**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Insurance Card**

|  |  |
| --- | --- |
| **Group 2** | |
| **Group members** | Đinh Quang Trung – SE60994  Nguyễn Hữu Phúc – SE60749  Phùng Quang Minh Trí – SE60746  Nguyễn Chí Kha – 60351 |
| **Supervisor** | Kiều Trọng Khánh |
| **Ext. Supervisor** | N/A |
| **Capstone Project Code** | MIC |

- Ho Chi Minh City, 12 May 2015 -

This page is intentionally left blank

Table of Contents

[A. Introduction 7](#_Toc419661993)

[1. Project Information 7](#_Toc419661994)

[2. Introduction 7](#_Toc419661995)

[3. Current Situation 7](#_Toc419661996)

[4. Problem Definition 7](#_Toc419661997)

[5. Proposed Solution 8](#_Toc419661998)

[5.1. Feature functions 8](#_Toc419661999)

[5.2. Advantages and disadvantages 8](#_Toc419662000)

[6. Functional Requirements 9](#_Toc419662001)

[7. Roles and Responsibility 10](#_Toc419662002)

[B. Software Project Management Plan 11](#_Toc419662003)

[1. Problem Definition 11](#_Toc419662004)

[1.1. Name of this Capstone Project 11](#_Toc419662005)

[1.2. Problem Abstract 11](#_Toc419662006)

[1.3. Project Overview 11](#_Toc419662007)

[2. Project organization 14](#_Toc419662008)

[2.1. Software Process Model 14](#_Toc419662009)

[2.2. Roles and responsibilities 15](#_Toc419662010)

[2.3. Tools and Techniques 16](#_Toc419662011)

[3. Project Management Plan 16](#_Toc419662012)

[3.1. Software development life cycle 16](#_Toc419662013)

[3.2. Phase Detail 18](#_Toc419662014)

[3.3. Task sheet 19](#_Toc419662015)

[3.4. All Meeting Minutes 19](#_Toc419662016)

[4. Coding Convention 19](#_Toc419662017)

[C. Software Requirement Specification 20](#_Toc419662018)

List of Tables

[Table 1: Definitions, Acronyms, and Abbreviations 6](#_Toc419662019)

[Table 2 Roles and Responsibility 10](#_Toc419662020)

[Table 3 Hardware requirement for continuous integrating server 13](#_Toc419662021)

[Table 4 Hardware requirement for web development 14](#_Toc419662022)

[Table 5 Hardware requirement for mobile development 14](#_Toc419662023)

[Table 6 Software requirement 14](#_Toc419662024)

[Table 7 Roles and responsibilities 16](#_Toc419662025)

[Table 8 Tools and Techniques 16](#_Toc419662026)

List of Figures

[Figure 1 Waterfall model 15](#_Toc420401184)

[Figure 2 Web Application Overview Use Case 22](#_Toc420401185)

Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Name | Definition |
| MIC | Motor Insurance Card |
| NFC | Near field communication |
|  |  |

Table 1: Definitions, Acronyms, and Abbreviations

# Introduction

## Project Information

* Project name: **Insurance Card**
* Project Code: **MIC**
* Product Type: **Website & Android Application**
* Start Date: **May 11th, 2015**
* End Date: **September 5­th,2015**

## Introduction

In this document we introduce a solution for motorbike insurance company. Current insurance company systems have some problems like delayed in renew contracts for customer or inconvenient in checking insurance card validation process. Based on our researches and analysis, we proposed a solution for insurance company in Vietnam and other developed countries.

We build a system which help the insurance companies to solve current problems. In the process of analysis we believe the NFC cards is capable to resolve the problem by using NFC card to save information about insurance contract. NFC cards are convenient to manage the contract information and checking, validating process. Beside of that we also provide an information system to manage NFC cards so that insurance companies will manage the contracts easier.

This document also describes our working process in 4 months includes our perspective in the system, component designs and detailed core workflows. We hope the system and our solution will help resolve the problems from insurance companies in Vietnam and other developed countries.

## Current Situation

When participating in traffic, vehicle owners is required to have compulsory insurance. Therefore, vehicle owners buy insurance from insurance companies or its agents. They pay insurance premium by cash or in online website and receive an insurance certificate with a term of one year, the term can be shorter in some specific situation. When their insurance out of date, they must buy a new insurance, old certificate will be useless. Traffic police will read insurance certificate to check traffic participants.

## Problem Definition

Below are disadvantages of current situation:

* **Forget insurance’s expired date**: Vehicle owners usually keeps their insurance certificate in wallet or somewhere on their vehicle. However, except in cases of necessity, people are not often check their insurance so they could forget its expired date. An expired insurance is not good while it be revealed by traffic officers and could get worse in case of traffic accident.
* **Hard for traffic officers to check and verify insurance**: Traffic officers must read insurance certificate to check and verify vehicle owner’s information. It can be difficult and hinder their work in some cases as at dark or handwriting illegible on insurance certificate.
* **Insurance certificate made of paper:** It could be torn, wet, smudged and especially is counterfeited.
* **Difficult to track and manage number of traffic violations and collisions:** In current scenario, insurance companies almost impossible knows vehicle owner’s history to adjust their insurance policy.

## Proposed Solution

The proposed solution is to build an insurance NFC card system, include a web application and 2 mobile applications with following functions:

### Feature functions

* Web application:
  + Register insurance: user can register a new insurance card with on website using online payment. A staff will contact the user to create contract and sends an insurance NFC card to him/her. If users already have a NFC card, they can use the website to renew current contract.
  + Check card information: user can login into the website and check for their card’s information.
  + Request compensation: user can fill data into the sample fields and sends compensation request to the company.
  + Make/manage contracts: staff can make and manage contracts.
  + Resolve compensation: staff can receive and resolve compensation requests.
  + Notify contract state: system will sends an email to notify the insured one when their insurance is expired.
  + Notify compensation state: system will sends an email to info the insured one when their compensation were accepted or rejected.
* Insurance card printer (mobile app):
  + Simulating NFC card printer: staff can print NFC card.
* Insurance card checker (mobile app):
  + Check card: traffic police and Police Department can check specified motor insurance card expired or not.
  + Update the punishment of violator: traffic police and Police Department can update the punishment of violator to the card information.

### Advantages and disadvantages

* Advantages:
  + The interaction between the insured one and the insurance company: the insured one and the company now are easier to communicate through the website when each person has an account.
  + Reduce risk of insurance card made of paper: the NFC insurance card will not be torn, wet or smudged. And it is difficult to be counterfeit than insurance card made of paper.
  + Support police to check valid insurance card easier.
* Disadvantages:
  + At the present time, not consistent with the law of Vietnam about insurance card issues.
  + Checking the valid of card can take a long time when the internet is slow.

## Functional Requirements

Function requirements of the system are listed as below:

* **User component:**
  + New contract request
  + Check card information.
  + Renew contract.
  + Request compensation.
  + Lost card request
  + Cancel contract
* **Staff component**
  + Create new contracts
  + Manage contracts.
  + Resolve compensation requests.
  + Resolve lost card request
* **System component** 
  + Manage contract states
* **Payment system**
  + Process payments
* **Notify component**
  + Notify contract expiration.
  + Notify compensation states (approved / rejected).
* **Checker mobile application**
  + Check card validation.
  + Update punishment information.
  + Retrieve card information.
* **Printer mobile application**
  + Get contract information from server.
  + Print NFC insurance card.

## Roles and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | khanhkt@fpt.edu.vn |
| 2 | Đinh Quang Trung | Developer | Leader | trungdqse60994@fpt.edu.vn |
| 3 | Nguyễn Hữu Phúc | Developer | Member | phucnhse60749@fpt.edu.vn |
| 4 | Phùng Quang Minh Trí | Developer | Member | tripqmse60746@fpt.edu.vn |
| 5 | Nguyễn Chí Kha | Developer | Member | khanc60351@fpt.edu.vn |

Table 2 Roles and Responsibility

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* **Official name**: Insurance Card
* **Vietnamese name**: Thẻ bảo hiểm
* **Abbreviation**: MIC

### Problem Abstract

As current in Viet Nam customer use Motor Insurance Certificate Paper when they get problems with their motor. Using the Motor Insurance Certificate Paper is inconvenient, for example it can be wet or to insert or update the information in to insurance certificate paper is complicate. So we use the NFC card we call it is insurance card to handle it. The NFC card is supplied by insurance company when the customer buy insurance. The card contain the information of customer, if the customer join with many insurance service they just use only one card.

We provide a software to check the validation of card, the deadline of card and some insurance services that customer joined. We also provide some other advantage that can help save time and costs in some process of company. For example, the software can automatic extend the insurance service, update the information about accidents of motor. In addition we also provide a system software to manage the information of customer via some insurance card we bought, this software will deploy at insurance company.

### Project Overview

#### Current Situation

Below is the problems encountered in this project:

* Security: currently, there is few possible problems encountered with NFC tags as NFC tags can be counterfeited, attacked during data transmission caused data loss, data corruption.
* Server crash: all the needed data is stored in the server. So if server crash, all the devices cannot get card information.
* Absence of team members: team members can get sick or unexpected problems.
* At the present time, not consistent with the law of Vietnam about insurance card issues.

#### The Proposed System

According to the technology researches, we found out that the NFC technology is very capable of resolve the current situations in insurance companies. We can use a feature of NFC tag to resolve the security problem from NFC card. The basic idea is to use a NFC tag (or NFC “card”) which contains a unique card ID as an insurance card instead of paper card currently.

We also build a high available webserver to maintain the main system to work 24/7 to make sure that if mobile applications need access to the information there will be always available.

We assign responsibility in vertical to make sure if any member in this problem cannot continue to work in our team there will be the least harmful to the project processes.

To resolve problem from Vietnam laws of insurance for motorbike, we support the insurance companies to propose new law sections about using technology devices to work with insurance certificate paper to make our system work legally in current situation.

Our system includes three main subsystems: an online website for company’s staffs, a mobile application for police officers and a mobile application to simulate the card printer.

##### Website

Website is a common communication portal for insurance company’s staffs and users (customers). Website provide following features:

* For users (customers):
  + Users can register new insurance card with online payment.
  + Users can look up information about their insurance card: compensation history, punishment history, expired date…
  + Users can renew current insurance contract with online payment.
  + Users can request compensations to insurance company when an accident occurs.
  + Users will be notified by emails when insurance card is nearly expired or a compensation request is approved/rejected.
* For staffs:
  + Staffs can create new contract for customer.
  + Staffs can manage contracts, see all insurance cards published and see statistics
  + Staffs can update compensation requests, resolve a compensation request when the case is done.

Beside above, website system also provides an API interface for two mobile applications to retrieve, update data from mobile applications.

##### Checker Mobile Application

This mobile application is used by traffic officer. This application do followings:

* Check if an insurance card (NFC card) is valid or not.
* Send punishment if the customer has law violations. Punishment information will be updated in server.

##### Printer Mobile Application

This is a simulating application to simulate the work of Card Printer. In reality the company who deploy this system need to have a NFC Card Printer to write information about the insurance company and customer information into an NFC card. However our system currently only support this as a simulating application. This application is used by company’s staffs and do followings:

* Retrieves insurance contract information and write data to a physical NFC card.

#### Boundaries of the System

This section suppose that the government law in local area supports the method of using NFC cards as insurance cards, and accept NFC insurance cards are legal.

* Every company who has Information System infrastructure can deploy this system.
* Companies who deployed this system has to equip enough devices for the system to run, includes:
  + Computer system with internet connection.
  + Smartphone devices with built-in NFC technology.
* The language of this system is Vietnamese
* The complete product includes:
  + Website application for staffs and users
  + Printer mobile application for staffs.
  + Checker mobile application for traffic police officers

#### Development Environment

##### Hardware requirement

* For continuous integrating server:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512Kbps | 8 Mbps |
| Operating System | Ubuntu Server 12 LTS | Ubuntu Server 14.04.2 LTS |
| Computer Processor | Intel® Pentium II | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 128MB of RAM | 3GB of RAM or more |

Table 3 Hardware requirement for continuous integrating server

* For web development:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512Kbps | 8 Mbps |
| Operating System | Windows Vista, 7, 8 | Windows 7, 8 |
| Computer Processor | 1 GHz | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 1GB of RAM | 3GB of RAM or more |

Table 4 Hardware requirement for web development

* For mobile development:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512Kbps | Wi-Fi Connection 12MB |
| Operating System | Android 4.0 | Android 4.0 |
| Hardware | NFC supported | NFC supported |
| Memory | 128MB of RAM | 1GB of RAM or more |

Table 5 Hardware requirement for mobile development

##### Software requirement

|  |  |
| --- | --- |
| Software | Name / Version |
| Operating system | Windows 7 or above |
| Environment | Java EE 6 |
| Modeling tool | Microsoft Visio 2013 |
| IDE | Netbeans 7.2.1, Intellij IDEA 14.1 |
| DBMS | MySQL 5.6 |
| Source control | TortoiseSVN 1.8.11 |
| Web browser | Chrome 42 or above |

Table 6 Software requirement

## Project organization

### Software Process Model

This project is developed under waterfall model, we applies waterfall model with customizes to capable with current situation in our team. We choose this model because the following reasons:

* This project is 4 months long due to the FPT University Capstone Project timeline, which can be consider a short project.
* Based on researches and clarify Vietnam laws of insurance for motorbike and current system in insurance companies, the requirements of this project are stable, clear, fixed and well understood by all team members.
* This project use NFC technology, which we have strong background knowledge and well practice skills. We also have experience in designing, building web and mobile application system.



Figure 1: Waterfall model

Reference: SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

We customize the waterfall model from the reference to make the process more capable with current situation of our team.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Kiều Trọng Khánh | Supervisor / Project Manager | - Clarify user requirement.  - Technical support and business analysis.  - Tracking development process.  - Review document and product. |
| 2 | Đinh Quang Trung | Team leader, BA, Developer, Tester | - Tracking process.  - Planning project, distribute tasks.  - Requirement analysis.  - Database design.  - Documentation.  - GUI Design.  - Coding.  - Testing.  - Deploy product. |
| 3 | Nguyễn Hữu Phúc | BA, Developer, Tester | - Requirement analysis.  - Database design.  - Documentation.  - GUI Design.  - Coding.  - Testing. |
| 4 | Phùng Quang Minh Trí | BA, Developer, Tester | - Requirement analysis.  - Database design.  - Documentation.  - GUI Design.  - Coding.  - Testing. |
| 5 | Nguyễn Chí Kha | BA, Developer, Tester | - Requirement analysis.  - Database design.  - Documentation.  - GUI Design.  - Coding.  - Testing. |

Table 7 Roles and responsibilities

### Tools and Techniques

|  |  |
| --- | --- |
| Tool / Technique | Name / version |
| Frontend | HTML, CSS, JavaScript, jQuery, Bootstrap |
| Backend | JavaEE, Servlet, JSP, Hibernate |
| Web server | Apache Tomcat 7 |
| Development tool | Netbeans 7.2.1, Intellij IDEA 14 |
| DBMS | MySQL 5.6 |
| Source control | TortoiseSVN 1.8.11 |
| Modeling tool | Microsoft Visio 2013 |
| Document tool | Microsoft Word 2013 |

Table 8 Tools and Techniques

## Project Management Plan

### Software development life cycle

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Description | Deliverables | Resource needed | Dependencies and Constrains | Risk |
| Requirements Definition | Identify and clarify system requirements. | Report No.1 Introduction. | 20 man-days | N/A | - Missing requirement.  - Project’s scope can be unclear.  - Lack of member share and understand. |
| System and Software Design | - Identify hardware and software requirements.  - Decide software architect and clarify software detail design.  - Design database. | Report No.2 Software Project Management Plan, Report No. 3 Software Requirement Specification and  Report No. 4 Software Design Description. | 50 man-days | Depend on Requirements Definition. | - Misunderstood or unclear system’s requirement.  - Lack of practical experience leading to unreasonable design. |
| Implementation and Unit Testing | - Implements all functions of system.  - Create test plan.  - Perform Unit testing. | Software package. | 120 man-days | - Base on Software Requirement Specification and Software Design Description.  - Coding try to follow coding convention. | - Member does not performs unit test.  - Lack of practical experience. |
| Integration and System Testing | - Perform integration test and system test. | Report No. 5 System Implementation & Test | 35 man-days | Implementation and Unit Testing are finished. | - Lack of testing experience leading to lack of test cases.  - Not enough time for performing test. |
| Operation and Maintenance | - Deploy the system  - Create the user’s manuals.  - Do routine maintenance activities. | Report No.6 Software User’s Manual | 15 man-days | Integration and System Testing are finished. | User’s manual may be difficult for user to understand and confuse. |

### Phase Detail

#### Phase 1: Requirements Definition

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Identify and clarify system requirements. | Research current systems to collect requirements.  Define main and needed functions the system must include. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |

#### Phase 2: System and Software Design

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Identify hardware and software requirements. | Find out the suitable hardware and software for the system, as well as its minimum and recommended requirements. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |
| Decide software architect and clarify software detail design. | - Define the major software components and interfaces.  - Draw core flow diagram, use case diagram, prototype …  - Group meeting to review and modify. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |
| Design database. | - Design database for the system. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |

#### Phase 3: Implementation and Unit Testing

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Implements all functions of system. | Coding all the components. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |
| Create test plan. | Planning for testing. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |
| Perform Unit testing. | - Write Unit test cases.  - Implement Unit tests. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |

#### Phase 4: Integration and System Testing

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Perform integration test and system test. | - Test groups of modules and test whole the system. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |

#### Phase 5: Operation and Maintenance

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Deploy the system | Deploy the system in client environment. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |
| Create the user’s manuals. | Create a guideline to instruct users using system. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |
| Do routine maintenance activities. | Do routine maintenance activities for client system. | Đinh Quang Trung  Nguyễn Hữu Phúc  Phùng Quang Minh Trí  Nguyễn Chí Kha |

### Task sheet

Refer to “Task sheet” folder.

### All Meeting Minutes

Refer to “Meeting minutes” folder.

## Coding Convention

This project follows “Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999”.

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

We use followings naming convention from the reference to capable with current situation in our team:

* Naming:
  + Class names must be in Pascal case.
  + Variable names must be in Camel case.
  + Each Java class belongs to a single file.
* Intentions:
  + Use 4 spaces intentions.
  + Avoid lines with more than 80 characters
* Declaration:
  + One declaration per line is recommended since it encourages commenting.
  + In absolutely no case should variables and functions be declared on the same line.
  + Do not put different types on the same line.

# Software Requirement Specification

## User Requirement Specification

### Customer requirement

### Staff requirement

### Police requirement

## System Requirement Specification

### External Interface Requirement

### System Overview Use Case

#### Web Application

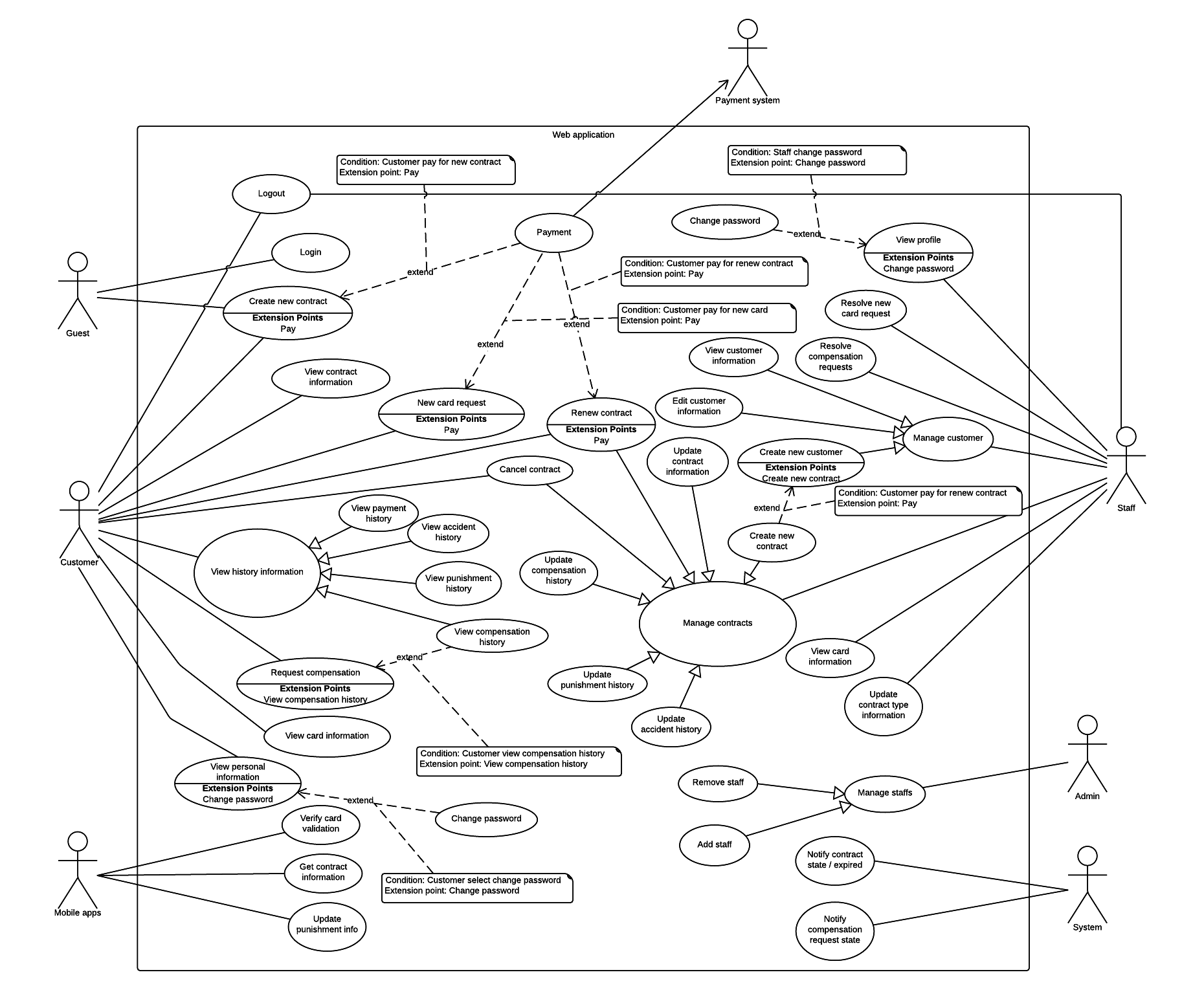


Figure 2: Web Application Overview Use Case

#### Checker Mobile Application

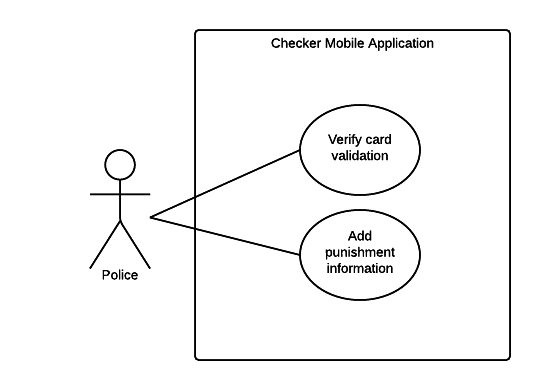


Figure 3: Checker Mobile Application

#### Printer Mobile Application

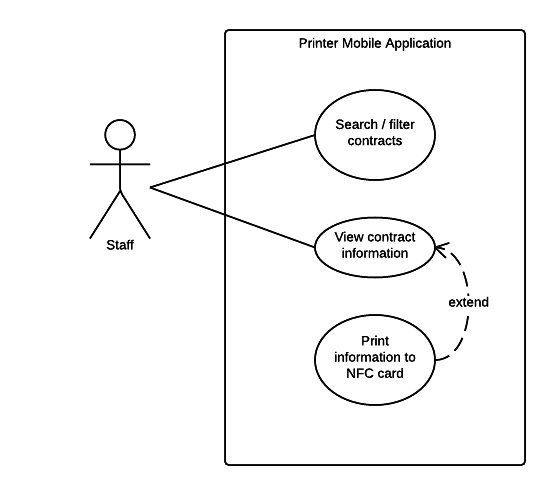


Figure 4: Printer Mobile Application

### List of Use Case

#### Web Application

##### <Guest> Overview Use Case

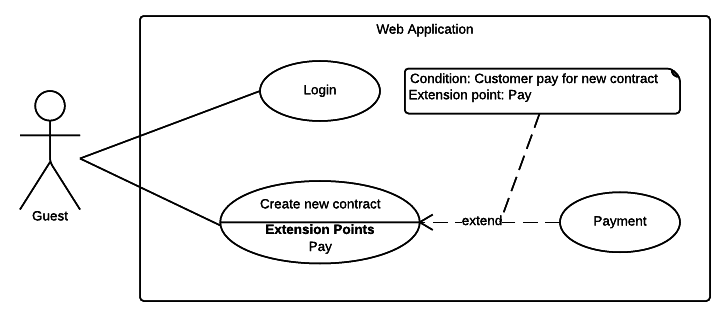


Figure 5: <Guest> Overview Use Case

###### <Guest> Login



Figure 6: <Guest> Login

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WG01 | | | |
| Use Case No. | WG01 | **Use Case Version** | 2.0 |
| Use Case Name | Login | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | Normal |
| Actor:   * Guest   Summary:   * This use case allows guest to log in the system.   Goal:   * Guest can log in the system.   Triggers:   * Guest sends the login command.   Preconditions:   * N/A   Post Conditions:   * Success: Guest login the system. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest goes to login view. | System requires identity information from Guest:   * Email or customer code: free text input, required, length 3 – 250 * Password: free text input, required, length 6 – 32 | | 2 | Guest inputs information. |  | | 3 | Guest sends command to login to system | Guest will login system with their specific role  [Alternative 1]  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest enter wrong identity information. | Wrong identity information, System shows error message. |   Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Missing of required fields | System notify guest to enter their identity information |   Relationships: N/A  Business Rules:   * After login to system, guest will be redirected to specific view based on their role on the system: staff or customer. * If role is “Customer”, the system will display to Customer view. * If role is “Staff”, the system will display to Staff Dashboard view. | | | |

###### <Guest> Create new contract request

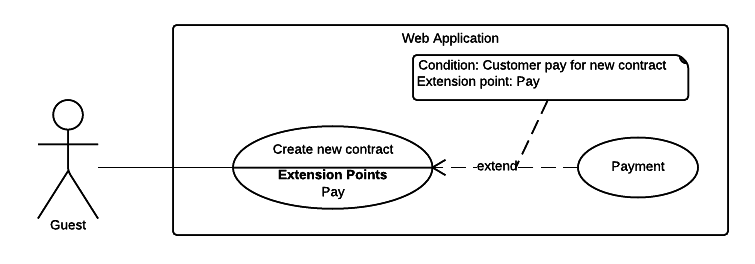


Figure 7: <Guest> Create new contract request

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WG01 | | | |
| Use Case No. | WG02 | **Use Case Version** | 2.0 |
| Use Case Name | Create new contract request | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | Normal |
| Actor:   * Guest   Summary:   * This use case allows guest to create new contract request.   Goal:   * Guest can create new contract request.   Triggers:   * Guest sends command to create contract request.   Preconditions:   * N/A   Post Conditions:   * Success: New contract will be created for guest. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest goes to new contract view. | System requires information from guest:  **Personal information**   * Name: free text input, required, length 3 – 80. * Address: free text input, required, length 3 – 250. * Email: free text input, required, length 3 – 250. * Phone number: free text input, required, length 8 – 15. * Personal ID: free text input, length 8 – 15.   **Contract information** (all information below are required)   * Contract’s type: select one of the options. * Start date: date time input, required. * Expired date: date time input, required. * Contract’s fee: text   **Vehicle information**   * Plate: free text input, required, length 4 – 15. * Brand: free text input, required, length 2 – 20. * Model code: free text input, required, 2 – 20. * Vehicle type: free text input, required, 2 – 20. * Color: free text input, required, length 2 – 20. * Engine: free text input, required, length 2 – 20. * Chassis: free text input, required, length 2 – 20. * Capacity: free text input, required, length 2 – 20. * Year of manufacture: number text input, required, value from 1900 to 2200. * Weight: free text input, required, value from 1 – 1000, unit: kilogram * Seat capacity: free text input, required, value from 1 – 100. * Certificate image: file upload input | | 2 | Guest inputs information. |  | | 3 | Guest sends command to create new contract request. | System validate information, display contract details and request for confirmation.  [Exception 1, 2, 3] | | 4 | Guest sends command to create new contract request. | Display new view let user select one of following payment gateways:  - PayPal payment gateway.  - Direct payment.  And show guest the fee:   * Contract’s fee: text. | | 5 | If user chooses PayPal gateway and sends confirm command.  [Alternative 1] | Forward to PayPal payment view to process the payment. | | 6 | User process the PayPal payment | If payment succeed:  Add new contract information to the system.  Show message created successful.  [Exception 4] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If user chooses direct payment method and sends confirm command. | Show list of company brands address. |   Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to create new contract request | System shows error message to ask user input missing required fields. | | 2 | Guest’s email is existed in the system | Show message to notify guest that their email is existed in the system. | | 3 | Guest’s vehicle plate is existed in the system | Show message to notify guest that their vehicle is existed in the system. | | 4 | If payment failed | Show message to notify user that payment failed and the renew request has been aborted. |   Relationships: Payment  Business Rules:   * New contract will be created in the system with inputted information. * Contract’s type must be 1 of 5 below:   + “Xe trên 50cc có BH cho người trên xe”   + “Xe trên 50cc không có BH cho người trên xe”   + “Xe dưới 50cc có BH cho người trên xe”   + “Xe dưới 50cc không có BH cho người trên xe”   + “Xe mô tô ba bánh, xe gắn máy và các loại xe tương tự” * The initial status of contract will be set to “Pending”. * When customer completed payment process, contract’s status would change from “Pending” to “No Card”. * Staff will receive a notification about new contract request, they verify contract’s information and issue a card for this contract, in this case, contract’s status would change from “No Card” to “Ready”. * If guest upload an image of vehicle registration certificate, they must not input vehicle’s information. * System must ensure has no duplicate customer or vehicle. * An email contains customer code and password will be sent to user, user can use this information to login to the system later. * Expired date and start date must not exceed 1 year. * Contract end date must not be earlier start date. * Contract price (per year) would be calculated from contract type and sets by staff. | | | |

##### <Customer> Overview Use Case

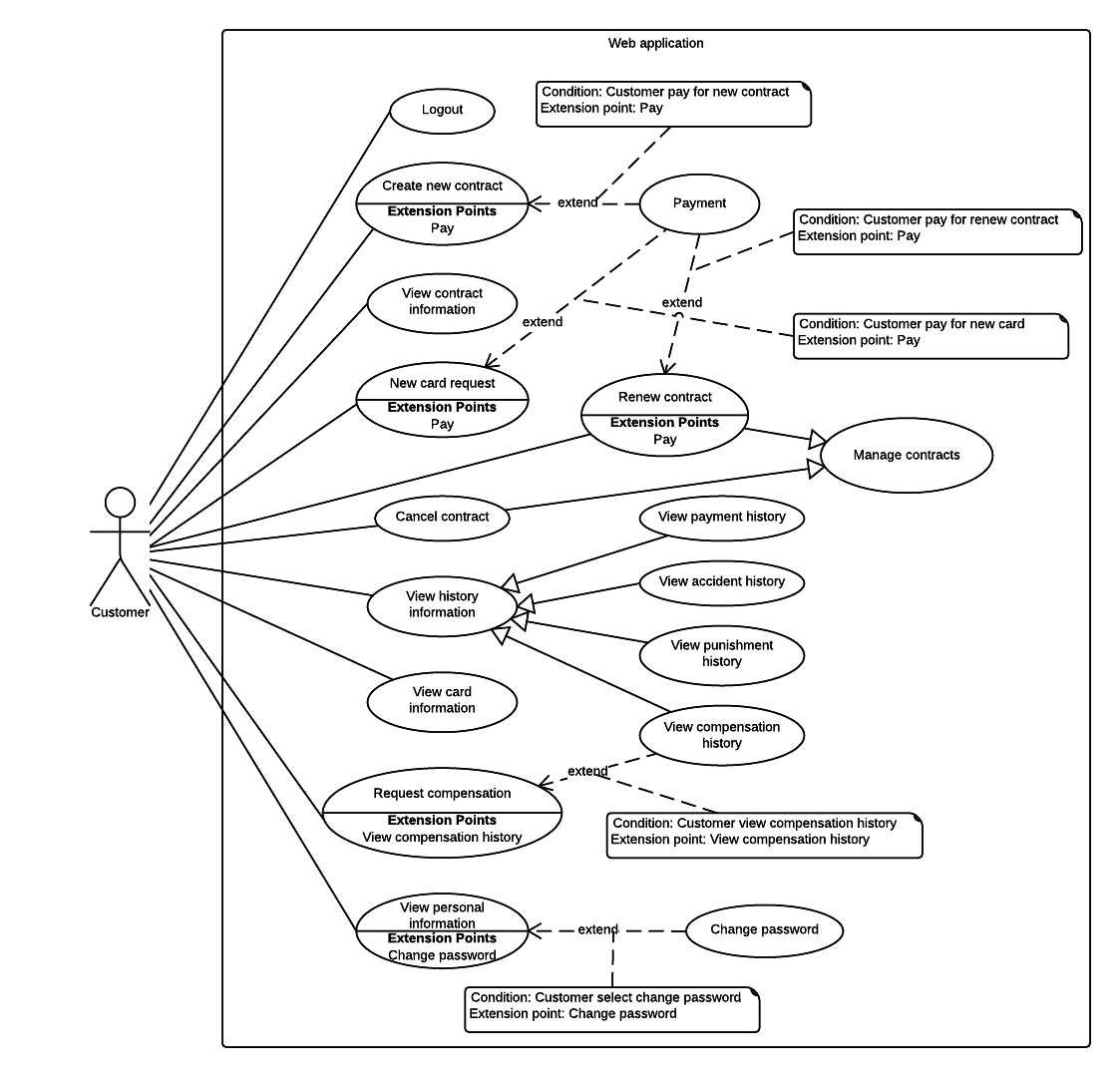


Figure 8: <Customer> Overview Use Case

###### <Customer> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC01 | | | |
| Use Case No. | WC01 | **Use Case Version** | 2.0 |
| Use Case Name | Logout | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer, staff.   Summary:   * This use case allows user to log out the system.   Goal:   * User can log out the system.   Triggers:   * User sends the logout command.   Preconditions:   * User logged in the system.   Post Conditions:   * Success: User logs out the system. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends logout command. | User exits from the system. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * After user logged out, return to the home. | | | |

###### <Customer> Renew contract

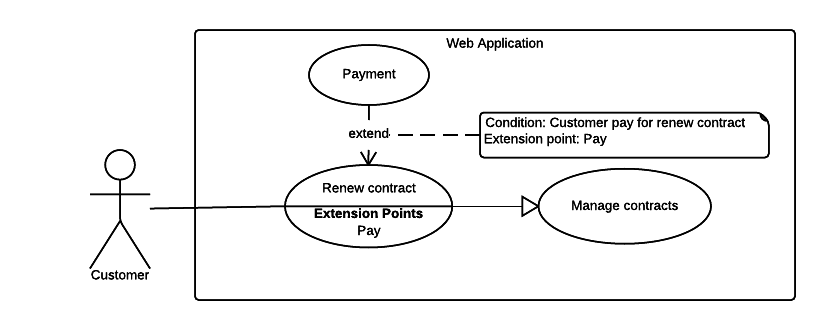


Figure 9: <Customer> Renew contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC02 | | | |
| Use Case No. | WC02 | **Use Case Version** | 2.0 |
| Use Case Name | Renew contract by user | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Customer.   Summary:   * This use case helps user to renew their contract.   Goal:   * User can renew their insurance contract.   Triggers:   * User sends renew contract command.   Preconditions:   * User must login into the system with role Customer. * Contract’s expired date is less than 2 months. * Contract’s status must be “No Card”, “Ready” or “Expired”.   Post Conditions:   * Success: Customer's contract is renewed or a renew contract request would be sent to the Staff. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends renew contract command. | Display new view let user select one of following payment gateways:  - PayPal payment gateway.  - Direct payment.  And show user the fee:   * Renewal fee: text. | | 2 | If user chooses PayPal gateway and sends confirm command.  [Alternative 1] | Forward to PayPal payment view to process the payment. | | 3 | User process the PayPal payment | If payment succeed:  Update information to the system. Renew user insurance contract.  Show message renew successful.  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If user chooses direct payment method and sends the confirm command. | Show list of company brands address. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If payment failed | Show message to notify user that payment failed and the renew request has been aborted. |   Relationships: Extend “Payment”.  Business Rules:   * If contract status is “Expired” and user paid the renew fee through PayPal, system automatically change the contract status to “Ready”. * If user paid the renew fee directly, contract’s status is “Pending” and Staff will update the payment for that contract and change contract status to “Ready”. | | | |

###### <Customer> Cancel contract

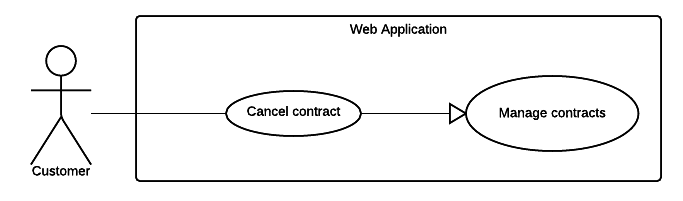


Figure 10: <Customer> Cancel contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC03 | | | |
| Use Case No. | WC03 | **Use Case Version** | 2.0 |
| Use Case Name | Cancel contract | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Customer.   Summary:   * This use case helps user cancel their contract.   Goal:   * Customer can cancel the contract.   Triggers:   * Customer sends cancel contract request.   Preconditions:   * User must login into the system with role Customer. * User’s contract has not expired. * Customer's contract status must not be "Cancelled" or “Request cancel”.   Post Conditions:   * Success: Send to the staff the cancel contract request. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to cancel contract view. | Display new view require user input some information:   * Reason to cancel the contract: can be optional selected from these values:   + “Xe cơ giới bị thu hồi đăng ký”   + “Xe cơ giới hết niên hạn”   + Other reason: free text input, required, length 1-250. * Attachment file upload: file upload input. | | 2 | User chooses the reason why he/she want to cancel contract and attach the related file (if any). |  | | 3 | User sends cancel contract request command. | * Change contract status. * Send request to the Staff.   [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If user didn't check any reason to cancel contract | Show message to notify user that they have to choose the reason for cancel contract. |   Relationships: N/A  Business Rules:   * Cancel contract request will be sent to the system with inputted information. * System update status of the contract from “Pending”, “No Card”, “Expired” or “Ready” to “Request cancel”. | | | |

###### <Customer> View payment history

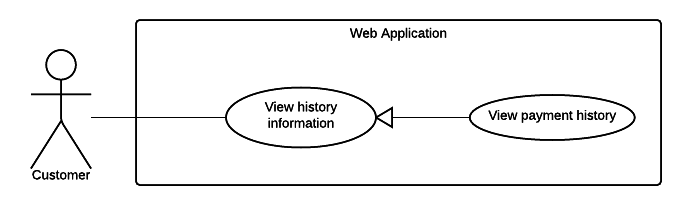


Figure 11: <Customer> View payment history

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC04 | | | |
| Use Case No. | WC04 | **Use Case Version** | 2.0 |
| Use Case Name | View payment history | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case shows user the history of customer’s transactions.   Goal:   * User can view list of transactions were made.   Triggers:   * User sends view payment history command.   Preconditions:   * User must login into the system with role Customer. * There is at least 01 transaction belongs to customer in the system.   Post Conditions:   * Success: Show the customer’s transactions history to the user. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends view payment history command. | Display new view that show user list of transaction history information includes:   * Date of transaction: text * Payment method: text * Service: text * Amount: text * Receiver: text * PayPal transaction ID: text   [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If there are no transaction was made. | Show message to notify that there are no transaction was made. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * The information is loaded from the system. * List of transactions will be sorted by date in descending order. * Pagination must be displayed if number of transactions larger than 10. * Search bar on the top help user finding transactions faster. | | | |

###### <Customer> View accident history

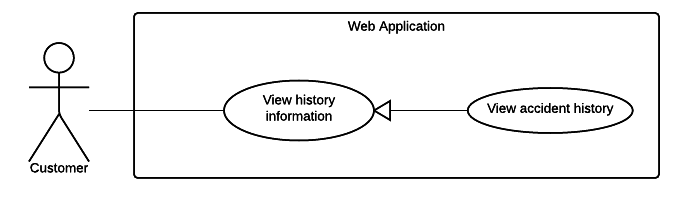


Figure 12: <Customer> View accident history

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC05 | | | |
| Use Case No. | WC05 | **Use Case Version** | 2.0 |
| Use Case Name | View accident history | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case shows user the history of customer’s accidents.   Goal:   * User can view list of their accidents history.   Triggers:   * User sends view accident history command.   Preconditions:   * User must login into the system with role Customer. * There is at least 01 accident belongs to customer in the system.   Post Conditions:   * Success: Show the customer’s accidents history to the user. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends view accident history command | Display new view show list of accidents information includes:   * Date of accident: text * Title: text * Attachment file: link to file.   [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There are no accident. | Show message to notify that there are no accident. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * The information is loaded from the system. * List of accidents will be sorted by date in descending order. * Pagination must be displayed if number of accidents larger than 10. * Search bar on the top help user to search accident by title. | | | |

###### <Customer> View punishment history

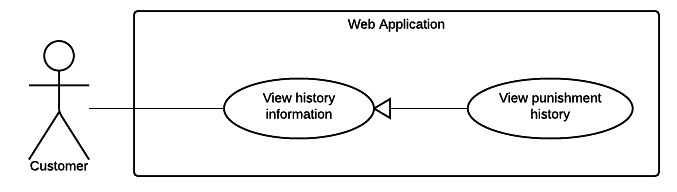


Figure 13: <Customer> View punishment history

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC06 | | | |
| Use Case No. | WC06 | **Use Case Version** | 2.0 |
| Use Case Name | View punishment history | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case shows user the history of the punishment.   Goal:   * User can view list of their punishment history.   Triggers:   * User sends view punishment history command.   Preconditions:   * User must login into the system with role Customer. * There is at least 01 punishment belongs to customer in the system.   Post Conditions:   * Success: Show the punishment history to the user. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends view punishment history command | Display a table shows list of punishments information includes:   * Date of punishment: text * Punishment title: text * Attachment file: link to file.   [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There are no punishment | Show message to notify that there are no punishment. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * The information is loaded from the system. * Punishments list is sorted by date order. * Search bar on the top help user finding punishment information faster. * Pagination must be displayed if number of punishments larger than 10. | | | |

###### <Customer> Request compensation

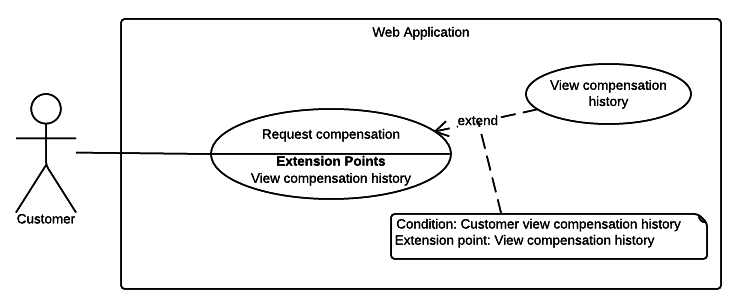


Figure 14: <Customer> Request compensation

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC07 | | | |
| Use Case No. | WC07 | **Use Case Version** | 2.0 |
| Use Case Name | Request compensation | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Customer.   Summary:   * This use case helps user to request compensation.   Goal:   * User can request compensation.   Triggers:   * User sends request compensation command.   Preconditions:   * User must login into the system with role Customer.   Post Conditions:   * Success: Store the compensation request to into the system. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to request compensation view. | Display new view ask user to input required information includes:   * Driver name: free text input, required, length 3-80. * License number: free text input, required, length 10-15. * License type: free text input, required, length 1-10. * Driver phone: free text input, required, length 8-15. * Vehicle capacity: free text input, required, length 1-20. * Driver address: free text input, required, length 3-250. * Plate number of accident motor: free text input, required, length 4-15. * Date of accident: date time input, required. * Place of accident: free text input, required, length 2-20. * Control Police Department: free text input, required, length 3-250. * Description: free text input, required, length 1-2000. * Human damage: free text input, required, length 1-2000. * Asset damage: free text input, required, length 1-2000. * Observer: free text input, required, length 3-80. * Compensation note: free text input, required, length 1-2000. * Attachment file: file upload input. | | 2 | User fill required information and attach the minutes of the accident (if any). |  | | 3 | User sends request compensation command | Show message to notify that request punishment succeed.  [Exception 1, 2] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User input missed one of requirement information. | Show message to notify user what required information is missing. | | 2 | The inputted information length is out of range. | Show message to notify user what information is out of range. |   Relationships: N/A  Business Rules:   * Compensation request will be sent to the system with inputted information. * Only permit user chooses the accident date before or at the request date. * New compensation request status will be set to “Đang xử lý”. * Compensation decision will be set to “Chưa quyết định”. | | | |

###### <Customer> View compensation history

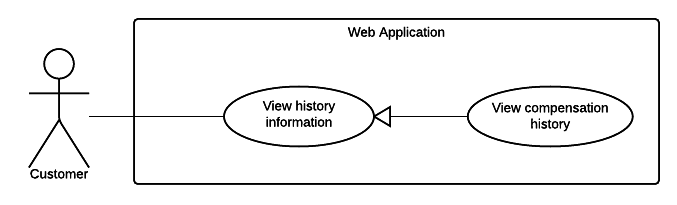


Figure 15: <Customer> View compensation history

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC08 | | | |
| Use Case No. | WC08 | **Use Case Version** | 2.0 |
| Use Case Name | View compensation history. | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case helps user to view compensation history.   Goal:   * User can view compensation history.   Triggers:   * User sends view compensation history command.   Preconditions:   * User must login into the system with role Customer. * There is at least 01 compensation belongs to customer in the system.   Post Conditions:   * Success: Show the compensation history to the user. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends view compensation history command. | Display new view show a table of compensations includes:   * Compensation code: text. * Contract code: text. * Customer name: text. * Request date: text. * Resolve date: text. | | 2 | User selects a compensation to view its details. | Display new view that show the detail of the selected compensation includes:   * Driver name: text. * License number: text. * License type: text. * Driver phone: text. * Vehicle capacity: text. * Driver address: text. * Plate number of accident motor: text. * Date of accident: text. * Place of accident: text. * Control Police Department: text. * Description: text. * Human damage: text. * Asset damage: text. * Observer: text. * Compensation note: text. * Attachment: link to file. * Created date: text. * Resolve date: text. * Decision: text. * Resolve note: text.   [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There are no compensation | Show message to notify that there are no compensation. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * The information is loaded from the system. * List of compensation is sorted by resolve date in decreasing order. The unsolved compensations will be the top rows. * Pagination must be displayed if number of compensations larger than 10. * Search bar on the top help user to finding compensation faster. | | | |

###### <Customer> New card request

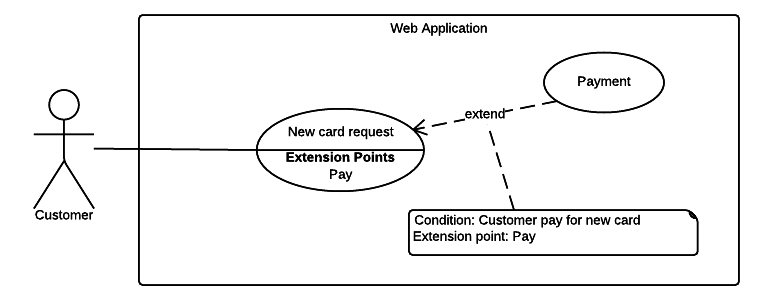


Figure 16: <Customer> New card request

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC09 | | | |
| Use Case No. | WC09 | **Use Case Version** | 2.0 |
| Use Case Name | New card request | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Customer.   Summary:   * This use case helps user to request a new card.   Goal:   * User can request a new card.   Triggers:   * User sends new card request command.   Preconditions:   * User must login into the system with role Customer. * User’s contract has not expired. * User must has a card in the system.   Post Conditions:   * Success: New card request will be sent to the system. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends new card request command. | Display a new view shows user:   * A text box to confirm by password: free text input, required, length 6-32. * Payment gateways: can be optional selected from these selections:   + PayPal payment gateway.   + Direct payment. * The new card fee: text. | | 2 | User enter password and choose the PayPal payment gateway. Then sends confirm command.  [Alternative 1] | Forward to PayPal payment process view.  [Alternative 2] | | 3 | User process the PayPal payment. | If payment succeed,  Show message to notify that the payment is succeed and the card will be sent to customer soon.  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User choose the direct payment method. | Show company information: address, telephone number.  [Alternative 2] | | 2 | If user enter wrong password | Show message to notify that user has entered wrong password. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If payment failed | Show message to notify that the payment failed. The request is aborted. |   Relationships: N/A  Business Rules:   * At a time, user can only has one new card request for each contract. | | | |

###### <Customer> View card information

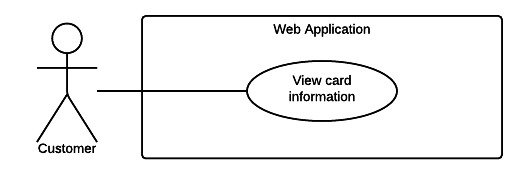


Figure 17: <Customer> View card information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC10 | | | |
| Use Case No. | WC10 | **Use Case Version** | 2.0 |
| Use Case Name | View card information | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case shows user their card information.   Goal:   * User can view their card information.   Triggers:   * User sends view card information command.   Preconditions:   * User must login into the system with role Customer. * There is at least 01 card belongs to customer in the system.   Post Conditions:   * Success: Show the user’s card information and the access history. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends view card information command | Display new view shows the card information includes:   * Card ID: text * User name: text * Card status: text * Activated date: text   And a table shows list of access history information includes:   * Access date: text * Access device: text * Request: text * Response: text |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There are no card | Show message to notify that there are no card. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * The information is loaded from the system. * List of access history is sorted by access date in descending order. * Pagination must be displayed if number of access logs larger than 10. * Filter bar on the top help user to finding access logs faster. | | | |

###### <Customer> View personal information



Figure 18: <Customer> View personal information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC11 | | | |
| Use Case No. | WC11 | **Use Case Version** | 2.0 |
| Use Case Name | View personal information | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case shows user their personal information.   Goal:   * User can review their card information.   Triggers:   * User sends view personal information command.   Preconditions:   * User must login into the system with role Customer.   Post Conditions:   * Success: Show the user’s personal information. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends view personal information command | Display new view shows the user information includes:   * User code: text * User name: text * Address: text * Email: text * Phone number: text * Personal ID: text |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules: N/A   * The information is loaded from the system. | | | |

###### <Customer> Change password



Figure 19: <Customer> Change password

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WC12 | | | |
| Use Case No. | WC12 | **Use Case Version** | 2.0 |
| Use Case Name | Change password. | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Customer.   Summary:   * This use case helps user to change their password.   Goal:   * User can change the password.   Triggers:   * User sends change password command.   Preconditions:   * User must login into the system.   Post Conditions:   * Success: Update new user’s password to the system. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to change password view. | Display new view contains textboxes for user to input their current and new password:   * Current password: free text input, required, length 6-32 * New password: free text input, required, length 6-32 * Confirm new password: free text input, required, length 6-32 | | 2 | User input their current password, new password and confirm new password. |  | | 3 | User sends change password command. | Show message to notify that the password has been changed successfully.  [Alternative 1, 2] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If user inputs the wrong current password | Show message to notify that the current password is wrong. | | 2 | If the new password and the confirm password not matched. | Show message to notify that the confirm password is not matched with the new password. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * The new password will be saved into the system if the provided information is correct. | | | |

##### <Staff> Overview Use Case

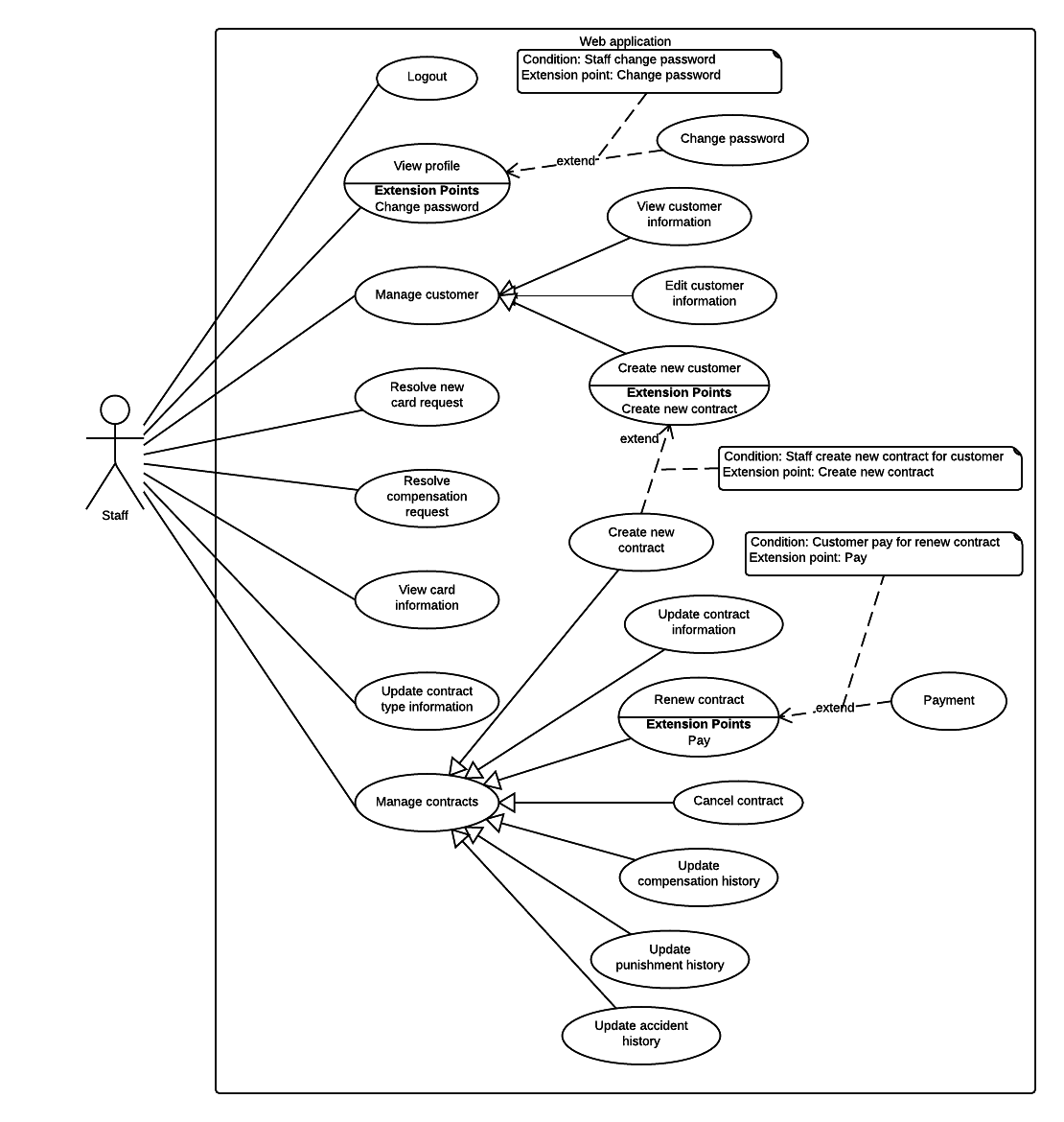


Figure 20: <Staff> Overview Use Case

###### <Staff> View profile



Figure 21: <Staff> View profile

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS01 | | | |
| Use Case No. | WS01 | **Use Case Version** | 2.0 |
| Use Case Name | View profile | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Normal |
| Actor:   * Staff   Summary:   * This use case allows staff view their profile.   Goal:   * Staff can view their detailed profile.   Triggers:   * Staff goes to view their profile.   Preconditions:   * Staff must login into the system with roles Staff.   Post Conditions:   * Success: Staff profile information is shown. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to view theirprofile. | Staff’s detail profile will be shown with following information:   * Staff’s code: text * Staff’s full name: text * Staff’s email: text * Staff’s phone: text |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * Staff’s profile information is always loaded from the system. | | | |

###### <Staff> Change password



Figure 22: <Staff> Change password

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS02 | | | |
| Use Case No. | WS02 | **Use Case Version** | 2.0 |
| Use Case Name | Change password | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Normal |
| Actor:   * Staff   Summary:   * This use case allows staff change their password.   Goal:   * Staff can change their account password.   Triggers:   * Staff sends change password command.   Preconditions:   * Staff must login into the system with roles Staff.   Post Conditions:   * Success: Staff’s new password is updated to the system. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to change password view. | Change password view will be shown with following information:   * Current password: free text input, required, length 6 – 32 * New password: free text input, required, length 6 – 32 * Confirm new password: free text input, required, length 6 – 32 | | 2 | User enters required information |  | | 3 | User sends change password command. | * Validate data   [Alternative 1, 2]  [Exception 1, 2]   * Update staff’s new password into the system. * Display message notify user their password has changed successfully. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Current password does not match with staff’s password in the system | Show message notify staff entered current password does not match with staff’s password in the system | | 2 | Confirm new password does not match with new password | Show message notify staff entered confirm password is not match with their new password |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Missing of required fields | Show message to notify staff input missed fields | | 2 | Length of new password not in range | Show message notify staff’s new password is out of range |   Relationships: N/A  Business Rules:   * In case of successful scenario, staff’s new password would be updated to the system. * The user should be able to know how strong their password is when they input their new password. * Passwords could be rated in one of three ways: “Weak”, “Moderate” and “Strong”. | | | |

###### <Staff> Resolve new card request

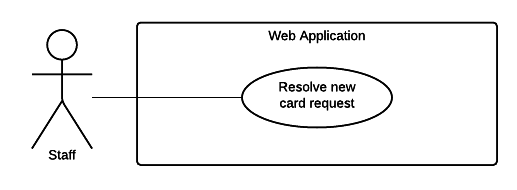


Figure 23: <Staff> Resolve new card request

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS03 | | | |
| Use Case No. | WS03 | **Use Case Version** | 2.0 |
| Use Case Name | Resolve new card request | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allows staff view and solve new card request.   Goal:   * Staff can view and solve request for new card from customers.   Triggers:   * Staff sends view and solve new card request command.   Preconditions:   * User must login into the system with role Staff. * There is at least 01 unsolved request for new card from customer. * The payment process for new card was completed.   Post Conditions:   * Success: A customer’s new card request is solved, a new card issued and the date request solved updated. * Fail: Request is unsolved.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to view all new card request view. | New card request view will be shown with following information:   * Date of request: text * Note of customer: text * ID of lost card: link to card details. * Name of card owner: link to detailed information of the customer who own this card. * Date a new card be issued for this customer: text * ID of new card: link to card details. | | 2 | Staff add new card for this customer | Update new card information to customer’s request and sends notification.   * Date new card issued. * ID of new card |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There are no unsolved new card request | Show message to notify staff there are no unsolved new card request. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * List of new card request is always loaded from the system. * List of new card request is sorted by date of request in descending order. * The lost card will changes status from “Ready” to “Deactivated”. * The new card will has status is “Ready”. * Search bar on the top help user finding card or customer faster. * Pagination must be display if number of requests larger than 10 and auto-change based on staff’s selection. * Staff can select how many requests should be displayed in user view, default is 10 requests. * A solved request must have code of new card and the date this card issued. | | | |

###### <Staff> Resolve compensation request

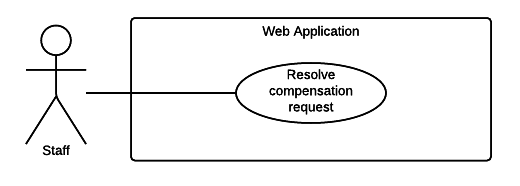


Figure 24: <Staff> Resolve compensation request

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS04 | **Use Case Version** | 2.0 |
| Use Case Name | Resolve compensation request | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Staff.   Summary:   * This use case allows staff view and solves compensation requests.   Goal:   * Staff can view and solve request for compensation from customers.   Triggers:   * Staff sends view and solve compensation request command.   Preconditions:   * User must login into the system with role Staff. * There is at least 01 unsolved request for compensation from customer. * The compensation’s decision must be negotiated with the customer, approved by the insurance company.   Post Conditions:   * Success: A customer’s compensation request solved, decision for this compensation has been made and the date request solved is updated. * Fail: Request is unsolved.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to view list of request for compensation. | List of compensation request will be shown with following information:   * Request’s code: link to request detail * Contract’s code: link to contract detail * Contract’s owner full name: link to customer detail * Get request date: text * Request solved date: text | | 2 | Staff select an unsolved request to view and resolve. | Compensation request detail will be shown with following information:   * Contract’s code: link to contract details. * Contract’s owner full name: link to customer details. * Driver’s full name: text * Driver’s address: text * Driver’s phone number: text * Driver’s license number: text * Driver’s license type: text * Vehicle’s capacity or seat capacity: text * Accident vehicle plate number: text * Time of the accident: text * Place of the accident: text * Police department solved accident: text * Description of the accident: text * Human damaged in the accident: text * Asset damaged in the accident: text * Name of the observer: text * Detail of compensation request from customer: text * Attachment: link to attach files * Note for compensation from staff: free text input. * Decision: select one of the options * Status of compensation: select one of the options. | | 3 | Staff input resolve compensation request information, included decision, description and request’s status. |  | | 4 | Staff sends save changes command | Update compensation status and notify to customer |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There are no unsolved compensation request | Show message to notify staff there are no unsolved compensation request. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * List of compensation request is always loaded from the system. * List of compensation request is sorted by get request date in descending order. * With unsolved request, its solved date does not exist. * A compensation request could have one of three decision is: * “Chưa quyết định” * “Chấp nhận bồi thường” * “Từ chối bồi thường” * A compensation request could have one of three status is: * “Đang xử lý” * “Hoàn tất” * “Hủy bỏ” * Search bar on the top help user finding contract or customer faster. * Pagination must be display if number of requests larger than 10 and auto-change based on staff’s selection. * Staff can select how many requests should be displayed in user view, default is 10 requests. | | | |

###### <Staff> View customer information

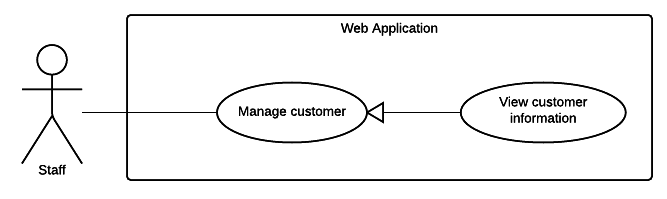


Figure 25: <Staff> View customer information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS05 | | | |
| Use Case No. | WS05 | **Use Case Version** | 2.0 |
| Use Case Name | View customer information | | |
| Author | KhaNC | | |
| Date | 20/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allows staff view customer details.   Goal:   * Staff can view customer’s detail information.   Triggers:   * Staff sends view customer information command.   Preconditions:   * User must login into the system with role Staff.   Post Conditions:   * Success: Customer’s detail information is shown. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to list of customer. | List of customer will be shown with following information:   * Customer’s code: link to customer detail * Customer’s full name: text * Customer’s phone: text * Customer’s NFC card ID: link to card detail. * Customer’s contract code: link to contract detail. | | 2 | Staff selects customer to view their detail information. | Customer detail will be shown with following information:   * Customer’s code: text * Customer’s full name: text * Customer’s address: text * Customer’s email address: text * Customer’s phone: text * Customer’s personal ID: text   Information of customer’s current card:   * Customer’s NFC card ID: link to card detail. * Card’s activated date: text * Card’s most recent access date: text   A list of customer’s contract as grid which show the following information:   * Customer’s contract code: link to contract detail. * Customer’s contract type: text * Contract’s start date: text * Contract’s expired date: text * Contract’s status: text |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * List of customer and their detail information is always loaded from the system. * Staff can view all customer’s card and each card’s access history. * Search bar on the top help user find customer faster. * Pagination must be display if number of requests larger than 10 and auto-change based on staff’s selection. * Staff can select how many requests should be displayed in user view, default is 10 requests. * Contract’s status is one of 6 below: * Pending * No Card * Ready * Expired * Request cancel * Cancelled | | | |

###### <Staff> Edit customer information

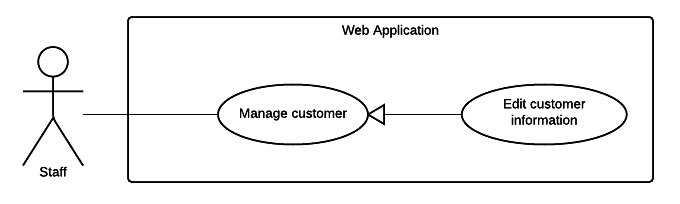


Figure 26: <Staff> Edit customer information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS06 | | | |
| Use Case No. | WS06 | **Use Case Version** | 2.0 |
| Use Case Name | Edit customer information | | |
| Author | KhaNC | | |
| Date | 20/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allows staff edit customer’s information.   Goal:   * Customer’s information is updated to the system.   Triggers:   * Staff sends update customer’s information command.   Preconditions:   * User must login into the system with role Staff.   Post Conditions:   * Success: Customer’s information is updated. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to edit customer information. | Edit customer view is shown with following labels and fields:   * Customer’s full name: free text input, required, length 3 – 80 * Customer’s address: free text input, required, length 3 – 250 * Customer’s email address: free text input, required, length 3 – 250 * Customer’s phone number: free text input, required, length 8 – 15 * Customer’s personal ID: free text input, length 8 – 15 | | 2 | Staff fills out the form. |  | | 3 | Staff sends update customer’s information command. | * Validate data   [Exception 1, 2, 3]   * Update customer’s new information to the system. * Reload customer detail. |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Missing of required fields | Show message notify staff input missed fields | | 2 | Length of field’s value is out of range | Show message notify staff which field’s value is out of range | | 3 | Entered email address is not a valid email | Show message notify entered email is not valid |   Relationships: N/A.  Business Rules:   * Customer’s detail information is always loaded from the system. * In case of success scenarios, customer new information would be updated to the system. * Reloaded customer detail will display customer updated information. * An email address must be validated by this regular expression:   /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/   * Customer’s current information must be shown in its respective fields. | | | |

###### <Staff> Create new customer

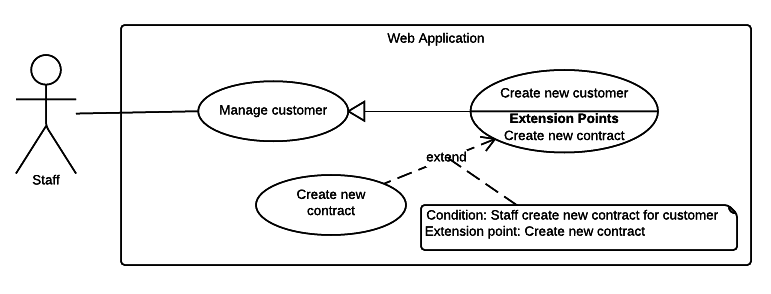


Figure 27: <Staff> Create new customer

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS07 | | | |
| Use Case No. | WS07 | **Use Case Version** | 2.0 |
| Use Case Name | Create new customer | | |
| Author | KhaNC | | |
| Date | 20/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allows staff create new customer.   Goal:   * A new customer is added to the system.   Triggers:   * Staff sends create new customer command.   Preconditions:   * User must login into the system with role Staff. * This customer is not existed in the system yet.   Post Conditions:   * Success: New customer is added to the system. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to create customer view. | Create customer view is shown with following labels and fields:   * Customer’s full name: free text input, required, length 3 – 80 * Customer’s address: free text input, required, length 3 – 250 * Customer’s email address: free text input, required, length 3 – 250 * Customer’s phone number: free text input, required, length 8 – 15 * Customer’s personal ID: free text input, length 8 – 15 | | 2 | Staff fills out the form. |  | | 3 | Staff sends create new customer command. | * Validate data   [Exception 1, 2, 3]   * Add new customer to the system. * Display create customer success message. |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Missing of required fields | Show message notify staff input missed fields | | 2 | Length of field’s value is out of range | Show message notify staff which field’s value is out of range | | 3 | Entered email address is not a valid email | Show message notify entered email is not valid |   Relationships: N/A  Business Rules:   * In case of success scenarios, a new customer would be added to the system. * An email address must be validated by this regular expression:   /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/   * System will suggest staff create new contract for the customer has just created. | | | |

###### <Staff> Create contract

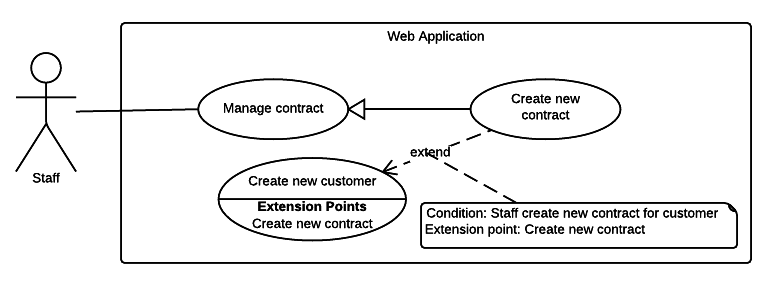


Figure 28: <Staff> Create contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS08 | | | |
| Use Case No. | WS08 | **Use Case Version** | 2.0 |
| Use Case Name | Create new contract | | |
| Author | KhaNC | | |
| Date | 20/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allows staff create new contract.   Goal:   * A new contract is added to the system.   Triggers:   * Staff sends create new contract command.   Preconditions:   * User must login into the system with role Staff. * There is at least 01 customer in the system.   Post Conditions:   * Success: New contract is added to the system * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to create new contract view. | Contract detail is shown with following label and fields:   * Customer’s full name: free text input, required, length 3 – 80   Contract’s information   * Contract’ type: select one of the options. * Contract’s start date: date time input, required * Contract’s expired date: date time input, required * Contract’s fee: free text input, required, length 1 – 10   Vehicle’s information:   * Vehicle’s plate number: free text input, required, length 4 – 15 * Vehicle’s brand: free text input, required, length 2 – 20 * Vehicle’s model code: free text input, required, length 2 – 20 * Vehicle’s type: free text input, required, length 2 – 20 * Vehicle’s color: free text input, required, length 2 – 20 * Vehicle’s engine: free text input, required, length 2 – 20 * Vehicle’s chassis: free text input, required, length 2 – 20 * Vehicle’s capacity: free text input, required, length 2 – 20 * Vehicle’s year of manufacture: free text input, required, length: 4 * Vehicle’s empty weight: free text input, required, length 1 – 4 * Vehicle’s seat capacity: free text input, required, length 1 – 3 * Image of vehicle registration certificate: file upload input, length 1 – 255   Payment information:   * Pay of charge date: text * Amount: free text input, required, length 1 – 10 * Receiver: free text input, required, length 3 – 80 | | 2 | Staff fills out the form. |  | | 3 | Staff sends create new contract command. | Display contract’s information to review and request for confirmation | | 4 | Staff confirms create new contract command. | * Validate data   [Exception 1, 2, 3]   * Add new contract’s information to the system. * Notify to staff created contract successfully. |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Missing of required fields | Show message notify staff input missed fields | | 2 | Length of field’s value is out of range | Show message notify staff which field’s value is out of range | | 3 | Entered vehicle is existed in the system | Show message notify staff this vehicle is existed. |   Relationships: N/A  Business Rules:   * In case of success scenarios, a new contract would be added to the system. * Staff can search and select a contract owner from available customers. * A new contract created successfully will has the initial status is “No Card”. * While staff issue a card for this contract, its status would change from “No Card” to “Ready”. * Contract’s start date must not be earlier than create contract date. * Contract’s term must not exceed 1 year. * Default contract’s term is 1 year. * Contract’s expired date must not be earlier than contract’s start date. * Contract’s type must be one of the following values: * “Xe trên 50cc có BH cho người trên xe” * “Xe trên 50cc không có BH cho người trên xe” * “Xe dưới 50cc có BH cho người trên xe” * “Xe dưới 50cc không có BH cho người trên xe” * “Xe mô tô ba bánh, xe gắn máy và các loại xe tương tự” | | | |

###### <Staff> Renew contract

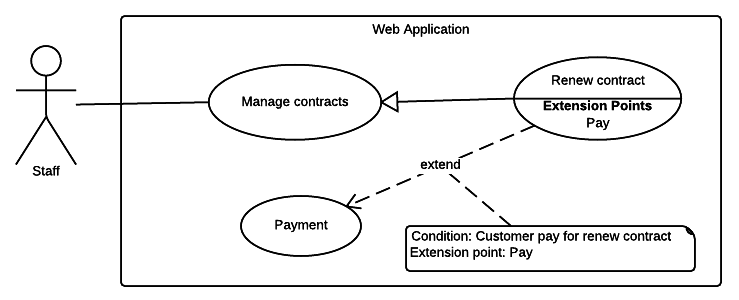


Figure 29: <Staff> Renew contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS09 | | | |
| Use Case No. | WS09 | **Use Case Version** | 2.0 |
| Use Case Name | Renew contract | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allows staff renew contract.   Goal:   * Contract’s new expired date will be updated to the system.   Triggers:   * Staff sends renew contract command.   Preconditions:   * User must login into the system with role Staff. * Contract’s status must be “No Card”, “Ready” or “Expired”. * The payment process for renew contract was completed.   Post Conditions:   * Success: Contract is renewed * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to renew contract view. | Renew contract detail is shown with following information:   * Contract’s new start date: date time input, required * Contract’s new expired date: date time input, required * Renew contract fee: free text input, required, length 1 – 10 * Pay of charge date: date time input, required * Description: free text input, length 1 – 2000 * Name of staff: free text input, required, length 3 – 80 | | 2 | Staff fills out the form. |  | | 3 | Staff sends renew this contract command | Display confirm renew contract view. | | 4 | Staff confirms renew this contract command. | * Validate data   [Exception 1, 2]   * Update contract’s information to the system. * Reload contract detail. |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Missing of required fields | Show message notify staff input missed fields | | 2 | Length of field’s value is out of range | Show message notify staff which field’s value is out of range |   Relationships: N/A  Business Rules:   * In case of success scenarios, a new expired date of contract would be updated to the system. * A contract renewed successfully will has status is “Ready”: * If contract’s status is “Ready”, there are no change. * In case of the contract has status is “No Card” or “Expired”, its will be changed to “Ready”. Contract’s start date must not be earlier than create contract date. * Contract’s term must not exceed 1 year. * Default contract’s term is 1 year. * Contract’s expired date must not be earlier than contract’s start date. | | | |

###### <Staff> Cancel contract

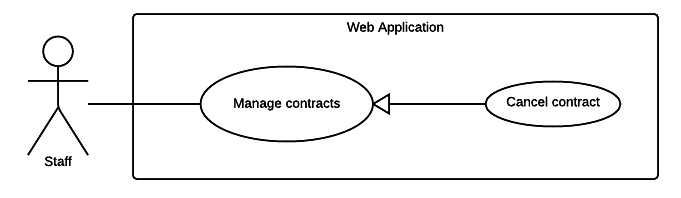


Figure 30: <Staff> Cancel contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS10 | **Use Case Version** | 2.0 |
| Use Case Name | Cancel contract | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allows staff cancel a contract.   Goal:   * Contract’s new status is updated to the system.   Triggers:   * Staff sends cancel contract command.   Preconditions:   * User must login into the system with role Staff. * Contract’s status must not be “Cancelled” * The expired date is less than 2 months. * The compensation for breach of contract was completed.   Post Conditions:   * Success: Contract is cancelled. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to cancel contract view. | Cancel contract detail is shown with following information:   * The reason why cancel contract: free text input, required, length 1 – 250 * Description: free text input, length 1 – 2000 * Name of staff: free text input, required, length 3 – 80 | | 2 | Staff fills out the form. |  | | 3 | Staff sends cancel contract command. | Display confirm cancel contract view. | | 4 | Staff confirms cancel contract command. | * Validate data   [Exception 1, 2]   * Update contract’s information to the system. * Reload contract detail. |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Missing of required fields | Show message notify staff input missed fields | | 2 | Length of field’s value is out of range | Show message notify staff which field’s value is out of range |   Relationships: N/A  Business Rules:   * No notification will be sends to user when a staff cancel a contract. * A contract cancelled successfully will change status from “Pending”, “No Card”, “Ready”, and “Expired” or “Request cancel” to “Cancelled”. | | | |

###### <Staff> Update contract information

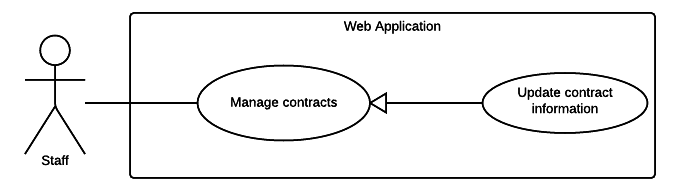


Figure 31: <Staff> Update contract information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS11 | | | |
| Use Case No. | WS11 | **Use Case Version** | 2.0 |
| Use Case Name | Update contract information | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Staff   Summary:   * This use case allows staff to update contract information.   Goal:   * Staff can update contract information.   Triggers:   * Staff sends command to update contract information.   Preconditions:   * User has to logged in to the system as Staff role * Contract is existed in the system and not deactivated   Post Conditions:   * Success: Contract information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to update contract information. | System list out information of contract:  **Personal information**   * Name: free text input, required, length 3 – 80. * Address: free text input, required, length 3 – 250. * Email: free text input, required, length 3 – 250. * Phone number: free text input, required, length 8 – 15. * Personal ID: free text input, length 15.   **Contract information** (all information below are required)   * Contract’s type: select one from the options. * Start date: date time input * Expired date: date time input   **Vehicle information**   * Plate: free text input, required, length 4 – 15. * Brand: free text input, required, length 2 – 20. * Model code: free text input, required, 2 – 20. * Vehicle type: free text input, required, 2 – 20. * Color: free text input, required, length 2 – 20. * Engine: free text input, required, length 2 – 20. * Chassis: free text input, required, length 2 – 20. * Capacity: free text input, required, length 2 – 20. * Year of manufacture: number text input, required, value from 1900 to 2200. * Weight: free text input, required, value from 1 – 1000, unit: kilogram * Seat capacity: free text input, required, value from 1 – 100. * Certificate image: file upload input | | 2 | Staff inputs required information. |  | | 3 | Staff sends command to save new information. | System shows message notify contract information is updated successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to save new information. | System shows error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Contract end date must not be earlier start date. * Expired date and start date must not exceed 1 year. * Contract information will be updated to system. * Contract’s type must be one of the following values: * “Xe trên 50cc có BH cho người trên xe” * “Xe trên 50cc không có BH cho người trên xe” * “Xe dưới 50cc có BH cho người trên xe” * “Xe dưới 50cc không có BH cho người trên xe” * “Xe mô tô ba bánh, xe gắn máy và các loại xe tương tự” | | | |

###### <Staff> Update compensation history

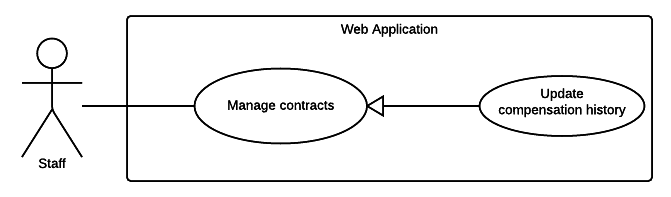


Figure 32: <Staff> Update compensation history

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS12 | | | |
| Use Case No. | WS12 | **Use Case Version** | 2.0 |
| Use Case Name | Update compensation history | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Staff   Summary:   * This use case allows staff to update compensation information.   Goal:   * Staff can update compensation information.   Triggers:   * Staff sends command to update compensation information.   Preconditions:   * User has to logged in to the system as Staff role * Compensation is existed in the system   Post Conditions:   * Success: Compensation information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to update compensation information. | System list out information of compensation:   * Driver name: free text input, required, length 3 – 80. * License number: free text input, required, length 10 – 15. * License type: free text input, required, length 1 – 10. * Driver phone: free text input, required, length 8 – 15. * Vehicle capacity: free text input, required, length 1 – 20. * Driver address: free text input, required, length 3 – 250. * Plate: free text input, required, length 4 – 15. * Date: date time input, required. * Place: free text input, required, length 4 – 15. * Control department: free text input, required, length 3 – 250. * Description: free text input, required, length 1 – 2000. * Human damage: free text input, required, length 1 – 2000. * Asset damage: free text input, required, length 1 – 2000. * Observer: free text input, required, length 3 – 80. * Compensation note: free text input, required, length 1 – 2000. * Attachment: file upload input * Created date: date time input, required. * Resolve date: date time input, required. * Decision: free text input, required, length 1 – 250. * Resolve note: free text input, required, length 1 – 2000. | | 2 | Staff inputs information |  | | 3 | Staff sends command to save new information. | System shows message notify compensation information is updated successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to save new information. | System shows error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Compensation information will be updated to system. | | | |

###### <Staff> Update punishment history

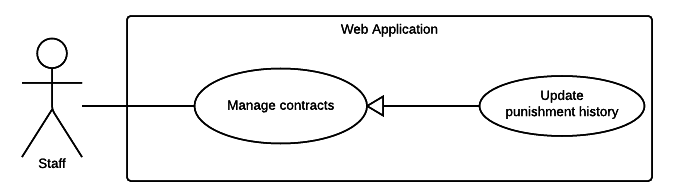


Figure 33: <Staff> Update punishment history

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS13 | | | |
| Use Case No. | WS13 | **Use Case Version** | 2.0 |
| Use Case Name | Update punishment history | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Staff   Summary:   * This use case allows staff to update punishment information.   Goal:   * Staff can update punishment information.   Triggers:   * Staff sends command to update punishment information.   Preconditions:   * User has to logged in to the system as Staff role * Punishment is existed in the system   Post Conditions:   * Success: Punishment information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to update punishment information. | System list out information of punishment:   * Date: date time input, required. * Title: free text input, required, length 1 – 250. * Attachment: file upload input, required. | | 2 | Staff inputs information |  | | 3 | Staff sends command to save new information. | System shows message notify punishment information is updated successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to save new information. | System shows error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Punishment information will be updated to system. * Punishment date must not exceed current date. | | | |

###### <Staff> Update accident history

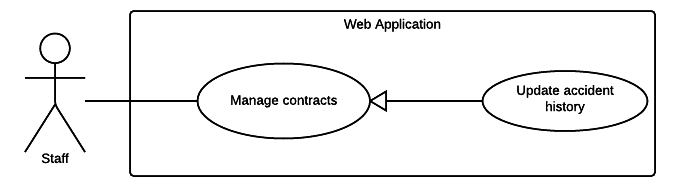


Figure 34: <Staff> Update accident history

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS14 | | | |
| Use Case No. | WS14 | **Use Case Version** | 2.0 |
| Use Case Name | Update accident history | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Staff   Summary:   * This use case allows staff to update accident information.   Goal:   * Staff can update accident information.   Triggers:   * Staff sends command to update accident information.   Preconditions:   * User has to logged in to the system as Staff role * Contract of the accident is existed in the system   Post Conditions:   * Success: Punishment information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to accident information. | System list out information of accident:   * Date: date time input, required. * Title: free text input, required, length 1 – 250. * Attachment: file upload input, required. | | 2 | Staff inputs information. |  | | 3 | Staff sends command to save new information. | System shows message notify compensation information is updated successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to save new information. | System shows error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Accident information will be updated to the system. * Accident date must not exceed current date. | | | |

###### <Staff> View cards information

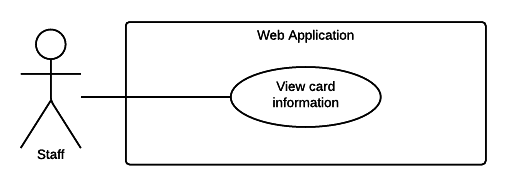


Figure 35: <Staff> View card information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS15 | | | |
| Use Case No. | WS15 | **Use Case Version** | 2.0 |
| Use Case Name | View cards information | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Staff   Summary:   * This use case allows staff to view published cards information.   Goal:   * Staff can view cards information.   Triggers:   * Staff goes to view cards information.   Preconditions:   * User has to logged in to the system as Staff role * Cards information is existed in the system   Post Conditions:   * Success: Cards information is shown * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to view cards information | System list out information of published cards:   * Card ID: text link to card detail * Card owner: text * Last access time: text * Status: text   [Exception 1] | | 2 | Staff sends command to view card detail information | System shows card detail information:   * Card ID: text * Card owner: text * Status: text * Activated date: text * Last access time: text * History of access   [Exception 2] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to view cards information | System shows no connection error message. | | 2 | Staff sends command to view card detail information | System shows no connection error message. |   Relationships:  Business Rules:   * Cards information will be loaded from the system. * History of access will show 10 results at a time, with pagination links at the bottom. | | | |

###### <Staff> Update contract type information



Figure 36: <Staff> Update contract type information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS16 | | | |
| Use Case No. | WS16 | **Use Case Version** | 2.0 |
| Use Case Name | Update contract type information | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Staff   Summary:   * This use case allows staff to update contract type.   Goal:   * Staff can update contract type.   Triggers:   * Staff sends command to update contract type.   Preconditions:   * User has to logged in to the system as Staff role * Contract type is existed in the system   Post Conditions:   * Success: Contract type information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to view contract type information. | System list out information of accident:   * Name: free text input, required, length 1 – 250. * Description: free text input, required, length 1 – 2000. * Price per year: free number input, required, value from 0 to 1 billion, unit: VND. | | 2 | Staff inputs information |  | | 3 | Staff sends command to save new information. | System shows message notify contract type information is updated successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to save new information. | System shows error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Contract type information will be updated to system. * Contract type price per year must be greater than 0 and less than 1 billion. | | | |

##### <Admin> Overview Use Case

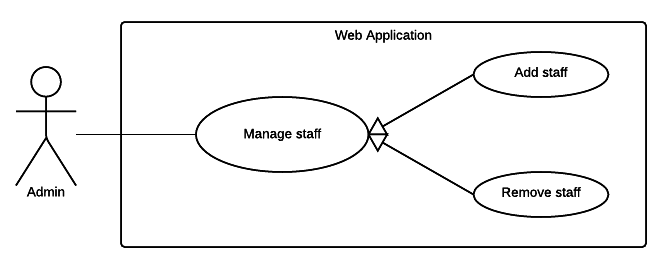


Figure 37: <Admin> Overview Use Case

###### < Admin> Remove staff

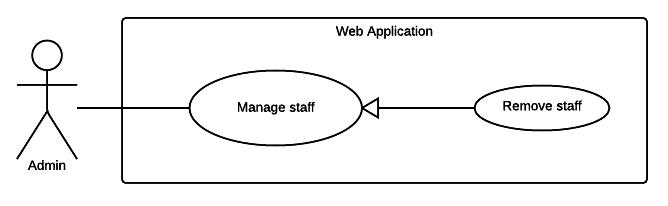


Figure 38: <Admin> Remove staff

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WA01 | | | |
| Use Case No. | WA01 | **Use Case Version** | 2.0 |
| Use Case Name | Remove staff | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Admin   Summary:   * This use case allows admin to remove other staff from the system.   Goal:   * Admin can remove staff from the system.   Triggers:   * Admin sends command to remove staff from the system.   Preconditions:   * User has to logged in to the system as Admin role   Post Conditions:   * Success: Admin be able to remove staff from the system * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin goes to manage members | System list out information of staffs in system:   * Staff code: text * Name: text * Email: text * Phone number: text | | 2 | Admin select the staff to delete |  | | 3 | Admin sends command to delete staff. | System shows message staff has been removed successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends command to delete staff. | System shows error message user cannot remove themselves from the system. |   Relationships: N/A  Business Rules:   * Their must be at least 01 staff in the system. | | | |

###### < Admin> Add staff

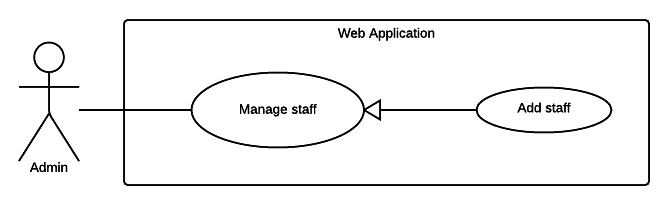


Figure 39: <Admin> Add staff

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WA02 | | | |
| Use Case No. | WA02 | **Use Case Version** | 2.0 |
| Use Case Name | Add staff | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Admin   Summary:   * This use case allows admin to add new staff to the system.   Goal:   * Admin can add new staff to the system.   Triggers:   * Admin sends command to add new staff to the system.   Preconditions:   * User has to logged in to the system as Admin role   Post Conditions:   * Success: Admin be able to add new staff to the system * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin goes to add staff. | System require information of staff:   * Staff code: text, required, length 3 - 80 * Name: text, required, length 3 – 80. * Email: text, required, length 3 – 250. * Phone number: text, required, length 8 – 15. | | 2 | Admin inputs information |  | | 3 | Admin sends command to add new staff. | System shows message staff has been added successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends command to add other staff. | System shows error message to ask user to enter missing required field. |   Relationships: N/A  Business Rules:   * Staff code is the code used in physical system maintained by insurance company, it probably already exists for each staff when join in the company. If the company has no staff code for their staff, this field can be enter personal id of the staff. * Staff code must not be duplicated. | | | |

##### <Mobile apps> Overview Use Case

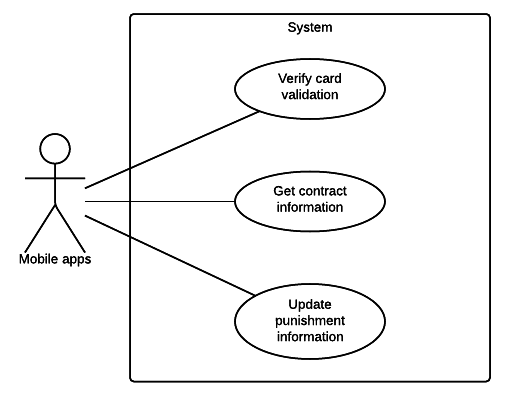


Figure 40: <Mobile apps> Overview Use Case

###### <Mobile apps> Verify card validation

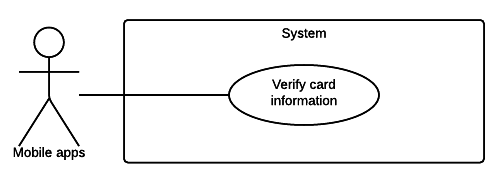


Figure 41: <Mobile apps> Verify card validation

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WM01 | | | |
| Use Case No. | WM01 | **Use Case Version** | 2.0 |
| Use Case Name | Verify card validation | | |
| Author | PhucNH | | |
| Date | 26/05/2015 | **Priority** | Medium |
| Actor:   * Mobile app   Summary:   * Mobile app can check the validation of card.   Goal:   * The card can be checked the validation.   Triggers:   * The NFC card has been put close to the mobile.   Preconditions:   * The mobile app can connect with server to recognize the information.   Post Conditions:   * Success: Mobile app notifies that card is valid. * Fail: Mobile notifies error.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Mobile app sends verify request to system | System returns the insurance contract and the motor information:   * Motor owner: text. * Owner address: text. * Phone number: text. * Plate number: text. * Engine: text. * Chassis: text. * Brand: text. * Model Code: text. * Type: text. * Color: text. * Capacity: text. * Year of manufacture: text * Weight: text. * Seat capacity: text. * Contract fee: text. * Start date: text * Expired date: text * Card Status: text.   [Alternative 1]  [Exception 1, 2] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If unable to read the card or the card is invalid | Mobile app shows error to notify police that mobile cannot read this card |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot read card. | Mobile app notifies that cannot read card | | 2 | Cannot connect to server. | Mobile app shows error connect to server is fail |   Relationships: N/A  Business Rules:   * Mobile application sends request about information card to server and receives information about contract. | | | |

###### <Mobile apps> Get contract information

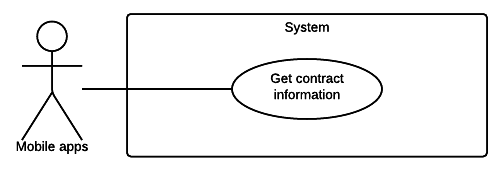


Figure 42: <Mobile apps> Get contract information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WM02 | | | |
| Use Case No. | WM02 | **Use Case Version** | 2.0 |
| Use Case Name | Get contract information | | |
| Author | PhucNH | | |
| Date | 26/05/2015 | **Priority** | Medium |
| Actor:   * Mobile app   Summary:   * Mobile app can get information about contract.   Goal:   * The information belong to card can be get and show on the screen of mobile.   Triggers:   * The application gets the information of contract.   Preconditions:   * The NFC card has checked the validation.   Post Conditions:   * Success: Mobile app returns information about contract. * Fail: Mobile app notifies error.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Mobile app sends request to get information about card | Mobile app returns the insurance contract and the motor information:   * Motor owner: text. * Owner address: text. * Phone number: text. * Plate number: text. * Engine: text. * Chassis: text. * Brand: text. * Model Code: text. * Type: text. * Color: text. * Capacity: text. * Year of manufacture: text. * Weight: text. * Seat capacity: text. * Contract fee: text. * Start date: text. * Expired date: text.   [Exception 1, 2] |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If unable to read the card or the card is invalid | Mobile shows error to notify police that mobile cannot read this card | | 2 | Cannot connect to server. | Mobile shows error connect to server is fail |   Alternative Scenario: N/A  Relationships: N/A  Business Rules:   * Mobile application sends request about information card to server and receives information about contract. | | | |

###### <Mobile apps> Update punishment information

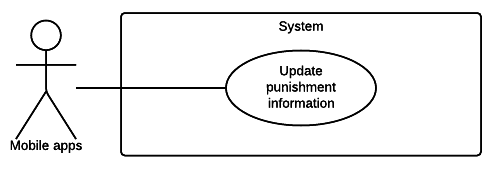


Figure 43: <Mobile apps> Update punishment information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WM03 | | | |
| Use Case No. | WM03 | **Use Case Version** | 2.0 |
| Use Case Name | Update punishment information | | |
| Author | PhucNH | | |
| Date | 26/05/2015 | **Priority** | High |
| Actor:   * Mobile application.   Summary:   * Mobile app can update punishment information.   Goal:   * Update punishment information of the driver.   Triggers:   * The mobile application can read NFC card successful.   Preconditions:   * The mobile application in Verify card screen. * The NFC card is valid.   Post Conditions:   * Success: Update punishment information of the driver * Fail: Mobile notifies error.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Mobile app sends command to update punishment | Mobile app returns some fill and request input fill | | 2 | Police chooses add more punishment. | Mobile required input description about punishment and picture about it.   * Description: free text input, required, length 1 - 200 * Picture: file upload input, required | | 3 | Mobile app get new information about punishment | Mobile sends punishment from police and notify this action is success  [Exception 1, 2] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Missing description and picture | Mobile reminds input description and picture. | | 2 | Cannot connect to server. | Warning that cannot connect to server. |   Relationships: N/A  Business Rules:   * Information about punishment will be sent in server and update in contract of user. | | | |

##### <System> Overview Use Case

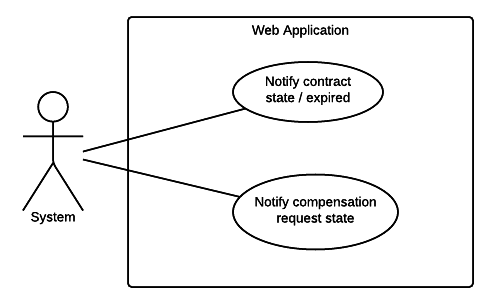


Figure 44: <System> Overview Use Case

###### <System> Notify contract state / expired

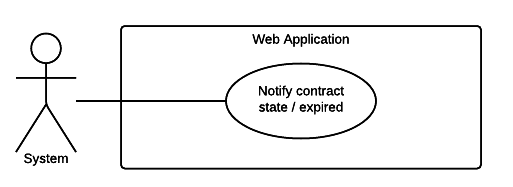


Figure 45: <System> Notify contract state / expired

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WY01 | | | |
| Use Case No. | WY01 | **Use Case Version** | 2.0 |
| Use Case Name | Notify contact state / expired | | |
| Author | PhucNH | | |
| Date | 26/05/2015 | **Priority** | Medium |
| Actor:   * System   Summary:   * This use case describes how the system sends notify about contract state / expired.   Goal:   * The information about the contract will be shown about notify to the user if user’s contract expires.   Triggers:   * System run a timer task that trigger check event.   Preconditions:   * Their must be at least one contract in the system.   Post Conditions:   * Success: Show the status of contract and notify to user if their contract close to the deadline. * Fail: System display error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System run timer task to check contract state and sends notify | User can see information show:   * Status of contract: text * Information about contract: text * Information about motorbike: text   [Exception 1] | |  |  |  |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System timer task is interrupted | No notification will be sent. Error detail will be tracked in a log file. |   Relationships: N/A  Business rules:   * System timer will send check event at 00:00 everyday. * Notification will repeat every 4 days if the user don’t renew the contract. * Contracts that are not renewed will has status “Expired” and have no value. * After 1 month of Expired, if the user don’t renew the contract, the contract’s status will be changed to “Cancelled”. | | | |

###### <System> Notify compensation request state

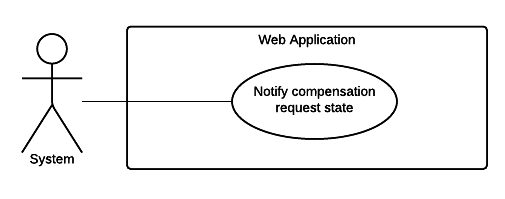


Figure 46: <System> Notify compensation request state

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WY02 | | | |
| Use Case No. | WY02 | **Use Case Version** | 2.0 |
| Use Case Name | Notify compensation request status. | | |
| Author | PhucNH | | |
| Date | 26/05/2015 | **Priority** | Medium |
| Actor:   * System   Summary:   * This use case describes how the system sends notify about compensation status.   Goal:   * The information about request compensation of user is saved.   Triggers:   * System run a timer task that trigger check event.   Preconditions:   * Their must be at least one compensation in the system.   Post Conditions:   * Success: Use received a notification about the compensation request when it is resolved. * Fail: System display error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System run timer task to check contract state and sends notify | Use can see information show:   * Status of compensation: text   [Exception 1] |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System timer task is interrupted | No notification will be sent. Error detail will be tracked in a log file. |   Relationships:  Business Rules:   * System timer will sends check event at 00:00 everyday. | | | |

##### <Payment system> Overview Use Case



Figure 46: <Payment> Payment

###### <Payment system> Payment



Figure 47: <Payment> Payment

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WP01 | | | |
| Use Case No. | WP01 | **Use Case Version** | 2.0 |
| Use Case Name | Payment | | |
| Author | KhaNC | | |
| Date | 03/06/2015 | **Priority** | High |
| Actor:   * Payment system.   Summary:   * This use case allows payment system process the payment.   Goal:   * Payment process is completed.   Triggers:   * Customer sends pay for charge by payment system command.   Preconditions:   * User must login into the system with role Customer.   Post Conditions:   * Success: Payment is completed successfully. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer select payment via PayPal. | Summary of customer payment:   * Description: text. * Amount: text. * Total: text.   Payment view is shown with following labels and fields:   * Customer’s PayPal account: free text input, required, length 3 – 250 * Customer’s PayPal password: free text input, required, length 6 – 32 | | 2 | Staff fills out the form. |  | | 3 | Staff sends pay command. | * Validate data   [Alternative 1]  [Exception 1, 2, 3]   * Process payment in PayPal |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Identity information is not match. | Show message notify customer input wrong email or password. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Missing of required fields | Show message notify staff input missed fields | | 2 | Length of field’s value is out of range | Show message notify staff which field’s value is out of range | | 3 | Entered email address is not a valid email | Show message notify entered email is not valid |   Relationships: N/A.  Business Rules:   * Use PayPal API to validate customer email and password. * PayPal will send notification to alert transaction is failed or succeeded. * An email address must be validated by this regular expression:   /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ | | | |

#### Checker Mobile Application

##### <Police> Overview Use Case

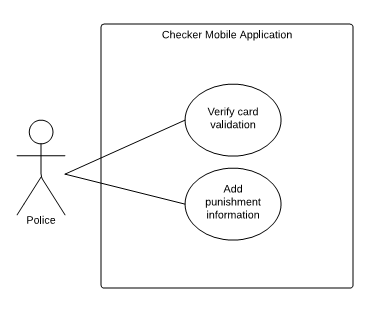


Figure 48: <Police> Overview Use Case

###### <Police> Verify card information

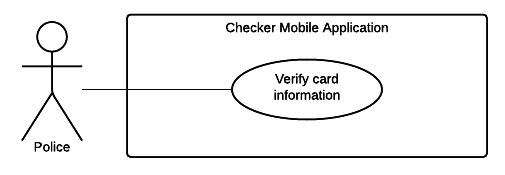


Figure 49: <Police> Verify card information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – CP01 | | | |
| Use Case No. | CP01 | **Use Case Version** | 2.0 |
| Use Case Name | Verify card information | | |
| Author | PhucNH | | |
| Date | 20/06/2015 | **Priority** | High |
| Actor:   * Police   Summary:   * Traffic Police and Police Department can use the device to view card’s information and check if the card is valid or not.   Goal:   * Check if the insurance card is valid or not.   Triggers:   * Police put the NFC card near the device to read card information.   Preconditions:   * The application in home screen.   Post Conditions:   * Success: Show the insurance card information. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Police put the NFC card close to the device. | Show the insurance contract and the motor information:   * Motor owner: text. * Owner address: text. * Phone number: text. * Plate number: text. * Engine: text. * Chassis: text. * Brand: text. * Model Code: text. * Type: text. * Color: text. * Capacity: text. * Year of manufacture: text. * Weight: text. * Seat capacity: text. * Contract fee: text. * Start date: text * Expired date: text * Card Status: text.   [Alternative 1]  [Exception 1, 2] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If unable to read the card or the card is invalid | Mobile app shows error to notify police that mobile cannot read this card |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot read card | Mobile app notifies that cannot read card | | 2 | Cannot connect to server. | Mobile app shows error connect to server is fail |   Relationships:  Business Rules:   * Mobile application sends request about information card to server and receives information about contract. * Highlight the status of the card include:   + Valid card   + Expired card   + Card is nearly expired, show remaining days | | | |

###### <Police> Add punishment information

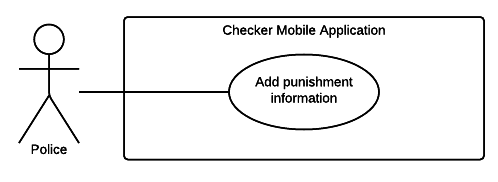


Figure 50: <Police> Add punishment information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – CP02 | | | |
| Use Case No. | CP02 | **Use Case Version** | 2.0 |
| Use Case Name | Add punishment information | | |
| Author | PhucNH | | |
| Date | 26/05/2015 | **Priority** | High |
| Actor:   * Police   Summary:   * Traffic Police and Police Department can use the device to add punishment information of the driver.   Goal:   * Add punishment information of the driver.   Triggers:   * Device reads NFC card successful.   Preconditions:   * The application in Verify card screen. * The NFC card is valid.   Post Conditions:   * Success: Add punishment information of the driver. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Police chooses new punishment | Mobile allows for police add new punishment. | | 2 | Police chooses add more punishment. | Mobile required police input description about punishment and picture about it.   * Description: free text input, required, length 1 - 200 * Picture: file upload input, required | | 3 | Police fill the punishment information and sends punishment. | Mobile sends punishment from police and notify this action is success  [Exception 1, 2] |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Police input missed one of punishment description and picture of punishment record. | Mobile reminds that police forgets take a photo for punishment. | | 2 | Cannot connect to server. | Warning that cannot connect to server. |   Relationships: N/A  Business Rules:   * Information about punishment will be sent in server and update in contract of user. | | | |

#### Printer Mobile Application

##### <Staff> Overview Use Case

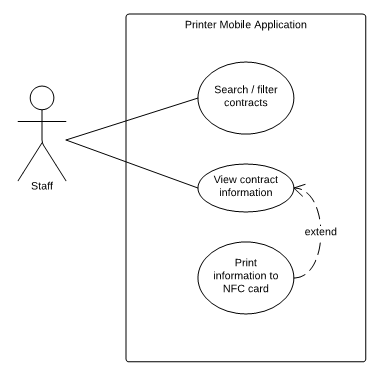


Figure 51: <Staff> Overview Use Case

###### <Staff> Search / filter contract

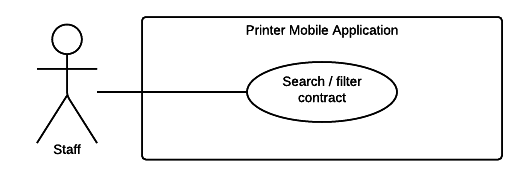


Figure 52: <Staff> Search / filter contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – PS01 | | | |
| Use Case No. | PS01 | **Use Case Version** | 2.0 |
| Use Case Name | Search / filter contract | | |
| Author | PhucNH | | |
| Date | 26/05/2015 | **Priority** | Medium |
| Actor:   * Staff   Summary:   * This use case helpss staff can search or filter the information of contract by contract code or customer name.   Goal:   * The information of contract is shown following the request search of staff.   Triggers:   * Staff sends command to search/ filter contract.   Preconditions:   * Staff must login successful.   Post Conditions:   * Success: The information of request search is shown. * Fail: System display error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends command to search/ filter contract  Staff chooses contract | System shows:  Information about request search.  System shows information about contract:   * Contract code: text * Customer name: text * Date start: text * Time of contract: text * Status: text   [Alternative 1]  [Exception 1] | |  |  |  |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System cannot finds the result from request | System notifies cannot find the request search. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | The request search is not valid. It is include:   * Invalid the value of textbox * Fill some especial character | Website show message to notify. |   Relationships:  Business Rules:  Search/ filter with some criterion:   * Search/ filter by contract code * Search/ filter by customer name * Search/ filter by start date * Search/ filter by time of contract | | | |

###### <Staff> View contract information



Figure 53: <Staff> View contract information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – PS02 | | | |
| Use Case No. | PS02 | **Use Case Version** | 2.0 |
| Use Case Name | View contract information | | |
| Author | PhucNH | | |
| Date | 21/05/2015 | **Priority** | Medium |
| Actor:   * Staff   Summary:   * This use case helps user view their contract’s information.   Goal:   * User can review their contract’s information.   Triggers:   * User chooses information insurance tab in the navigation bar.   Preconditions:   * User must login into the system.   Post Conditions:   * Success: Show contract’s information to user. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User chooses information insurance in the navigation bar. | Show information about the insurance and the motor’s information:   * Motor owner: text. * Owner address: text. * Phone number: text * Plate number: text. * Engine: text. * Chassis: text. * Brand: text. * Model Code: text. * Type: text. * Color: text. * Capacity: text. * Year of manufacture: text * Weight: text. * Seat capacity: text. * Contract fee: text. * Start date: text * Expired date: text |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot connect to server. | System shows error that cannot connect to server. |   Relationships:  Business Rules:   * System application shows information about contract. | | | |

###### <Staff> Print information to NFC card



Figure 54: <Staff> Print information to NFC card

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – PS03 | | | |
| Use Case No. | PS03 | **Use Case Version** | 2.0 |
| Use Case Name | Print information to NFC card | | |
| Author | PhucNH | | |
| Date | 26/05/2015 | **Priority** | High |
| Actor:   * Staff   Summary:   * This use case helpss staff can print information of contract into NFC card.   Goal:   * The information of NFC can be printed from application on mobile   Triggers:   * Staff finds the information of contract then command to request print information into card.   Preconditions:   * Staff must login successful in mobile application.   Post Conditions:   * Success: Notify that print successfully. * Fail: Notify that print fail.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff filters by name or contract code to find contract | Show information of customer:   * Motor owner: text * Address: text * Phone Number: text * Plate: text * Chassis: text * Engine: text * Type: text * Time: text * Insurance premium: text * Fee payer: text * People issue: text * Date issue: text * Hot line: text | | 2 | Staff choose print information into card | Notify print success  [Alternative 1]  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff commands to search/ filter but cannot see the result of contract | Staff commands a request to server to notify |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If the information of contract cannot print into the NFC card | Show message print fail |   Relationships:  Business Rules:   * The information of staff who prints that card will be saved and information of card will be printed and transfer for user. | | | |

## Software System Attribute

### Usability

### Reliability

### Availability

### Security

### Maintainability

### Portability

### Performance

## Conceptual Diagram