ThanhTT, TuanTL |
| **2. Identify and clarify main functions.** | Define main flows which system will build to adapt, and main functions for system. | TienLX, QuocNVH, ThanhTT, TuanTL |
| **3. Create project management plan** | Define the overview of the project  Create project plan | TienLX, QuocNVH |

Table 7: Phase 1: Requirement Analysis

#### Phase 2: Design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Defined overall system design** | Design basis system architecture design | TienLX |
| **2. Create entity-relationship diagram** | From the defined use cases, create the ERD for the system | TienLX, QuocNVH |
| **3. Create logical database diagram** | Define the logical database diagram based on ERD | ThanhTT, TuanTL |
| **4. Create class diagrams and definition** | Create the classes with attributes and functions, along with its definition | TienLX, QuocNVH, ThanhTT, TuanTL |
| **5. Create interaction diagrams** | Include sequences diagrams and activity diagrams | TienLX, QuocNVH, ThanhTT, TuanTL |
| **6. Create interface design** | Design the user interface for the system applications | TienLX, QuocNVH, ThanhTT, TuanTL |
| **7. Define main algorithms** | Analyze and define the required algorithms for the system | TienLX, QuocNVH, ThanhTT, TuanTL |

Table 8: Phase 2: Design

#### Phase 3: Implement

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Design physical database** | Define the physical database design on SQL Server, and create database dictionary | TienLX, QuocNVH, ThanhTT, TuanTL |
| **2. Define base structure** | Define the base structure for the program | TienLX |
| **3. Coding admin functions** | Include manage account | TuanTL |
| **4. Coding manager functions** | Include manage repair list and manage contract list | ThanhTT, QuocNVH |
| **5. Coding staff functions** | Include manage office | TienLX |
| **6. Coding customer functions** |  |  |
| **6.1. Coding request office flow** | Include search office, request office, request appointment | BE: QuocNVH  FE: TienLX |
| **6.2. Coding contract signing flow** | Include the assign flow for manager and view assigned flow for staff | ThanhTT |
| **6.3. Coding contract extend/cancel flow** | Include request extend contract and cancel contract | QuocNVH |
| **6.4. Coding request repair flow** | Include request repair office for customers | TuanTL |
| **7. Coding additional mobile app** | Include view assigned job for staff, view assigned and update the repair job for staff | BE: QuocNVH, TuanTL  FE: TienLX, ThanhTT |

Table 9: Phase 3: Implement

#### Phase 4: Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Create test plan** | Create the test plan document for the system | TienLX |
| **2. Create unit testing** | Create the test cases for each functions describe in the implementation |  |
| **2.1. For admin functions** | Create test cases for functions in manage account | TienLX |
| **2.2. For manager functions** | Create test cases for functions in manage repair and contract | TuanTL |
| **2.3. For staff functions** | Create test cases for functions in manage office | ThanhTT |
| **2.4. For customer functions** | Create test cases for functions for customer | QuocNVH |
| **2.5. For mobile app** | Create test cases for functions for mobile app | TuanTL |
| **3. Testing and fix bugs** | Base on the test cases, test the functions and fix bugs | TienLX, QuocNVH, ThanhTT, TuanTL |
| **4. System integration test** | Test the overall of the system, measure the performance of the system | TienLX, QuocNVH, ThanhTT, TuanTL |

Table 10: Phase 4: Testing

#### Phase 5: Operation and maintenance

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Create installation guide** | Define the system requirement for the server and client. document installation guild | TienLX, QuocNVH |
| **2. Create User guide** | Define the user manual for operating the system | TienLX, QuocNVH, ThanhTT, TuanTL |
| **3. Maintenance and fix bugs** | Tracking the system and fix the remains bugs while installation | ThanhTT, TuanTL |

Table 11: Phase 5: Operation and maintenance

### All Meeting Minutes

Refer to Meeting Minutes folder.

## Coding Convention

**Indentation:**

* Four spaces should be used as the unit of indentation
* Tabs must be set exactly every 8 spaces
* Avoid lines longer than 80 characters
* Break after a comma.

**Declarations:**

* One declaration per line
* Put declarations only at the beginning of blocks
* No space between a method name and the parenthesis "(" starting its parameter list
* Open brace "{" appears at the end of the same line as the declaration statement
* Closing brace "}" starts a line by itself indented to match its corresponding opening statement, except when it is a null statement the "}" should appear immediately after the "{"

**White Space:**

* Between sections of a source file
* Between class and interface definitions
* Between methods

**Naming Conventions:**

* Class names should be noun
* Methods should be verbs

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

# Software Requirements Specifications (SRS)

## User Requirement Specification

The system has five actors including: guest, customer, staff, admin and system.

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Guest | Person join to website but not login into system. |
| Customer | Person who want to rent offices and logged into system |
| Admin | Person who manage account and ban/ unban account |
| Manager | Person who manage request, contact customer, schedule staff…. |
| Staff | Person who in charge of create office, receive and process the appointment/ repair/ rental/ return office request |
| System | System will manage the schedule to process contract time, send notification, suggest office for customer |

Table 12: Overall user requirement specification

### Guest requirement

* **Register:** Guest need to register to become member of system.
* **Search office:** Customer can search suitable office by criteria such as price, location, name….
* **Login**: Guest uses email and password to login into the system to search or request office.

### Customer requirement

* **Search office:** Customer can search suitable office by criteria such as price, location, name….
* **Send request:** Customer can send request about book appointment, repair something in office, rental, and request cancel before expire…
* **Rate and comment office:** Customer can rate and comment satisfaction about office they renting.

### Admin requirement

* **Logout:** When finish all activities at website they can log out of system.
* **Manager member:** Admin can add, edit, remove or ban/ unban account.

### Manager requirement

* **Contact with customer:** Manager can notify customer when task is done.
* **Manager contract:** Manager can create, edit contract.
* **Monitor request:** Manager can confirm about request repair, rental, appointment.
* **Schedule staff:** manager can schedule staff to contact with customer when request has been approved by manager.

### Staff requirement

* **Logout:** When finish all activities at website they can log out of system.
* **Contact with customer:** staff have to check task list about the appointment with customer and change status of the appointment when finish his task.
* **Monitor office:** staff can add, edit or remove comment about office.
* **Repair amenities:** staff have to check task list about the request repair amenities and change status of that request after repair complete.

### System requirement

* **Send notification mail:** system will send notification mail to customer when appointment has been approved and scheduled or request repair has been accepted.
* **Send notification SMS (short message service):** system will send notification mail to customer when their contract will be expired in one month.
* **Suggest office:** system will suggest some offices which is nearly suitable with what customer searching.
* **Schedule request:** system will auto schedule all request about appointment and repair when those request has been approved by manager.

## System Requirement Specification

### External Interface Requirements

#### User Interfaces

* User interface must be friendly, simple.
* All functions should be showed clearly and don’t make user confuse.

#### Hardware Interfaces

* The system will use the standard hardware and data communications resources of a standard computer.

#### Software Interfaces

* The website run in Firefox and Chrome browsers.

#### Communications Protocol

* The website uses:
* HTTP/HTTPS protocol for communication between the web browser and the web server.
* TCP/IP network protocol for communication with HTTP protocol.
* WAP protocol for sending message for customer.

### System Overview Use Case:

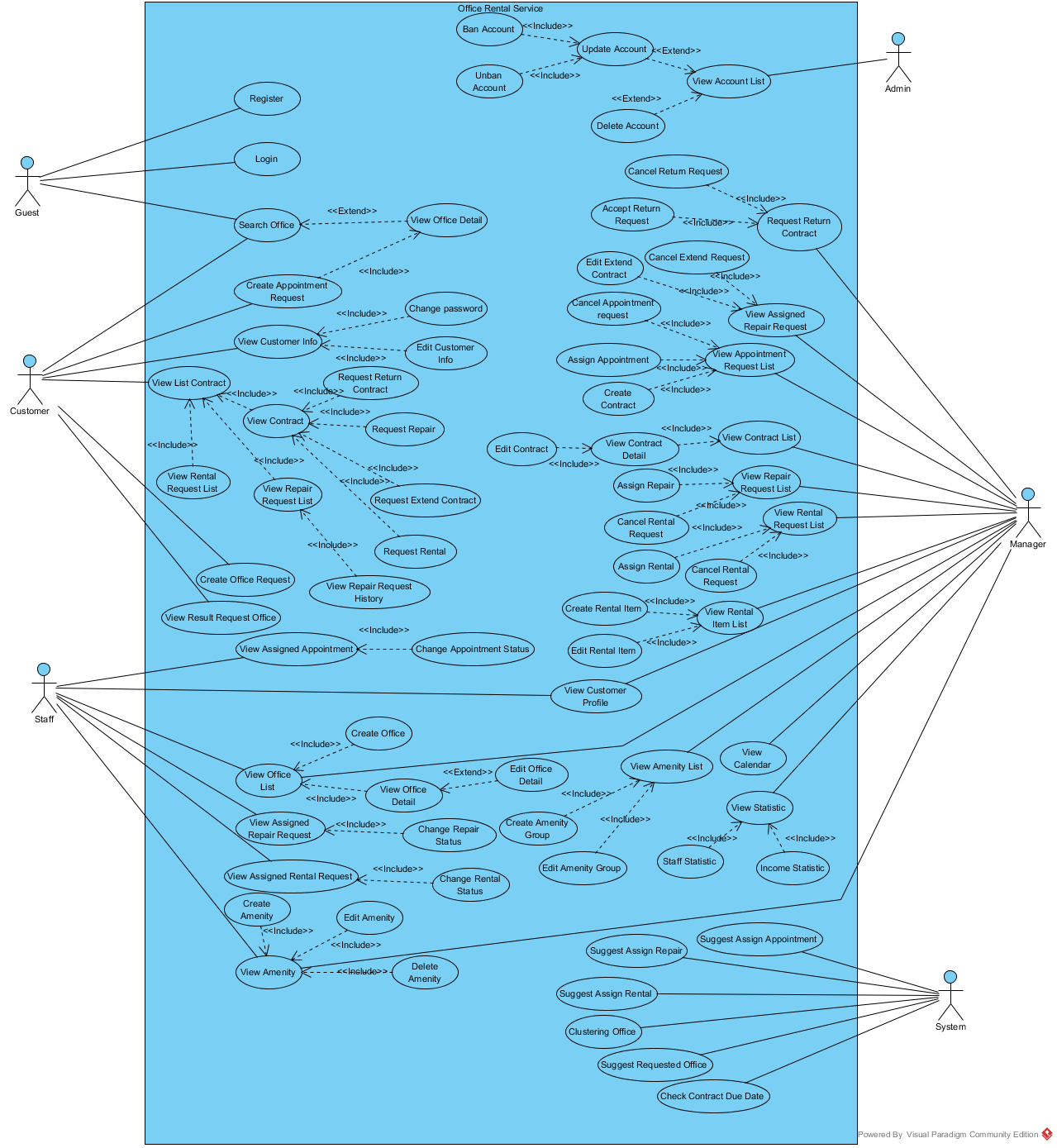


Figure 2: Use Case Overview

### List of Use Case:

#### <Guest>Overview Use Case

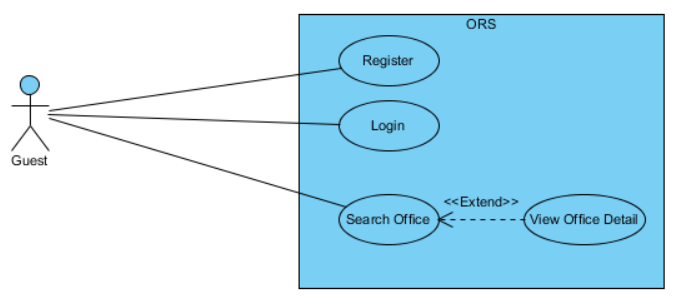


Figure 3: <Guest>Overview Use Case

##### <Guest> Register

**Use Case Diagram**



Figure 4: <Guest> Register

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – REGISTER** | | | | |
| **Use-case No.** | UC001 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Register | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Guest.   **Summary:**   * This use case allows guest create account.   **Goal:**   * Help guest to create new account.   **Trigger:**   * Click “Đăng ký” button.   **Pre-conditions:**   * N/A.   **Post conditions:**   * **Success**: The new account will be added into database. * **Failure**: Cannot create account. System displays error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Đăng ký” button. | System show [Register pop-up], contain a register form:   * Tên đăng nhập (\*): Textbox (min length: 3, max length: 50).   [Exception 1, 9, 10]   * Mật khẩu (\*): Password Filed (min length: 8, max length: 20).   [Exception 2, 10]   * E-mail (\*): Textbox (max length: 30).   In e-mail format.  [Exception 3, 10]   * Xưng hô: Drop-down list. * Họ và tên (\*): Textbox (min length: 6, max length: 50).   [Exception 4, 10]   * Công ty: Textbox (max length: 50).   [Exception 5]   * Số điện thoại (\*): Textbox (min length: 10, max length: 10).   In format: /^[0-9]\*$/  [Exception 6, 10]   * Địa chỉ: Textbox (max length: 100).   [Exception 7]   * Ngày sinh: Date time picker.   [Exception 8]   * Captcha: Checkbox. * Đăng ký: Button. * Hủy: Button. | | 2 | Input information.  Clicks “Đăng ký” button.  [Alternative 1] | Close [Register pop-up] and displays message:”Đăng kí thành công. Mời đăng nhập và kiểm tra email để hoàn tất đăng kí”. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | | 1 | Click on the “Hủy” button. | Close [Register pop-up]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | “Tên đăng nhập” is less than 3 or more than 50 characters. | Display error message “Nhập ít nhất 3 ký tự” or “Số ký tự tối đa là 50”. | | 2 | “Mật khẩu” is less than 8 or more than 20 characters. | Display error message “Nhập ít nhất 8 ký tự” or “Số ký tự tối đa là 20”. | | 3 | “E-mail” is more than 30 characters. | Display error message “Số ký tự tối đa là 30”. | | 4 | “Họ và tên” is less than 6 or more than 50 characters. | Display error message “Nhập ít nhất 6 ký tự” or “Số ký tự tối đa là 50”. | | 5 | “Công ty” is more than 50 characters. | Display error message “Số ký tự tối đa là 50”. | | 6 | “Số điện thoại” is not a number or less than 10 numbers or greater than 11 numbers. | Display error message “Nhập ít nhất 10 ký tự” or “Số ký tự tối đa là 11”. | | 7 | “Địa chỉ” is greater than 100 characters. | Display error message “Số ký tự tối đa là 100”. | | 8 | “Ngày sinh” is invalid. | Display error message “Ngày không hợp lệ”. | | 9 | “Tên đăng nhập” exist in database. | Display message “Tên đăng nhập đã được sử dụng. Hãy chọn tên khác”. | | 10 | “Tên đăng nhập”, “Mật khẩu”, “E-mail”, “Họ và tên”, “Số điện thoại” are empty. | Display error message “Bắt buộc”. |   **Relationships:** N/A.  **Business Rules**:   * The role of new created account is “Customer” and status is “Normal”. * Username must be different from existed username in system. * Use captcha to prevent spammer. * “Số điện thoại” used to contact and receive message when our website find suitable office for customer. | | | | |

Table 13: <Guest> Register

##### <Guest> Login

**Use Case Diagram**

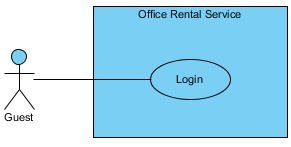


Figure 5: <Guest> Login

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – LOGIN** | | | | |
| **Use-case No.** | UC002 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Login | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Guest   **Summary:**   * This use case allows user to login to website.   **Goal:**   * Guest can login to website.   **Triggers:**   * Click “Đăng nhập” button on menu bar.   **Preconditions:**   * Guest already has an account on website.   **Post Conditions:**   * **Success:** Guest successfully login into the website. * **Failure:** Guest cannot login.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Đăng nhập” button. | System show [Login pop-up], contain a login form:   * Tên tài khoản: Textbox.   [Exception 1]   * Mật khẩu: Password Filed.   [Exception 2]   * Giữ trạng thái đăng nhập: Checkbox. * Quên mật khẩu: Link. * Đăng nhập: Button.   [Exception 3, 4]   * Hủy: Button. | | 2 | Input username and password.  Click “Đăng Nhập” button.  [Alternative 1, 2, 3] | Close [Login pop-up] and show message “Đăng nhập thành công”. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | | 1 | Click “Giữ trạng thái đăng nhập” checkbox, then click “Đăng Nhập” button. | Close [Login pop-up] and show message “Đăng nhập thành công”. | | 2 | Click “Quên mật khẩu” link. |  | | 3 | Click “Hủy” button. | Close [Login pop-up]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | “Tên tài khoản” is empty. | Display error message “Vui lòng nhập tên đăng nhập”. | | 2 | “Mật khẩu” is empty. | Display error message “Vui lòng nhập mật khẩu”. | | 3 | Cannot login due to database connection. | Display error message “Có lỗi xảy ra. Xin thử lại”. | | 4 | “Tên tài khoản” or “Mật khẩu” not correct. | Display error message “Tên đăng nhập hoặc mật khẩu không chính xác, xin thử lại”. |   **Relationships:** N/A  **Business Rules:**   * Guest can only login in website and cannot login in system page. * Staff, manager and admin can’t login in website. | | | | |

Table 14: <Guest> Login

##### <Guest, Customer> Search office

**Use Case Diagram**

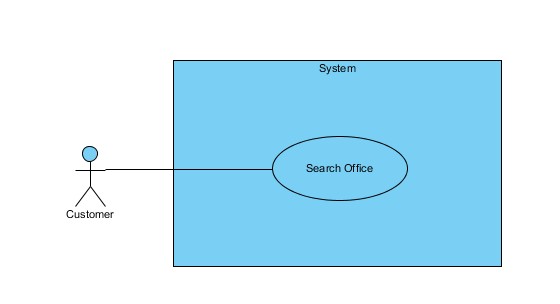


Figure 6: <Guest, Customer> Search office

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – SEARCH OFFICE** | | | | |
| **Use-case No.** | UC013 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Search office | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 18/05/2015 | **Priority** | Normal | |
| **Actor:**   * Guest * Customer   **Summary:**   * This use case allows customer search office.   **Goal:**   * Customer can find success an office.   **Trigger:**   * Fill information in text fill or choice some available information. * Click “Tìm kiếm” in home page.   **Pre-conditions:**  **Post conditions:**   * **Success:** Search office successful. * **Fail:** Show message error and show which information error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Fill information in home page. | Show error message or success message  Tên văn phòng: Textbox  Giá thuê: Dropdown list  Quận: Dropdown list  Thành phố: Dropdown list  Số người: Dropdown list | | 2 | Click “Tìm kiếm” button.  [Alternative 1] | [Exceptions 1]  Show result page. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | | 1 | Click to “Xoá trắng” button. | All information in register form is cleared. |   **Exceptions:**   |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể sửa thông tin, vui lòng thử lại sau.” |   **Relationships:** N/A  **Business Rules**: . | | | | |

Table 15: Search office

##### <Guest, Customer> View office detail

**Use Case Diagram**

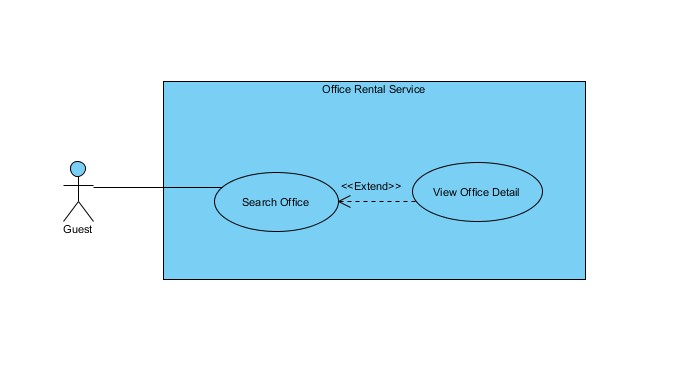


Figure 7: <Guest, Customer> View office detail

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW OFFICE DETAIL** | | | | |
| **Use-case No.** | UC014 | **Use-case Version** | | 2.0 |
| **Use-case Name** | View Office Detail | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Guest * Customer   **Summary:**   * This use case allows guest search office information.   **Goal:**   * Guest can search information.   **Triggers:**   * Fill information in the textbox or chose some suggest in search form. * Click “Tìm kiếm” button.   **Preconditions:**  **Post Conditions:**   * **Success**: Office result is showed successfully. * **Fail:** Cannot show office information.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on suggest offices. | Show office information include:  Tên văn phòng: Label.  Giới thiệu văn phòng: Text Area.  Quận: Label  Thành phố: Label  Số người: Label  Giá Thuê: Label  Thông tin thêm: Text Area.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể xem thông tin, vui lòng thử lại sau.” |   **Relationships:** Office  **Business Rules:** | | | | |

Table 16: <Guest, Customer> View office detail

#### <Customer>Overview Use Case

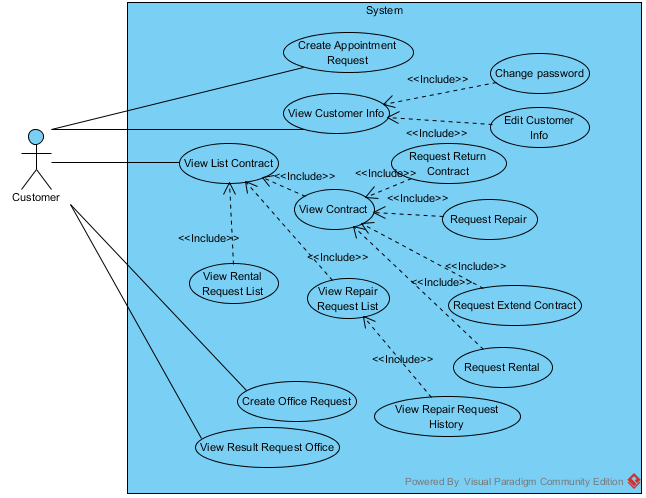
****

Figure 8: <Customer>Overview Use Case

##### <Customer> View customer info

**Use Case Diagram**

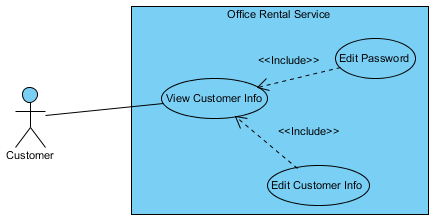


Figure 9: <Customer> View customer info

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW CUSTOMER INFO** | | | | |
| **Use-case No.** | UC003 | **Use-case Version** | | 2.0 |
| **Use-case Name** | View customer Info | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer view customer information.   **Goal:**   * Show profile of customer.   **Triggers:**   * Click “Thông tin cá nhân” in drop-down menu.   **Preconditions:**   * User must login in customer role.   **Post Conditions:**   * **Success**: Customer information is showed successfully. * **Fail:** Cannot view customer information.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Thông tin cá nhân” in drop-down menu. | Show customer information.  Display customer details includes:   * Xưng hô: Text * Họ và tên: Text * Công ty: Text * Số điện thoại: Text * Địa chỉ: Text * Ngày sinh: Text * Sửa thông tin: Button * Sửa mật khẩu: Button |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** Edit customer info.  **Business Rules:**   * Customer information must exist in database. | | | | |

Table 17: <Customer> View customer info

##### <Customer> Edit customer info

**Use Case Diagram**

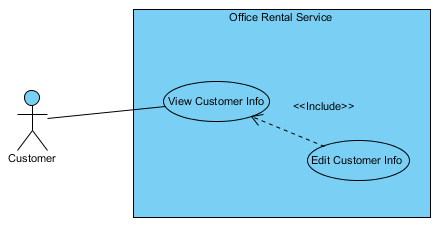


Figure 10: <Customer> Edit customer info

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – EDIT CUSTOMER INFO** | | | | |
| **Use-case No.** | UC004 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Edit customer info | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer   **Summary:**   * This use case allows customer edit information.   **Goal:**   * Customer can edit information.   **Triggers:**   * Click “Thông tin cá nhân” in drop-down menu. * Click “Sửa thông tin” button.   **Preconditions:**   * Customer must be login.   **Post Conditions:**   * **Success**: customer information is edited successfully. * **Fail:** Cannot edit customer information.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Sửa thông tin” on the screen. | Show customer information, includes:   * Họ và tên (\*): Textbox.   [Exception 1, 6]   * Công ty: Textbox.   [Exception 2]   * Số điện thoại (\*): Textbox.   [Exception 3, 6]   * Địa chỉ Email: Textbox.   [Exception 4]   * Ngày sinh: Textbox.   [Exception 5]   * Chỉnh sửa: Button. * Hủy: Button. | | 2 | Click “Chỉnh sửa” button.  [Alternative 1] | Show message “Chỉnh sửa thành công” and return to [Profile page]. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Return to [Profile page]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | “Họ và tên” is less than 6 or more than 50 characters. | Display error message “Nhập ít nhất 6 ký tự” or “Số ký tự tối đa là 50”. | | 2 | “Công ty” is more than 50 characters. | Display error message “Số ký tự tối đa là 50”. | | 3 | “Số điện thoại” is not a number or less than 10 numbers or greater than 11 numbers. | Display error message “Nhập ít nhất 10 ký tự” or “Số ký tự tối đa là 11”. | | 4 | “Địa chỉ” is more than 100 characters. | Display error message “Số ký tự tối đa là 100”. | | 5 | “Ngày sinh” is invalid. | Display error message “Ngày không hợp lệ”. | | 6 | “Họ và tên” or “Số điện thoại” is empty. | Display error message “Bắt buộc”. |   **Relationships:** View customer info  **Business Rules:**   * Only customer can edit information. | | | | |

Table 18: <Customer> Edit customer info

##### <Customer> Change password

**Use Case Diagram**

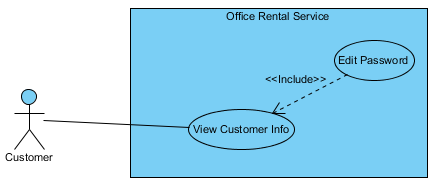
****

Figure 11: <Customer> Change password

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CHANGE PASSWORD** | | | | |
| **Use-case No.** | UC005 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Change password | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer   **Summary:**   * This use case allows customer change password.   **Goal:**   * Customer can change password.   **Triggers:**   * Click “Thông tin cá nhân” in drop-down menu. * Click “Sửa mật khẩu” button.   **Preconditions:**   * Customer must be login.   **Post Conditions:**   * **Success**: password of customer is changed successfully. * **Fail:** Cannot change password.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Sửa mật khẩu” on the screen. | Show information, includes:   * Mật khẩu hiện tại: Password field.   [Exception 1, 4]   * Mật khẩu mới: Password field.   [Exception 2, 4]   * Nhập lại mật khẩu mới: Password field.   [Exception 3, 4]   * Chỉnh sửa: Button. * Hủy: Button. | | 2 | Click “Chỉnh sửa” button.  [Alternative 1] | Show message “Chỉnh sửa thành công” and return to [Profile page]. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Return to [Profile page]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | “Mật khẩu hiện tại” is less than 8 or more than 20 characters. | Display error message “Nhập ít nhất 8 ký tự” or “Số ký tự tối đa là 20”. | | 2 | “Mật khẩu mới” is less than 8 or more than 20 characters. | Display error message “Nhập ít nhất 8 ký tự” or “Số ký tự tối đa là 20”. | | 3 | “Nhập lại mật khẩu mới” is less than 8 or more than 20 characters. | Display error message “Nhập ít nhất 8 ký tự” or “Số ký tự tối đa là 20”. | | 4 | “Mật khẩu hiện tại”, “Mật khẩu mới” or “Nhập lại mật khẩu mới” is empty. | Display error message “Bắt buộc”. |   **Relationships:** View customer info  **Business Rules:**   * Only customer can change password. | | | | |

Table 19: <Customer> Change password

##### <Customer> Create appointment request

**Use Case Diagram**

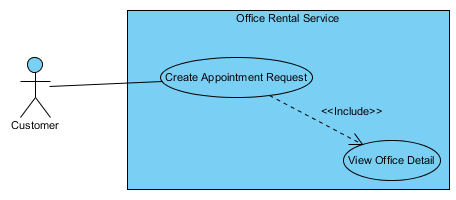


Figure 12: <Customer> Create appointment request

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CREATE APPOINTMENT REQUEST** | | | | |
| **Use-case No.** | UC006 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Create appointment request | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer request appointment.   **Goal:**   * Customer can request appointment after they found suitable office.   **Trigger:**   * After choosing suitable office, click “Gửi yêu cầu”.   **Pre-conditions:**   * Customer must be login.   **Post conditions:**   * **Success:** Request appointment successful. * **Fail:** Show message error and can’t send request.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Gửi yêu cầu” button. | Redirect to [Request page], contain:   * Thời gian [?]: Date time picker. * Đặt lịch hẹn: Button.   [Exception 1]   * Hủy: Button. | | 2 | Input date and time.  Click “Đặt lịch hẹn” button.  [Alternative 1] | Return to [Detail office page] and displays message:”Đặt lịch hẹn thành công”. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | | 1 | Click “Hủy” button. | Return to [Detail office page]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | |  | Actor action | System Response | | 1 | “Thời gian” is empty | Displays error message:”Không thành công. Có lỗi xảy ra, xin thử lại”. |   **Relationships:** View office detail.  **Business Rules**:   * Customer can request appointment which offices have status available. * Valid time from 8 am to 6 pm. * Valid date from Monday to Friday. | | | | |

Table 20: <Customer> Create appointment request

##### <Customer> View rental request list

**Use Case Diagram**



Figure 13: <Customer> View rental request list

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW RENTAL REQUEST LIST** | | | | |
| **Use-case No.** | UC007 | **Use-case Version** | | 2.0 |
| **Use-case Name** | View rental request list | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer view list rental request.   **Goal:**   * Show information about request rental of customer.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar. * Then click “Danh sách thuê thiết bị”.   **Preconditions:**   * User must login in customer role. * Customer at least has one contract and one request rental.   **Post Conditions:**   * **Success**: List request rental is showed successfully. * **Fail:** Cannot view list rental request.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Danh sách thuê thiết bị” on the screen. | Show list rental request information, includes:   * Tên thiết bị: Text. * Mô tả: Text. * Ngày tạo: Text. * Trạng thái: Text. * Ngày giao thiết bị: Text. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** View list contract.  **Business Rules:**   * Only show request has status “Đang xử lý” and “Chờ khách hàng xác nhận”. * Can’t view rental request list of other customer. | | | | |

Table 21: <Customer> View rental request list

##### <Customer> View repair request list

**Use Case Diagram**

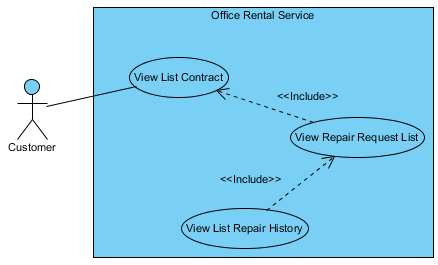


Figure 14: <Customer> View repair request list

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW REPAIR REQUEST LIST** | | | | |
| **Use-case No.** | UC008 | **Use-case Version** | | 2.0 |
| **Use-case Name** | View repair request list | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer view list repair request.   **Goal:**   * Show information about request repair of customer.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar. * Then click “Danh sách sửa chữa”.   **Preconditions:**   * User must login in customer role. * Customer at least has one contract and one request repair.   **Post Conditions:**   * **Success**: List request repair is showed successfully. * **Fail:** Cannot view list repair request.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Danh sách sửa chữa” on the screen. | Show list repair request information, includes:   * Mô tả: Text. * Ngày tạo: Text. * Tìng trạng: Text. * Ngày đã sửa chữa: Text. * Lịch sử sữa chửa: Button. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** View list contract, view list repair history.  **Business Rules:**   * Only show request has status “Đang xử lý” and “Chờ khách hàng xác nhận”. * Can’t view repair request list of other customer. | | | | |

Table 22: <Customer> View repair request list

##### <Customer> View list repair history

**Use Case Diagram**

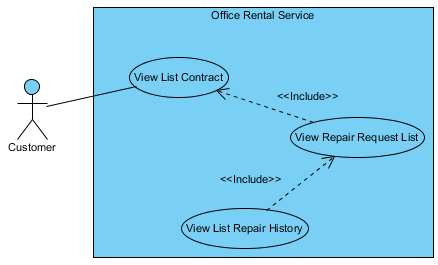


Figure 15: <Customer> View list repair history

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW LIST REPAIR HISTORY** | | | | |
| **Use-case No.** | UC009 | **Use-case Version** | | 2.0 |
| **Use-case Name** | View list repair history | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer view list repair history.   **Goal:**   * Show information about request repair which had repaired.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar. * Click “Danh sách sửa chữa”. * Click “Lịch sử sửa chữa”.   **Preconditions:**   * User must login in customer role. * Customer at least has one contract and one request repair.   **Post Conditions:**   * **Success**: List repair history is showed successfully. * **Fail:** Cannot view list repair history.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Lịch sử sửa chữa” button. | Show list request that had repaired, includes:   * Mô tả: Text. * Ngày tạo: Text. * Tìng trạng: Text. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** View repair request list.  **Business Rules:**   * Show request has status “Hoàn thành” and “Hủy”. * Can’t view repair request list of other customer. | | | | |

Table 23: <Customer> View list repair history

##### <Customer> Request extend contract

**Use Case Diagram**

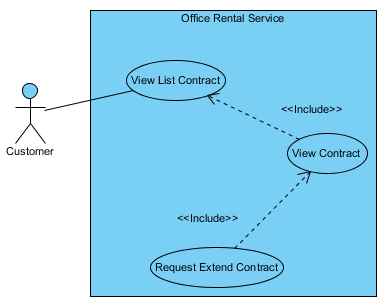
****

Figure 16: <Customer> Request extend contract

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – REQUEST EXTEND CONTRACT** | | | | |
| **Use-case No.** | UC010 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Request extend contract | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer request extend the contract.   **Goal:**   * Customer can request extend the contract if they want to continue hiring.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar. * Click “Chi tiết”. * Then click link “Gia hạn”   **Preconditions:**   * User must login in customer role. * Customer at least has one contract.   **Post Conditions:**   * **Success**: Send request extend successfully. * **Fail:** Cannot send request extend.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click link “Gia hạn” in line “Ngày hết hạn”. | System show [Contract extend pop-up], contain:   * Bạn đang yêu cầu gia hạn hợp đồng. Chúng tôi sẽ sớm liên lạc với bạn để xác nhận. Nhấn "Đồng ý" để xác nhận: Text. * Đồng ý: Button. * Hủy: Button. | | 2 | Click “Đồng ý” button.  [Alternative 1] | Close [Contract extend pop-up] and show message: “Đã gửi yêu cầu gia hạn hợp đồng” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Hủy” button. | Close [Contract extend pop-up] and return to [Contract detail page] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** View contract.  **Business Rules:**   * After receive request, manager will contact with customer and edit contract. * If customer send request extend, they can’t send request return until manager check. | | | | |

Table 24: <Customer> Request extend contract

##### <Customer> Request return contract

**Use Case Diagram**

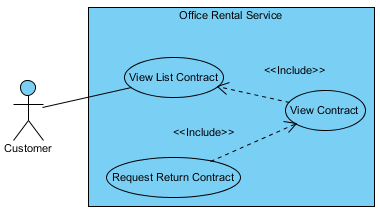
****

Figure 17: <Customer> Request return contract

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – REQUEST RETURN CONTRACT** | | | | |
| **Use-case No.** | UC011 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Request return contract | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer request return the contract.   **Goal:**   * Customer can request return the contract if they want to stop hiring.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar. * Click “Chi tiết”. * Then click link “Hủy hợp đồng trước hạn”   **Preconditions:**   * User must login in customer role. * Customer at least has one contract.   **Post Conditions:**   * **Success**: Send request return successfully. * **Fail:** Cannot send request return.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click link “Trả trước hạn” in line “Ngày hết hạn”. | System show [Contract return pop-up], contain:   * Bạn đang yêu cầu hủy hợp đồng trước hạn. Chúng tôi sẽ liên lạc với bạn để xác nhận và bàn giao văn phòng trong thời gian sớm nhất. Nhấn "Đồng ý" để xác nhận: Text. * Đồng ý: Button. * Hủy: Button. | | 2 | Click “Đồng ý” button.  [Alternative 1] | Close [Contract return pop-up] and show message: “Đã gửi yêu cầu hủy hợp đồng trước hạn”. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Hủy” button. | Close [Contract return pop-up] and return to [Contract detail page] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** View contract.  **Business Rules:**   * After staff contact with customer to check the office, manager accept the request. * If customer send request return, they can’t send request extend until manager check. | | | | |

Table 25: <Customer> Request return contract

##### <Customer> View list contract

**Use Case Diagram**

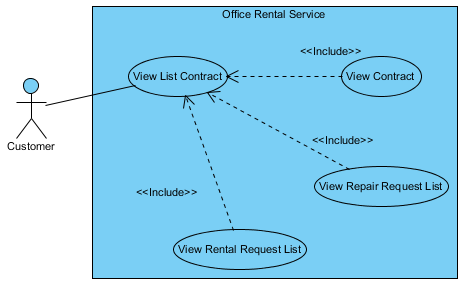


Figure 18: <Customer> View list contract

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW LIST CONTRACT** | | | | |
| **Use-case No.** | UC012 | **Use-case Version** | | 2.0 |
| **Use-case Name** | View list contract | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer view list contract.   **Goal:**   * Show information about contract of customer.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar.   **Preconditions:**   * User must login in customer role. * Customer at least has one contract.   **Post Conditions:**   * **Success**: List contract is showed successfully. * **Fail:** Cannot view list contract.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Hợp đồng của bạn” in menu bar. | Show list contract information, includes:   * Tên văn phòng: Text. * Ngày bắt đầu: Text. * Ngày kết thúc: Text. * Giá (VNĐ): Text. * Thời hạn thanh toán: Text. * Chi tiết: Button. * Danh sách sửa chữa: Button. * Danh sách thuê thiết bị: Button. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** View contract, view rental request list, view repair request list  **Business Rules:**   * Only show list contract of customer who is logging. | | | | |

Table 26: <Customer> View rental request list

##### <Customer> View contract

**Use Case Diagram**

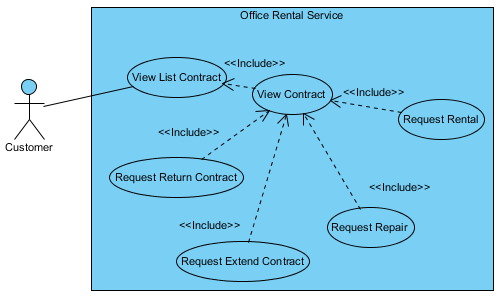


Figure 19: <Customer> View contract

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW CONTRACT** | | | | |
| **Use-case No.** | UC013 | **Use-case Version** | | 2.0 |
| **Use-case Name** | View contract | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer view detail contract.   **Goal:**   * Show information about contract of customer.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar. * Click “Chi tiết” button.   **Preconditions:**   * User must login in customer role. * Customer at least has one contract.   **Post Conditions:**   * **Success**: Contract information is showed successfully. * **Fail:** Cannot view list contract.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Chi tiết” in the screen. | Show contract information, includes:   * Tên văn phòng: Text. * Yêu cầu sửa chữa: Link. * Ngày bắt đầu: Text. * Ngày hết hạn: Text. * Hủy hợp đồng trước hạn: Link. * Gia hạn hợp đồng: Link * Điều khoản giá: Text. * Danh sách thuê tết bị hiện tại:   Tên thiết bị: Text.  Mô tả: Text.  Giá thuê (VNĐ): Text.  Số lượng (cái): Text.  Thành tiền (VNĐ): Text.  Tổng cộng (VNĐ): Text.   * Yêu câu thuê thiết bị: Button. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** Request extend contract, request return contract, request repair, request rental, view list contract.  **Business Rules:**   * Can’t view contract of other customer. * If contract expire, customer can’t view it. | | | | |

Table 27: <Customer> View rental request list

##### <Customer> Request repair

**Use Case Diagram**

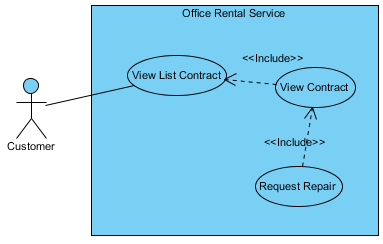


Figure 20: <Customer> Request repair

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – REQUEST REPAIR** | | | | |
| **Use-case No.** | UC014 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Request repair | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer   **Summary:**   * This use case allows customer to send request repair.   **Goal:**   * Customer can send request repair.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar. * Click “Chi tiết” button. * Then click “Yêu cầu sửa chữa” link.   **Preconditions:**   * User must login in customer role. * Customer at least has one contract.   **Post Conditions:**   * **Success**: Request repair is sent successfully. * **Fail:** Cannot send request repair.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Yêu cầu sửa chữa” link on the screen. | Show [Request repair page], includes:   * Tên văn phòng: Text. * Sữa chữa: List checkbox. * Mô tả yêu cầu sửa chữa: Text area.   [Exception 1]   * Gửi yêu cầu: Button.   [Exception 2]   * Hủy: Button. | | 2 | Click “Gửi” button on screen.  [Alternative 1] | Show message “Thành công. Đã gửi yêu cầu sửa chữa” and return to [Contract list page]. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Hủy” button. | Return to [Contract list page]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | “Mô tả yêu cầu sửa chữa” is empty. | Show error message: “Vui lòng điền mô tả”. | | 2 | Not click checkbox in “Sửa chữa”. | Show error message: “Vui lòng chọn ít nhất một tiện nghi để sửa”. |   **Relationships:** View contract  **Business Rules:**   * Status of request repair is “Chờ xử lý”. | | | | |

Table 28: <Customer> Request repair

##### <Customer> Request rental

**Use Case Diagram**

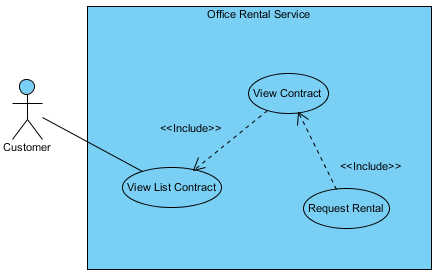


Figure 21: <Customer> Request repair

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – REQUEST REPAIR** | | | | |
| **Use-case No.** | UC015 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Request repair | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer   **Summary:**   * This use case allows customer to send request rental.   **Goal:**   * Customer can choose rental item and send request rental.   **Triggers:**   * Click “Hợp đồng của bạn” in menu bar. * Click “Chi tiết” button. * Then click “Yêu cầu thuê thiết bị” link.   **Preconditions:**   * User must login in customer role. * Customer at least has one contract.   **Post Conditions:**   * **Success**: Request rental is sent successfully. * **Fail:** Cannot send request rental.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Yêu cầu thuê thiết bị” button on the screen. | Show [Rental item page], includes:   * Xem giỏ hàng: Button. * Số lượng: Number (> 0).   [Exception 1]   * Thuê: Button. | | 2 | Choose quantity of the item you want to hire.  Click “Thuê” button on screen. | List item will update on left screen. | | 3 | Click “Xem giỏ hàng” button. | Redirect to [Request rental page]. | | 4 | Click “Gửi yêu cầu” button. | Show message “Gửi yêu cầu thành công” and redirect to [Home page]. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Input number > 0 |  |   **Relationships:** View contract  **Business Rules:**   * Status of request rental is “Chờ xử lý”. | | | | |

Table 29: <Customer> Request rental

##### <Customer> Create office request

**Use Case Diagram**

Figure 22: <Customer> Create office request

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CREATE OFFICE REQUEST** | | | | |
| **Use-case No.** | UC016 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Create office request | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows customer request office.   **Goal:**   * Customer can request office if they can’t find suitable office.   **Trigger:**   * Click “Đặt yêu cầu văn phòng” at home page.   **Pre-conditions:**   * Customer must be login.   **Post conditions:**   * **Success:** Request office successful. * **Fail:** Show message error and can’t send request.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Đặt yêu cầu văn phòng” button. | Redirect to [Request office page], contain:   * Loại văn phòng: Drop-down list. * Giá (VNĐ): Number.   [Exception 1]   * Diện tích (m2): Number.   [Exception 1]   * Quận: Textbox.   [Exception 1]   * Danh sách tiện nghi: Textbox. * Thêm: Button. * Đặt văn phòng: Button.   [Exception 2]   * Hủy: Button. | | 2 | Input information and click “Đặt văn phòng” button.  [Alternative 1] | Return to [Home page] and displays message:”Đặt lịch hẹn thành công”. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | | 1 | Click “Hủy” button. | Return to [Home page]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | |  | Actor action | System Response | | 1 | “Giá”, “Diện tích” and “Quận” are empty. | Display error message “Bắt buộc”. | | 2 | “Loại văn phòng” is empty | Button “Đặt văn phòng” is disable. |   **Relationships:** N/A  **Business Rules**:   * “Loại văn phòng”, “Giá”, “Diện tích” and “Quận” can’t empty. | | | | |

Table 30: <Customer> Create office request

##### <Customer> View Result Request Office

**Use Case Diagram**



Figure 23: <Customer> View Result Request Office

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE VIEW RESULT REQUEST OFFICE** | | | | |
| **Use-case No.** | UC025 | **Use-case Version** | | 2.0 |
| **Use-case Name** | View Result Request Office | | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | | |
| **Date** | 31/05/2015 | **Priority** | Normal | |
| **Actor:**   * Customer   **Summary:**   * This use case allows customer to see all his/her requests office and offices that is suggest for them   **Goal:**   * All request office and suggest office of customer are shown   **Triggers:**   * Click “Yêu cầu văn phòng của bạn” in menu after login.   **Preconditions:**   * Customer must have an available request office.   **Post Conditions:**   * **Success**: All request and office suggest of this customer will be shown. * **Fail:** Cannot load request and office.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Yêu cầu văn phòng của bạn” in menu after login. | Show page display all request and  office includes:   * Details of request: * Loại văn phòng [text] * Diện tích [number] * Giá [number] * Địa chỉ [text] * Ngày tạo [Date] * “Hủy yêu cầu” button * List of office that suitable of the request   [Exception 1], [Exception 2] | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot get request office | Show error message: “Có lỗi xảy ra, xin thử lại.” | | 2 | Cannot get suggest office | Show error message: “Có lỗi xảy ra, xin thử lại.” |   **Relationships:** RequestOffice  **Business Rules:**   * Customer can see all request office themselves and suggest offices that suitable with their request. * Customer also can remove their request. | | | | |

Table 31: <Customer> Create office request

#### <Staff> Overview Use Case

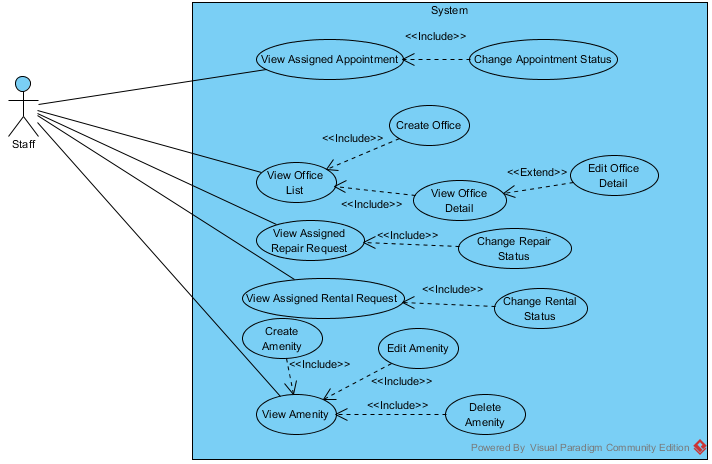
****

Figure 24: <Staff> Overview Use Case

##### <Staff>View Assigned Appointment

**Use Case Diagram**

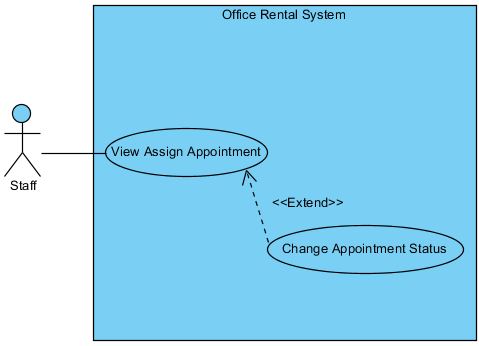
**

Figure 25: <Staff>View Assigned Appointment

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW ASSIGNED APPOINTMENT** | | | |
| **Use Case No.** | UC031 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Assigned Appointment | | |
| **Author** | Trương Tiến Thành | | |
| **Date** | 29/05/2015 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to view time to meet customer at the office they want to rent.   **Goal:**   * Staff can view time to meet customer.   **Triggers:**   * Click “Xem công việc” in menu. * Choose “Lịch gặp khách hàng” in drop down menu.   **Preconditions:**   * User must log in the system with Staff role.   **Post Conditions:**   * **Success:** Display appointment list. * **Fail:** Cannot load appointment list.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Choose “Lịch gặp khách hàng” in drop down menu at staff page. | Display all appointment with status.  [Alternative 1] [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Choose “Lịch gặp khách hàng” in drop down menu at staff page. | Display message: “Không có lịch gặp khách hàng.” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot view list appointment to database connection. | Display error message: “Không thể tải lịch gặp khách hàng. Vui lòng thử lại sau.” |   **Relationships:** Change Appointment Status  **Business Rules:**   * List of appointment is sorted by status with the status “Incomplete” on top of list. | | | |

Table 32: <Staff>View Assigned Appointment

##### <Staff> Change Appointment Status

**Use Case Diagram**

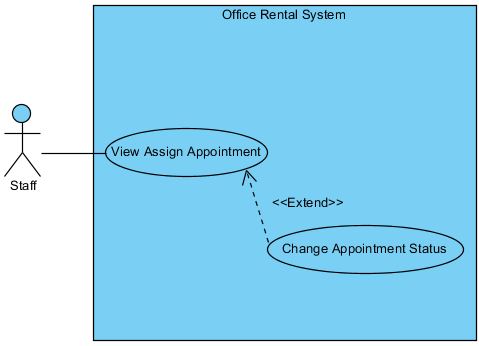
**

Figure 26: <Staff>View Assigned Appointment

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE CHANGE APPOINTMENT STATUS** | | | |
| **Use Case No.** | UC032 | **Use Case Version** | 1.0 |
| **Use Case Name** | Change Appintment Status | | |
| **Author** | Trương Tiến Thành | | |
| **Date** | 29/05/2015 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to change status of appointment when they finish meeting customer at office.   **Goal:**   * Staff change status from incomplete to complete.   **Triggers:**   * Click “Xem công việc” in menu. * Choose “Lịch gặp khách hàng” in drop down menu. * Click “Hoàn thành” button.   **Preconditions:**   * User must log in the system with Staff role. * Staff can change status after they meet customer. * At least there is one appointment with customer.   **Post Conditions:**   * **Success:** Appointment’s status is changed to “Hoàn thành”. * **Fail:** Appointment’s status isn’t updated in database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | On appointment page, clicks “Hoàn thành” button next to the appointment that he/she finished. | Display message “Đã thay đổi trang thái sang hoàn thành.”  [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot change status due to database connection. | Display error message: “Không thể thay đổi trang thái. Vui lòng thử lại sau.” |   **Relationships:** View Assigned Appointment.  **Business Rules:**   * Cannot change status “Hoàn thành” to “Chưa hoàn thành”. * Staff should notify to manage before change status. | | | |

Table 33: <Staff>View Assigned Appointment

##### <Staff>View Assigned Repair Request

**Use Case Diagram**

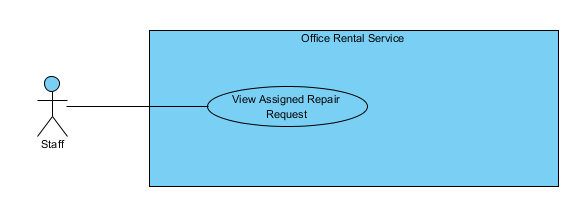


Figure 27: <Staff>View Assigned Repair Request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW ASSIGNED REPAIR REQUEST** | | | |
| **Use Case No.** | UC038 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Assigned Repair Request | | |
| **Author** | Trương Tiến Thành | | |
| **Date** | 28/05/2015 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to view task repairing that manager assign after customers request.   **Goal:**   * Staff can view task repairing.   **Triggers:**   * Click “Xem công việc” in menu. * Choose “Sửa chữa” in drop down menu.   **Preconditions:**   * User must log in the system with Staff role.   **Post Conditions:**   * **Success:** Display repair request. * **Fail:** Cannot load repair request.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Choose “Sửa chữa” in drop down menu at staff page. | Display all repair request with status in repairing page.  Display details includes:   * Tên văn phòng: Label * Khách hàng: Label * Ngày sửa chữa: Label * Mô tả: Label * “Xem chi tiết”: Link   [Alternative 1] [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Choose “Sửa chữa” in drop down menu at staff page. | Display message: “Không có yêu cầu sửa chữa nào.” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot view repair request due to database connection. | Display error message: “Không thể tải yêu cầu sửa chữa. Vui lòng thử lại sau.” |   **Relationships:** Change Repair Status  **Business Rules:**   * List of repair request is sorted by status with the status “Incomplete” on top of list. | | | |

Table 19: <Staff>View Assigned Repair Request

##### <Staff>Change Repair Status

**Use Case Diagram**

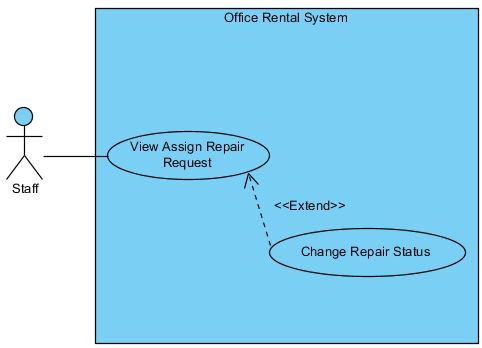
**

Figure 28: <Staff>Change Repair Status

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE CHANGE REPAIR STATUS** | | | |
| **Use Case No.** | UC039 | **Use Case Version** | 1.0 |
| **Use Case Name** | Change Repair Status | | |
| **Author** | Trương Tiến Thành | | |
| **Date** | 28/05/2015 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to change status of request when they finish repair.   **Goal:**   * Staff can change status after they complete task repairing. * Staff change status from incomplete to complete.   **Triggers:**   * Click “Xem công việc” in menu. * Choose “Sửa chữa” in drop down menu. * Click “Hoàn thành” button.   **Preconditions:**   * User must log in the system with Staff role. * At least there is one request repair from customer.   **Post Conditions:**   * **Success:** Repair request’s status is changed to “Hoàn thành”. * **Fail:** Repair request’s status isn’t updated in database.  |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Xem chi tiết” in submenu | Show page detail  Display details includes:   * Tên văn phòng: Label * Khách hàng: Label * Ngày tạo: Label * Mô tả: Label * Ngày sửa chữa: Label * Tình trạng: Label * “Đồng ý sửa” : Button * “Không đồng ý sửa” : Button | | 2 | Click “Đồng ý sửa” button  [Alternative 1] | Show successful message “Chấp nhận yêu cầu thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Không đồng ý sửa” button | Show successful message “Chập nhận yêu cầu thành công.” on screen.  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** View Assigned Repair Request  **Business Rules:**   * Cannot change status “Hoàn thành” to “Chưa hoàn thành”. * Staff should notify to manager after change status. | | | |

Table 20: <Staff>Change Repair Status

##### <Staff, Manager> Create office

**Use Case Diagram**

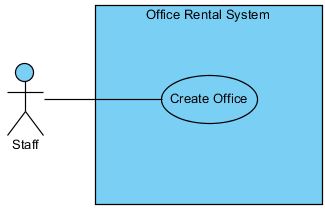


Figure 29: <Staff, Manager> Create Office

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CREATE OFFICE** | | | | |
| **Use-case No.** | UC037 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Create Office | | | |
| **Author** | Lê Xuân Tiến | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Staff * Manager   **Summary:**   * This usecase allow staff to create a new office.   **Goal**:   * Staff can add new office to system.   **Triggers:**   * Click “Thêm mới văn phòng” button.   **Preconditions:**   * User logged in successfully with Staff or Manager role.   **Post Conditions:**   * **Success:** A successful message will be displayed, new office will be added. After reloaded staff page, new row of office will be display. * **Fail:** Display error message and cannot add new office to database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Clicks “Thêm mới văn phòng” button on staff page. | Show the page with title “Thêm mới văn phòng”. The create office page includes:   * Tên văn phòng: Text box (Name, length [3, 50], required)   [Exception 2]   * Địa chỉ: Text box (Address, min length: 6, required) * Detail of address, include district and city, and the map of current adress * Hình ảnh: Add image button * List of image, each of image added has the “Xóa” button * Mô tả văn phòng: Text area (Description, length [3, 500], required)   [Exception 3]   * Loại văn phòng: Drop down list (Category, required) * Giá thuê: Text box (Price, type number, required)   [Exception 4]   * Đơn vị giá: Drop down list (PriceTerm, require) * Số tầng: Text box (Floor, type number, required)   [Exception 5]   * Tiện ích: Tags with typeahead input * Tổng diện tích: Text box (Area, type number, min: 50, required)   [Exception 6, 7]   * Tạo mới: button   [Exception 1, 8]   * Hủy: button | | 2 | Inputs information into all fields and clicks “Tạo mới” button.  [Alternative 1] | Show successfully message: “Đã thêm thành công văn phòng!” and redirect to staff page. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff clicks “Hủy” button. | The data in all fields will be reset and redirect to staff page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | “Tên văn phòng”, “Địa chỉ”, “Hình ảnh”, “Trạng thái”, “Mô tả văn phòng”, “Giá thuê”, “Đơn vị giá”, “Số tầng”, “Loại văn phòng”, “Tiện ích”, “Tổng diện tích” are empty. | Shows error message “Vui lòng nhập đầy đủ thông tin yêu cầu!” | | 2 | Name is less than 3 characters or more than 50 characters. | Shows error message “Tên văn phòng từ 3 đến 50 ký tự!” | | 3 | Description is less than 3 characters or more than 500 characters. | Shows error message “Mổ tả văn phòng từ 3 đến 500 ký tự!” | | 4 | Price is not a number. | Shows error message “Giá thuê không hợp lệ!” | | 5 | Floor is not a number. | Shows error message “Số tầng không hợp lệ!” | | 6 | Area is smaller than 50. | Shows error message “Tổng diện tích tối thiểu cho thuê là 50m2!” | | 7 | Area is not a number. | Shows error message “Tổng diện tích không hợp lệ!” | | 8 | Cannot create new office | Shows error message “Không thể thêm mới văn phòng. Vui lòng thử lại sau” | | 9 |  |  |   **Relationships:** View Office Detail  **Business Rules:**   * When select office type “Văn phòng nguyên căn”, the price term available “Nguyên căn” and “Liên hệ”, while type “Tòa nhà văn phòng”, the price term available: “/m2” , “/người”, and “Liên hệ” * Staffs can add new office to system when there is a new office for lease contact with us. * Office name should be different from existed office of system. * Office information should be clearly and correctly. | | | | |

Table 30: <Staff>Create Office

##### <Staff, Manager>View Office Detail

**Use Case Diagram**

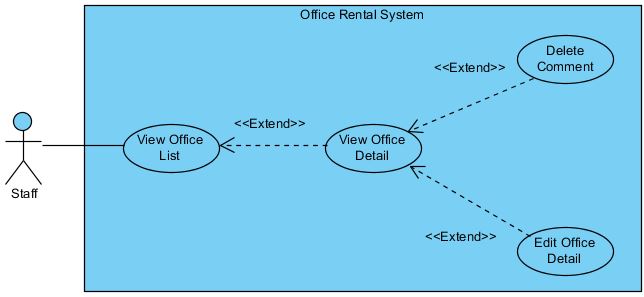
**

Figure 31: <Staff, Manager>View office detail

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW OFFICE DETAIL** | | | |
| **Use Case No.** | UC034 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Office Detail | | |
| **Author** | Trương Tiến Thành | | |
| **Date** | 28/05/2015 | **Priority** | Normal |
| **Actor:**   * Staff, manager   **Summary:**   * This use case allows staff or manager to view information of office.   **Goal:**   * Staff or manager can view information of office.   **Triggers:**   * Click “Xem văn phòng” in menu. * Click “Xem chi tiết văn phòng” button.   **Preconditions:**   * User must log in the system with Staff role. * At least there is one office in database.   **Post Conditions:**   * **Success:** Display detail information of office. * **Fail:** Cannot load information of office.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Xem chi tiết văn phòng” button next to the office that user want to view. | Display office detail page.  [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot view detail of office due to database connection. | Display error message: “Không thể tải trang chi tiết văn phòng. Vui lòng thử lại sau.” |   **Relationships:** View Office List, Edit Office Detail, Delete Comment.  **Business Rules:**   * The display is different from the customer view | | | |

Table 16: <Staff, Manager>View Office Detail

##### <Staff, Manager>Edit Office Detail

**Use Case Diagram**

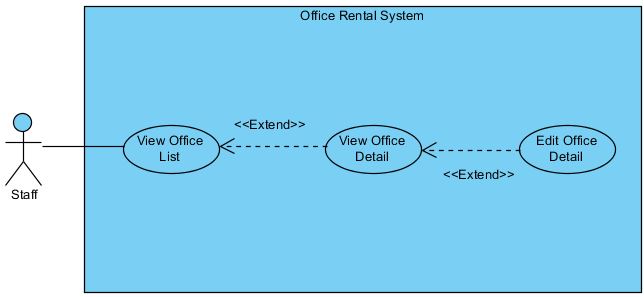


Figure 32: <Staff, Manager>Edit Office Detail

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE EDIT OFFICE DETAIL** | | | |
| **Use Case No.** | UC036 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Office Detail | | |
| **Author** | Trương Tiến Thành | | |
| **Date** | 28/05/2015 | **Priority** | Normal |
| **Actor:**   * Staff, Manager   **Summary:**   * This use case is used to edit information of office.   **Goal:**   * Staff or manager can edit information of a office.   **Triggers:**   * Click “Xem văn phòng” in menu. * Click “Xem chi tiết văn phòng” button. * Click “Cập nhật văn phòng” button   **Preconditions:**   * User must be logged in the system with Staff/Manager role. * At least there is one office in database.   **Post Conditions:**   * **Success:** New information of office is updated. * **Fail:** Cannot update information of office.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Clicks “Cập nhật văn phòng” button on the row of office that he/she wants to edit. | Redirect to update office page. It includes:   * Tên văn phòng: Text box (Name, length [3, 50], required)   [Exception 2]   * Địa chỉ: Text box (Address, min length: 6, required). * Quận: Drop down list (District, required) * Hình ảnh: Button (Image] * Trạng thái: Drop down list (Status, required) * Mô tả văn phòng: Text area (Description, length [3, 500], required)   [Exception 3]   * Giá thuê: Text box (Price, type number, required)   [Exception 4]   * Đơn vị giá: Drop down list (PriceTerm, require) * Số tầng: Text box (Floor, type number, required)   [Exception 5]   * Loại văn phòng: Drop down list (Category, required) * Tiện ích: Check box (Amenity) * Tổng diện tích: Text box (Area, type number, min length: 50, required)   [Exception 6, 7]   * Cập nhật: button   [Exception 1, 8]   * Hủy: button | | 3 | Edit information and clicks on “Cập nhật” button.  [Alternative 1] | Show popup:   * “Cập nhập thành công” message. * “Đóng” button | | 4 | Clicks on “Đóng” button | Redirect to office detail page. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Clicks on “Hủy” button | Redirect to office detail page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot update office. | Shows error message “Không thể cập nhật văn phòng. Vui lòng thử lại sau” | | 2 | Name is less than 3 characters or more than 50 characters. | Shows error message “Tên văn phòng từ 3 đến 50 ký tự!”. | | 3 | Description is less than 3 characters or more than 500 characters. | Shows error message “Mổ tả văn phòng từ 3 đến 500 ký tự!”. | | 4 | Price is not a number. | Shows error message “Giá thuê không hợp lệ!” | | 5 | Floor is not a number. | Shows error message “Số tầng không hợp lệ!” | | 6 | Area is smaller than 50. | Shows error message “Tổng diện tích tối thiểu cho thuê là 50m2!” | | 7 | Area is not a number. | Shows error message “Tổng diện tích không hợp lệ!” | | 8 | “Tên văn phòng”, “Địa chỉ”, “Hình ảnh”, “Trạng thái”, “Mô tả văn phòng”, “Giá thuê”, “Đơn vị giá”, “Số tầng”, “Loại văn phòng”, “Tiện ích”, “Tổng diện tích” are empty. | Shows error message “Vui lòng nhập đầy đủ thông tin yêu cầu!” |   **Relationships:** View Office Detail  **Business Rules:**   * Information of office should be clearly and correctly. | | | |

Table 18: <Staff, Manager>Edit Office Detail

##### <Manager, Staff, Admin> Logout

**Use Case Diagram**

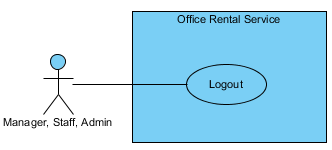


Figure 33: <Manager, Staff, Admin> Logout

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – LOGIN** | | | | |
| **Use-case No.** | UC017 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Logout | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Manager, Staff, Admin   **Summary:**   * This use case allows manager, staff or admin to logout.   **Goal:**   * Manager, staff or admin can logout.   **Triggers:**   * Click “Đăng xuất” button on drop-down menu.   **Preconditions:**   * Manager, staff or admin already login on system.   **Post Conditions:**   * **Success:** Manager, staff or admin successfully logout. * **Failure:** Manager, staff or admin cannot logout.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Đăng xuất” button. | Return to [Admin login page]. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** N/A  **Business Rules:** N/A | | | | |

Table 18: <Manager, Staff, Admin> Logout

##### <Manager, Staff> Edit rental item

**Use Case Diagram**

Figure 34: <Manager, Staff> Edit rental item

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – EDIT RENTAL ITEM** | | | | |
| **Use-case No.** | UC019 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Edit rental item | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer   **Summary:**   * This use case allows customer edit rental item.   **Goal:**   * Customer can edit rental item.   **Triggers:**   * Click “Quản lý thiết bị” in menu. * Click “Chỉnh sửa” link.   **Preconditions:**   * User must login in customer or staff role. * At least has one item.   **Post Conditions:**   * **Success**: Rental item information is edited successfully. * **Fail:** Cannot edit rental item information.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Chỉnh sửa” on the screen. | Show rental item information, includes:   * Tên: Textbox. * Mô tả: Text area. * Hình ảnh: Textbox. * Giá: Textbox. * Số lượng: Textbox. * Lưu: Button. * Quay về: Button. | | 2 | Click “Lưu” button.  [Alternative 1] | Return to [List rental item page]. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Quay về” button. | Return to [List rental item page]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Relationships:** View rental item list  **Business Rules:**   * Only manager and staff can edit rental item. | | | | |

Table 18: <Manager, Staff> Edit customer info

##### <Manager, Staff> Create rental item

**Use Case Diagram**

Figure 35: <Manager, Staff> Create rental item

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CREATE RENTAL ITEM** | | | | |
| **Use-case No.** | UC020 | **Use-case Version** | | 2.0 |
| **Use-case Name** | Create rental item | | | |
| **Author** | Trương Tiến Thành | | | |
| **Date** | 20/07/2015 | **Priority** | Normal | |
| **Actor:**   * Customer.   **Summary:**   * This use case allows manager or staff add rental item.   **Goal:**   * Manager or staff can add new rental item.   **Trigger:**   * Click “Quản lý thiết bị” in menu. * Click “Thêm mới thiết bị” link.   **Pre-conditions:**   * User must login in customer or staff role.   **Post conditions:**   * **Success:** Add new rental item successful. * **Fail:** Can’t add rental item to database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Thêm mới thiết bị” on the srceen. | Redirect to [New rentalItem page], contain:   * Tên: Textbox.   [Exception 1]   * Mô tả: Text area.   [Exception 1]   * Hình ảnh: Textbox. * Giá: Textbox.   [Exception 1]   * Số lượng: Textbox.   [Exception 1]   * Tạo mới: Button. * Hủy: Button. | | 2 | Input information and click “Tạo mới” button.  [Alternative 1] | Return to [List rental item page]. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | | 1 | Click “Hủy” button. | Return to [List rental item page]. |   **Exceptions:**   |  |  |  | | --- | --- | --- | |  | Actor action | System Response | | 1 | “Tên”, “Mô tả”, “Giá” or “Số lượng” are empty | Displays message:”Vui long điền vào trường này”. |   **Relationships:** View rental item list.  **Business Rules**:   * Only manager and staff can edit rental item. | | | | |

Table 18: <Manager, Staff> Create rental item

#### <Admin> Overview Use Case

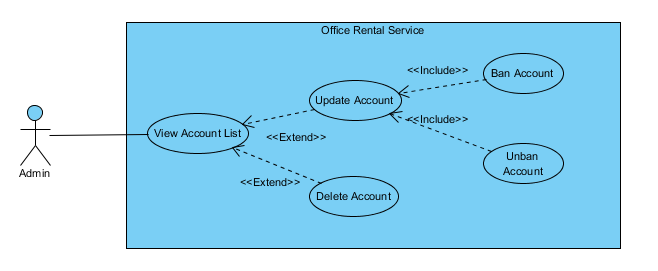


Figure 36: <Admin> Overview Use Case

##### <Admin> View account list

**Use Case Diagram**

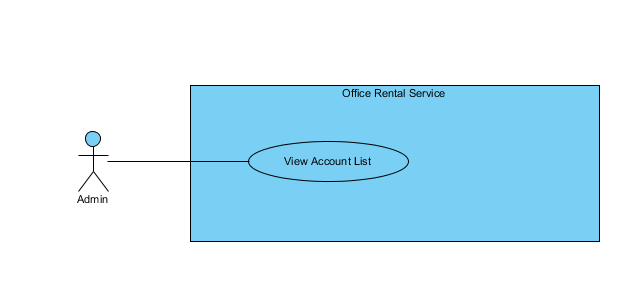


Figure 37: <Admin> View account list

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW ACCOUNT LIST** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | View Account List | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to view account list.   **Goal:**   * Admin can view account list successful.   **Triggers:**   * Click “Người dùng” in menu.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: create view account list successfully. * **Fail:** Cannot view account list   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Người dùng” on menu. | Show page account management  Display account details includes:   * Thêm mới tài khoản: Link * Tên đăng nhập: Label * Email: Password * Chức vụ: Dropdown list * Tình trạng: Label * Tác vụ:   + “Xoá”: Button  +”Sửa”: Button  [Exception 1] | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Role Account  **Business Rules:** | | | | |

Table 16: <Admin> View account list

##### <Admin> Ban account

**Use Case Diagram**

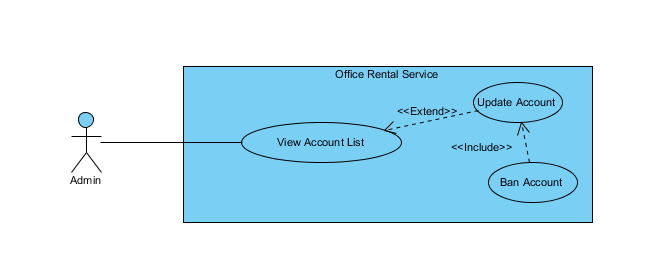


Figure 38: <Admin> Ban account

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – BAN ACCOUNT** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Request repair | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to ban an account.   **Goal:**   * Admin can ban an account successful.   **Triggers:**   * Click “Người dùng” in menu. * Click “Sửa” icon in menu. * Click “Khoá tài khoản” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: account has been banned successfully. * **Fail:** Cannot ban account.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Sửa” icon on submenu. | Show page account detail  Display account details includes:   * Tên tài khoản: Label * Mật khẩu: Password * Email: Textbox * Chức vụ: Dropdown list   “Huỷ” : button  “Cập nhật” : button  “Khoá tài khoản” : button | | 2 | Click “Khoá tài khoản” button on screen.  [Alternative 1] | Show successful message “Khoá tài khoản thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Close page and go to account management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Role Account  **Business Rules:** | | | | |

Table 16: <Admin> Ban account

##### <Admin> Create account

**Use Case Diagram**

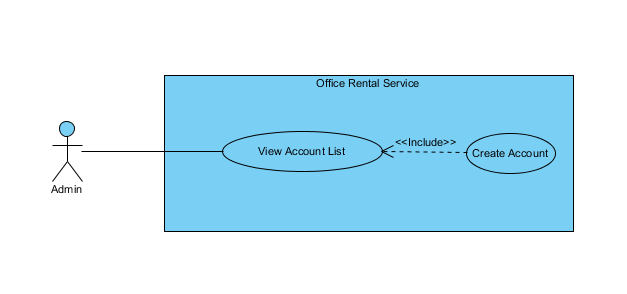


Figure 39: Create account use case diagram

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CREATE ACCOUNT** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Create Account | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to create account.   **Goal:**   * Admin can create account successful.   **Triggers:**   * Click “Người dùng” in menu. * Click “Thêm mới tài khoản” in menu.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: create account successfully. * **Fail:** Cannot create account   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Người dùng” on menu. | Show page account management  Display account details includes:   * Thêm mới tài khoản: Link * Tên đăng nhập: Label * Email: Password * Chức vụ: Dropdown list * Tình trạng: Label * Tác vụ:   + “Xoá”: Button  +”Sửa”: Button | | 2 | Click “Thêm mới tài khoản” on menu. | Show page account management  Display account details includes:   * Tên tài khoản: Label * Mật khẩu: Password * Email: Textbox * Chức vụ: Dropdown list   “Huỷ” : Button  “Tạo mới” : Button | | 3 | Click “Tạo mới” button on screen.  [Alternative 1] | Show successful message “Tạo tài khoản thành công.” on screen.  [Exception 1] | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Close page and go to account management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Role Account  **Business Rules:** | | | | |

Table 16: <Admin> Create Account

##### <Admin> Delete account

**Use Case Diagram**

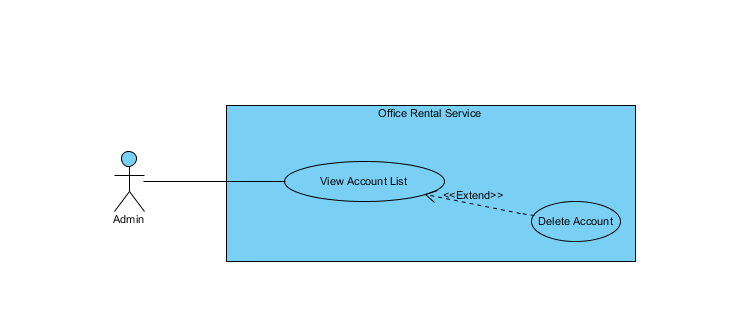


Figure 40: Delete account use case diagram

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – DELETE USER** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Delete user | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to delete account.   **Goal:**   * Admin can delete account successful.   **Triggers:**   * Click “Người dùng” in menu. * Click “Xoá tài khoản” in menu.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: delete account successfully. * **Fail:** Cannot delete account   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Người dùng” on menu. | Show page account management  Display account details includes:   * Tên đăng nhập: Label * Email: Label * Chức vụ: Label * Tình trạng: Label * Tác vụ: Label   “Xoá” : Button  “Sửa” : Button | | 2 | Click “Xoá” button on screen. | Show successful message “Xoá tài khoản thành công.” on screen.  [Exception 1] | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Role Account  **Business Rules:** | | | | |

Table 16: <Admin>Delete Account

##### <Admin> Update account

**Use Case Diagram**

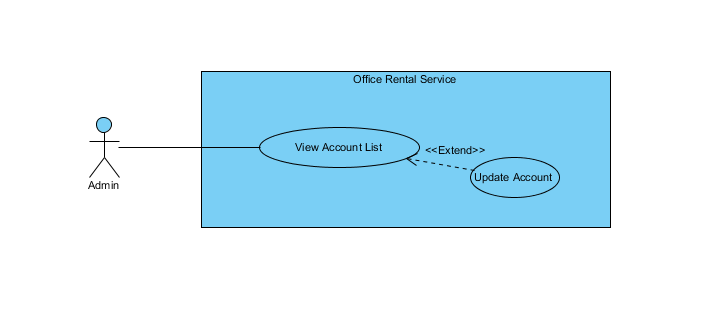


Figure 41: Update account use case diagram

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – UPDATE ACCOUNT** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Update Account | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to edit account.   **Goal:**   * Admin can edit account successful.   **Triggers:**   * Click “Người dùng” in menu. * Click “Sửa” in menu.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: create account successfully. * **Fail:** Cannot create account   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Người dùng” on menu. | Show page account management  Display account details includes:   * Tên đăng nhập: Label * Email: Label * Chức vụ: Label * Tình trạng: Label * Tác vụ: Label   “Xoá” : Button  “Sửa” : Button | | 2 | Click “Sửa” on menu. | Show page account management  Display account details includes:   * Tên tài khoản: Label * Mật khẩu: Password * Email: Textbox * Chức vụ: Dropdown list   “Huỷ” : Button  “Tạo mới” : Button | | 3 | Click “Cập nhật” button on screen.  [Alternative 1] | Show successful message “Cập nhật tài khoản thành công.” on screen.  [Exception 1] | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Close page and go to account management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Role Account  **Business Rules:** | | | | |

Table 16: <Admin> Update account

##### <Admin> Unban account

**Use Case Diagram**

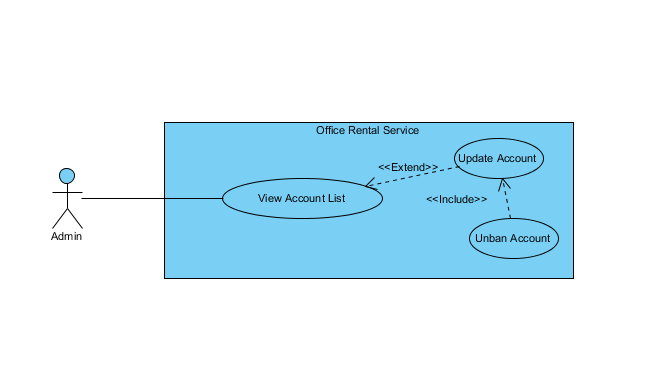


Figure 42: Unban account use case diagram

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – UNBAN ACCOUNT** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Unban Account | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to unban an account.   **Goal:**   * Admin can unban an account successful.   **Triggers:**   * Click “Người dùng” in menu. * Click “Sửa” icon in menu. * Click “Mở tài khoản” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: account has been banned successfully. * **Fail:** Cannot ban account.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Người dùng” on menu. | Show page account management  Display account details includes:   * Tên đăng nhập: Label * Email: Label * Chức vụ: Label * Tình trạng: Label * Tác vụ: Label   “Xoá” : Button  “Sửa” : Button | | 2 | Click “Sửa” icon on submenu. | Show page account detail  Display account details includes:   * Tên tài khoản: Label * Mật khẩu: Password * Email: Textbox * Chức vụ: Dropdown list   “Huỷ” : button  “Cập nhật” : button  “Mở tài khoản” : button | | 3 | Click “Mở tài khoản” button on screen.  [Alternative 1] | Show successful message “Mở tài khoản thành công.” on screen.  [Exception 1] | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Close page and go to account management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Role Account  **Business Rules:** | | | | |

Table 16: <Admin> Unban account

#### <Manager> Overview Use Case

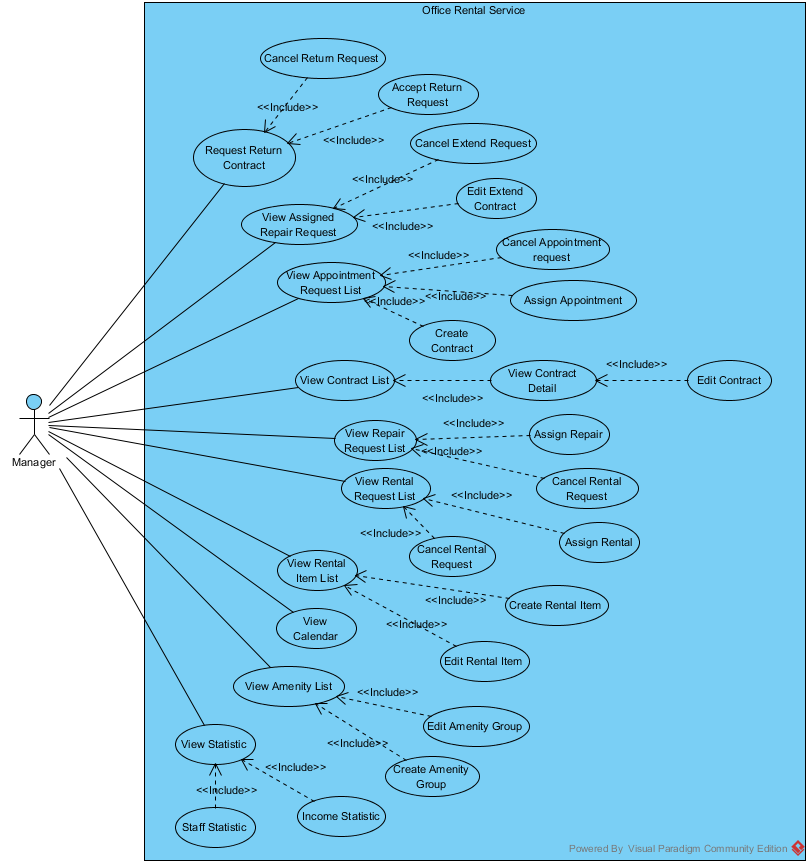
****

Figure 43: <Manager> Overview Use Case

##### <Manager> View request extend contract

**Use Case Diagram**

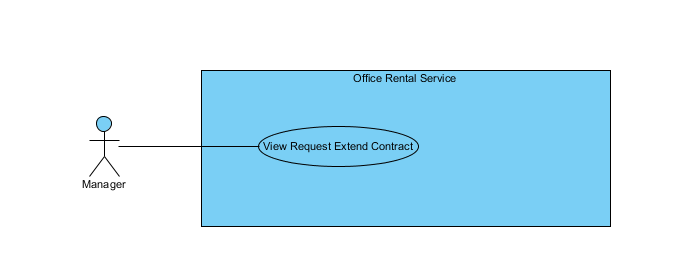


Figure 44: View request extend contract

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW REQUEST EXTEND CONTRACT** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | View Request Extend Contract | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager   **Summary:**   * This use case allows user to view request extend contract list.   **Goal:**   * User can view request extend contract list successful.   **Triggers:**   * Click “Yêu cầu khách hàng” in menu. * Click “Gia hạn” in submenu   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Request extend contract list has been viewed successfully. * **Fail:** Cannot view request extend contract list.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Hợp đồng” on menu. |  | | 2 | Click “Gia hạn” in submenu | Show page detail  Display details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày bắt đầu: Label * Ngày kết thúc: Label * Giá: Label * Thời hạn thanh toán * “Xem chi tiết”: Link   [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Contract  **Business Rules:** | | | | |

Table 16: <Manager> View Request extend contract

##### <Manager> Cancel extend request

**Use Case Diagram**

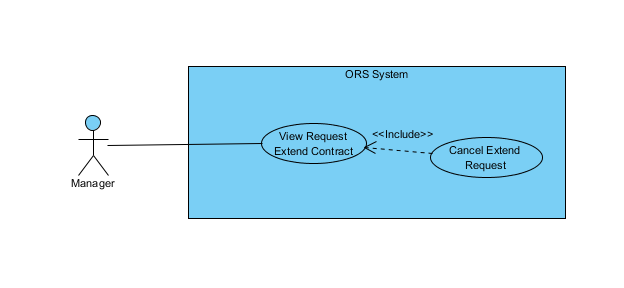


Figure 45: Cancel extend request use case diagram

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CANCEL EXTEND REQUEST** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Cancel Extend Request | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager to cancel request.   **Goal:**   * Manager can cancel request successful.   **Triggers:**   * Click “Yêu cầu của khách hàng” in menu. * Click “Xem chi tiết” icon in menu. * Click “Huỷ yêu cầu” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: contract has been canceled successfully. * **Fail:** Cannot cancel request.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Yêu cầu khách hàng” on menu. |  | | 2 | Click “Gia hạn” on submenu. | Show page request detail  Display request details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày bắt đầu: Label * Ngày kết thúc: Label * Giá: Label * Thời hạn thanh toán: Label * Xem chi tiết: Link | | 3 | Click “Xem chi tiết” link on screen. | Display request details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày kết thúc: Textbox * Thời hạn thanh toán: Dropdown list * Giá: Textbox   “Xác nhận” : button  “Huỷ yêu cầu” : button  “Quay về” : button | | 4 | Click “Huỷ yêu cầu” button on screen.  [Alternative 1] | Show successful message “Huỷ yêu cầu thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Quay về” button. | Close request page and go to management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Contract  **Business Rules:** | | | | |

Table 16: <Manager> Cancel extend request

##### <Manager> Accept return contract

**Use Case Diagram**

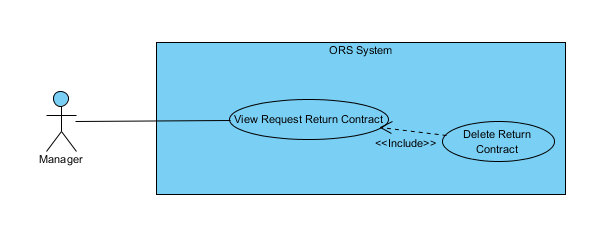


Figure 46: <Manager> Accept return contract

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE –ACCEPT RETURN CONTRACT** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Accept Return Contract | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager to delete contract.   **Goal:**   * Manager can delete contract successful.   **Triggers:**   * Click “Yêu cầu của khách hàng” in menu. * Click “Xem chi tiết” icon in menu. * Click “Xác nhận” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: contract has been deleted successfully. * **Fail:** Cannot delete contract.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Yêu cầu khách hàng” on menu. |  | | 2 | Click “Huỷ trước hạn” on submenu. | Show page request detail  Display request details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày bắt đầu: Label * Ngày kết thúc: Label * Giá: Label * Thời hạn thanh toán: Label * Xem chi tiết: Link | | 3 | Click “Xem chi tiết” link on screen. | Display request details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày kết thúc: Textbox * Thời hạn thanh toán: Dropdown list * Giá: Textbox   “Xác nhận” : button  “Huỷ yêu cầu” : button  “Quay về” : button | | 4 | Click “Xác nhận” button on screen.  [Alternative 1] | Show successful message “xoá thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Quay về” button. | Close request page and go to management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Contract  **Business Rules:** | | | | |

Table 16: <Manager> Accept return contract

##### <Manager> View request return contract

**Use Case Diagram**

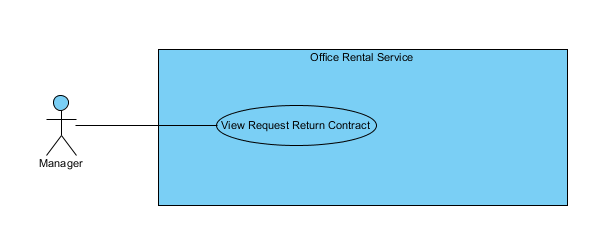


Figure 47: <Manager> View request return contract

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW REQUEST RETURN CONTRACT** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | View Request Return Contract | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager   **Summary:**   * This use case allows user to view request return contract list.   **Goal:**   * User can view request return contract list successful.   **Triggers:**   * Click “Yêu cầu khách hàng” in menu. * Click “Huỷ trước hạn” in submenu   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Return request contract list has been viewed successfully. * **Fail:** Cannot view return request contract list.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Hợp đồng” on menu. |  | | 2 | Click “Huỷ trước hạn” in submenu | Show page detail  Display details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày bắt đầu: Label * Ngày kết thúc: Label * Giá: Label * Thời hạn thanh toán: Label * “Xem chi tiet”: Link   [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Contract  **Business Rules:** | | | | |

Table 16: <Manager> View request return contract

##### <Manager> Cancel return request

**Use Case Diagram**

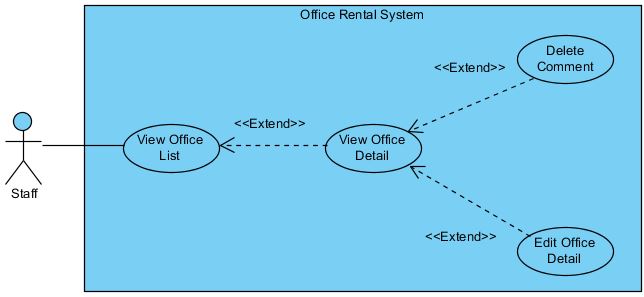
**

Figure 48: <Manager> Cancel return request

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CANCEL RETURN REQUEST** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Cancel Return Request | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager to cancel request.   **Goal:**   * Manager can cancel request successful.   **Triggers:**   * Click “Yêu cầu của khách hàng” in menu. * Click “Xem chi tiết” icon in menu. * Click “Huỷ yêu cầu” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: contract has been canceled successfully. * **Fail:** Cannot cancel request.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Yêu cầu khách hàng” on menu. |  | | 2 | Click “Huỷ trước hạn” on submenu. | Show page request detail  Display request details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày bắt đầu: Label * Ngày kết thúc: Label * Giá: Label * Thời hạn thanh toán: Label * Xem chi tiết: Link | | 3 | Click “Xem chi tiết” link on screen. | Display request details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày kết thúc: Textbox * Thời hạn thanh toán: Dropdown list * Giá: Textbox   “Xác nhận” : button  “Huỷ yêu cầu” : button  “Quay về” : button | | 4 | Click “Huỷ yêu cầu” button on screen.  [Alternative 1] | Show successful message “Huỷ yêu cầu thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Quay về” button. | Close request page and go to management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Contract  **Business Rules:** | | | | |

Table 16: <Manager> Cancel return request

##### <Manager> Edit extend contract

**Use Case Diagram**

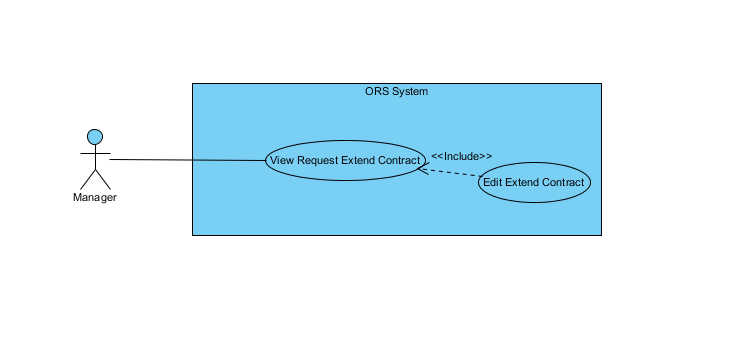


Figure 49: <Manager> Edit extend contract

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – EDIT EXTEND CONTRACT** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Delete Extend Contract | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager to edit extend contract.   **Goal:**   * Manager can edit extend contract successful.   **Triggers:**   * Click “Yêu cầu của khách hàng” in menu. * Click “Xem chi tiết” icon in menu. * Click “Xác nhận” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: contract has been edited successfully. * **Fail:** Cannot edit contract.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Yêu cầu khách hàng” on menu. |  | | 2 | Click “Huỷ trước hạn” on submenu. | Show page request detail  Display request details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày bắt đầu: Label * Ngày kết thúc: Label * Giá: Label * Thời hạn thanh toán: Label * Xem chi tiết: Link | | 3 | Click “Xem chi tiết” link on screen. | Display request details includes:   * Khách hàng: Label * Văn phòng: Label * Ngày kết thúc: Textbox * Thời hạn thanh toán: Dropdown list * Giá: Textbox   “Xác nhận” : button  “Huỷ yêu cầu” : button  “Quay về” : button | | 4 | Click “Xác nhận” button on screen.  [Alternative 1] | Show successful message “Gia hạn thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Quay về” button. | Close request page and go to management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Contract  **Business Rules:** | | | | |

Table 16: <Manager> Edit extend contract

##### <Manager> View Appointment Request List

**Use Case Diagram**

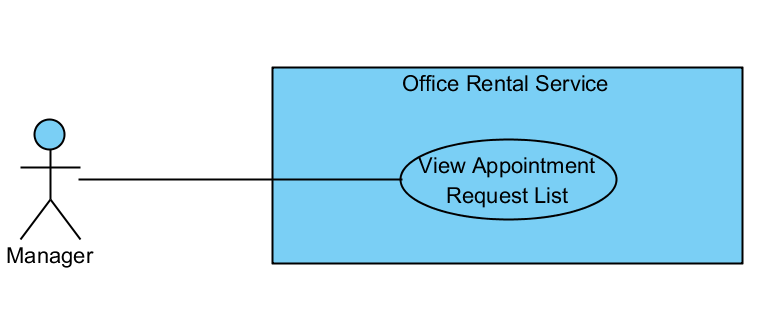


Figure 50: <Manager> View Appointment Request List

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW APPOINTMENT REQUEST LIST** | | | |
| **Use Case No.** | UC041 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Appointment Request List | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager see all appointment request   **Goal:**   * Show all appointment request   **Triggers:**   * From the menu bar on admin page:   + Select “ Lịch hẹn ” menu.   **Preconditions:**   * There are at least request appointment from customer * Requests from customers are valid and send successful   **Post Conditions:**   * **Success:** All appointment request are shown * **Fail:** Cannot get appointment list from database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Appointment” button from menu bar on admin page | * Show all request appointment list in which includes:   + Khách hàng   + Văn phòng   + Thời gian gặp   + Đề xuất nhân viên   [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  | . |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load appointment list from database | Display error message: “Cannot get request appointment list. Please try again” |   **Relationships:** Delete Appointment request, Assign Appointment, View Customer Info  **Business Rules:**   * Manager can view all list appointment from customer’s request. * On list appointment system will suggest staff suitable. | | | |

##### <Manager> Cancel Appointment request

**Use Case Diagram**

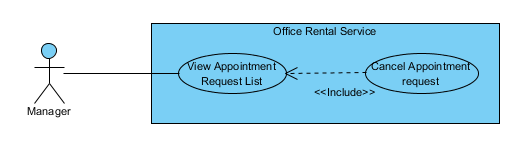


Figure 51: <Manager> Cancel Appointment request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE CANCEL APPOINTMENT REQUEST** | | | |
| **Use Case No.** | UC042 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel Appointment request | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager cancel appointment request which is not suitable.   **Goal:**   * Cancel appointment request successful * Customer receive SMS with reject message   **Triggers:**   * From the menu bar:   + Select “Lịch hẹn” menu.   + Choose appointment   + Click “Hủy lịch hẹn” button on appointment detail page   **Preconditions:**   * There are request appointment from customer * Requests from customers are valid and send successful * Status of request is ‘Chờ xử lý’ or ‘Chấp nhận’   **Post Conditions:**   * **Success:** Appointment request is cancel successful * **Fail:** Cancel error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Lịch hẹn” button from menu bar on admin page | * Show all request appointment list   + Khách hàng   + Văn phòng   + Thời gian gặp   + Đề xuất nhân viên   [Exception 1] | | 2 | Choose the appointment in appointment list. | * Show detail of this appointment   + Khách hàng   + Tên văn phòng   + Nhân viên được giao   + Thời gian gặp   + Tình trạng   [Exception 2] | | 3 | Click “Hủy lịch hẹn” button on appointment detail page | * Show popup input comment | | 4 | Input comment and click “Yes” button on warning popup  [Alternative 1] | * Return to appointment list   [Exception 3] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Click “No” button on warning popup | * Close comment popup |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load appointment list from database | Display error message: “Cannot get request appointment list. Please try again” | | 2 | Cannot get detail of chosen appointment from database | Display error message: “Cannot get data this appointment. Please try again” | | 3 | Delete error | Display error message: “Cannot delete this appointment. Please try again” |   **Relationships:** Appointment Request List  **Business Rules:**   * Manager can cancel appointment from customer’s request if there are some error with customer’s error. * When cancel, system will send sms to customer. * Manager only cancel appointment which is have status ‘Chờ xử lý’ or ‘Chấp nhận’ | | | |

##### <Manager> Assign Appointment

**Use Case Diagram**

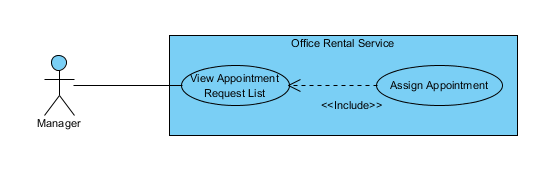


Figure 52: <Manager> Assign Appointment

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE ASSIGN APPOINTMENT** | | | |
| **Use Case No.** | UC043 | **Use Case Version** | 1.0 |
| **Use Case Name** | Assign Appointment | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager assign appointment request to staff   **Goal:**   * Assign appointment to staff successful   **Triggers:**   * From the menu bar:   + Select “Lịch hẹn” menu.   + Click “Giao việc”   **Preconditions:**   * There are request appointment from customer * Requests from customers are valid and send successful * Manager checked the appointment is suitable   **Post Conditions:**   * **Success:** Appointment request is assign successful * **Fail:** Assign error.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Lịch hẹn” button from menu bar on admin page | * Schedule staff who suitable * Show all request appointment list   + Khách hàng   + Văn phòng   + Thời gian gặp   + Đề xuất nhân viên   [Exception 1] | | 2 | Click “Giao việc” button on each row of appointment | [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Click “close” button on list staff | * Show appointment detail page |     **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load appointment list from database | Display error message: “Cannot get request appointment list. Please try again” | | 2 | Cannot get detail of chosen appointment from database | Display error message: “Cannot get data this appointment. Please try again” | | 3 | Assign error | Display error message: “Cannot assign this appointment to staff. Please try again” |   **Relationships:** Appointment Request List  **Business Rules:**   * Manager can assign appointment from customer’s request to staff if the request is suitable. * When assign, system will send email to staff’ and add the request to staff’s task list. | | | |

##### <Manager> View Contract List

**Use Case Diagram**

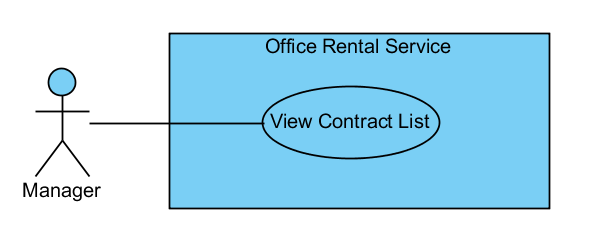


Figure 53: <Manager> View Contract List

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW CONTRACT LIST** | | | |
| **Use Case No.** | UC045 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Contract List | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager view contract list   **Goal:**   * Show all contract list   **Triggers:**   * From the menu bar on admin page:   + Click “Lịch hẹn” menu.   **Preconditions:**   * Login as manager role * There is at least one contract create successful   **Post Conditions:**   * **Success:** Show all contracts in database * **Fail:** Cannot get data of the contract in database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Lịch hẹn” button on menu bar of manager role | * Show all contract in list:   + Khách hàng [text]   + Văn phòng [text]   + Ngày bắt đầu [Date]   + Ngày kết thúc [Date]   + Giá [Number]   + Thời hạn thanh toán [text]   [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |     **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load contract list from database | Display error message: “Cannot get contract list. Please try again” |   **Relationships:** Contract  **Business Rules:**   * Only manager can see all contracts in database | | | |

##### <Manager> View Contract Detail

**Use Case Diagram**

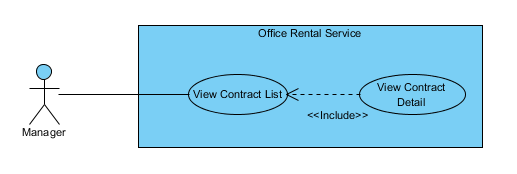


Figure 54: <Manager> View Contract Detail

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW CONTRACT DETAIL** | | | |
| **Use Case No.** | UC046 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Contract Detail | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager view contract detail   **Goal:**   * Show contract detail   **Triggers:**   * From the menu bar:   + Select “Lịch hẹn” menu.   + Click button “Thông tin” on each row   **Preconditions:**   * Contract is not out of date   **Post Conditions:**   * **Success:** Show detail of chosen contract * **Fail:** Cannot get data of the contract in database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Lịch hẹn” button on menu bar of manager role | * Show all contract in list   [Exception 1] | | 2 | Choose contract in contract list and click button “Thông tin” | * Show all detail of the contract includes:   + Khách hàng [text]   + Văn phòng [text]   + Ngày bắt đầu [Date]   + Ngày kết thúc [Date]   + Giá [Number]   + Thời hạn thanh toán [text]   [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |     **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load contract list from database | Display error message: “Cannot get contract list. Please try again” | | 2 | Cannot load data of the chosen contract | Display error message: “Cannot get data of this contract. Please try again” |   **Relationships:** Contract  **Business Rules:**   * Manager can see detail of the contract * The contract is still have time if not system will change status of contract in database and send email to customer | | | |

##### <Manager> Create Contract

**Use Case Diagram**

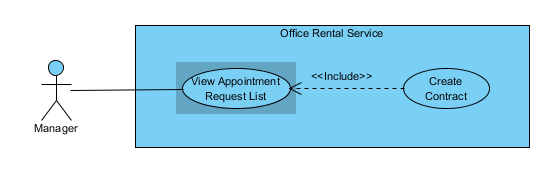


Figure 55: <Manager> Create Contract

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE CREATE CONTRACT** | | | |
| **Use Case No.** | UC048 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create Contract | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager create contract   **Goal:**   * Contract is create and add to database   **Triggers:**   * From the menu bar:   + Click “Lịch hẹn” on menu.   + On “Chấp nhận” tab click “Đến kí/ hủy hợp đồng”   + Click “Tạo hợp đồng” button   + Click “Tạo mới” button after input information   **Preconditions:**   * Appointment has been accepted by customer and status is accepted   **Post Conditions:**   * **Success:** Contract is saved to database * **Fail:** Create fail   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Lịch hẹn” on menu | * Show all appointment list   [Exception 1] | | 2 | On “Chấp nhận” tab click “Đến kí/ hủy hợp đồng” | * Show detail of the appointment   + Khách hàng [text]   + Văn phòng [text]   + Nhân viên được giao [text]   + Thời gian gặp [Date]   + Tình trạng [Text] * Group button:   + Tạo hợp đồng   + Hủy lịch hẹn [Alternative 1]   + Quay về [Alternative 2] | | 3 | Click “Tạo hợp đồng” | * Show input field and detail of contract:   + Khách hàng [text]   + Văn phòng [text]   + Ngày bắt đầu [Date]   + Ngày kết thúc [Date]   + Giá [Number]   + Thời hạn thanh toán [text] * Button “Tạo mới” | | 4 | Input information and click “Tạo mới” button | * Validate input data [Exception 2] * Show successful message [Exception 3] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Click “Hủy lịch hẹn” button | * Show popup input comment and return to appointment list page after manager input comment | | 2 | Click “Quay về” button | * Return to appointment list page |     **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load contract list from database | Display error message: “Cannot get contract list. Please try again” | | 2 | From date > End Date | Display error message: “From date or end date is invalid. Please check again” | | 3 | Cannot save new contract to database |  |   **Relationships:** Contract  **Business Rules:**   * Manager will meet customer after customer decide to rent office * Manager will create contract after the meeting | | | |

##### <Manager> Edit Contract

**Use Case Diagram**

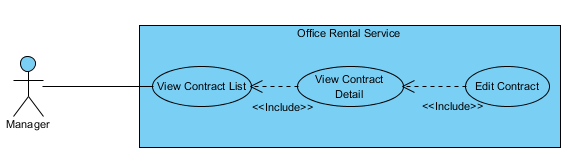


Figure 56: <Manager> Edit Contract

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE EDIT CONTRACT** | | | |
| **Use Case No.** | UC047 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Contract | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager edit contract   **Goal:**   * Contract is edited   **Triggers:**   * Click “Chỉnh sửa” button on contract detail page   **Preconditions:**   * Contract is not out of date   **Post Conditions:**   * **Success:** Edit contract successful * **Fail:** Edit error   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “thông tin” button on contract list page | * Show all detail of the contract   + Khách hàng [text]   + Văn phòng [text]   + Ngày bắt đầu [Date]   + Ngày kết thúc [Date]   + Giá [Number]   + Thời hạn thanh toán [text]   [Exception 1] | | 2 | Click “Chỉnh sửa” button | * Show field to edit information   + Khách hàng [text]   + Văn phòng [text]   + Ngày bắt đầu [Date]   + Ngày kết thúc [Date]   + Giá [Number]   + Thời hạn thanh toán [text] * Button group:   + Cập nhật   + Hủy [Alternative 1] | | 3 | Edit contract info | * Validate input show error if data invalid   [Exception 2] | | 4 | Click “Cập nhật” button | * Save to database * Return to contract list |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Click “Hủy” button | * Return to contract list page |     **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load contract list from database | Display error message: “Cannot get contract list. Please try again” | | 2 | Cannot load data of the chosen contract | Display error message: “Cannot get data of this contract. Please try again” | | 3 | From date > End Date | Display error message: “From date or end date is invalid. Please check again” |   **Relationships:** Contract  **Business Rules:**   * Manager edit contract when there is error in contract. * Manager only edit contract which is not expired. | | | |

##### <Manager> View Repair Request List

**Use Case Diagram**

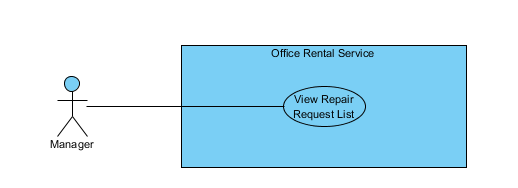


Figure 57: <Manager> View Repair Request List

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW REPAIR LIST** | | | |
| **Use Case No.** | UC141 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Repair Request List | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager see all repair requests   **Goal:**   * Show suggest staff and suggest time * Show all repair request   **Triggers:**   * From the menu bar on admin page:   + Select “ Sửa chữa ” on the menu.   **Preconditions:**   * At lease exist one repair request in database   **Post Conditions:**   * **Success:** All request repair are shown * **Fail:** Cannot get repair list from database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Sửa chữa” button on menu bar of manager role | * Show all repair requests list in which includes:   + Tên văn phòng [text]   + Khách hàng [text]   + Ngày tạo yêu cầu [Date]   + Mô tả [text]   + Đề xuất nhân viên [dropdown list]   + Đề xuât thời gian [Date]   [Exception 1]   * Button group   + Giao việc   + Chi tiết | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  | . |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load repair request list from database | Display error message: “Không lấy được dữ liệu toàn bộ yêu cầu sửa chữa. Xin vui lòng thử lại” |   **Relationships:** Request repair  **Business Rules:**   * Manager can view all request repair for assigning to staff | | | |

##### <Manager> Assign Repair

**Use Case Diagram**

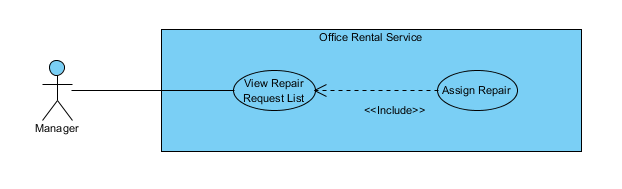


Figure 58: <Manager> Assign Repair

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE ASSIGN REPAIR** | | | |
| **Use Case No.** | UC141 | **Use Case Version** | 1.0 |
| **Use Case Name** | Assign Repair | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager assign repair to staff   **Goal:**   * Show suggest staff and suggest time * Request is assigned to staff successful   **Triggers:**   * From the menu bar on admin page:   + Select “Sửa chữa ” on the menu.   + Click “Giao việc” button   **Preconditions:**   * At lease exist one repair request in database * At lease exist one staff in database   **Post Conditions:**   * **Success:** Request is assigned to staff successful * **Fail:** Assign failed   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Sửa chữa” button on menu bar of manager role | * Show all repair requests list in which includes:   + Tên văn phòng [text]   + Khách hàng [text]   + Ngày tạo yêu cầu [Date]   + Mô tả [text]   + Đề xuất nhân viên [dropdown list]   + Đề xuât thời gian [Date]   [Exception 1]   * Button group   + Giao việc   + Chi tiết | | 2 | Click “Giao việc” button | * Change status of request repair * Change detail of request repair   [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  | . |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load repair request list from database | Display error message: “Không lấy được dữ liệu toàn bộ yêu cầu sửa chữa. Xin vui lòng thử lại” | | 2 | Cannot assign to staff |  |   **Relationships:** Request repair  **Business Rules:**   * Manager can view staff is suggested for the repair request and can change staff manually | | | |

##### <Manager> Cancel Repair

**Use Case Diagram**

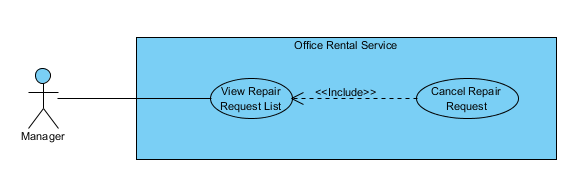


Figure 59: <Manager> Cancel Repair

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE CANCEL REPAIR** | | | |
| **Use Case No.** | UC141 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel Repair | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager cancel repair request   **Goal:**   * The request has been canceled.   **Triggers:**   * From the menu bar on admin page:   + Select “ Sửa chữa ” on the menu.   + Choose one request   + Click “Từ chối sửa chữa” button on repair detail page   **Preconditions:**   * At lease exist one repair request in database * The request status is “Cần giao việc”   **Post Conditions:**   * **Success:** Request is assigned to staff successful * **Fail:** Assign failed   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Sửa chữa” button on menu bar of manager role | * Show all repair requests list in which includes:   + Tên văn phòng [text]   + Khách hàng [text]   + Ngày tạo yêu cầu [Date]   + Mô tả [text]   + Đề xuất nhân viên [dropdown list]   + Đề xuât thời gian [Date]   [Exception 1]   * Button group   + Giao việc   + Chi tiết | | 2 | Click “Chi tiết” button | * Show detail of repair request:   + Tên văn phòng [text]   + Khách hàng [text]   + Ngày tạo yêu cầu [Date]   + Mô tả [text]   + Đề xuất nhân viên [dropdown list]   + Đề xuât thời gian [Date] * Button group:   + Giao việc   + Từ chối sửa chữa   + Quay về [Alternative 1] | | 3 | Click “Từ chối sửa chữa” button | * Change status of request repair * Return to request repair list page |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Click “Quay về” button | * Return to request repair list page |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load repair request list from database | Display error message: “Không lấy được dữ liệu toàn bộ yêu cầu sửa chữa. Xin vui lòng thử lại” | | 2 | Cannot cancel request |  |   **Relationships:** Request repair  **Business Rules:**   * Manager can cancel request repair if the request has invalid info. | | | |

##### <Manager> View Rental Request List

**Use Case Diagram**

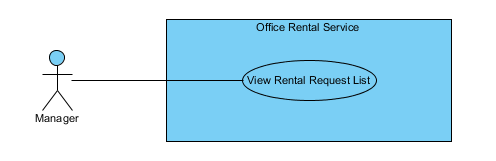


Figure 60: <Manager> View Rental Request List

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW RENTAL REQUEST LIST** | | | |
| **Use Case No.** | UC141 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Rental Request List | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager see all rental requests   **Goal:**   * Show suggest staff and suggest time * Show all rental request   **Triggers:**   * From the menu bar on admin page:   + Select “Thuê thiết bị” on the menu.   **Preconditions:**   * At lease exist one rental request in database   **Post Conditions:**   * **Success:** All request rental are shown * **Fail:** Cannot get rental request list from database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Thuê thiết bị” button on menu of manager role | * Show all rental requests list in which includes:   + Tên văn phòng [text]   + Khách hàng [text]   + Ngày tạo yêu cầu [Date]   + Mô tả [text]   + Đề xuất nhân viên [dropdown list]   + Đề xuât thời gian [Date]   [Exception 1]   * Button group   + Giao việc   + Chi tiết | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  | . |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load rental request list from database | Display error message: “Không lấy được dữ liệu toàn bộ yêu cầu thuê thiết bị. Xin vui lòng thử lại” |   **Relationships:** Rental, Rental Detail, Rental Item, Rental Status  **Business Rules:**   * Manager can view all request rental for assigning to staff | | | |

##### <Manager> Assign Rental

**Use Case Diagram**

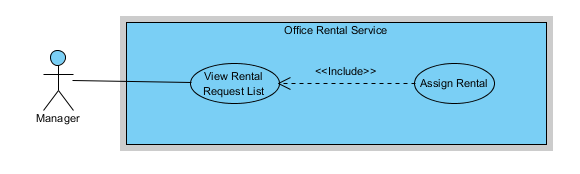


Figure 61: <Manager> Assign Rental

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE ASSIGN RENTAL** | | | |
| **Use Case No.** | UC141 | **Use Case Version** | 1.0 |
| **Use Case Name** | Assign Rental | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager assign rental request to staff   **Goal:**   * Show suggest staff and suggest time * Request is assigned to staff successful   **Triggers:**   * From the menu bar on admin page:   + Select “Thuê thiết bị ” on the menu.   + Click “Giao việc” button   **Preconditions:**   * At lease exist one rental request in database * At lease exist one staff in database   **Post Conditions:**   * **Success:** Request is assigned to staff successful * **Fail:** Assign failed   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Thuê Thiết bị” button on menu bar of manager role | * Show all rental requests list in which includes:   + Tên văn phòng [text]   + Khách hàng [text]   + Ngày tạo yêu cầu [Date]   + Mô tả [text]   + Đề xuất nhân viên [dropdown list]   + Đề xuât thời gian [Date]   [Exception 1]   * Button group   + Giao việc   + Chi tiết | | 2 | Click “Giao việc” button | * Change status of request rental * Change detail of request rental   [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  | . |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load repair request list from database | Display error message: “Không lấy được dữ liệu toàn bộ yêu cầu thuê thiết bị. Xin vui lòng thử lại” | | 2 | Cannot assign to staff |  |   **Relationships:** Rental, Rental Detail, Rental Item, Rental Status  **Business Rules:**   * Manager can view staff is suggested for the rental request and assign staff for the rental request | | | |

##### <Manager> Cancel Rental Request

**Use Case Diagram**

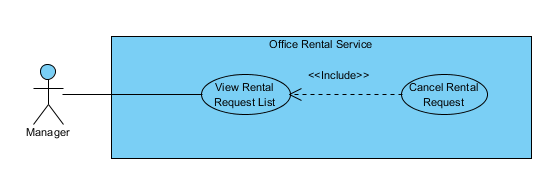


Figure 62: <Manager> Cancel Rental Request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE CANCEL RENTAL REQUEST** | | | |
| **Use Case No.** | UC141 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel Rental Request | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager cancel rental request   **Goal:**   * The rental request has been canceled.   **Triggers:**   * From the menu bar on admin page:   + Select “Thuê thiết bị” on the menu.   + Choose one request   + Click “Từ chối” button on rental detail page   **Preconditions:**   * At lease exist one rental request in database * The request status is “Cần giao việc”   **Post Conditions:**   * **Success:** Request is assigned to staff successful * **Fail:** Assign failed   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Thuê thiết bị” button on menu of manager role | * Show all rental requests list in which includes:   + Tên văn phòng [text]   + Khách hàng [text]   + Ngày tạo yêu cầu [Date]   + Mô tả [text]   + Đề xuất nhân viên [dropdown list]   + Đề xuât thời gian [Date]   [Exception 1]   * Button group   + Giao việc   + Chi tiết | | 2 | Click “Chi tiết” button | * Show detail of rental request:   + Tên văn phòng [text]   + Khách hàng [text]   + Ngày tạo yêu cầu [Date]   + Mô tả [text]   + Đề xuất nhân viên [dropdown list]   + Đề xuât thời gian [Date] * Button group:   + Giao việc   + Từ chối   + Quay về [Alternative 1] | | 3 | Click “Từ chối” button | * Change status of rental repair * Return to rental repair list page |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Click “Quay về” button | * Return to rental request list page |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load repair request list from database | Display error message: “Không lấy được dữ liệu toàn bộ yêu cầu thuê thiết bị. Xin vui lòng thử lại” | | 2 | Cannot cancel request |  |   **Relationships:** Rental, Rental Detail, Rental Item, Rental Status  **Business Rules:**   * Manager can cancel rental request if that request has invalid info. | | | |

##### <Manager> View amenity list

**Use Case Diagram**

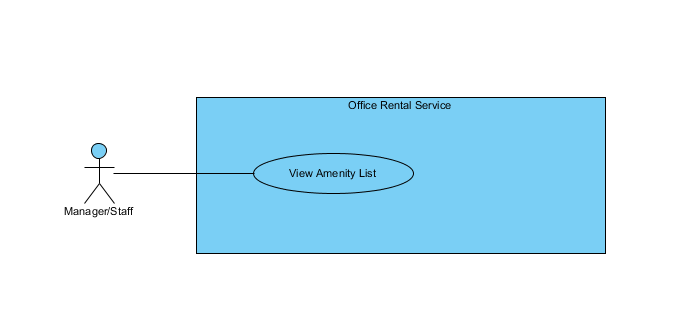


Figure 63: <Manager>View amenity list

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW AMENITY LIST** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | View Amenity List | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager, Staff   **Summary:**   * This use case allows user to view amenity list.   **Goal:**   * User can view amenity list successful.   **Triggers:**   * Click “Quản lí tiện nghi” in menu.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Amenity list has been viewed successfully. * **Fail:** Cannot view amenity list.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý tiện nghi” on menu. | Show page detail  Display details includes:   * Tên tiện nghi: Label * Mô tả: Label * Trọng số: Label * Độ ưu tiên: Label   “Cập nhật” : button  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Amenity AmenityGroup  **Business Rules:** | | | | |

Table 16: <Manager>View amenity list

##### <Manager> Create amenity

**Use Case Diagram**

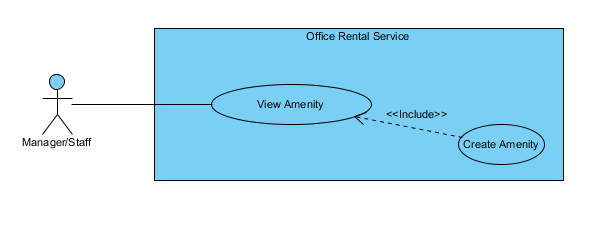


Figure 64: <Manager> Create amenity

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CREATE AMENITY** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Create Amenity | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager, Staff   **Summary:**   * This use case allows user to create amenity.   **Goal:**   * User can create amenity successful.   **Triggers:**   * Click “Quản lý tiện nghi” in menu. * Click “Thêm mới tiện nghi” in menu. * Fill amenity information. * Click “Tạo mới” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Amenity has been created successfully. * **Fail:** Cannot create new amenity.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý tiện nghi” on menu. | Show page detail  Display details includes:   * Thêm mới tiện nghi: Link * Tên tiện nghi: Label * Mô tả: Label * Trọng số: Label * Độ ưu tiên: Label   “Cập nhật” : button | | 2 | Click “Thêm mới tiện nghi” on menu. | Show page detail  Display details includes:   * Tên tiện nghi: Textbox * Mô tả: Textbox * Trọng số: Textbox * Độ ưu tiên: Textbox * Nhóm thiết bị: Dropdown List   “Huỷ” : button  “Tạo mới” : button | | 3 | Click “Tạo mới” button on screen.  [Alternative 1] | Show successful message “Thêm thiết bị thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Close page and go to management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Amenity AmenityGroup  **Business Rules:** | | | | |

Table 16: <Manager> Create amenity

##### <Manager> View amenity detail

**Use Case Diagram**

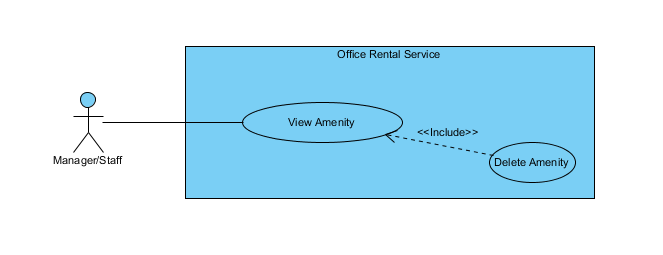


Figure 65: <Manager> View amenity detail

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – DELETE AMENITY** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Delete Amenity | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager, Staff   **Summary:**   * This use case allows user to delete amenity.   **Goal:**   * User can delete amenity successful.   **Triggers:**   * Click “Quản lý tiện nghi” in menu. * Click “Xoá” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Amenity has been created successfully. * **Fail:** Cannot create new amenity.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý tiện nghi” on menu. | Show page detail  Display details includes:   * Thêm mới tiện nghi: Link * Tên tiện nghi: Label * Mô tả: Label * Trọng số: Label * Độ ưu tiên: Label   “Xoá” : button  “Cập nhật” : button | | 2 | Click “Xoá” button on screen. | Show successful message “Xoá thiết bị thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Amenity AmenityGroup  **Business Rules:** | | | | |

Table 16: <Manager> View amenity detail

##### <Manager> Edit amenity

**Use Case Diagram**

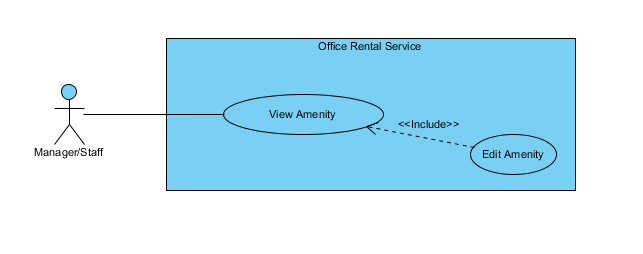


Figure 66: <Manager> Edit amenity

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – EDIT AMENITY** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Edit Amenity | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager, Staff   **Summary:**   * This use case allows user to edit amenity.   **Goal:**   * User can edit amenity successful.   **Triggers:**   * Click “Quản lý tiện nghi” in menu. * Click “Cập nhật” in menu. * Fill amenity group information. * Click “Cập nhật” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Amenity has been created successfully. * **Fail:** Cannot create new amenity.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý tiện nghi” on menu. | Show page detail  Display details includes:   * Thêm mới tiện nghi: Link * Tên tiện nghi: Label * Mô tả: Label * Trọng số: Label * Độ ưu tiên: Label   “Cập nhật” : button | | 2 | Click “Thêm mới tiện nghi” on menu. | Show page request detail  Display request details includes:   * Tên tiện nghi: Label * Mô tả: Textbox * Trọng số: Textbox * Độ ưu tiên: Textbox   “Huỷ” : button  “Cập nhật” : button | | 3 | Click “Cập nhật” button on screen.  [Alternative 1] | Show successful message “Cập nhật thiết bị thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Close page and go to management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Amenity AmenityGroup  **Business Rules:** | | | | |

Table 16: <Manager> Edit amenity

##### <Manager> View amenity group list

**Use Case Diagram**

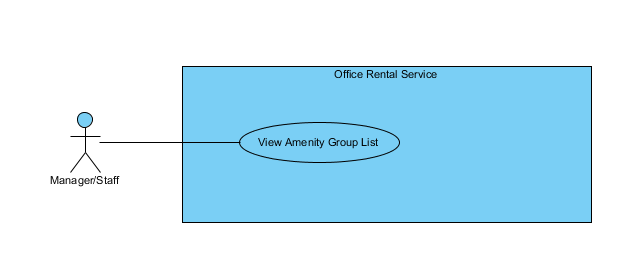


Figure 67: <Manager> View amenity group list

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – VIEW AMENITY GROUP LIST** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | View Amenity Group List | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager, Staff   **Summary:**   * This use case allows user to view amenity group list.   **Goal:**   * User can view amenity group list successful.   **Triggers:**   * Click “Quản lý nhóm tiện nghi” in menu.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Amenity list has been viewed successfully. * **Fail:** Cannot view amenity list.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý nhóm tiện nghi” on menu. | Show page detail  Display details includes:   * Tên nhóm: Label * Mô tả: Label   “Xoá” : button  “Cập nhật” : button  [Exception 1] | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Amenity AmenityGroup  **Business Rules:** | | | | |

Table 16: <Manager> View amenity group list

##### <Manager> Create amenity group

**Use Case Diagram**

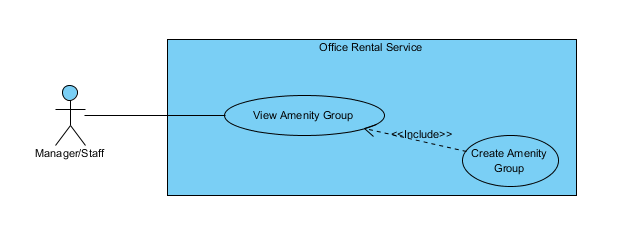


Figure 68: <Manager> Create amenity group

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – CREATE AMENITY GROUP** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Create Amenity Group | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager, Staff   **Summary:**   * This use case allows user to create amenity group.   **Goal:**   * User can create amenity group successful.   **Triggers:**   * Click “Quản lý nhóm tiện nghi” in menu. * Click “Thêm mới nhóm tiện nghi” in menu. * Fill amenity group information. * Click “Tạo mới” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Amenity has been created successfully. * **Fail:** Cannot create new amenity.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý nhóm tiện nghi” on menu. |  | | 2 | Click “Thêm mới tiện nghi” on menu. | Show page detail  Display details includes:   * Tên tiện nghi: Textbox * Mô tả: Textbox   “Huỷ” : button  “Tạo mới” : button | | 3 | Click “Tạo mới” button on screen.  [Alternative 1] | Show successful message “Thêm nhóm thiết bị thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Close page and go to management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Amenity AmenityGroup  **Business Rules:** | | | | |

Table 16: <Manager> Create amenity group

##### <Manager> Edit amenity group

**Use Case Diagram**

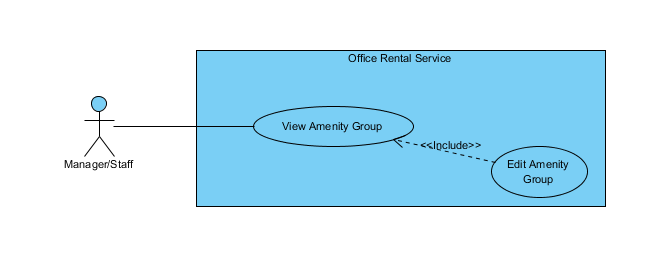


Figure 69: <Manager> Edit amenity group

**Use Case Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE – EDIT AMENITY GROUP** | | | | |
| **Use-case No.** |  | **Use-case Version** | | 2.0 |
| **Use-case Name** | Edit Amenity Group | | | |
| **Author** | Trần Lê Tuấn | | | |
| **Date** | 25/05/2015 | **Priority** | Normal | |
| **Actor:**   * Manager, Staff   **Summary:**   * This use case allows user to create amenity group.   **Goal:**   * User can create amenity group successful.   **Triggers:**   * Click “Quản lý nhóm tiện nghi” in menu. * Click “Cập nhật” in menu. * Fill amenity group information. * Click “Cập nhật” button.   **Preconditions:**   * User must login in system.   **Post Conditions:**   * **Success**: Amenity has been created successfully. * **Fail:** Cannot create new amenity.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Quản lý nhóm tiện nghi” on menu. | Show page detail  Display details includes:   * Thêm mới nhóm tiện nghi: Link * Tên nhóm: Label * Mô tả: Label   “Xoá” : button  “Cập nhật” : button  [Exception 1] | | 2 | Click “Cập nhật” on menu. | Show page request detail  Display request details includes:   * Tên tiện nghi: Label * Mô tả: Textbox   “Huỷ” : button  “Cập nhật” : button | | 3 | Click “Cập nhật” button on screen.  [Alternative 1] | Show successful message “Cập nhật nhóm thiết bị thành công.” on screen.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Click “Huỷ” button. | Close page and go to management page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Cannot send request due to database connection | Show error message: “Không thể gửi yêu cầu, vui lòng thử lại sau.” |   **Relationships:** Amenity AmenityGroup  **Business Rules:** | | | | |

Table 16: <Manager> Edit amenity group

##### <Manager> View Customer Info

**Use Case Diagram**

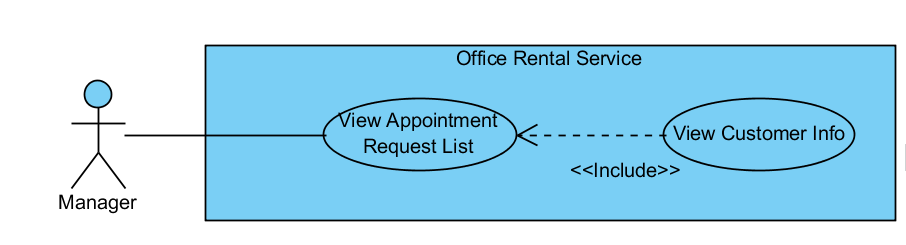


Figure 70: <Manager> View Customer Info

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE VIEW CUSTOMER INFO** | | | |
| **Use Case No.** | UC044 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Customer Info | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * Manager   **Summary:**   * This use case allows manager view customer info from appointment list   **Goal:**   * View customer profile   **Triggers:**   * From the menu bar:   + Select “Appointment” menu.   + Choose appointment   + Click to customer’s name in appointment detail page   **Preconditions:**   * There are request appointment from customer * Requests from customers are valid and send successful   **Post Conditions:**   * **Success:** Show customer profile * **Fail:** Cannot get data of the customer in database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Appointment” button on menu bar of manager role | * Show all request appointment list   [Exception 1] | | 2 | Choose the appointment in appointment list. | * Show detail of this appointment   [Exception 2] | | 3 | Click to name of customer in appointment detail page | * Show customer profile includes:   + Username   + Email   + Phone   + Address   [Exception 3] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |     **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot load appointment list from database | Display error message: “Cannot get request appointment list. Please try again” | | 2 | Cannot get detail of chosen appointment from database | Display error message: “Cannot get data this appointment. Please try again” | | 3 | Cannot get detail of chosen customer from database | Display error message: “Cannot get data this customer. Please try again” |   **Relationships:** Appointment Request List  **Business Rules:**   * Manager can find customer profile base on appointment request list | | | |

Table 24: <Manager> View Customer Info

#### <System>Overview Use Case

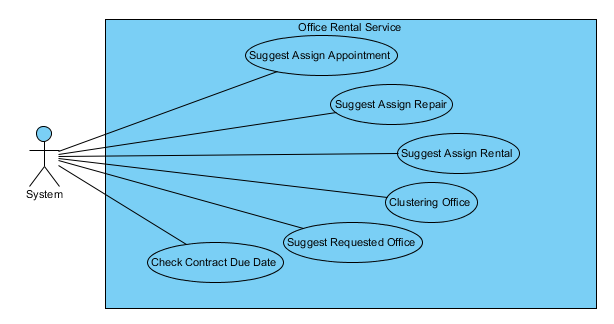


Figure 71: <System>Overview Use Case

##### <System> Suggest Requested Office

**Use Case Diagram**

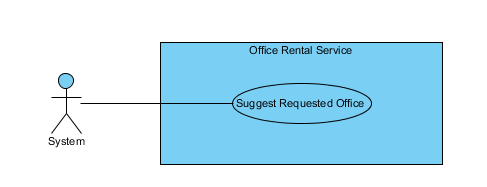


Figure 72: <System> Suggest Requested Office

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SUGGEST REQUESTED OFFICE** | | | |
| **Use Case No.** | UC041 | **Use Case Version** | 1.0 |
| **Use Case Name** | Suggest Requested Office | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * System   **Summary:**   * This use case show system find offices that suitable with customer’s requests.   **Goal:**   * Find offices that suitable with customer’s requests.   **Triggers:**   * Schedule start   **Preconditions:**   * There are at least one request office from customer * There are at least one office in database   **Post Conditions:**   * **Success:** Office that suitable with customer’s requests is saved to database. * **Fail:** Cannot find office although there is the office suitable.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Start schedule | * Find all office in database * Find all request office in database * Check office with request * Find office that suitable * Save id of office to request office |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Relationships:** RequestOffice, Office  **Business Rules:**   * System will automatic check all requests office of customer and find which office in database suitable. * System will check 5 criteria:   + Amenity:     - If there is one request amenity in request, system will find if office which has that amenity will be suitable.     - If there are two or more request amenities, system will find if office which has two or more amenities in request will be suitable.   + Category:     - System will find which office suitable with request’s category.   + Price:     - System will find which office’s price is in range of request’s price: from (request’s price \* 80%) to (request’s price \* 120%)   + Area:     - If request category is “Văn phòng nguyên căn”: office’s area in range is suitable: from (request’s area \* 80%) to (request’s area \* 120%)     - If request category is “Tòa nhà văn phòng”: office’s area is larger than request’s area is suitable   + District:     - System will check which office in district of customer’s request | | | |

Table 16: <System> Suggest requested office

##### <System> Check Contract Due Date

**Use Case Diagram**

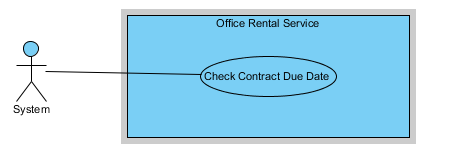


Figure 73: <System> Check Contract Due Date

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE CHECK CONTRACT DUE DATE** | | | |
| **Use Case No.** | UC041 | **Use Case Version** | 1.0 |
| **Use Case Name** | Check Contract Due Date | | |
| **Author** | Nguyễn Vũ Hoàng Quốc | | |
| **Date** | 31/05/2015 | **Priority** | Normal |
| **Actor:**   * System   **Summary:**   * This use case show system check contract and send notification mail to customer.   **Goal:**   * Find contract which nearly expired and contract is expired in day and send mail to customer.   **Triggers:**   * Schedule start   **Preconditions:**   * There are at least one contract in database   **Post Conditions:**   * **Success:** Customer receive mail that notify about their contract will be expired of expired. * **Fail:** Send mail failed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Start schedule | * Find all contract in database * Check contract with is nearly or expired in day * Send notification mail to customer |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Relationships:** Contract, account  **Business Rules:**   * System check if contract has 7 days before expired will send mail to customer base on customer info. * If contract is due date it will auto change status to expired and send mail to customer. | | | |

Table 16: <System> Check contract due date

## Software System Attribute

### Usability

* Staff should need less than one training week to interact with system.

### Reliability

* Information should be updated every day.

### Availability

* System uptime at about 99%

### Security

* All input data should be validated before saving to database.
* All privacy information, such as password, should be encrypted to ensure security.
* User should be authenticated and authorized when accessing to the system.

### Maintainability

* The system should be divided into modules and code. It would be easy to maintain and upgrade.

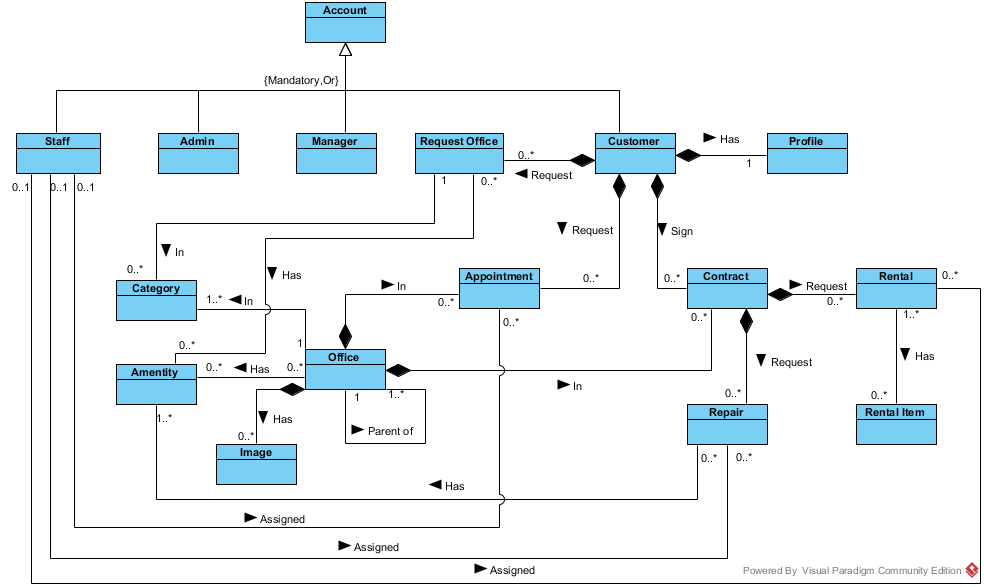
### Portability

* N/A

### Performance

* All requests should be response in no more than 1 minute.

## Conceptual Diagram



**Data dictionary:**

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

# Report 4: Software Design Description

## Design Overview

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

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

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

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

Document overview:

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

## System Architectural Design

### Choice of System Architecture

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

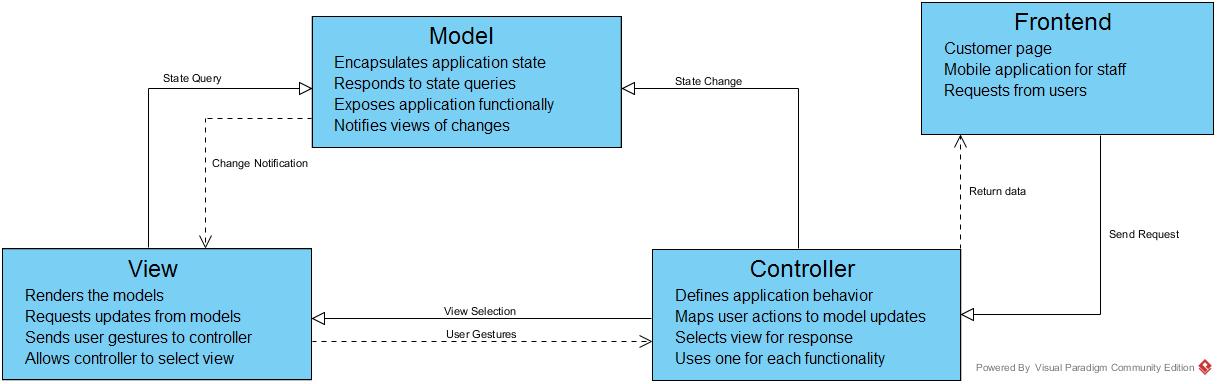


Figure 74: System architecture

## Component Diagram

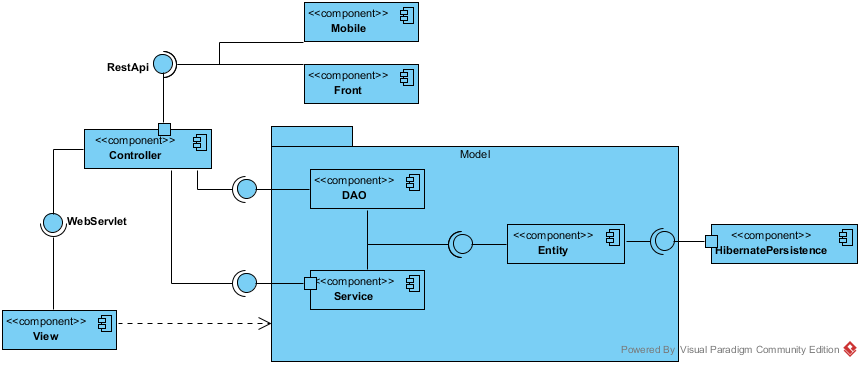


Figure 75: Component Diagram

## Detailed Description Explanation

### Class Diagram

#### Entity Package Class Diagram

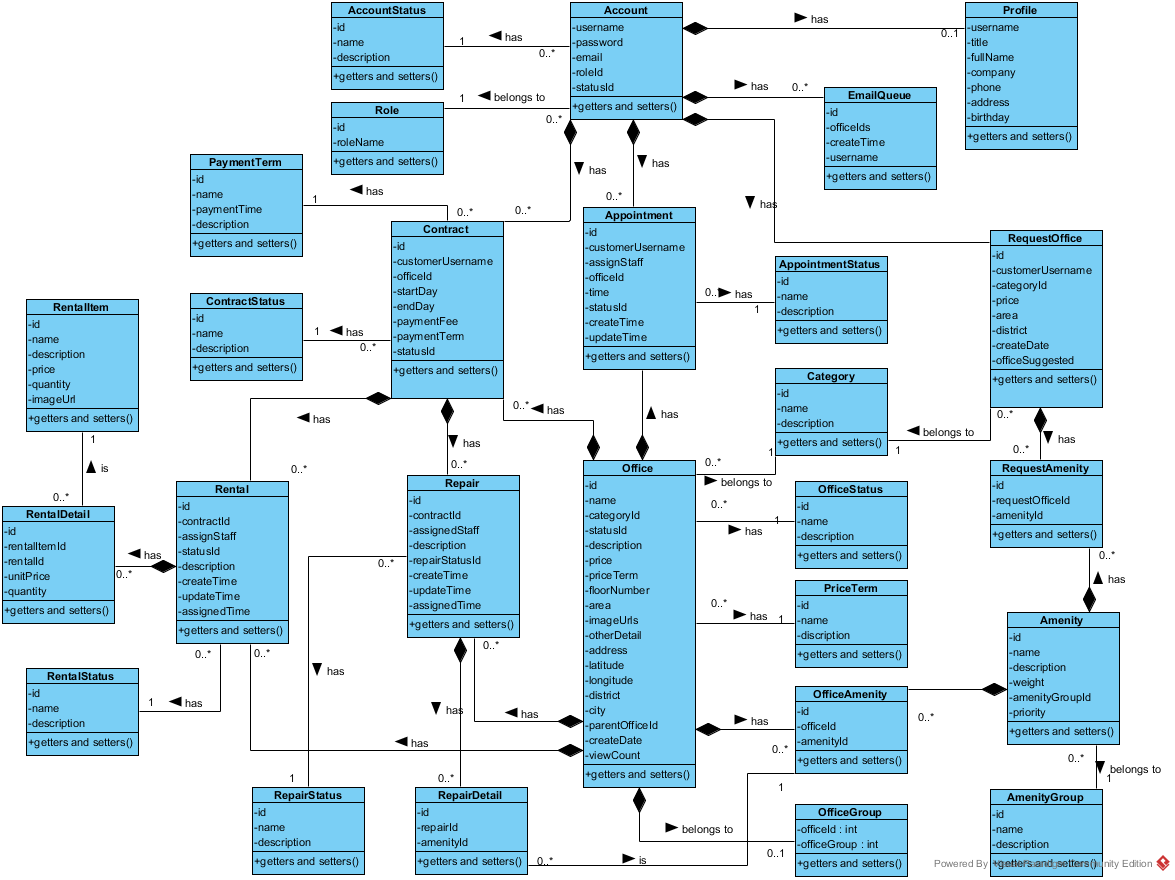


Figure 76: Class Diagram

**Class Diagram Explanation**

##### RepairStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the status of repair |
| name | String | private | The name of the status of repair |
| description | String | private | The description of the status of repair |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### RepairDetail

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the repair detail |
| repairId | int | private | The id of the repair |
| amenityId | int | private | The id of the amenity |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Repair

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the repair |
| contractId | int | private | The id of the repair |
| assignedStaff | String | private | Username of staff who responsibility for this repair request |
| repairStatusId | int | private | The id of the status of repair |
| description | String | private | The description of the repair request |
| createTime | Timestamp | private | Time when customer create this repair request |
| updateTime | Timestamp | private | Time when manger or staff change status |
| assignedTime | Timestamp | private | Time when manager assign to staff |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### PaymentTerm

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the payment term |
| name | String | private | The name of the payment term |
| paymentTime | int | private | Number months of the payment |
| description | String | private | The description of payment term |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### ContractStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the contract status |
| name | String | private | The name of the contract status |
| description | String | private | The description of the contract status |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Contract

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the contract |
| customerUsername | String | private | Username of customer |
| officeId | int | private | The id of the office |
| startDate | date | private | Date when contract starts |
| endDate | date | private | Date when contract ends |
| paymentFee | int | private | Payment fee for the contract |
| paymentTerm | int | private | Payment term for the contract |
| statusId | int | private | The id of contract status |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### OfficeStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the office status |
| name | String | private | The name of the office status |
| description | String | private | The description of the office status |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### OfficeGroup

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| officeId | int | private | The id of the office |
| officeGroup | int | private | Group of office for searching |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### PriceTerm

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the price term |
| name | String | private | The name of the price term |
| description | String | private | The description of the price term |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### EmailQueue

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the email queue |
| username | String | private | Username of the customer |
| officeIds | String | private | The id of list office that system suggests to customer |
| createTime | date | private | Time when customer create request |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Office

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the office |
| name | String | private | The name of the office |
| categoryId | int | private | The id of the category |
| statusId | int | private | The id of the status office |
| description | String | private | The description of the office |
| price | long | private | The price of the office |
| priceTerm | int | private | The price term of the office |
| floorNumber | int | private | The number of floor in the office |
| area | double | private | The area of the office |
| imageUrls | String | private | Link of the images of office |
| otherDetail | String | private | Some detail of the office |
| address | String | private | The address of the office |
| latitude | Double | private | Latitude of the office |
| longitude | Double | private | Longitude of the office |
| district | String | private | The district of the office |
| city | String | private | The city of the office |
| parentOfficeId | int | private | The parent office id of the office |
| createDate | Timestamp | private | Date when office create in database |
| viewCount | int | private | Number views when customer see detail of office |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Appointment

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the appointment |
| customerUsername | String | private | The username of customer |
| assignedStaff | String | private | Staff who is responsibility for the appointment |
| officeId | int | private | Office where customer want to make appointment |
| time | Timestamp | private | Time for appointment |
| statusId | int | private | The status of the appointment |
| createTime | Timestamp | private | Time when customer create appointment |
| updateTime | Timestamp | private | Time when manager update the appointment |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Account

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | private | Unique identifier of an account |
| password | String | private | Password for login to system |
| email | String | private | Email of this account |
| roleId | int | private | Role of this account |
| statusId | int | private | Status of this account |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### AccountStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of an account status |
| name | String | private | The name of this status |
| description | String | private | The description for this status |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Role

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of a role |
| roleName | String | private | The name of role |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Profile

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | private | The id of an account |
| title | String | private | Gender of the profile |
| fullName | String | private | Full name of this profile |
| company | String | private | Company of this profile |
| phone | String | private | Phone of this profile |
| address | String | private | Address of this profile |
| birthday | Timestamp | private | Birthday of this profile |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### AppointmentStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of an appointment status |
| name | String | private | The name of appointment status |
| description | String | private | The description for appointment status |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Category

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of category |
| name | String | private | The name of category |
| description | String | private | The description of category |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### OfficeAmentity

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of office amenity |
| officeId | int | private | The id of office |
| amentityId | int | private | The id of amenity |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Rental

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of rental |
| contractId | int | private | The id of contract |
| assignStaff | String | private | Username of staff who responsibility for this rental request |
| statusId | int | private | The status id of this rental request |
| description | String | private | The description of rental |
| createTime | Timestamp | private | Time when customer create this rental request |
| updateTime | Timestamp | private | Time when manger or staff change status |
| assignedTime | Timestamp | private | Time when manager assign to staff |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### RequestOffice

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the request office |
| customerUsername | String | private | Username of customer |
| categoryId | int | private | The id of category |
| price | int | private | Price of the office |
| area | int | private | Area of the office |
| district | int | private | District of the office |
| createTime | Timestamp | private | Time when customer create this rental request |
| scheduleDate | Timestamp | private | Date when system check this request |
| OfficeSuggested | String | private | List offices that are suitable with request |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### RequestAmenity

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of request amenity |
| requestOfficeId | int | private | The id of request office |
| amentityId | int | private | The id of amenity |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### Amenity

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of amenity |
| name | String | private | Name of this amenity |
| description | String | private | Description for this amenity |
| weight | int | private | Weight of amenity in one group |
| amenityGroupId | int | private | Group of this amenity |
| priority | int | private | Priority of amenity in one group |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### AmenityGroup

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of amenity group |
| name | String | private | Name of amenity group |
| description | String | private | Description for amenity group |
|  |  |  |  |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### RentalStatus

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of the status of rental |
| name | String | private | The name of the status of rental |
| description | String | private | The description of the status of rental |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### RentalItem

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of rental item |
| name | String | private | Name of the item |
| description | String | private | Description for the item |
| price | double | private | Price of the item |
| quantity | int | private | Quantity of the item |
| imageUrl | String | private | Link image of the item |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

##### RentalDetail

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | The id of rental detail |
| rentalItemId | int | private | Item rental for the rental request |
| rentalId | int | private | Id of request rental |
| unitPrice | double | private | Price of one item |
| quantity | int | private | Quantity of rental item for the rental request |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getters | attribute type | public | Get the respective attribute value |
| setters | void | public | Set the respective attribute with the value |

#### DAO Package Class Diagram

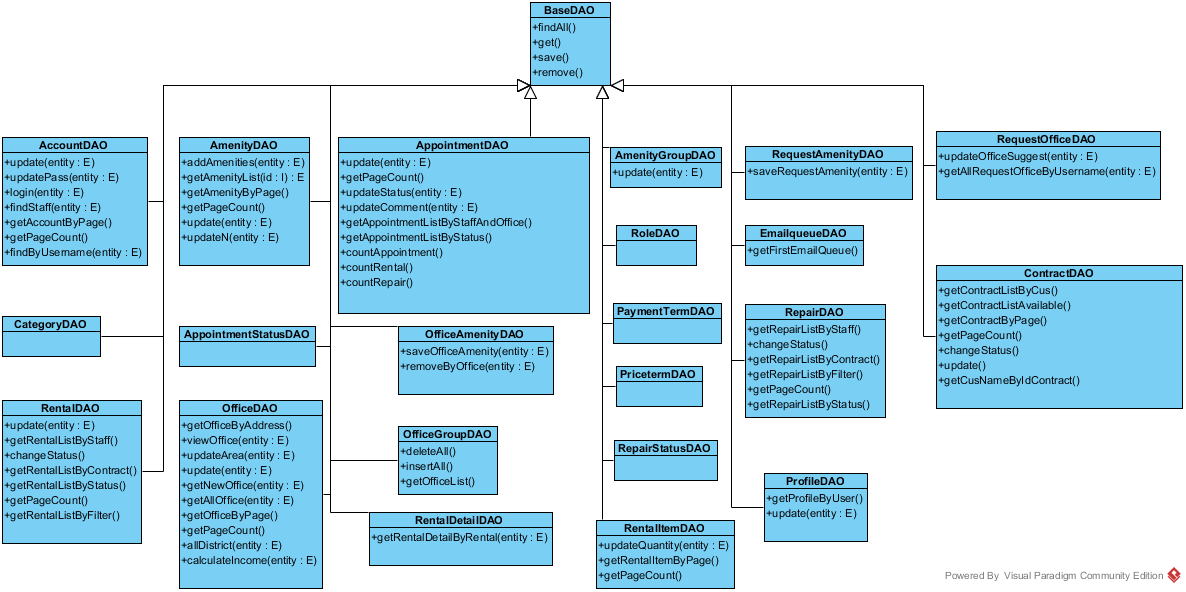


Figure 77: Class Diagram

**Class Diagram Explanation**

##### Class BaseDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| findAll() | List<E> | public | Get all record on the equivalent entity class E of DAO class |
| get(I) | E | public | Get the instance of entity E which has the id I |
| save(E) | boolean | public | Save the instace of entity E to database. Return true if save success |
| remove(E) | boolean | public | Remove the object E from database. Return true if remove success |

##### Class AccountDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| update() | void | public | Update all information of exist account. |
| updatePass() | boolean | public | Update password of exist account |
| delete() | boolen | public | Delete an exist account |
| login() | Account | public | Login into system with exist account |
| findStaff() | List<Account> | public | List array of staff account |
| getAccountByPage() | List<Account> | public | Get a list of account |
| getPageCount() | int | public | Count number of page |

##### Class AmenityDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| addAmenities() | boolean | public | Add amenity into database |
| getAmenityList() | List<Amenity> | public | Get a list of amenity |
| getPageCount() | int | public | Count number of page |
| update() | Boolean | public | Update amenity information |
| updateN() | Boolean | public | Update amenity information |
| getAmenityByPage() | List<Amenity> | public | Get a list of amenity |

##### Class AmenityGroupDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| update() | boolean | public | update amenity group information |

##### Class AppointmentDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| countAppointment() | int | public | count number of appointment |
| getAppointmentListByStatus() | List<Appointment> | public | Get appointment list which available in current time |
| getPageCount() | int | public | Get number of page |
| update() | void | public | Update information of appointment |
| updateStatus() | Boolean | public | Update status of appointment |
| updateComment() | Boolean | public | Update comment of appointment |
| getAppointmentListByStaffAndOffice () | List<Appointment> | public | Get appointment list follow staff and office |
| countRental() | int | public | Count number of rental |
| countRepair() | int | public | Coun number of repair |

##### Class ContractDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getContractListByCus() | List<Contract> | public | Get list of contract follow name of customer |
| getContractListAvailable() | List<Contract> | public | Get list of contract follow status |
| getContractByPage() | int | public | Get list of contract |
| getPageCount() | int | public | Get number of page |
| changeStatus() | boolean | public | Change status of contract |
| update() | void | public | Update information of contract |
| getCusNameByIdContract() | Object | public | Get name of customer follow contract |

##### Class OfficeAmenityDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| saveOfficeAmenity() | boolean | public | Add new amenity into office |
| removeByOffice() | boolean | public | delete amenity in office |

##### Class OfficeDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getOfficeByAddress() | List<Office> | public | Get list of office by address |
| viewOffice() | Office | public | Get information about office and give to view |
| updateArea() | boolean | public | Update information of area |
| update() | boolean | public | update information of office |
| getNewOffice() | List<Office> | public | Get 5 office have the largest of number of view |
| getAllOffice() | List<Office> | public | Get a list of office |
| getOfficeByPage() | List<Office> | public | Get a list of office |

##### Class OfficeGroupDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| deleteAll() | int | public |  |
| insertAll() | boolean | public |  |
| getOfficeList() | List<OfficeGroup> | public |  |

##### Class ProfileDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getProfileByUser() | Object | public | Get information of profile by name of user |
| update() | boolean | public | Update profile of user |

##### Class RentalDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| update() | void | public | Update information if rental |
| getRentalListByStaff() | List<Rental> | public | Get a list of rental by staff name |
| changeStatus() | boolean | public | Change status of rental |
| getRentalListByContract() | List<Rental> | public | Get a list of rental by contract |
| getRentalListByStatus() | List<Rental> | public | Get a list of rental by status |
| getPageCount() | int | public | Get number of page |
| getRentalListByFilter() | List<Rental> | public | Get a list of rental by information in filter |

##### Class RentalDetailDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| getRentalDetailByRental() | List<RentalDetail> | public | Get a list of rental detail by rental name |

##### Class RentalItemDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| update() | void | public | Update information of rental item |
| updateQuantity() | void | public | Update quantity of rental item |
| getRentalItemByPage() | List<RentalItem> | public | Get a list of rental item |
| getPageCount() | int | public | Get number of page |

##### Class RepairDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| update() | void | public | Update information of repair |
| getRepairListByStaff() | List<Repair> | public | Get a list of repair by staff |
| changeStatus() | boolean | public | Change status of repair |
| getRepairListByContract() | List<Repair> | public | Get a list of repair by contract |
| getRepairListByFilter() | List<Repair> | public | Get a list of repair by information in filter |
| getRepairListByFilter() | List<Repair> | public | Get a list of repair by information in filter |
| getPageCount() | int | public | Get number of page |
| getRepairListByStatus() | List<Repair> | public | Get a list of repair by current status of repair |

##### Class RepairDetailDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| saveRepairDetail() | boolean | public | Save information of repair detail |

##### Class RequestAmenityDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| saveRequestAmenity() | boolean | public | Save information of request amenity |

##### Class RequestOfficeDAO

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| updateOfficeSuggest() | boolean | public | Update information of office suggest |
| getAllRequestOfficeByUsername() | List<RequestOffice> | public | Get a list off request office by username |

##### Class AppointmentStatusDAO

<No attribute and method>

##### Class CategoryDAO

<No attribute and method>

##### Class PaymentTermDAO

<No attribute and method>

##### Class PriceTermDAO

<No attribute and method>

##### Class RepairStatusDAO

<No attribute and method>

##### Class RoleDAO

<No attribute and method>

#### Service Package Class Diagram

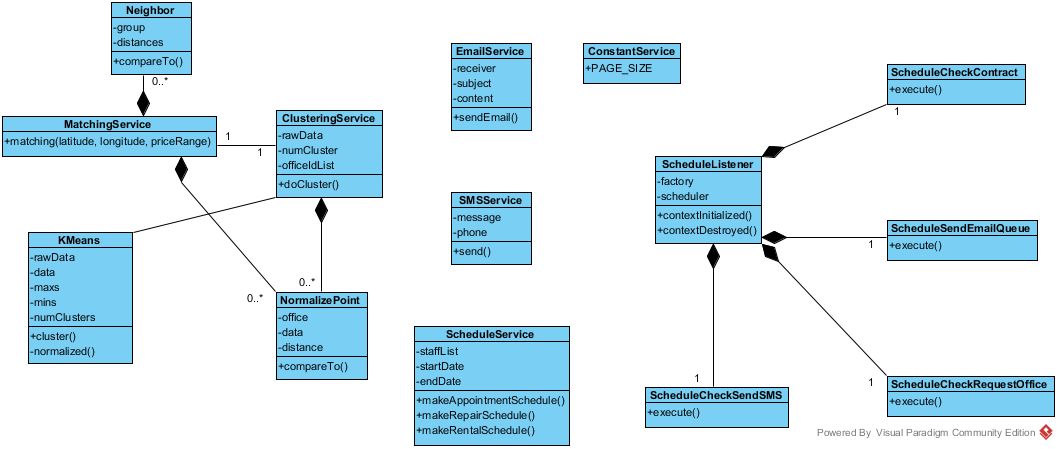


Figure 78: Service Package Class Diagram

**Class Diagram Explanation**

##### EmailService

Attribute

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

Method

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

##### SMSService

Attribute

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

Method

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

##### ConstantService

Attribute

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

##### ScheduleService

Attribute

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

Method

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

##### KMeans

Attribute

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

Method

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

##### ClusteringService

Attribute

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

Method

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

##### NormalizePoint

Attribute

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

Method

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

##### MatchingService

Method

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

##### Neighbor

Attribute

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

Method

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

##### ScheduleListener

Attribute

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

Method

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

##### ScheduleCheckContract

Method

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

##### ScheduleCheckRequestOffice

Method

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

##### ScheduleCheckSendSMS

Method

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

##### ScheduleCheckEmailQueue

Method

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

#### Controller Package Class Diagram

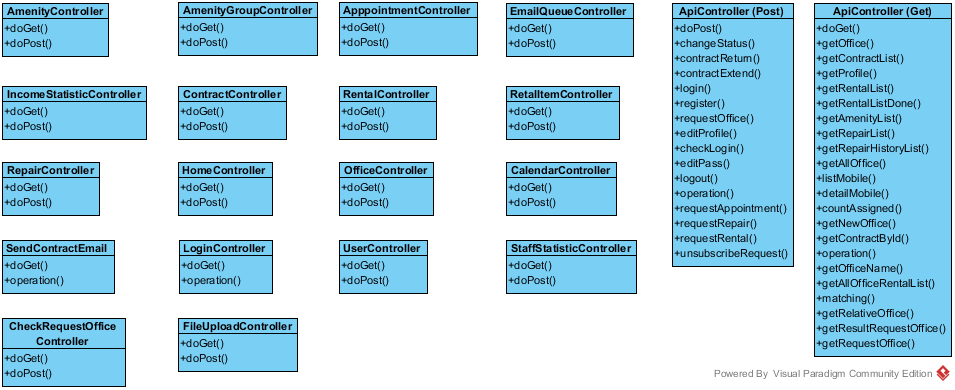


Figure 79: Class Diagram

**Class Diagram Explanation**

##### Controller

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| doGet() | void | public | Hanlde the get request from user |
| doPost | void | public | Handle the post request from user |

##### ApiController

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| getOffice() | void | public | Get the office json object by office id |
| getContractList() | void | public | Get the contract list json object by customer in session |
| getProfile() | void | public | Get the profile json object by customer in session |
| getRentalList() | void | public | Get the rental list json object by office id |
| getRentalListDone() | void | public | Get the rental list json object by office id where status is done |
| getAmenityList() | void | public | Get all the amenity list |
| getRepairList() | void | public | Get the repair list json object by office id |
| getRepairHistoryList() | void | public | Get the repair list json object by office id where status is done |
| getAllOffice() | void | public | Get list json of all office |
| listMobile() | void | public | Get the list of job json for mobile app |
| detailMobile() | void | public | Get the detail of job json for mobile app |
| countAssigned() | void | public | Count the assigned job for mobile |
| searchOfficeByAddress() | void | public | Get the office list filter by address |
| getNewOffice() | void | public | Get the office list has most viewed |
| getContractById() | void | public | Get the contract json by id |
| amenity() | void | public | Get the amenity list |
| officeName() | void | public | Get the list of all office name |
| getAllOfficeRentalList() | void | public | Get the list of all rental json by office |
| matching() | void | public | Get the matching office list by search criteria |
| getRelativeOffice() | void | public | Get the relative office by current office |
| getResultRequestOffice() | void | public | Get the result office list that had requested |
| getRequestOffice() | void | public | Get the list of request office by customer |
| changeStatus() | void | public | Change the status of the job in mobile app |
| contractReturn() | void | public | Request return office |
| contractExtend() | void | public | Request extend contract |
| login() | void | public | Request login in front end app |
| register() | void | public | Request register in front end app |
| requestOffice() | void | public | Request office |
| editProfile() | void | public | Edit profile |
| checkLogin() | void | public | Check the status login in current session |
| editPass() | void | public | Change password of current customer |
| logout() | void | public | Customer log out |
| requestAppointment() | void | public | Request appointment by customer |
| requestRepair() | void | public | Request repair by customer |
| requestRental() | void | public | Request rental by customer |
| unsubscribeRequest() | void | public | Unsubscribe the office request |

#### View Package Class Diagram

Figure 80: Class Diagram

**Class Diagram Explanation**

##### Class 1

Attribute

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

Method

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

##### Class 2

Attribute

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

Method

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

#### Front Package Class Diagram

Figure 81: Class Diagram

**Class Diagram Explanation**

##### Class 1

Attribute

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

Method

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

##### Class 2

Attribute

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

Method

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

#### Mobile Package Class Diagram

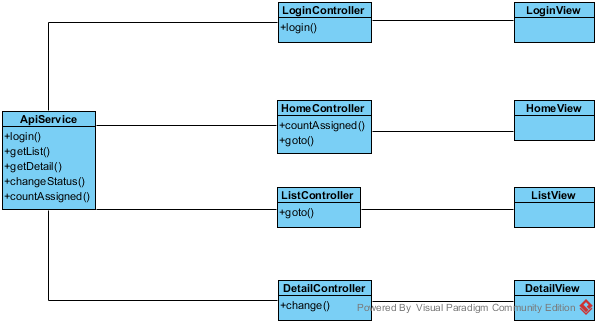


Figure 82: Mobile Package Class Diagram

**Class Diagram Explanation**

##### ApiService

Method

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

##### LoginController

Attribute

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

Method

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

##### LoginView

This is the presentation view of LoginController

##### HomeController

Attribute

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

Method

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

##### HomeView

This is the presentation view of HomeController

##### ListController

Attribute

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

Method

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

##### ListView

This is the presentation view of ListController

##### DetailController

Attribute

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

Method

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

##### DetailView

This is the presentation view of DetailController

### Interaction Diagram

#### <Guest> Login

**Summary:** This diagram used to describe the process of guest login

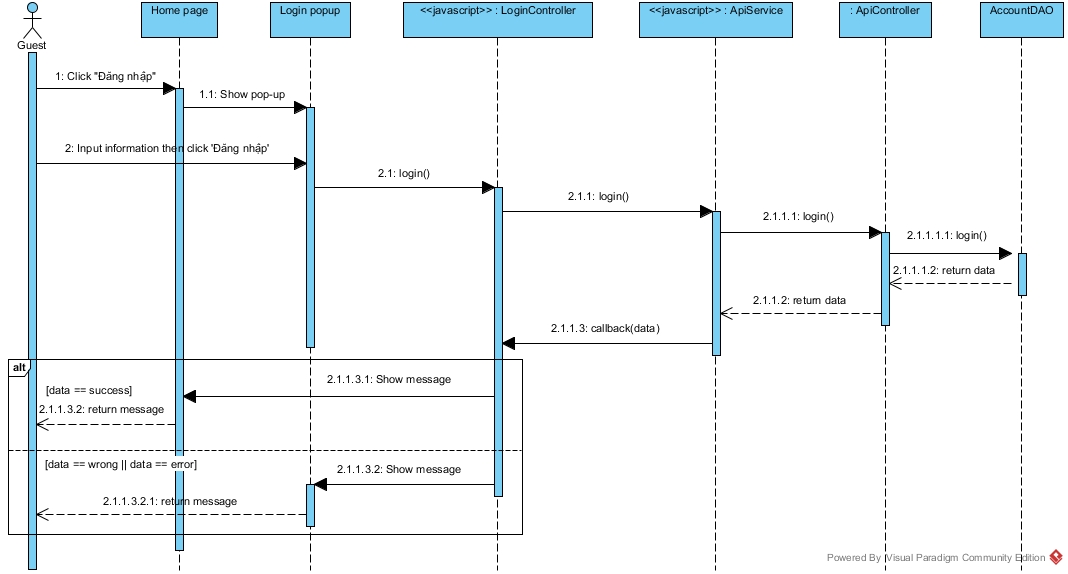


Figure 74: Login Sequence Diagram

#### <Guest> Register

**Summary:** This diagram used to describe the process of guest register account

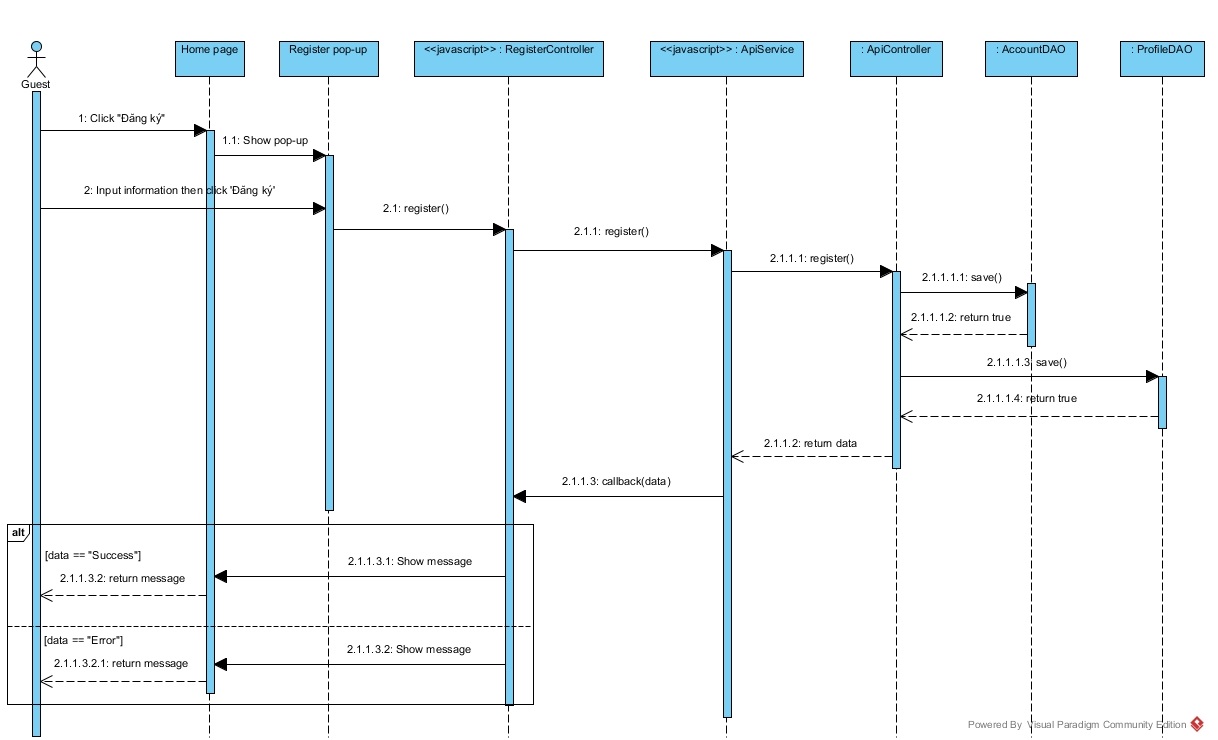


Figure 74: Register Sequence Diagram

#### <Customer> Get Office List

**Summary:** This diagram used to describe the process of customer get new office list

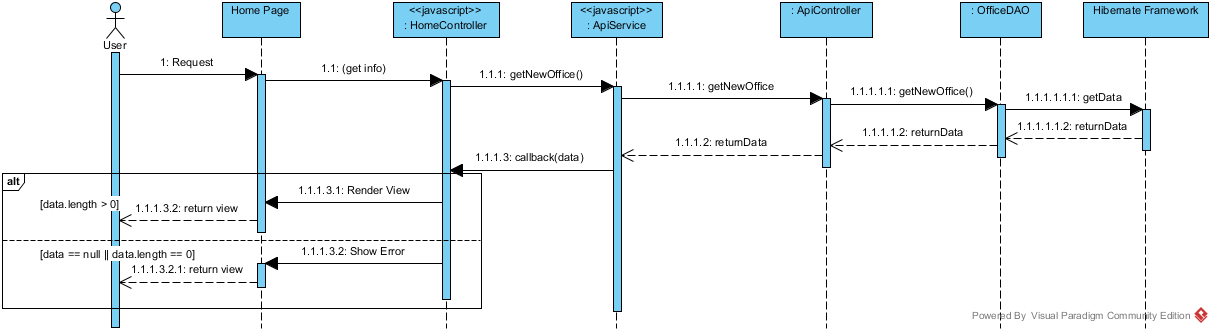


Figure 84: Get office list Sequence Diagram

#### <Customer> Request Appointment

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

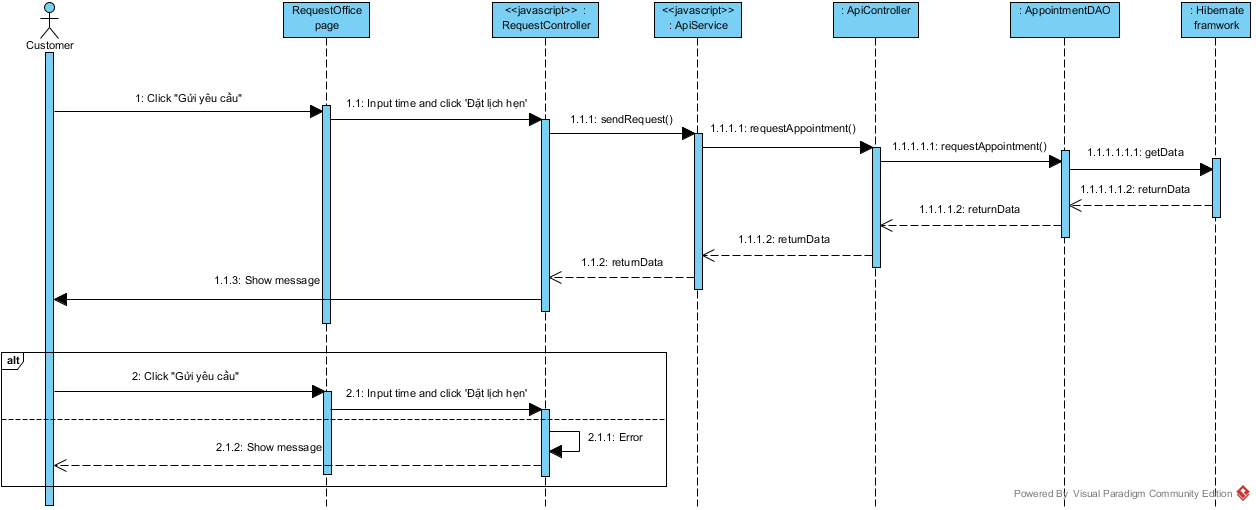


Figure 85: Request Appointment Sequence Diagram

#### <Customer> Create Request new Office

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

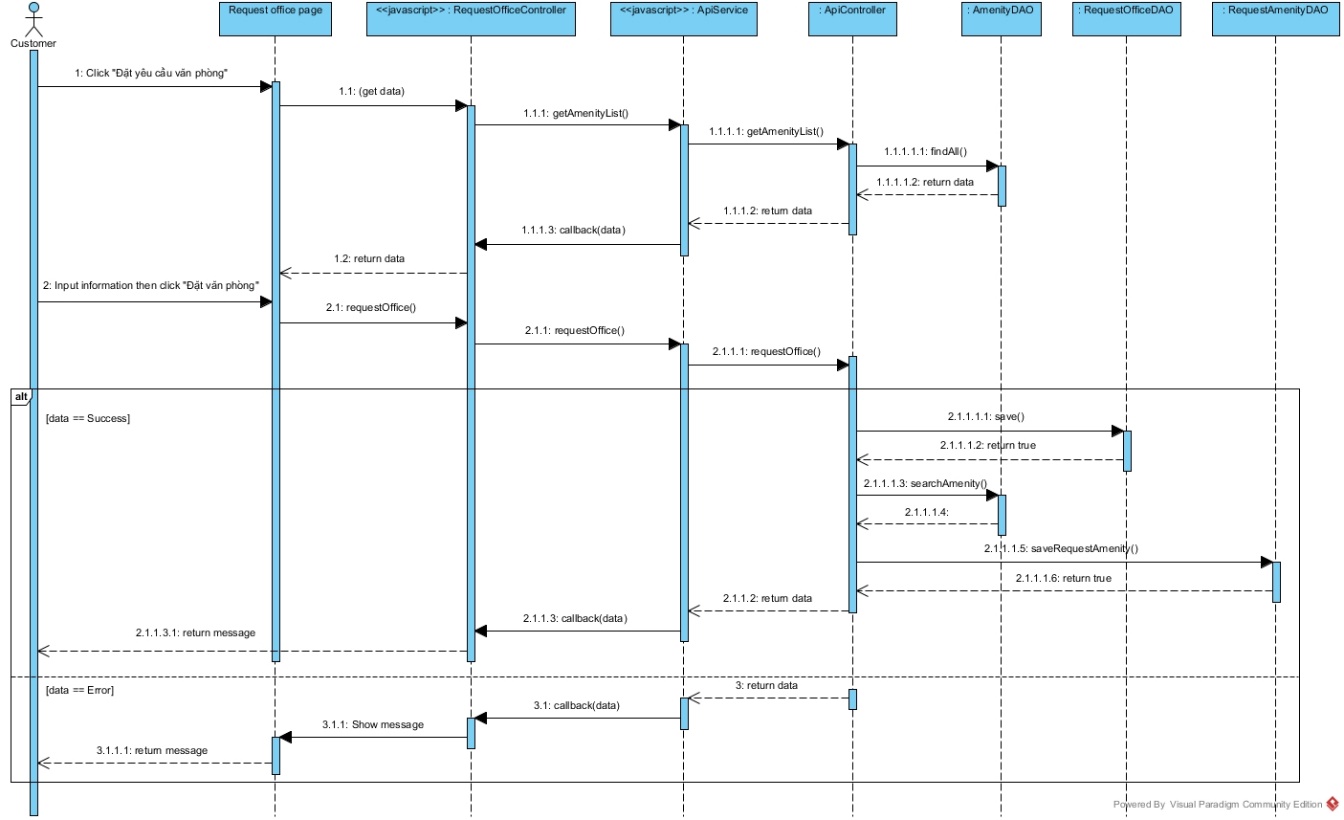


Figure 86: Create Appointment Request Sequence Diagram

#### <Customer> View List Contract

**Summary:** This diagram used to describe the process of customer view list contract

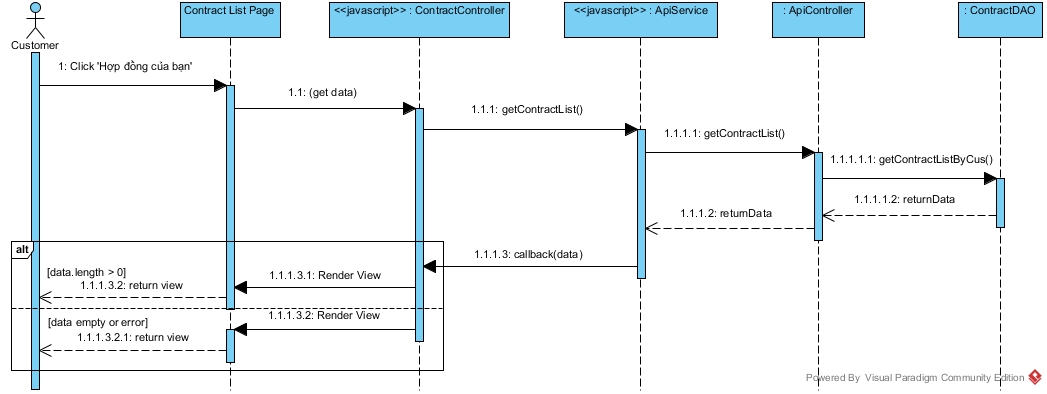


Figure 74: View List Contract Sequence Diagram

#### <Customer> View contract detail

**Summary:** This diagram used to describe the process of customer view contract detail

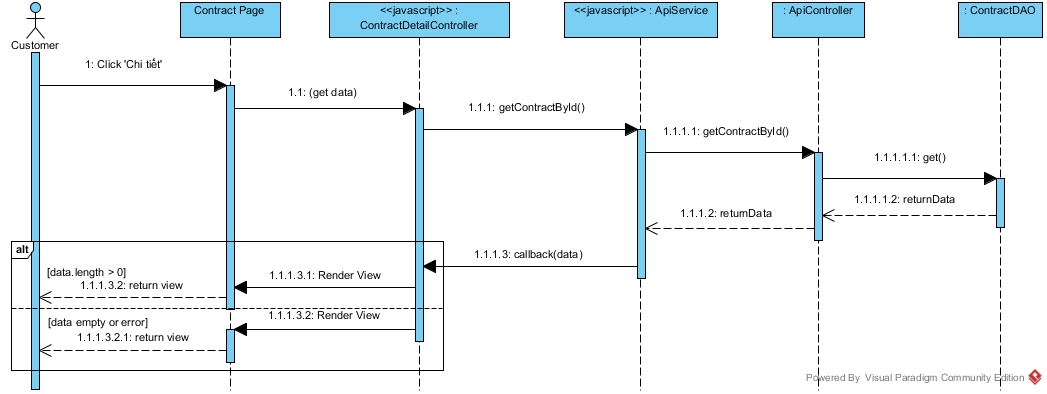


Figure 74: View Contract Sequence Diagram

#### <Customer> Request Extend Contract

**Summary:** This diagram used to describe the process of customer request extend contract

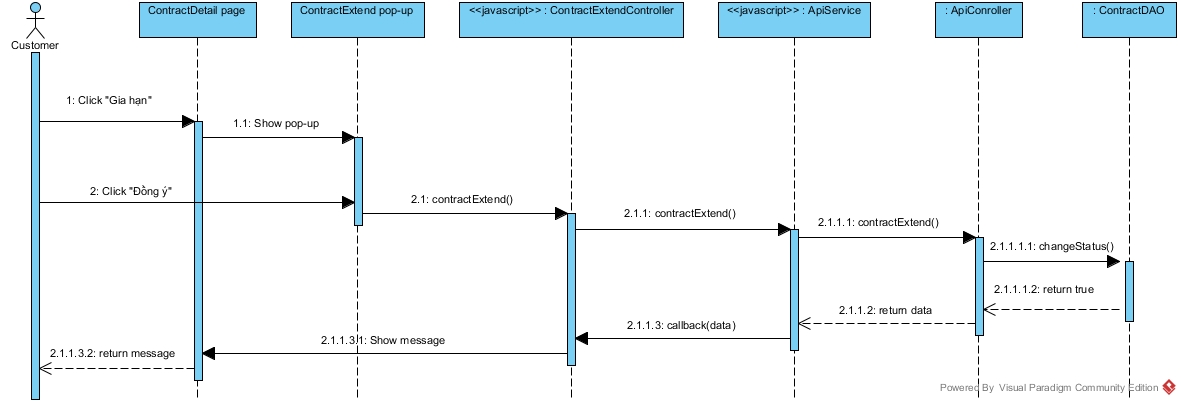


Figure 87: Request Extend Contract Sequence Diagram

#### <Customer> Request Return Contract

**Summary:** This diagram used to describe the process of customer request return contract

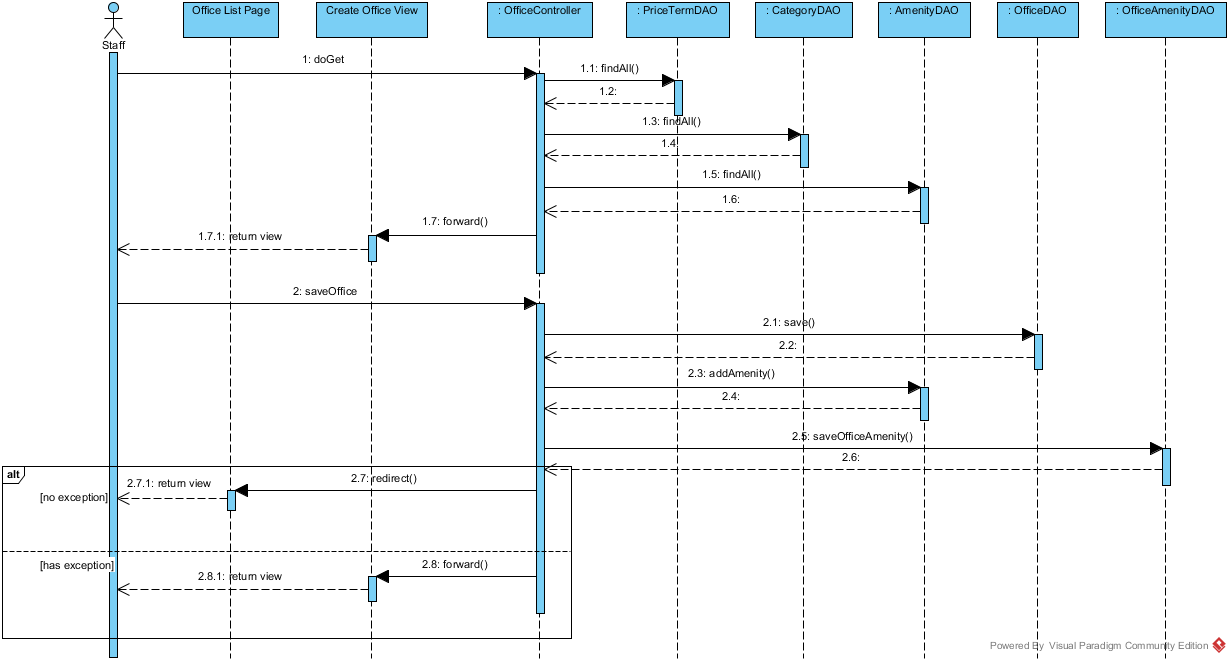


Figure 74: Request Return Contract Sequence Diagram

#### <Customer> Request Rental

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

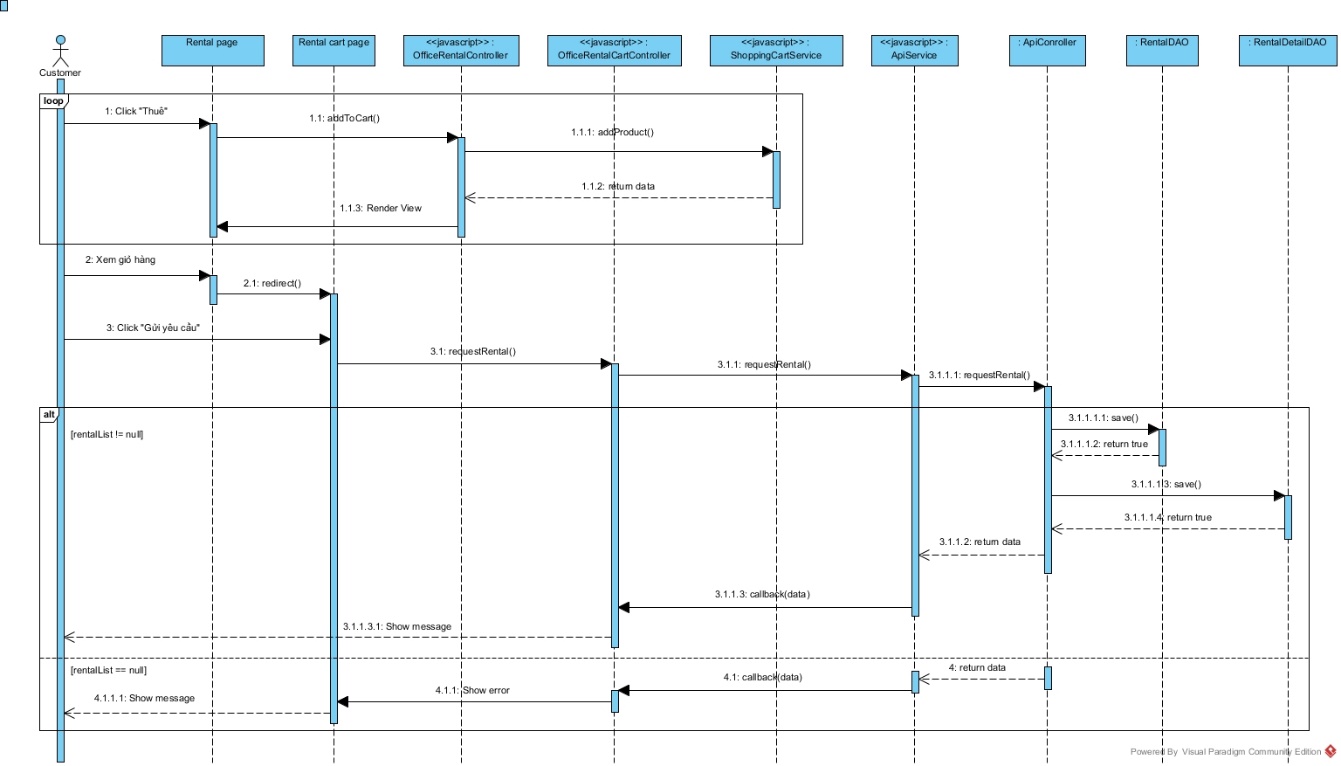


Figure 74: Create office Sequence Diagram

#### <Customer> Request Repair

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

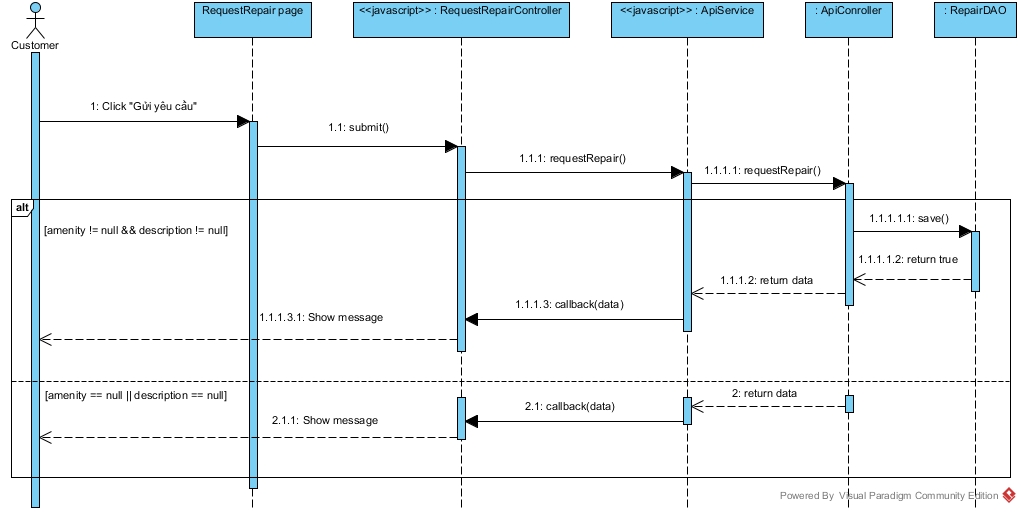


Figure 74: Request Repair Sequence Diagram

#### <Customer> View Rental Request List

**Summary:** This diagram used to describe the process of customer view rental request list



Figure 74: View Rental Request List Sequence Diagram

#### <Customer> View Repair Request List

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

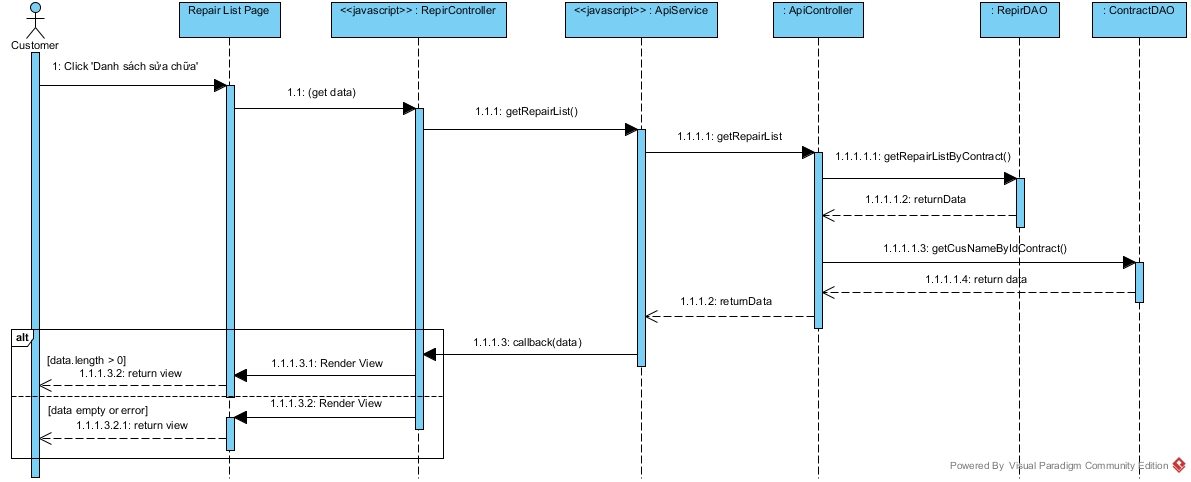


Figure 74: View Repair Request List Sequence Diagram

#### <Customer> View List Repair History

**Summary:** This diagram used to describe the process of customer view list repair history

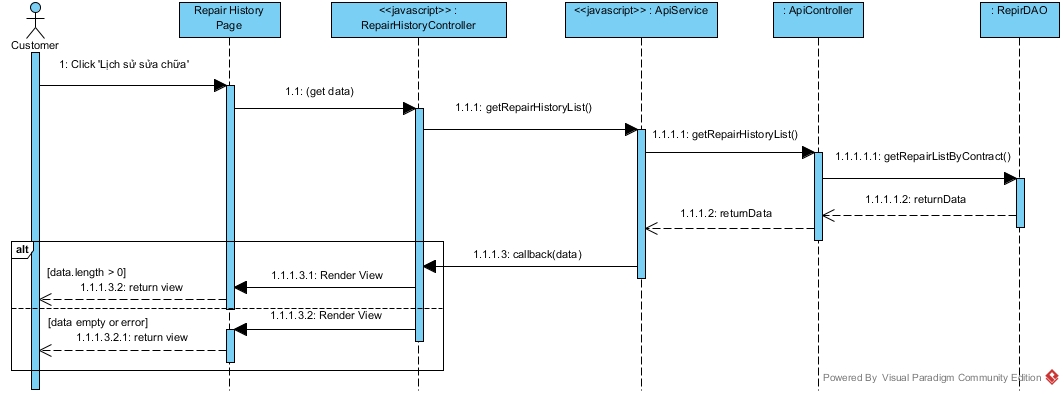


Figure 74: View List Repair History Sequence Diagram

#### <Customer> View Customer Info

**Summary:** This diagram used to describe the process of customer view profile

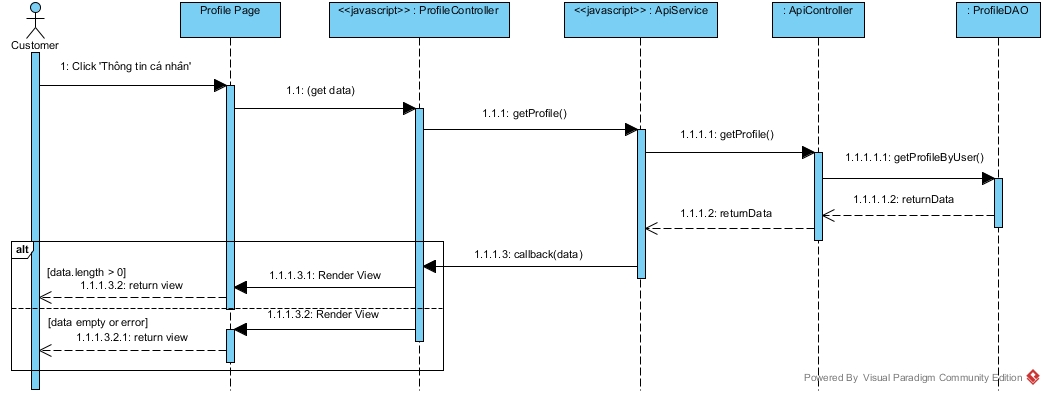


Figure 74: View Customer Info Sequence Diagram

#### <Customer> Create Office Request

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

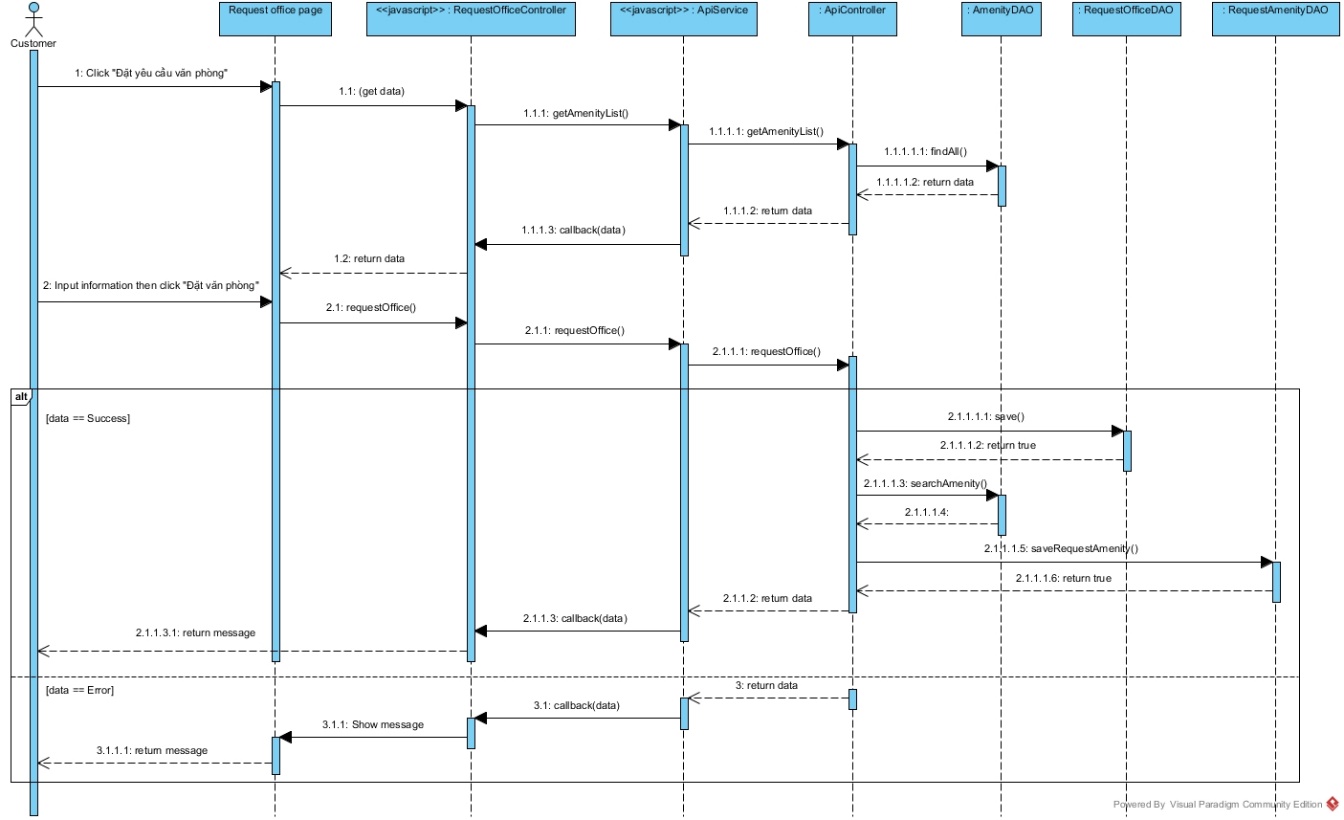


Figure 74: Create Office Request Sequence Diagram

#### <Customer> Change Password

**Summary:** This diagram used to describe the process of customer change password

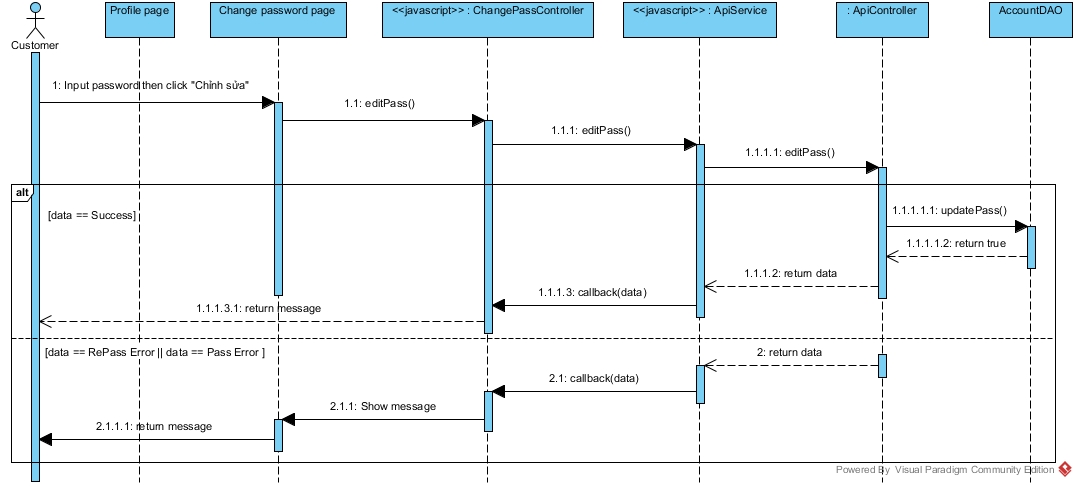


Figure 89: Change Password Sequence Diagram

#### <Customer> Edit Customer Info

**Summary:** This diagram used to describe the process of customer edit profile

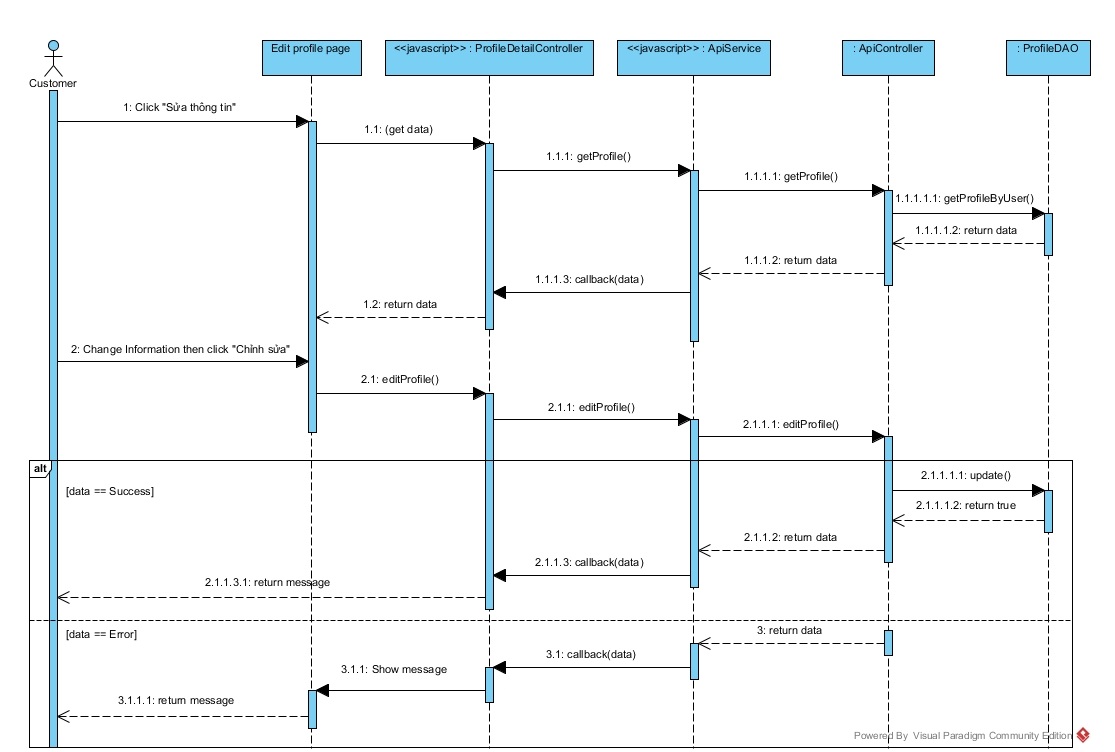


Figure 90: Edit Customer Info Sequence Diagram

#### <Manager, Staff> View Rental Item List

**Summary:** This diagram used to describe the process of manager or staff view rental item list

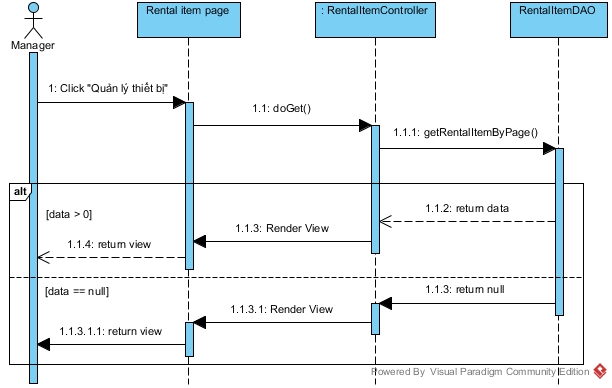


Figure 90: View Rental Item List Sequence Diagram

#### <Manager, Staff> Create Rental Item

**Summary:** This diagram used to describe the process of manager or staff create rental item

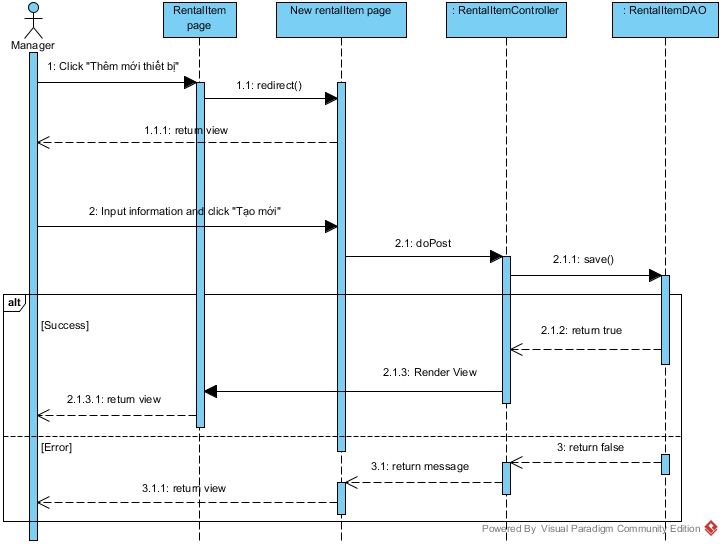


Figure 91: Create Rental Item Sequence Diagram

#### <Manager, Staff> Edit Rental Item

**Summary:** This diagram used to describe the process of manager or staff edit rental item

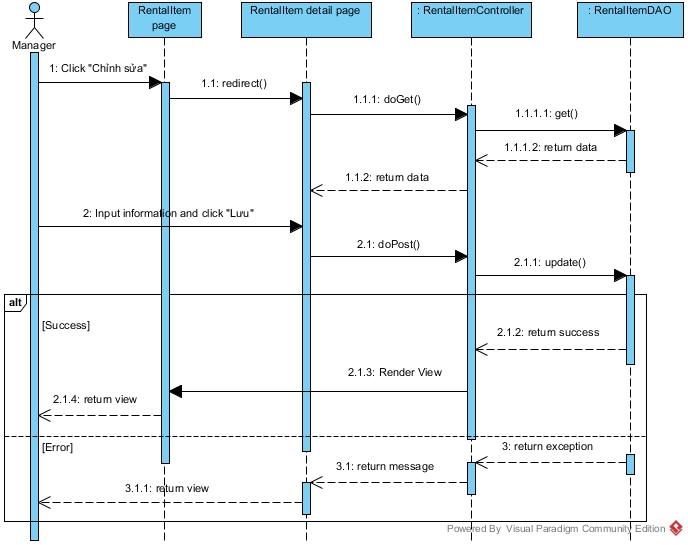


Figure 92: Edit Rental Item Sequence Diagram

#### <Manager, Staff, Admin> Log out

**Summary:** This diagram used to describe the process of manager, staff or admin logout

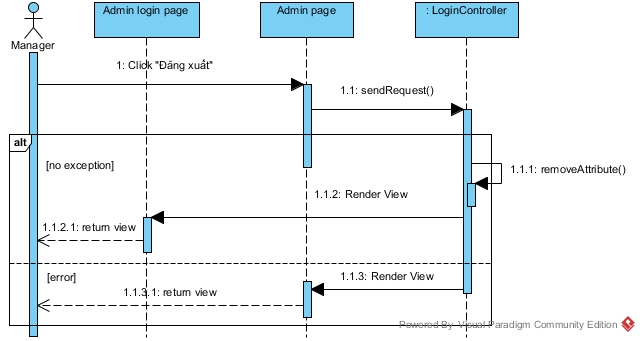


Figure 74: Log out Sequence Diagram

#### <Staff, Manager> Create office

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

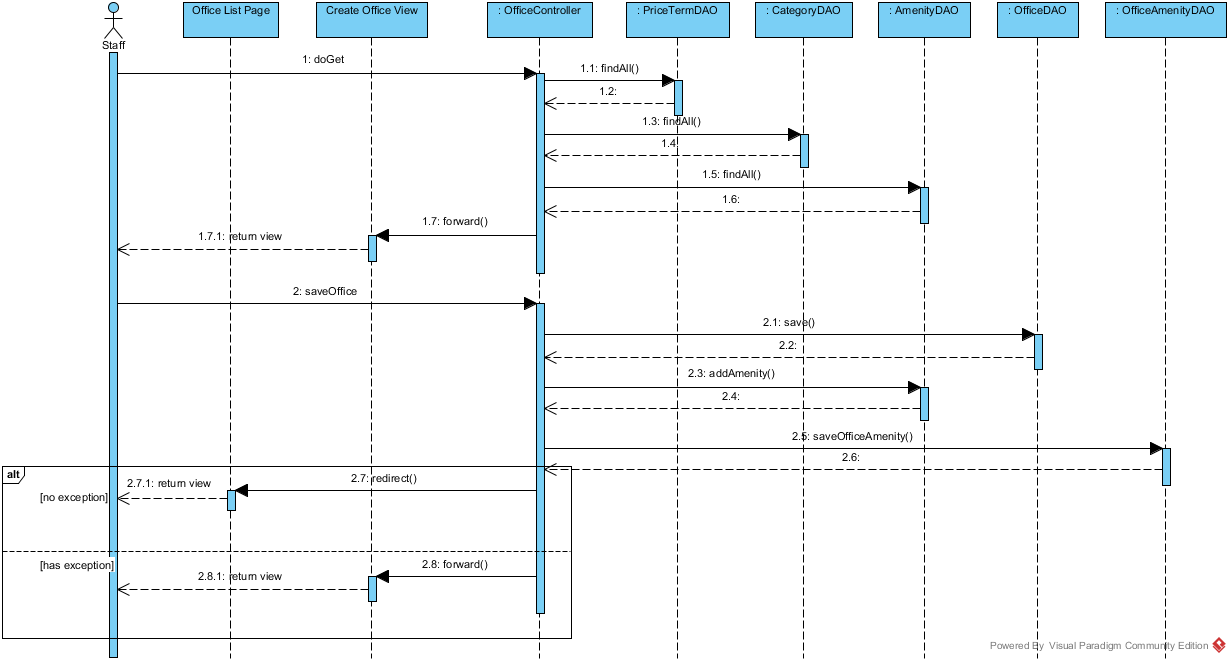


Figure 93: Create office Sequence Diagram

#### <Manager> Create contract

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

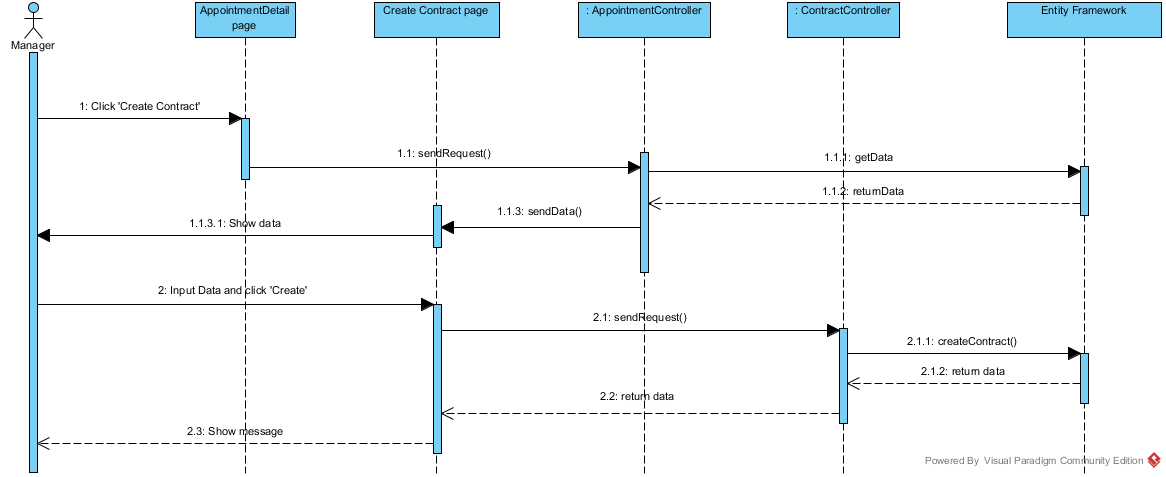


Figure 94: Create Contract Sequence Diagram

#### <Manager> Assign Appointment

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

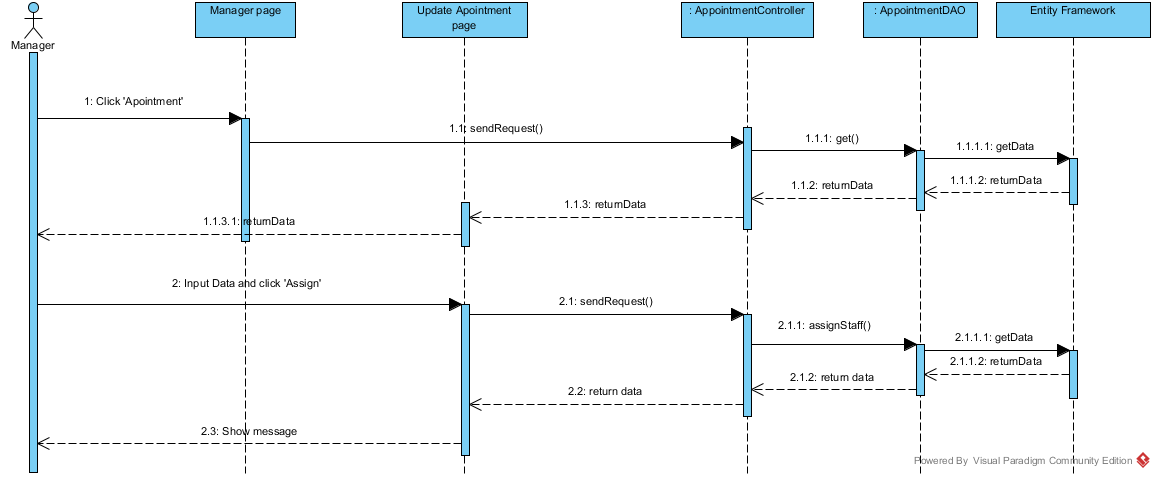


Figure 95: Assign Appointment Sequence Diagram

#### <Manager> Update Appointment

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

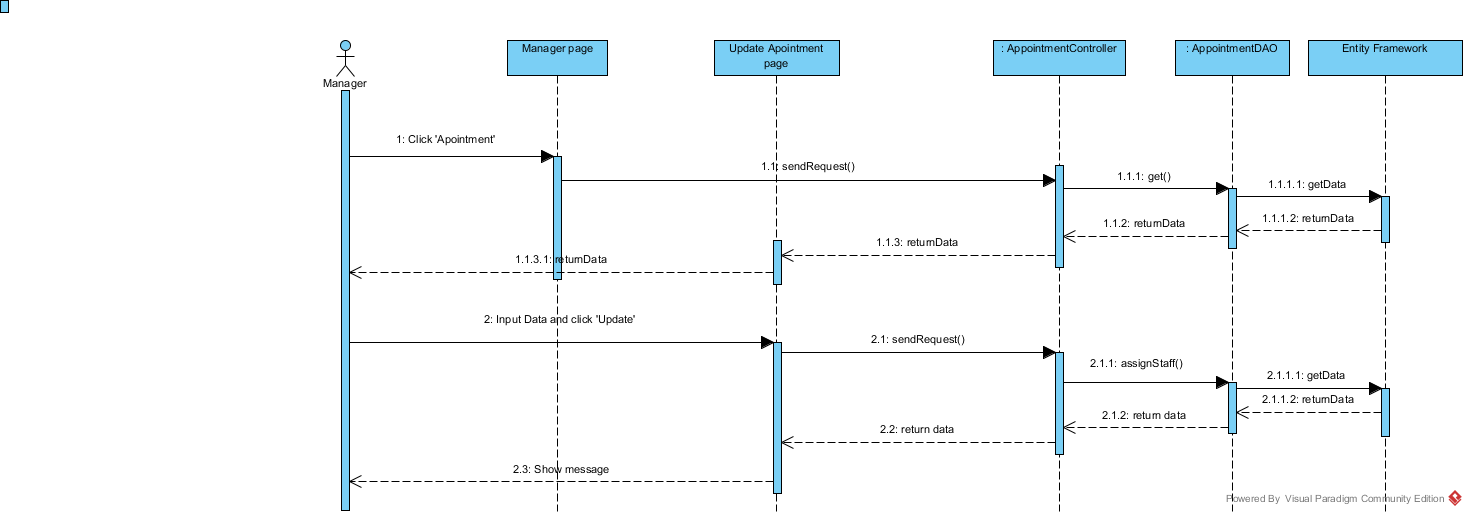


Figure 96: Update Appointment Sequence Diagram

#### <Manager> View Appointment Request List

**Summary:** This diagram used to describe the process of manager view all appointment requests.

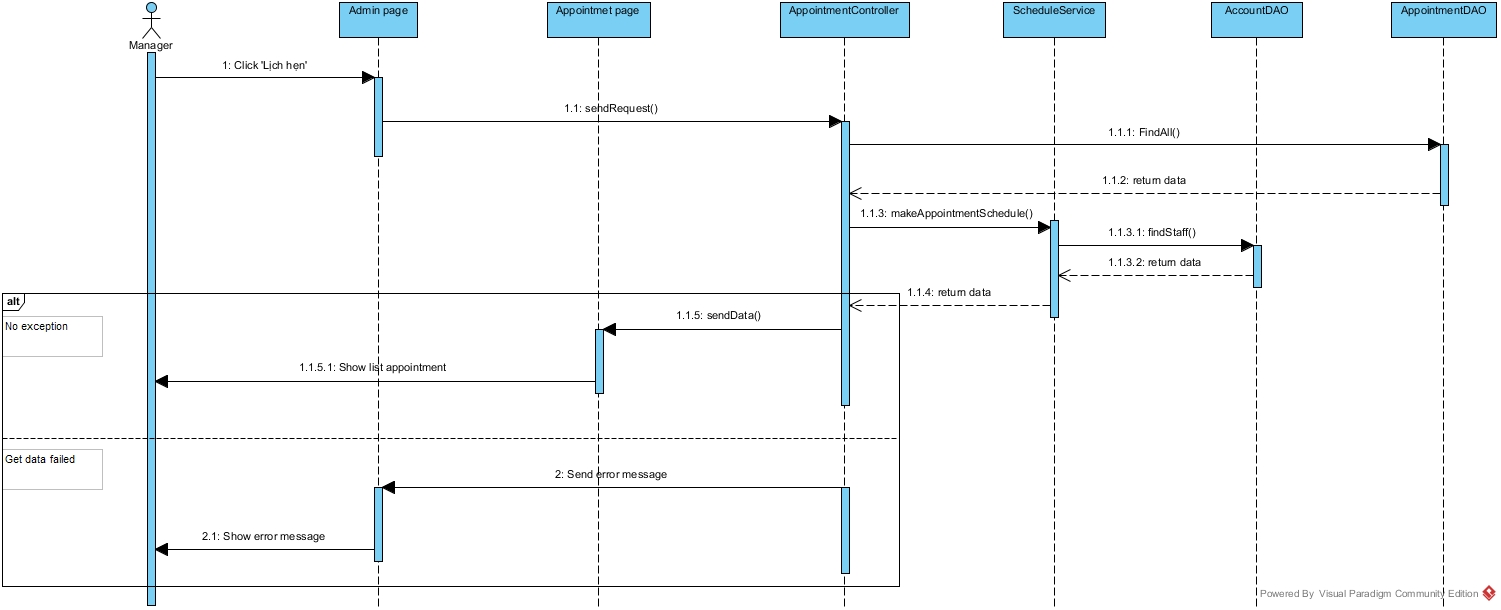


Figure 97: View Appointment Request List

#### <Manager> Assign Appointment

**Summary:** This diagram used to describe the process of manager assign appointment requests to staff.

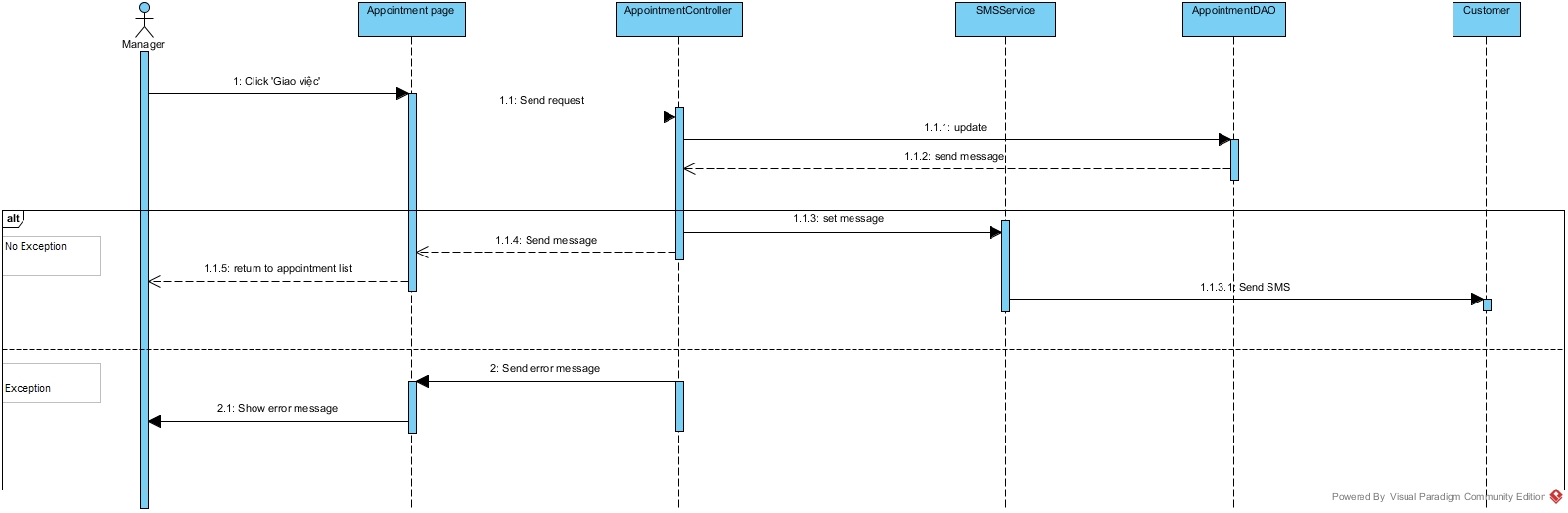


Figure 98: Assign Appointment

#### <Manager> Cancel Appointment

**Summary:** This diagram used to describe the process of manager cancel appointment

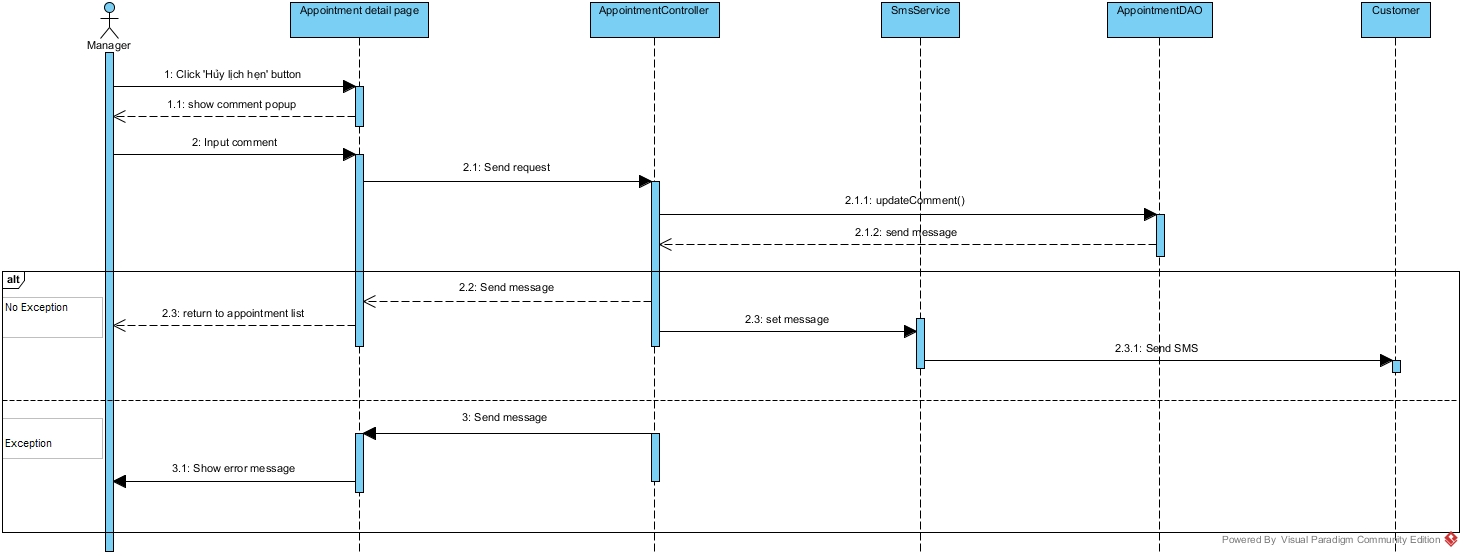


Figure 99: Cancel Appointment

#### <Manager> Create Contract

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

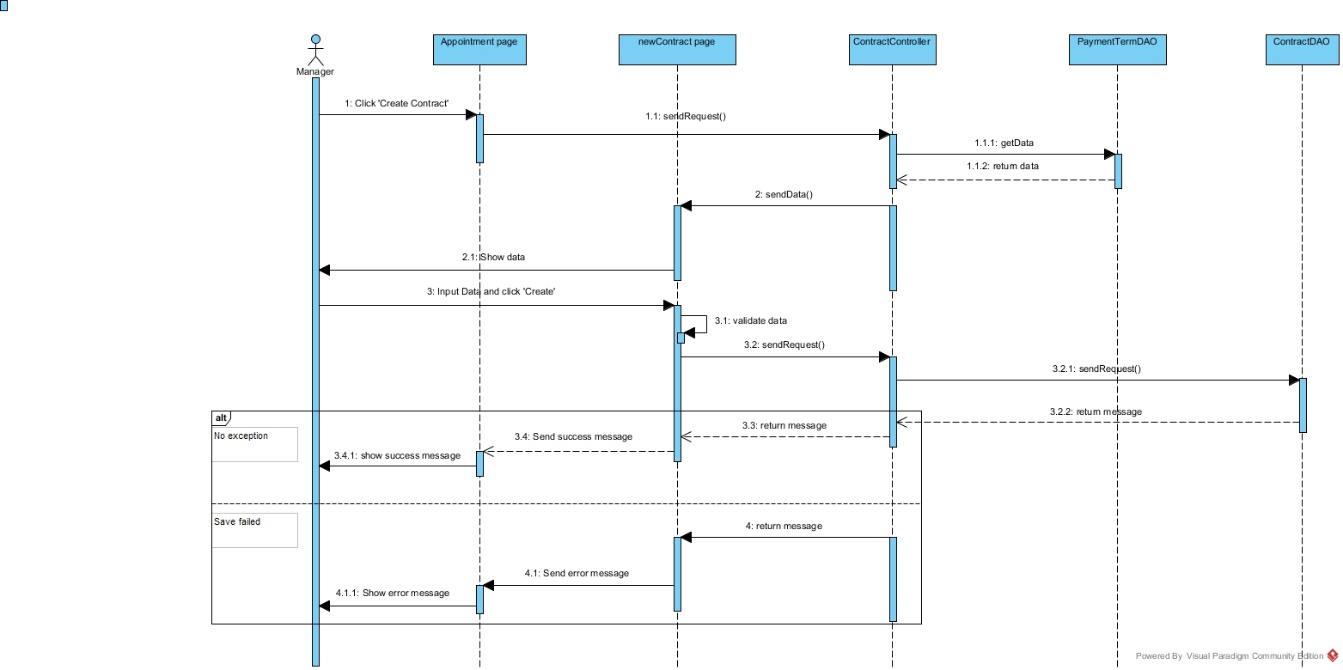


Figure 100: Create Contract

#### <Manager> View Contract List

**Summary:** This diagram used to describe the process of manager view contract list

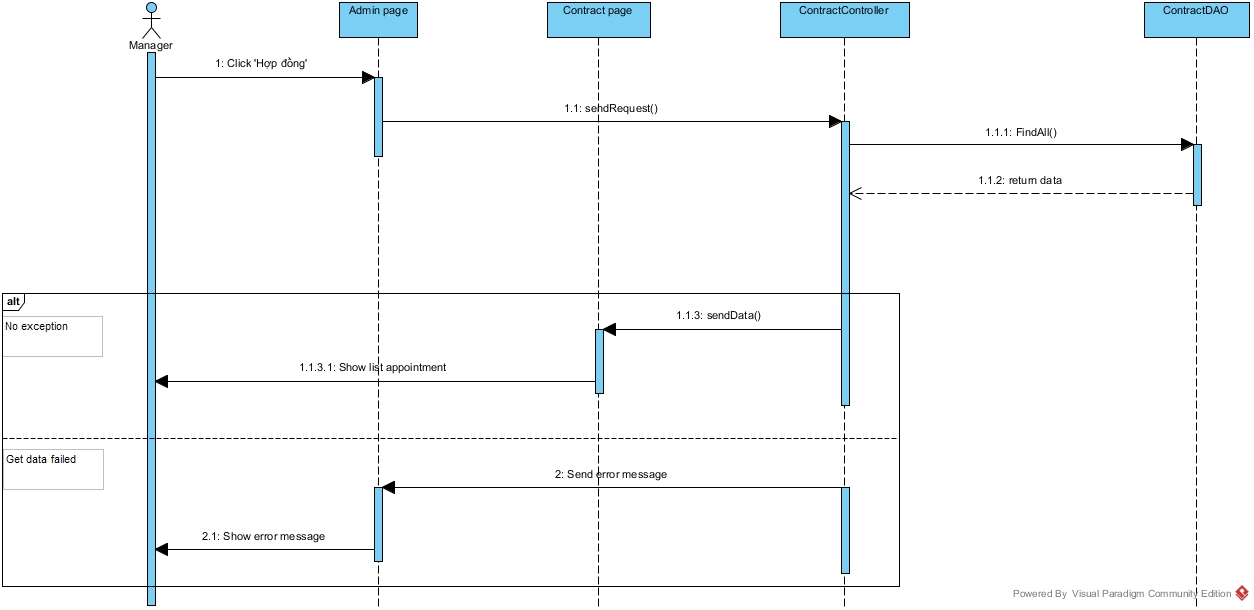


Figure 101: View Contract List

#### <Manager> View Contract Detail

**Summary:** This diagram used to describe the process of manager view contract detail

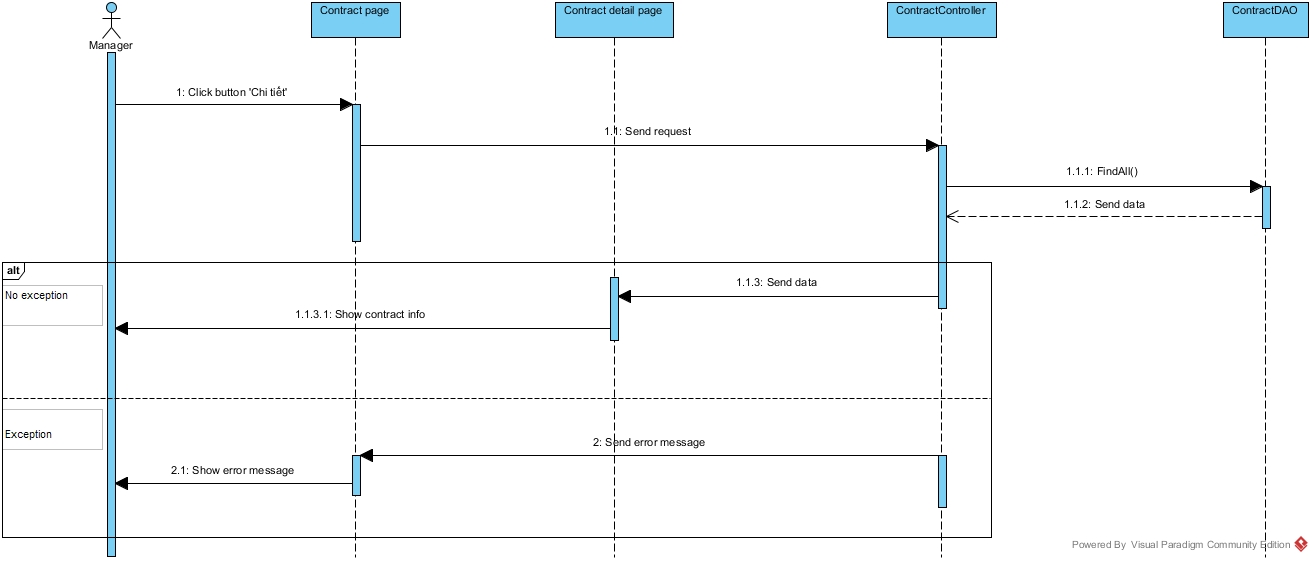


Figure 102: View Contract Detail

#### <Manager> Edit Contract

**Summary:** This diagram used to describe the process of manager edit contract.

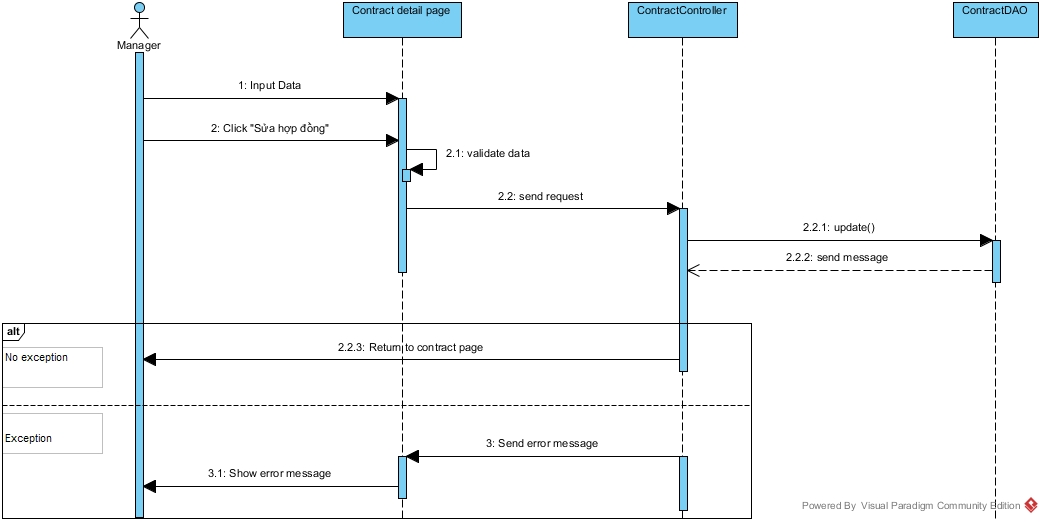


Figure 103: Edit Contract

#### <Manager> View Repair Request

**Summary:** This diagram used to describe the process of manager view repair requests.

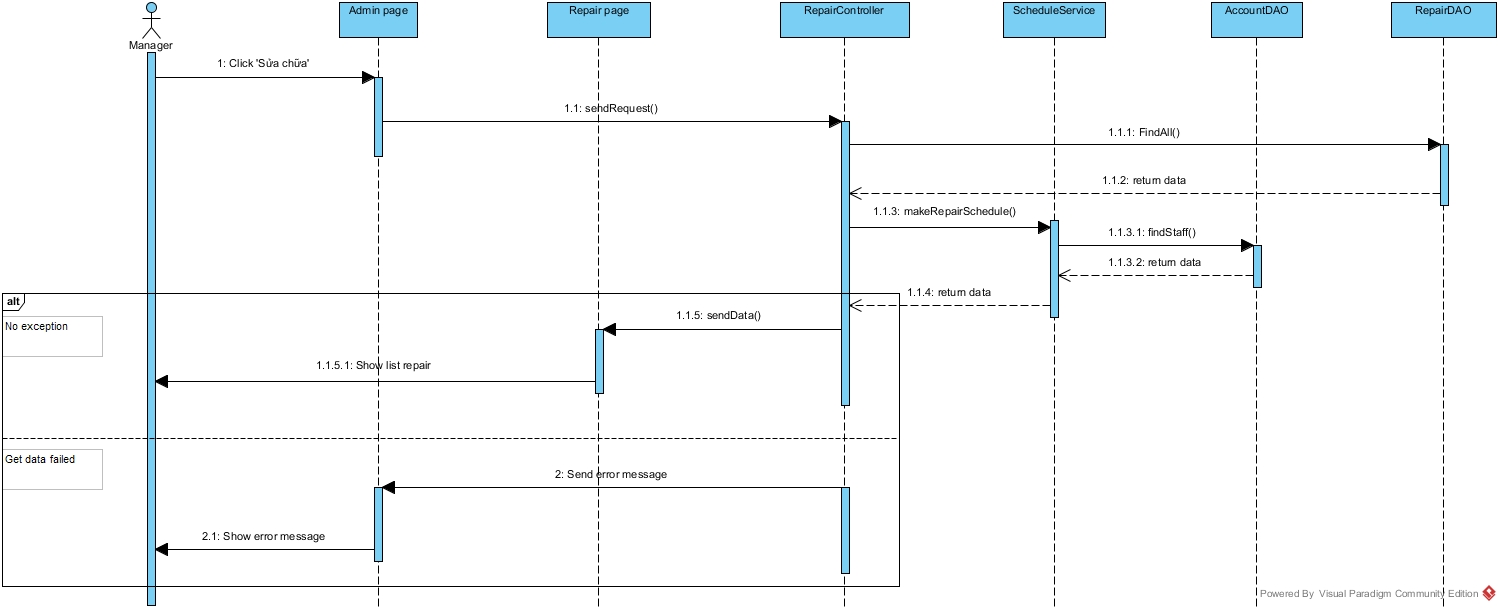


Figure 104: View Repair Request

#### <Manager> Assign Repair Request

**Summary:** This diagram used to describe the process of manager assign repair requests to staff.

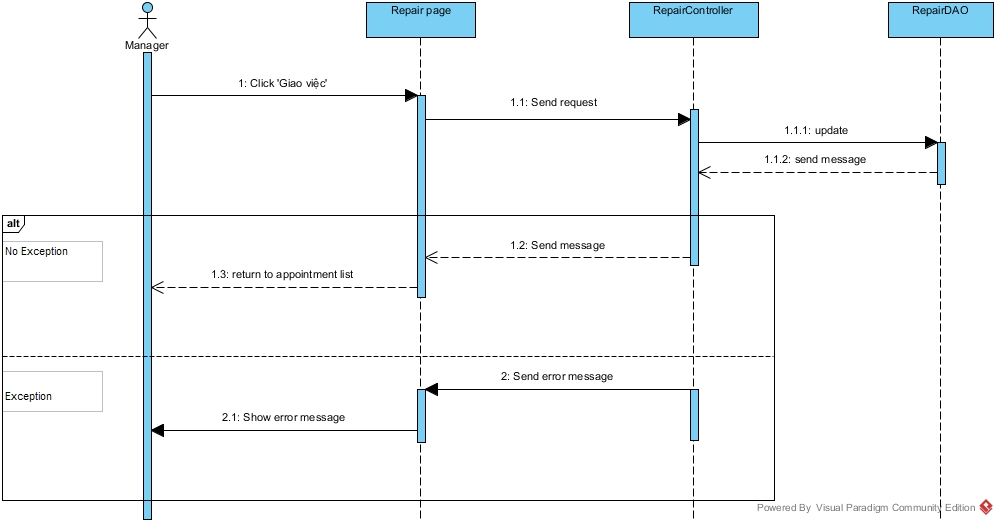


Figure 105: Assign Repair Request

#### <Manager> Cancel Repair Request

**Summary:** This diagram used to describe the process of manager cancel repair requests.

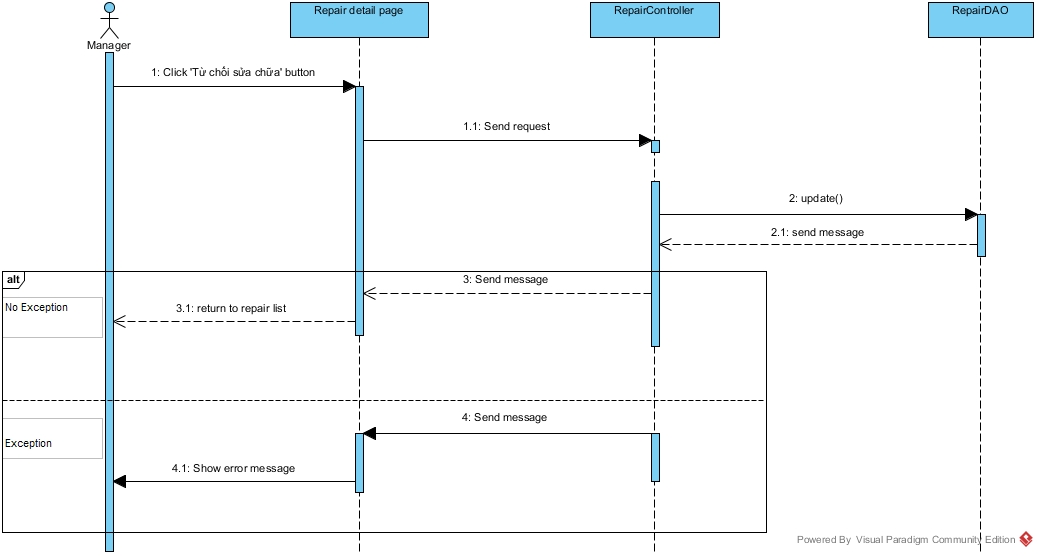


Figure 106: Cancel Repair Request

#### <Manager> View Rental Request

**Summary:** This diagram used to describe the process of manager view rental requests.

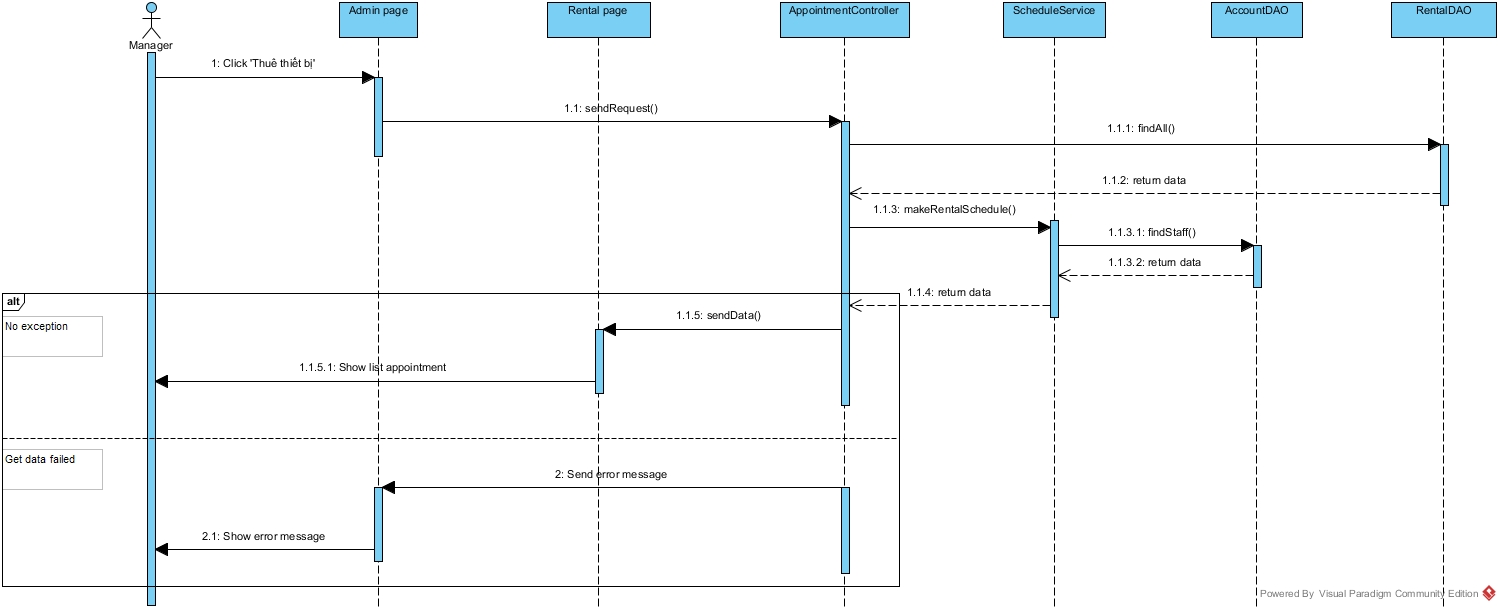


Figure 107: View Rental Request

#### <Manager> Assign Rental Request

**Summary:** This diagram used to describe the process of manager assign rental requests to staff.

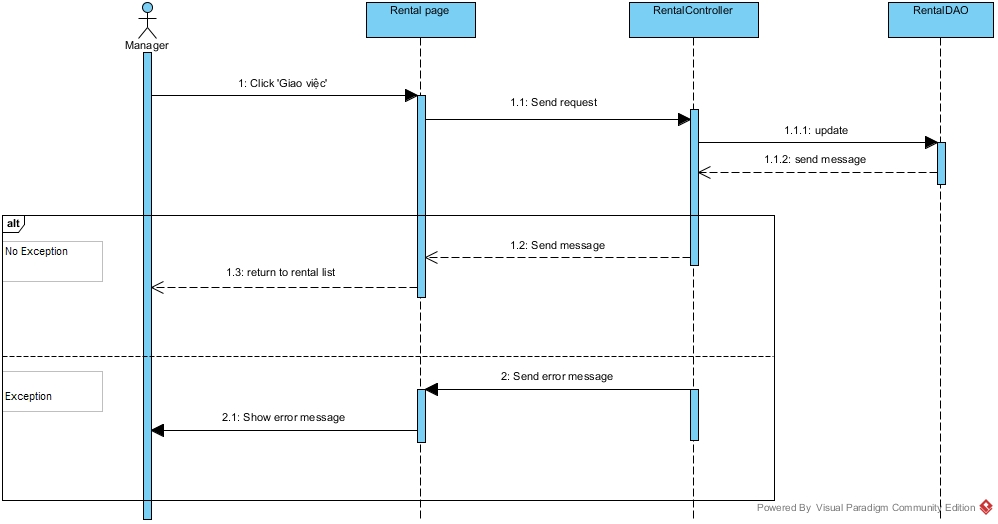


Figure 108: Assign Rental request

#### <Manager> Cancel Rental Request

**Summary:** This diagram used to describe the process of manager cancel rental request.

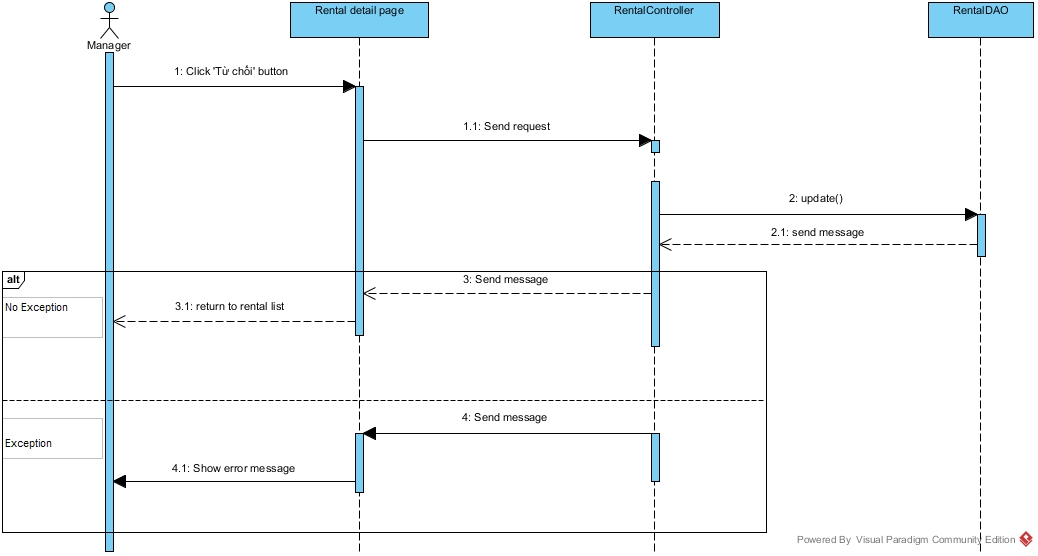


Figure 109: Cancel Rental Request

#### <Customer> View Result Request Office

**Summary:** This diagram used to describe the process of customer view result request office.

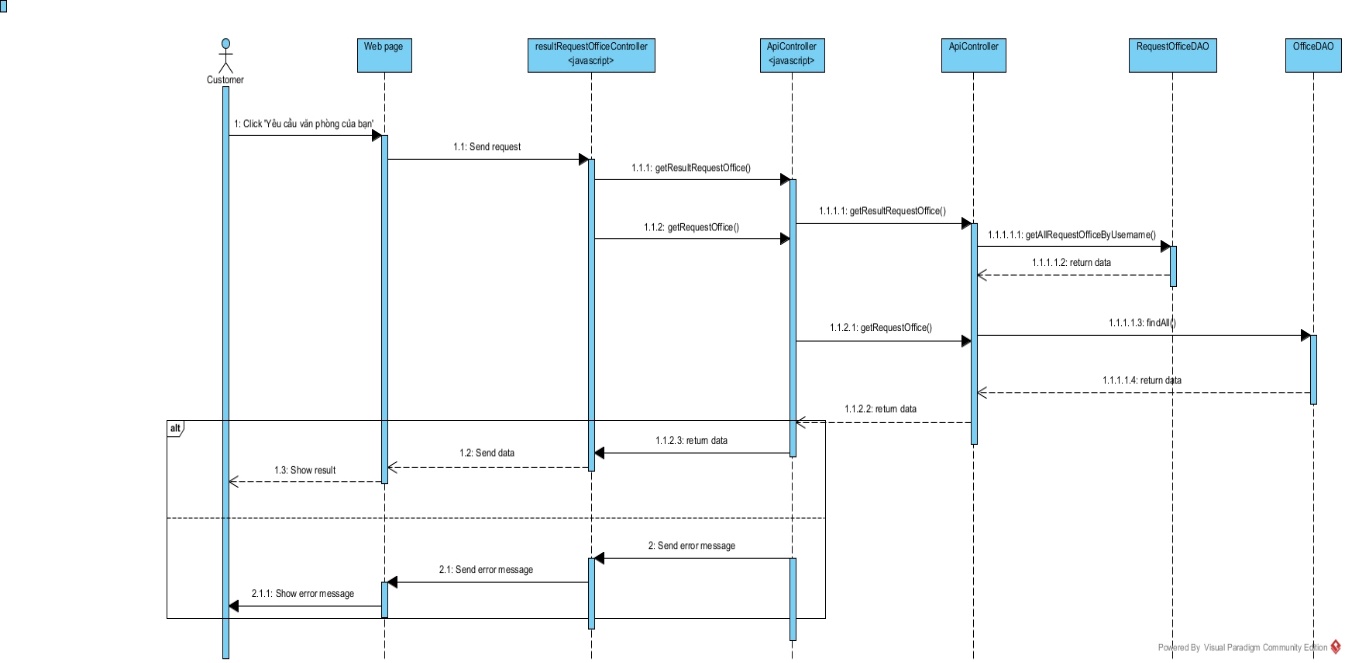


Figure 110: View Result Request Office

#### <System> Check Contract Due Date

**Summary:** This diagram used to describe the process of system check contract due date.

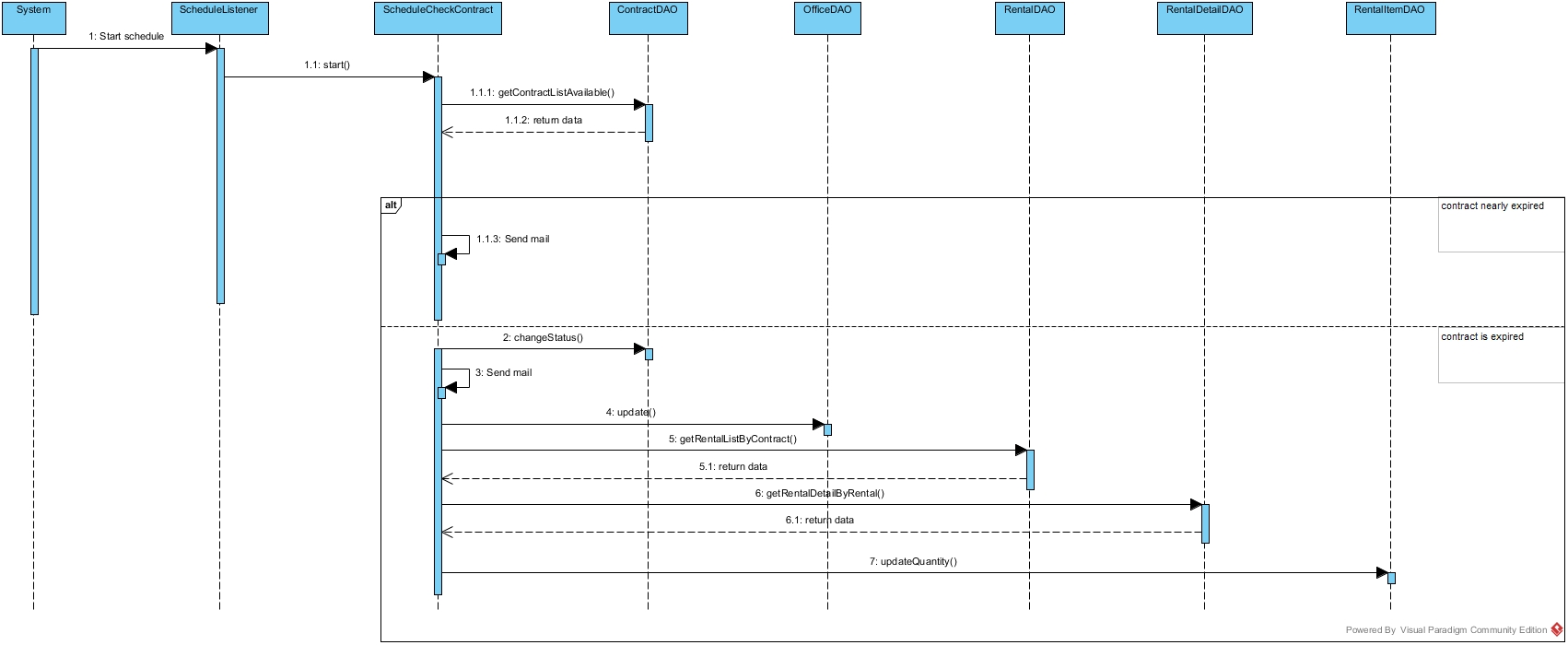


Figure 111: Check Contract Due Date

#### <System> Suggest Request Office

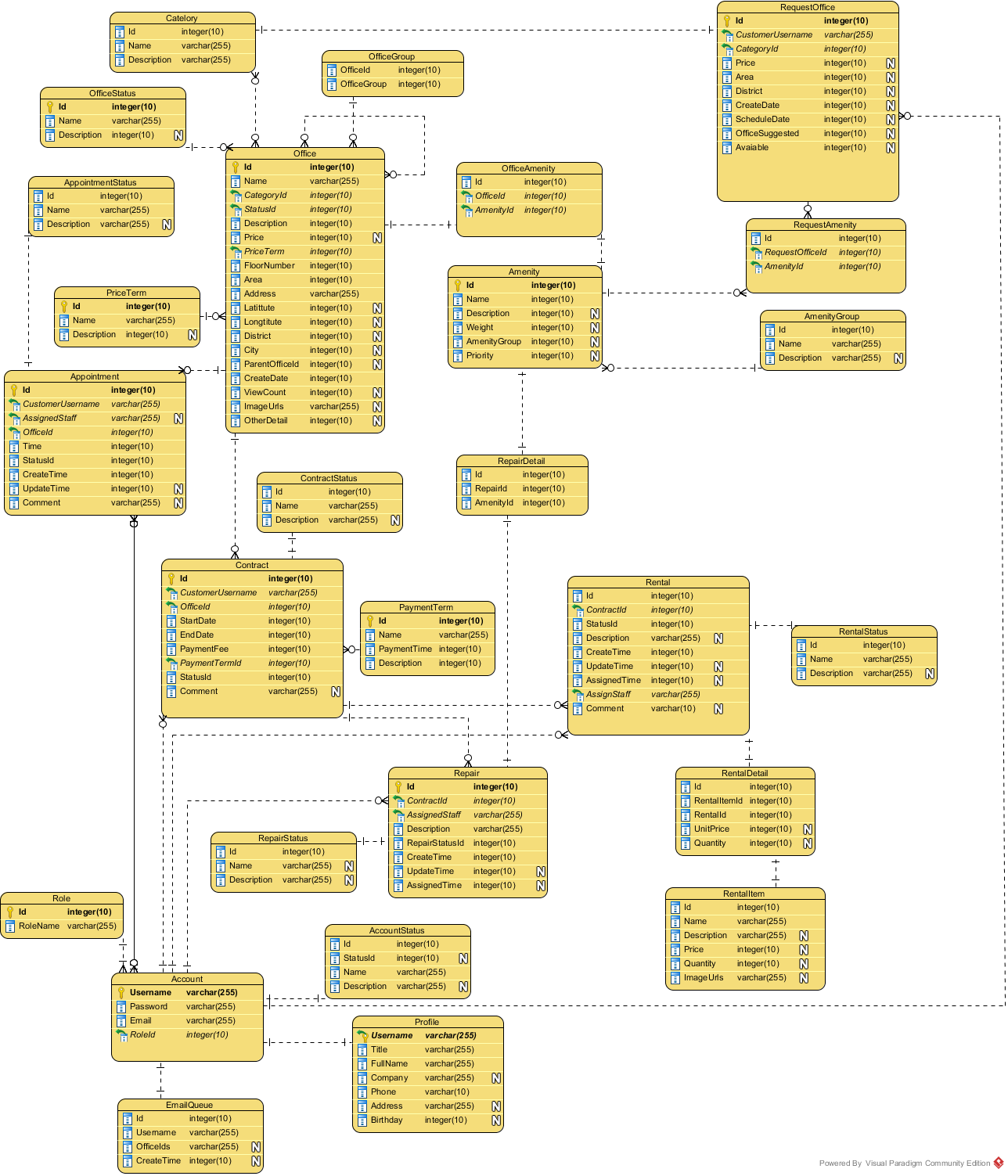
**Summary:** This diagram used to describe the process of system suggest request office to customer.



Figure 112: Suggest Request Office

## Database Design

### Logical Diagram



**Figure 113: Logical Database Diagram**

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| Account | Describe all accounts of users in the system. |
| Account Status | Describe status of account. One account has only one status. |
| Amenity | Describe all amenities in system. |
| Amenity Group | Describe groups of amenity. One amenity is only in one group. |
| Appointment | Describe all appointment requests of customer. |
| Appointment Status | Describe status of appointment request. One appointment has only one status. |
| Category | Describe category of office. One office is only in one category. |
| Contract | Describe all contracts in the system. |
| Contract Status | Describe status of contract. One contract has only one status. |
| Email Queue | Describe all emails have to send to customer in the system. |
| Office | Describe all offices in the system. |
| Office Amenity | Describe all amenities in one office. One office can have more than one amenity. |
| Office Group | Describe group of the office for searching office. One office is only in one group |
| Office Status | Describe status of the office. One office has only one status. |
| Payment Term | Describe time for the payment |
| Price Term | Describe the way that money is calculated base on. |
| Profile | Describe information of the account |
| Rental | Describe all rental requests from customer |
| Rental Detail | Describe all rental items for one rental request |
| Rental Item | Describe all rental items in system |
| Rental Status | Describe status of rental request. One rental request has only one status. |
| Repair | Describe all repair requests from customer |
| Repair Detail | Describe all amenities need to repair for one repair request |
| Repair Status | Describe status of repair request. One repair request has only one status. |
| Request Amenity | Describe all amenities in one request office |
| Request Office | Describe all requests office from customer |
| Role | Describe role of account. One account has only one role. |

#### Table Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Account | Username{PK} | Unique identifier of an account, username for login to system | nvarchar(50) | No |
| Password | Password for login to system | varchar(50) | No |
| Email | Email of this account | varchar(50) | No |
| RoleId{FK} | Role of this account. Foreign key references to table Role | int | No |
| StatusId{FK} | Status of this account. Foreign key references to table Account Status | int | No |
| Unique Key: Email | | | | |

#### Table Account Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Account Status | Id{PK} | Unique identifier of an account status, automatically increase | int | No |
| Name | Name of this status | nvarchar(50) | No |
| Description | Description for this status | nvarchar(50) | Yes |

#### Table Amenity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Amenity | Id{PK} | Unique identifier of an amenity, automatically increase | int | No |
| Name | Name of this amenity | nvarchar(50) | No |
| Description | Description for this amenity | nvarchar(250) | Yes |
| Weight | Weight of amenity in one group | int | Yes |
| AmenityGroupId{FK} | Group of this amenity. Foreign key references to table Amenity Group | int | Yes |
| Priority | Priority of amenity in one group | int | Yes |
| Unique Key: Weight | | | | |

#### Table Amenity Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Amenity Group | Id{PK} | Unique identifier of an amenity group, automatically increase | int | No |
| Name | Name of amenity group | nvarchar(50) | No |
| Description | Description for amenity group | nvarchar(250) | Yes |

#### Table Appointment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Appointment | Id{PK} | Unique identifier of an appointment, automatically increase | int | No |
| CustomerUsername{FK} | Username of customer. Foreign key references to table Account | nvarchar(50) | No |
| AssignedStaff{FK} | Staff who is responsibility for the appointment. Foreign key references to table Account | nvarchar(250) | Yes |
| OfficeId{FK} | Office where customer want to make appointment. Foreign key references to table Office. | int | No |
| Time | Time for appointment | datetime | No |
| StatusId{FK} | Status of the appointment. Foreign key references to table appointment status. | int | No |
| CreateTime | Time when customer create appointment | datetime | No |
| UpdateTime | Time when manager update the appointment | datetime | Yes |
| Comment | Comment for the appointment | nvarchar(250) | Yes |

#### Table Appointment Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Appointment Status | Id{PK} | Unique identifier of an appointment status, automatically increase | int | No |
| Name | Name of appointment status | nvarchar(50) | No |
| Description | Description for appointment status | nvarchar(50) | Yes |

#### Table Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Category | Id{PK} | Unique identifier of a category for office, automatically increase | int | No |
| Name | Name of category | nvarchar(50) | No |
| Description | Description for category | nvarchar(250) | No |
| Unique Key: Name | | | | |

#### Table Contract

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Contract | Id{PK} | Unique identifier of a contract, automatically increase | int | No |
| CustomerUsername{FK} | Username of customer. Foreign key references to table Account | nvarchar(50) | No |
| OfficeId{FK} | Office where customer want to make contract. Foreign key references to table Office. | int | No |
| StartDate | Date when contract starts. | date | No |
| EndDate | Date when contract ends. | date | No |
| PaymentFee | Fee for the contract | int | No |
| PaymentTerm{FK} | Payment term for the contract. Foreign key references to table Payment Term. | int | No |
| StatusId{FK} | Status of the contract. Foreign key references to table contract status. | int |  |
| Comment | Comment for the appointment | nvarchar(250) | Yes |

#### Table Contract Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Contract Status | Id{PK} | Unique identifier of a contract status, automatically increase | int | No |
| Name | Name of contract status | nvarchar(50) | No |
| Description | Description for contract status | nvarchar(50) | Yes |

#### Table Email Queue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Email Queue | Id{PK} | Unique identifier of an email queue, automatically increase | int | No |
| Username{FK} | Username of the customer. Foreign key references to table Account. | nvarchar(50) | No |
| OfficeIds | List office that system suggests to customer | nvarchar(50) | Yes |
| CreateTime | Time when customer create request | datetime | Yes |

#### Table Office

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Office | Id{PK} | Unique identifier of an office, automatically increase | int | No |
| Name | Name of the office. | nvarchar(50) | No |
| CategoryId{FK} | Category of the office. Foreign key references to table Category. | int | No |
| StatusId | Status of the office. Foreign key references to table Office Status. | int | No |
| Description | Description for the office | ntext | No |
| Price | Price of the office | bigint | Yes |
| PriceTerm{FK} | Price term of the office. Foreign key references to table Price Term. | int | No |
| FloorNumber | Number of floor in the office | int | No |
| Area | Area of the office | float | No |
| ImageUrls | Link of the images of office | text | Yes |
| OtherDetail | Some detail of the office | ntext | Yes |
| Address | Address of the office | nvarchar(150) | No |
| Latitude | Latitude of the office | float | Yes |
| Longitude | Longitude of the office | float | Yes |
| District | District of the office | nvarchar(50) | Yes |
| City | City of the office | nvarchar(50) | Yes |
| ParentOfficeId{FK} | Parent Office ID of the office. Foreign key references to table Office. | int | Yes |
| CreateDate | Date when office create in database | datetime | No |
| ViewCount | Number views when customer see detail of office | int | Yes |

#### Table Office Amenity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Office Amenity | Id{PK} | Unique identifier of a contract status, automatically increase | int | No |
| OfficeId{FK} | Id of office. Foreign key references to table Office. | int | No |
| AmenityId{FK} | Amenity of this office Foreign key references to table Amenity. | int | No |

#### Table Office Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Office Group | OfficeId{PK, FK} | Id of office. Foreign key references to table Office. | int | No |
| OfficeGroup | Group after system run algorithm of this office | int | No |

#### Table Office Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Office Status | Id{PK} | Unique identifier of an office status, automatically increase | int | No |
| Name | Name of office status | nvarchar(50) | No |
| Description | Description for office status | nvarchar(250) | Yes |

#### Table Payment Term

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Payment Term | Id{PK} | Unique identifier of a payment term, automatically increase | int | No |
| Name | Name of term for payment | nvarchar(50) | No |
| PaymentTime | Number months of the payment | int | No |
| Description | Description for payment term | nvarchar(100) | Yes |
| Unique Key: PaymentTime | | | | |

#### Table Price Term

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Price Term | Id{PK} | Unique identifier of a price term, automatically increase | int | No |
| Name | Name of term for price | nvarchar(50) | No |
| Description | Description for price term | nvarchar(250) | Yes |
| Unique Key: Name | | | | |

#### Table Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Profile | Username{PK, FK} | Unique identifier of an account. Foreign key references to table Account. | nvarchar(50) | No |
| Title | Gender of the profile | char(4) | No |
| FullName | Full name of this profile | nvarchar(50) | No |
| Company | Company of this profile | nvarchar(100) | Yes |
| Phone | Phone of this profile | varchar(15) | No |
| Address | Address of this profile | nvarchar(150) | Yes |
| Birthday | Birthday of this profile | datetime | Yes |

#### Table Rental

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Rental | Id{PK} | Unique identifier of a rental request, automatically increase | int | No |
| ContractId{FK} | Contract which customer send rental request. Foreign key references to table Contract. | int | No |
| AssignStaff{FK} | Username of staff who responsibility for this rental request. Foreign key references to table Account. | nvarchar(50) | Yes |
| StatusId{FK} | Status of this rental request. Foreign key references to table Rental Status. | int | No |
| Description | Description of rental | ntext | Yes |
| CreateTime | Time when customer create this rental request | datetime | No |
| UpdateTime | Time when manger or staff change status | datetime | Yes |
| AssignedTime | Time when manager assign to staff | datetime | Yes |

#### Table Rental Detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Rental Detail | Id{PK} | Unique identifier of a rental detail, automatically increase | int | No |
| RentalItemId{FK} | Item rental for the rental request. Foreign key references to table Rental Item. | int | No |
| RentalId{FK} | Id of Request rental. Foreign key references to table Rental. | int | No |
| UnitPrice | Price of one item | float | Yes |
| Quantity | Quantity of rental item for the rental request | int | Yes |

#### Table Rental Item

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Rental Item | Id{PK} | Unique identifier of a rental item, automatically increase | int | No |
| Name | Name of the item | nvarchar(150) | No |
| Description | Description for the item | ntext | Yes |
| Price | Price of the item | float | Yes |
| Quantity | Quantity of the item | int | Yes |
| ImageUrl | Link image of the item | nvarchar(250) | Yes |

#### Table Rental Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Rental Status | Id{PK} | Unique identifier of a rental status, automatically increase | int | No |
| Name | Name of rental status | nvarchar(50) | No |
| Description | Description for rental status | nvarchar(50) | Yes |

#### Table Repair

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Repair | Id{PK} | Unique identifier of a repair request, automatically increase | int | No |
| ContractId{FK} | Contract which customer send repair request. Foreign key references to table Contract. | int | No |
| AssignStaff{FK} | Username of staff who responsibility for this repair request. Foreign key references to table Account. | nvarchar(50) | Yes |
| RepairStatusId{FK} | Status of this repair request. Foreign key references to table Repair Status. | int | No |
| Description | Description of repair | ntext | Yes |
| CreateTime | Time when customer create this repair request | datetime | No |
| UpdateTime | Time when manger or staff change status | datetime | Yes |
| AssignedTime | Time when manager assign to staff | datetime | Yes |

#### Table Repair Detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Repair Detail | Id{PK} | Unique identifier of a repair detail, automatically increase | int | No |
| RepairId {FK} | Id of repair request. Foreign key references to table Repair. | int | No |
| AmenityId{FK} | Amenity of this repair request. Foreign key references to table Amenity. | int | No |

#### Table Repair Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Repair Status | Id{PK} | Unique identifier of a repair status, automatically increase | int | No |
| Name | Name of repair status | nvarchar(50) | No |
| Description | Description for repair status | nvarchar(50) | Yes |

#### Table Request Amenity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Request Amenity | Id{PK} | Unique identifier of a request amenity, automatically increase | int | No |
| RequestOfficeId{FK} | Id of request office. Foreign key references to table Request Office. | int | No |
| AmenityId{FK} | Amenity of this request. Foreign key references to table Amenity. | int | No |

#### Table Request Office

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Request Office | Id{PK} | Unique identifier of a request office, automatically increase | int | No |
| CustomerUsername{FK} | Username of customer. Foreign key references to table Account | nvarchar(50) | No |
| CategoryId{FK} | Category of the office. Foreign key references to table Category. | int | Yes |
| Price | Price of the office. | int | Yes |
| Area | Area of the office | float | Yes |
| District | District of the office | nvarchar(50) | Yes |
| CreateDate | Time when customer create this request. | datetime | No |
| ScheduleDate | Date when system check this request | datetime | Yes |
| OfficeSuggested | List offices that are suitable with request | nvarchar(250) | Yes |

#### Table Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Type** | **Null** |
| Role | Id{PK} | Unique identifier of a role, automatically increase | int | No |
| RoleName | Name of role | nvarchar(50) | No |
| Unique Key: RoleName | | | | |

## Algorithms

### Clustering data

#### Definition

Given the offices which have different location and price range. Place those offices into different groups that have similarity

#### Define Problem

* Office in near location may have similarity in price range. But the location may vary and hard to define all location.
* Group the office by location and price range will make the user easier to find the similarity offices

#### Solution

To solve the problem, we use the well-known K-Means algorithm.

The step to perform the algorithm:

1. Defined the k number (number of cluster)
   * The k number is calculate by the square root of total of data (n). This make sure that there is group, and each group has about data
2. Normalize the data into the n-dimensions Euclidean space, all data based on [0, 1] distance. Each of office data is called “point”
   * In this problem, we use 3 statics: latitude, longitude and price range of the office for 3-dimensions
   * The price range is divided into 5 range: Call as 0, 100.000 VND – 200.000 VND as 1, etc., larger than 500.000 VND as 5
   * In each dimension, calculate the min and max. Then the normalize dimension data is calculate by:

normalize: dimension normalized data

data: the raw data in dimension

1. Select random point into k group, each group has at least 1 point
2. Calculate the mean points of each group
3. Calculate the distance of each data point to each group’s mean point, and change the group of data point to the group that has the nearest mean
   * The distance between A(a1, a2, … an) and X(x1, x2, … xn) is calculate by Euclidean distance:

Where n is the number of dimension

1. Repeat step 4 until the group in step 5 is unchanged

#### Complexity

* In total, the complexity of this algorithm is NP-hard. But with current algorithm maximum complexity limit to n\*10 (n is total of data)

#### Flow chart

Calculate K

Normalize data

Group data in K groups

Calculate the mean of each group

Group based on minimum distance to mean

Group changed?

End

Start

Yes

No

### Matching data

#### Definition

Given the search office criteria, take out the result group (as calculate above) that have the most similarity with the criteria.

#### Define Problem

* The search office may be different from all of the office in system, and user want to make their decision with vary of result.

#### Solution

With the clustering data using K-Means, the best solution is KNN (K-Nearest-Neighbor) algorithm.

The steps to perform the algorithm:

1. Define k number. In this solution, we choose k at least by haft of the average office in each group of K-Means algorithm. Particular 2\* .
2. Normalize the query point (input data) to the same dimension of sample data (by above function)
3. Calculate the distance between query point and each of data in sample (all office) by Euclidean distance (as above).
4. Sort the calculated distance, select the minimum k distances for comparison.
5. The group of query point will be group that have the most frequent existence in above k distances. If there is more than 1 group that have largest existence, the selected group will be the group that have minimum total distance (in the k selected distances only)

#### Complexity

* In total, the complexity of this algorithm is O(n)