**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 : 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 send 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 send 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 send an email to notify the insured one when their insurance is expired.
  + Notify compensation state: system will send 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 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 can’t 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 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 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 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 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 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 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 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

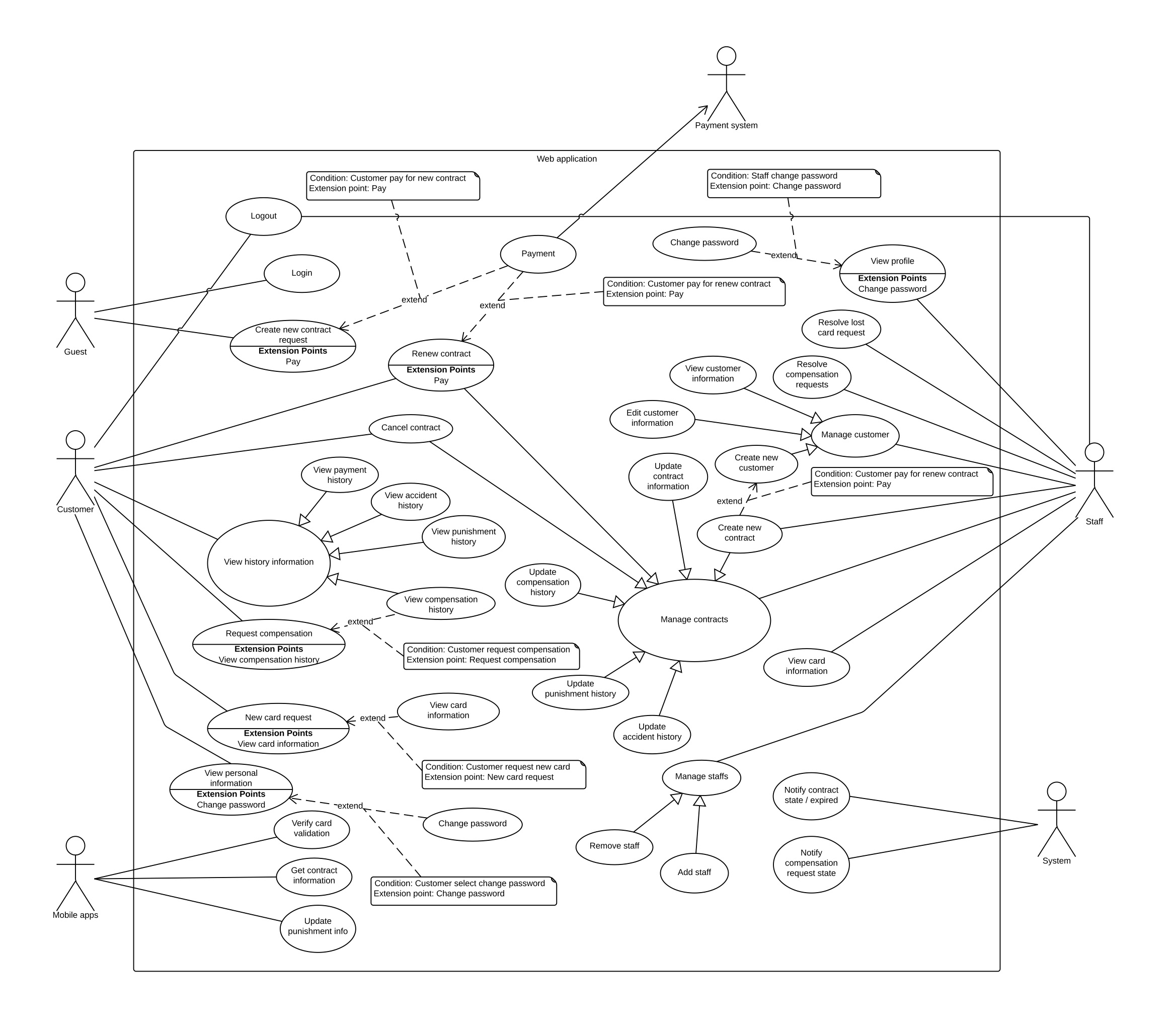


Figure Web Application Overview Use Case

#### Checker Mobile Application

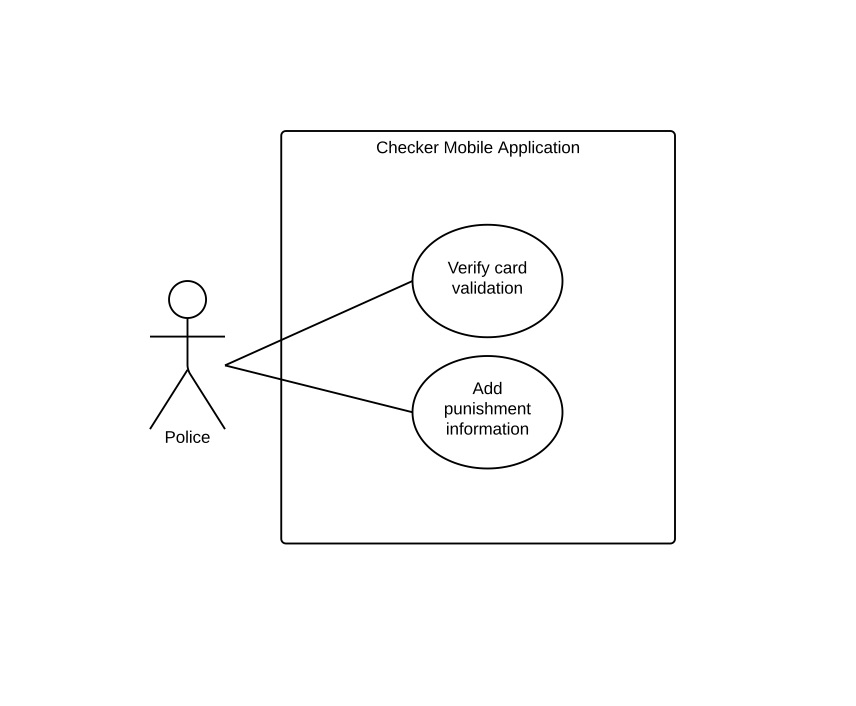


Figure Checker Mobile Application

#### Printer Mobile Application

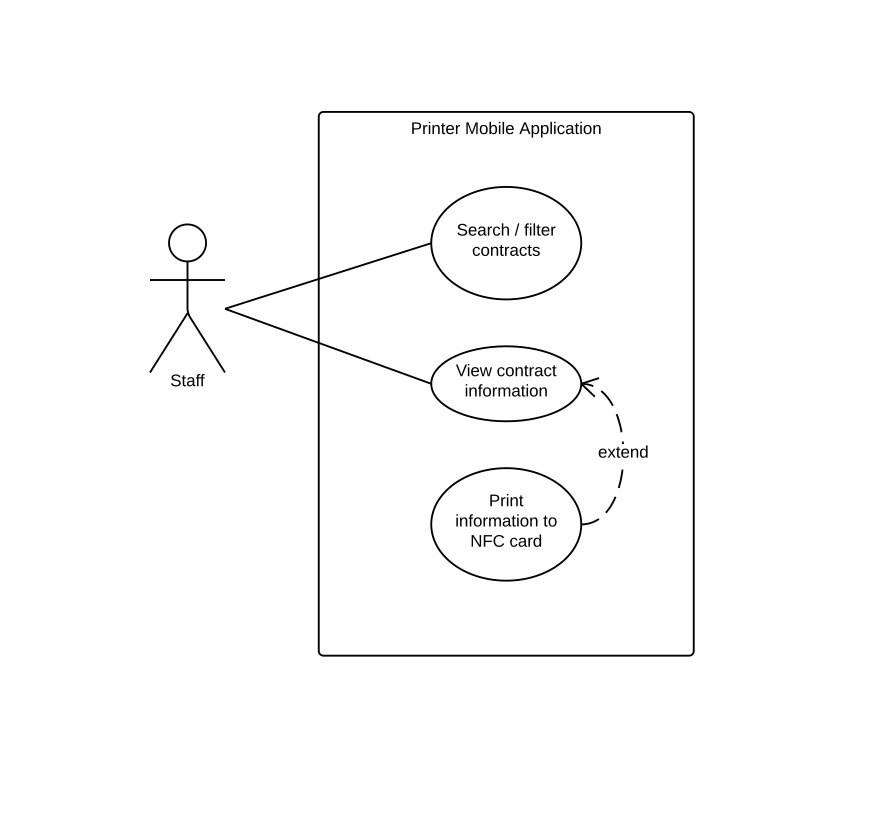


Figure Printer Mobile Application

### List of Use Case

#### Web Application

##### <Guest> Overview Use Case

###### <Guest> Login

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Login | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | Normal |
| Actor:  Guest  Summary:  This use case allow guest to login to the system.  Goal:  Guest can login to the system.  Triggers:  Guest send command to login.  Preconditions:  N/A  Post Conditions:  Success: Guest will be logged in to the system.  Fail: Show error message.  Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest go to login page | System requires identity information from Guest:  Email or customer code: free text input, required, min length: 1, max length: 254.  Password: free text input, required, min length: 1, max length: 128. | | 2 | Guest input information |  | | 3 | Guest send command to login to system | Guest will login system with their specific role  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest send command to login to system | Wrong identity information, system show error message |   Relationships: N/A  Business Rules:   * After login to system, guest will be redirected to specific page based on their role on the system: staff or customer. * If role is “Customer”, the page will redirect to Customer page * If role is “Staff”, the page will redirect to Staff Dashboard page | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow guest to create new contract request.  Goal:  Guest can create new contract request.  Triggers:  Guest send 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 go to new contract page | System requires information from guest:  **Personal information**   * Name (required): free text input * Address (required): free text input * Email (required): free text input * Phone number (required): free text input * Personal ID: free text input   **Contract information** (all information below are required)   * Contract type: can be optional selected from these 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” * Start date: date input * Expired date: date input   **Vehicle information** (all information below are required)   * Plate: free text input * Brand: free text input * Model code: free text input * Vehicle type: free text input * Color: free text input * Engine: free text input * Chassis: free text input * Capacity: free text input * Year of manufacture: number text input * Weight: free text input * Seat capacity: free text input * Certificate image: file upload input | | 2 | Guest input information |  | | 3 | Guest send command to create new contract request. | System navigates to payment page  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest send command to create new contract request | System show error message to ask user input missing required fields. |   Relationships: Payment  Business Rules:   * New contract will be created in the system with inputted information. * If user paid for the contract, status of the contract will be “Ready” * If user don’t paid for the contract yet, status of the contract will be “Pending” * An email contains customer code and password will be sent to user, user can use this information to login to the system later. * Staff will receive a notification about new contract. * Expired date and start date must not exceed 1 year. * Contract end date must not be earlier start date. * Contract price (per year) will be calculated from contract type and is set by staff. | | | |

##### <Customer> Overview Use Case

###### <Customer> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Logout | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | Normal |
| Actor:   * User, 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.  Return to the home page. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * System clear user session and logout user. | | | |

###### <Customer> Renew contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 help 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.   Post Conditions:   * Success: User renew contract or send a renew contract base on the payment way he/she used. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends renew contract command. | Display new page show user the payment gateways and the renewal fee:   * Payment gateways:   + PayPal: radio button.  + Direct payment: radio button.   * Renewal fee: currency. | | 2 | If user chooses PayPal gateway and send confirm command.  [Alternative 1] | Forward to PayPal payment page to process the payment. | | 3 | User process the PayPal payment | If payment succeed,  Update information to database. Renew user insurance contract.  Show message renew successful.  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If user chooses direct payment and send 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 “PayPal payment”.  Business Rules:   * If user paid the renew fee through PayPal, system automatically change the contract status to Ready. * If user paid the renew fee directly, Staff will update the payment for that contract and change contract status to Ready. * If the card expired date is more than 2 months, the user can’t make a renew contract request. | | | |

###### <Customer> Cancel contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Cancel contract | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Customer.   Summary:   * This use case help 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 must be valid.   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 sends cancel contract request command. | Display new page with radio buttons ask user to choose the reason why he/she cancel contract request and a button to attach file:   * List of reasons to cancel contract: radio buttons. * Other reason: input text free. * Attach file: button. | | 2 | User chooses the reason why he/she want to cancel contract and attach the related file (if any). Then send confirm command.  [Alternative 1] | Show success message.  [Exception 1]  [Exception 2] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User send exit request. | Return to the contract information page. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If user doesn’t check any radio button | Show message to notify user that they have to choose the reason for cancel contract. | | 2 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * The inputted data from user is stored into database. * System update status of the contract to “Request cancel” * A notification will be sent to staff to notify about the cancel request, staff will approve to cancel the contract with customer. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 the 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.   Post Conditions:   * Success: Show the 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 page that show user list of transaction history information includes:   * Date of transaction: date * Method: text * Service: text * Amount: currency * Receiver: text * PayPal transaction ID: text   [Alternative 1]  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If there is no transaction were made | Show message to notify that there is no transaction were made. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * System will load transactions from database. * Transactions list is sorted by date order. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 the 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.   Post Conditions:   * Success: Show the 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 page show list of accidents information includes:   * Date: date input * Title: free text input * Attachment: file upload input   [Alternative 1]  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no accident | Show message to notify that there is no accident. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * System will load accidents from database. * Accidents list is sorted by date order. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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.   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: date * Punishment title: text * Punishment record: link   [Alternative 1]  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | There is no punishment | Show message to notify that there is no punishment. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * Punishments will be loaded from database. * Punishments list is sorted by date order. | | | |

###### <Customer> Request compensation

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Request compensation | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Customer.   Summary:   * This use case help 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. * User’s contract must be valid.   Post Conditions:   * Success: Store the compensation request to into database. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends request compensation command. | Display new page ask user to input required information includes:   * Driver name: input text free, required, length 1- 50. * License number: input text free, required, length 12. * License type: input text free, required, length 2. * Driver phone: input text free, required, length 10 - 20 * Vehicle capacity: number, required * Driver address: input text free, required, length 10 - 256 * Plate number of accident motor: input text free, required, length 3- 12 * Date of accident: date, required * Place of accident: input text free, required, length 10 - 256 * Control Police Department: input text free, required, length 5 - 256 * Description: input text free, required, length 10 – max. * Human damage: input text free, required, length 10 - 256 * Asset damage: input text free, required, length 10 - 256 * Observer: input text free, required, length 5 - 50 * Compensation note: input text free, max length 256. * Attachment file: button. | | 2 | User fill required information and attach the minutes of the accident (if any). Then send confirm command.  [Alternative 1] | Store information into database.  Show message to notify that request punishment successful.  [Exception 1]  [Exception 2] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User send cancel command. | Return to the compensation history page. |   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 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * Inputted data is stored into database. * System will notify staffs about new compensation request. * After insurance company resolve the compensation, staff will update information and resolve the compensation in the system. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 help 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.   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 page that show user the history of compensations 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: date * Place of accident: date * Control Police Department: text * Description: text * Human damage: text * Asset damage: text * Observer: text * Compensation note: text * Attachment: link * Created date: date * Resolve date: date * Decision: text * Resolve note: text   [Alternative 1]  [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If there is no compensation | Show message to notify that there is no compensation. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * Information is loaded from database. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 help 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.   Post Conditions:   * Success: New card request is stored into database. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends new card request command. | Display a new page shows user:   * A text box to confirm by password: password input, required, length 8-20 * Payment gateways:   + Direct payment: radio button  + PayPal: radio button   * The fee: currency | | 2 | User enter password and choose the PayPal payment gateway. Then send confirm command.  [Alternative 1] | Forward to PayPal payment process page.  [Exception 1] | | 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 2]  [Exception 3] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User enter password and choose the direct payment. Then send confirm command. | Show company information: address, telephone number. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | If user enter wrong password | Show message to notify that user has entered wrong password. | | 2 | If payment failed | Show message to notify that the payment failed. The request is aborted. |   Relationships:  Business Rules:   * At a time, user can only has one new card request for each contract. * System will notify to staffs about new card request. * When staff print new card for customer, the status of the request will be changed automatically. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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.   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 page shows the card information includes:   * Card ID: text * User name: text * Card status: text * Activated date: date   And a table shows list of access history information includes:   * Access date: date * Access device: text * Request: text * Response: text   [Exception 1] | |  |  |  |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * System loads all card information from database. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send view personal information command | Display new page shows the user information includes:   * User code: text * User name: text * Address: text * Email: text * Phone number: text * Personal ID: text   [Exception 1] | |  |  |  |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * System loads all personal information from database. | | | |

###### <Customer> Change password

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Change password. | | |
| Author | TriPQM | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:   * Customer.   Summary:   * This use case help user to change their password.   Goal:   * User can change the password.   Triggers:   * User sends change password command.   Preconditions:   * User must login into the system with role Customer.   Post Conditions:   * Success: Update new user’s password to database. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends change password command. | Display new page contains textboxes for user to input their current and new password:   * Current password: password input, length 8-20 * New password: password input, length 8-20 * Confirm new password: password input, length 8-20 | | 2 | User input their current password, new password and confirm new password. Then send confirm command. | Show message password changed successfully.  [Exception 1]  [Exception 2]  [Exception 3] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   Exceptions:   |  |  |  | | --- | --- | --- | | 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 invalid. | | 1 | Couldn’t connect to server. | Show message to notify that couldn’t connect to server. |   Relationships:  Business Rules:   * System will update new password for user into database if provided information is correct. | | | |

##### <Staff> Overview Use Case

###### <Staff> View profile

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | View profile | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Normal |
| Actor:   * Staff   Summary:   * This use case allow user view their profile.   Goal:   * User can view their detailed profile.   Triggers:   * User sends request to view profile.   Preconditions:   * User must login into the system with roles Customer of Staff.   Post Conditions:   * Success: User profile information is shown. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send requests to view profile. | User profile page will be shown with following information:   * User’s code: text * User’s full name: text * User’s address: text * User’s email: text |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * User’s detail profile is always loaded from database. * User’s code (username) must be highlighted. | | | |

###### <Staff> Change password

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Change password | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Normal |
| Actor:   * Staff   Summary:   * This use case allow user change their password.   Goal:   * User can change their account password.   Triggers:   * User sends change password command.   Preconditions:   * User must login into the system with roles Customer of Staff.   Post Conditions:   * Success: User’s new password is updated to database. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends change password command. | Change password page will be shown with following information:   * Current password: password input field, required, min length: 8, max length: 30 * New password: password input field, required, min length: 8, max length: 30 * Confirm new password: password input field, required, min length: 8, max length: 30 | | 2 | User fill out the form |  | | 3 | User submits their new password and approve to change  [Alternative 1] | * Validate data   [Exception 1, 2, 3, 4, 5, 6]   * Update user’s new password to database. * Display message notify user their password has changed successfully. * Write log file. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User abort this action. | Return to view profile page. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Current password field is empty | Show message notify user enter current password | | 2 | Current password is not match | Show message notify user entered current password is not match | | 3 | New password field is empty | Show message notify user enter new password | | 4 | Length of new password is not in range | Show message notify user entered new password is not a valid password | | 5 | Confirm new password field is empty | Show message notify user enter their new password again | | 6 | Confirm new password is not match with new password | Show message notify user entered confirm password is not match |   Relationships: Login  Business Rules:   * In case of successful scenario, user’s new password would be updated to database. * 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” and it has color for each value while displaying. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow 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.   Post Conditions:   * Success: List of new card request is shown and available to resolve. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends view and solve new card request command. | New card request page will be shown as a grid with following information:   * Ordinal number: positive integer, ascending * Date of request: dd/mm/yyyy * Note of customer: text * ID of lost card: link to card detail information page. * Name of card owner: link to page has detailed information of the customer who own this card. * Date a new card be issued for this customer: dd/mm/yyyy * ID of new card: link to card detail information page. | | 2 | Staff add new card for this customer | Update new card information.   * Date new card issued. * ID of new card |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * List of new card request is always loaded from database. * List of new card request is sorted by date of request in descending. * The lost card will changes status from “Ready” to “Deactivated”. * The new card will changes status from “Deactivated” to “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. * Allow staff select how many requests should be displayed in one page, default is 10 requests per page. * Staff can click on card ID link and customer’s full name link to view their detail information. * A solved request must have code of new card and the date this card be issued. | | | |

###### <Staff> Resolve compensation

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Resolve compensation request | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allow staff view and solve compensation request.   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.   Post Conditions:   * Success: List of new card request is shown and available to resolve. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends view and solve compensation request command. | Compensation request detail page will be shown with following information:   * Contract’s code: link to contract details page * Contract’s owner full name: link to customer details page * Driver’s full name: text * Driver’s address: text * Driver’s phone number: text * Driver’s license number: positive integer * Driver’s license type: text * Vehicle’s capacity or seat capacity: number * 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 * Note for compensation from staff: free input field. * Decision: select from a list. * Status of compensation: select from a list. | | 2 | Staff make decision and note about it | Update compensation status and notify to customer |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * List of compensation request is always loaded from database. * List of compensation request is sorted by date of request in descending. * A compensation request could have one of three decision is “Chưa quyết định”, “Chấp nhận bồi thường” and “Từ chối bồi thường”. * A compensation request could have one of three status is “Đang xử lý”, “Hoàn tất” and “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. * Allow staff select how many requests should be displayed in one page, default is 10 requests per page. * Staff can click on links to view its detail information. * A solved request must have solved date. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | View customer details | | |
| Author | KhaNC | | |
| Date | 20/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allow staff view customer details.   Goal:   * Staff can view customer’s detail information.   Triggers:   * Staff sends view customer information commands.   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 sends view customer information commands. | Customer details page 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 * Customer’s NFC card ID: link to detail information page. * 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:   * Ordinal number: positive integer, ascending * Customer’s contract code: link to contract detail page. * Customer’s contract type: text * Contract’s start date: dd/mm/yyyy * Contract’s expired date: dd/mm/yyyy * Contract’s status: text |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: Edit customer information.  Business Rules:   * Customer’s detail information is always loaded from database. * Separate customer’s information into 3 parts: personal information, customer NFC tag and customer’s contract for better vision. * Staff can clicks on NFC tag ID link and contract code link to view their detail information. * Contract’s status is colored by different colors. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow staff update 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. Log file is generated. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends update customer’s information command. | Edit customer page is shown with following labels and fields:   * Enter customer’s full name: free input field, required, min length: 3, max length: 80 * Enter customer’s address: free input field, required, min length: 5, max length: 100 * Enter customer’s email address: free input field, required, min length: 5, max length: 100 * Enter customer’s phone number: free input field, required, min length: 10, max length: 11 * Enter customer’s personal ID: free input field, length: 9 | | 2 | Staff fill out the form. |  | | 3 | Staff approve to update customer’s information.  [Alternative 1] | * Validate data   [Exception 1, 2, 3, 4, 5, 6, 7, 8, 9]   * Update customer’s information to database. * Refresh customer details page. * Write log file. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff abort this action. | Return to customer details screen. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Customer’s full name field is empty | Show message notify staff enter customer’s full name | | 2 | Length of customer’s full name is out of range | Show message notify staff entered customer’s full name is not valid | | 3 | Customer’s address field is empty | Show message notify staff enter customer’s address | | 4 | Length of customer’s address is out of range | Show message notify staff entered customer’s address is not valid | | 5 | Customer’s email address field is empty | Show message notify staff enter customer’s email address | | 6 | Customer’s email address field’s value is not a valid email | Show message notify entered email is not valid | | 7 | Customer’s phone number field is empty | Show message notify staff enter customer’s phone number | | 8 | Customer’s phone number field is not a valid phone number | Show message notify entered phone number is not valid | | 9 | Length of customer’s personal ID is different from 9 | Show message notify customer’s personal ID must contains 9 digit. |   Relationships: View customer details.  Business Rules:   * In case of success scenarios, customer new information would be updated to database. * Reloaded customer details page will display customer updated information. * An email address must be validated by this regular expression: (Add later) * All required fields must have the \* symbol belongs with its label. * Customer’s current information must be shown in fields. * Exception must not violate. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow staff create new customer.   Goal:   * A new customer is added to the system.   Triggers:   * Staff sends request to create new customer.   Preconditions:   * User must login into the system with role Staff.   Post Conditions:   * Success: New customer is added to the system. Log file is generated. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request to create new customer. | Create customer page is shown with following labels and fields:   * Enter customer’s full name: free input field, required, min length: 3, max length: 80 * Enter customer’s address: free input field, required, min length: 5, max length: 100 * Enter customer’s email address: free input field, required, min length: 5, max length: 100 * Enter customer’s phone number: free input field, required, min length: 10, max length: 11 * Enter customer’s personal ID: free input field, length: 9 | | 2 | Staff fill out the form. |  | | 3 | Staff approve to create new customer.  [Alternative 1] | * Validate data   [Exception 1, 2, 3, 4, 5, 6 , 7, 8, 9]   * Add new customer’s information to database. * Display create customer success page. * Write log file. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff abort this action | Return to customer management screen. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Customer’s full name field is empty | Show message notify staff enter customer’s full name | | 2 | Length of customer’s full name is out of range | Show message notify staff entered customer’s full name is not valid | | 3 | Customer’s address field is empty | Show message notify staff enter customer’s address | | 4 | Length of customer’s address is out of range | Show message notify staff entered customer’s address is not valid | | 5 | Customer’s email address field is empty | Show message notify staff enter customer’s email address | | 6 | Customer’s email address field’s value is not a valid email | Show message notify entered email is not valid | | 7 | Customer’s phone number field is empty | Show message notify staff enter customer’s phone number | | 8 | Customer’s phone number field is not a valid phone number | Show message notify entered phone number is not valid | | 9 | Length of customer’s personal ID is different from 9 | Show message notify customer’s personal ID must contains 9 digit. |   Relationships: N/A  Business Rules:   * In case of success scenarios, a new customer would be added to database. * An email address must be validated by this regular expression: (Add later) * All required fields must have the \* symbol belongs with its label. * Exception must not violate. | | | |

###### <Staff> Create contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow 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.   Post Conditions:   * Success: New contract is added to the system. Log file is generated. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends create new contract command. | Contract detail information page is shown with following label and fields:   * Enter customer’s full name: free input field, required, min length: 3, max length: 80 * Contract’s start date: date time picker, required * Contract’s expired date: date time picker, required * Contract’s sign date: date time picker, required * Contract’s sign place: free input field, required, min length: 5, max length: 100 * Name of staff signed contract: free input field, required, min length: 3, max length: 80 * Enter vehicle’s plate number: free input field, required, min length: 8, max length: 10 * Enter vehicle’s brand: free input field, required, min length: 2, max length: 30 * Enter vehicle’s model code: free input field, required, min length: 2, max length: 30 * Enter vehicle’s type: free input field, required, min length: 2, max length: 15 * Enter vehicle’s color: free input field, required, min length: 2, max length: 30 * Enter vehicle’s engine: free input field, required, length: 7 * Enter vehicle’s chassis: free input field, required, length: 7 * Enter vehicle’s capacity: free input field, required, min length: 2, max length: 3 * Enter vehicle’s year of manufacture: free input field, required, length: 4 * Enter vehicle’s empty weight: free input field, required, min length: 2, max length: 3 * Enter vehicle’s seat capacity: free input field, required, length: 2 * Image of vehicle registration certificate: free input field | | 2 | Staff fill out the form. | Display review contract’s information page. | | 3 | Staff approve to create this contract  [Alternative 1] | * Validate data   [Exception 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]   * Add new contract’s information to database. * Display create contract success page. * Write log file. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff abort this action | Return to contract management screen. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Customer’s full name field is empty | Show message notify staff enter customer’s full name | | 2 | Length of customer’s full name is out of range | Show message notify staff entered customer’s full name is not valid | | 3 | Contract’s start date field is empty | Show message notify staff pick contract’s start date | | 4 | Contract’s expired date field is empty | Show message notify staff pick contract’s expired date | | 5 | Contract’s sign date field is empty | Show message notify staff pick contract’s sign date | | 6 | Contract’s sign place field is empty | Show message notify staff enter contract’s sign place | | 7 | Length of contract’s sign place is out of range | Show message notify staff entered contract’s sign place is not valid | | 8 | Sign contract staff field is empty | Show message notify staff enter name of staff signed contract | | 9 | Length of name of staff signed contract is out of range | Show message notify staff entered name of staff sign is not valid | | 10 | Vehicle’s plate number field is empty | Show message notify staff enter vehicle’s plate number | | 11 | Length of vehicle’s plate number is out of range | Show message notify staff entered vehicle’s plate number is not valid | | 12 | Vehicle’s brand field is empty | Show message notify staff enter vehicle’s brand | | 13 | Length of vehicle’s brand is out of range | Show message notify staff entered vehicle’s brand is not valid | | 14 | Vehicle’s model code field is empty | Show message notify staff enter vehicle’s model code | | 15 | Length of vehicle’s model code is out of range | Show message notify staff entered vehicle’s model code is not valid | | 16 | Vehicle’s type field is empty | Show message notify staff enter vehicle’s type | | 17 | Length of vehicle’s type is out of range | Show message notify staff entered vehicle’s type is not valid | | 18 | Vehicle’s color field is empty | Show message notify staff enter vehicle’s color | | 19 | Length of vehicle’s color is out of range | Show message notify staff entered vehicle’s color is not valid | | 20 | Vehicle’s engine field is empty | Show message notify staff enter vehicle’s engine | | 21 | Length of vehicle’s engine is out of range | Show message notify staff entered vehicle’s engine is not valid | | 22 | Vehicle’s chassis field is empty | Show message notify staff enter vehicle’s chassis | | 23 | Length of vehicle’s chassis is out of range | Show message notify staff entered vehicle’s chassis is not valid | | 24 | Vehicle’s capacity field is empty | Show message notify staff enter vehicle’s capacity | | 25 | Length of vehicle’s capacity is out of range | Show message notify staff entered vehicle’s capacity is not valid | | 26 | Vehicle’s year of manufacture field is empty | Show message notify staff enter vehicle’s year of manufacture | | 27 | Length of vehicle’s year of manufacture is out of range | Show message notify staff entered vehicle’s year of manufacture is not valid | | 28 | Vehicle’s empty weight field is empty | Show message notify staff enter vehicle’s empty weight | | 29 | Length of vehicle’s empty weight is out of range | Show message notify staff entered vehicle’s empty weight is not valid | | 30 | Vehicle’s seat capacity field is empty | Show message notify staff enter vehicle’s seat capacity | | 31 | Length of vehicle’s seat capacity is out of range | Show message notify staff entered vehicle’s seat capacity is not valid |   Relationships: N/A  Business Rules:   * In case of success scenarios, a new contract would be added to database. * Staff can search and select a contract owner from available customers. * A new contract created successfully will has status is “Ready”. * Default contract term is 01 year. * All required fields must have the \* symbol belongs with its label. * Exception must not violate. | | | |

###### <Staff> Renew contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Renew contract | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allow staff renew contract.   Goal:   * Contract’s new expired date is updated to the system.   Triggers:   * Staff sends renew contract command.   Preconditions:   * User must login into the system with role Staff.   Post Conditions:   * Success: Contract is renewed. Log file is generated. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends renew contract command. | Renew contract detail information page is shown with following information:   * New contract’s start date: date time picker, required * New contract’s expired date: date time picker, required * Amount of paid renew contract fee: free input field, required, min length: 5, max length: 10 * Pay of charge date: date time picker, required * Description: free input field * Name of staff: free input field, required, min length: 3, max length: 80 | | 2 | Staff fill out the form. | Display review contract’s information page. | | 3 | Staff approve to renew this contract  [Alternative 1] | * Validate data   [Exception 1, 2, 3, 4, 5, 6, 7]   * Update contract’s information to database. * Reload contract detail page. * Write log file. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff abort this action | Return to contract management screen. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | New contract’s start date field is empty | Show message notify staff pick new contract’s start date | | 2 | New contract’s expired date field is empty | Show message notify staff pick new contract’s expired date | | 3 | Amount of paid renew contract fee field is empty | Show message notify staff enter amount of paid renew contract fee | | 4 | Length of amount of paid renew contract fee is out of range | Show message notify staff entered amount of paid renew contract fee is not valid | | 5 | Pay of charge date field is empty | Show message notify staff pick pay of charge date | | 6 | Sign contract staff field is empty | Show message notify staff enter name of staff signed contract | | 7 | Length of name of staff signed contract is out of range | Show message notify staff entered name of staff sign is not valid |   Relationships: N/A  Business Rules:   * In case of success scenarios, a new expired date of contract would be updated to database. * A contract renewed successfully will has status is “Ready”. * Default contract term is 01 year. * All required fields must have the \* symbol belongs with its label. * Exception must not violate. | | | |

###### <Staff> Cancel contract

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Cancel contract | | |
| Author | KhaNC | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Staff.   Summary:   * This use case allow 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.   Post Conditions:   * Success: Contract is cancelled. Log file is generated. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends cancel contract command. | Cancel contract detail information page is shown with following information:   * The reason why cancel contract: select from a list, required * Description: free input field * Name of staff: free input field, required, min length: 3, max length: 80 | | 2 | Staff fill out the form. | Display review contract’s information page. | | 3 | Staff approve to cancel this contract  [Alternative 1] | * Validate data   [Exception 1, 2, 3, 4, 5, 6, 7]   * Update contract’s information to database. * Reload contract detail page. * Write log file. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff abort this action | Return to contract management screen. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Reason to cancel contract field is empty | Show message notify staff select a reason to cancel contract | | 2 | Cancel contract staff field is empty | Show message notify staff enter name of staff cancelled contract | | 2 | Length of name of staff cancel contract is out of range | Show message notify staff entered name of staff cancelled is not valid |   Relationships: N/A  Business Rules:   * When a staff cancel a contract, other activities (fee refund, negotiations…) must be done outside of the system. No notification will be send to user when a staff cancels a contract. * A contract cancelled successfully will has status is “Cancel”. * All required fields must have the \* symbol belongs with its label. * Exception must not violate. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow staff to update contract information.  Goal:  Staff can update contract information.  Triggers:  Staff send command to update contract information.  Preconditions:   * User has to logged in to the system as Staff role * Contract is existed in database 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 navigates to update contract information page | System list out information of contract:  **Personal information**   * Name (required): free text input * Address (required): free text input * Email (required): free text input * Phone number (required): free text input * Personal ID: free text input   **Contract information** (all information below are required)   * Contract type: can be optional selected from these 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” * Start date: date input * Expired date: date input   **Vehicle information** (all information below are required)   * Plate: free text input * Brand: free text input * Model code: free text input * Vehicle type: free text input * Color: free text input * Engine: free text input * Chassis: free text input * Capacity: free text input * Year of manufacture: number text input * Weight: free text input * Seat capacity: free text input * Certificate image: file upload input | | 2 | Staff input information |  | | 3 | Staff send command to save new information. | System show message contract information is updated success.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to save new information. | System show 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 database. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow staff to update compensation information.  Goal:  Staff can update compensation information.  Triggers:  Staff send command to update compensation information.  Preconditions:   * User has to logged in to the system as Staff role * Compensation is existed in database   Post Conditions:   * Success: Compensation information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff navigates to update compensation information page | System list out information of compensation:   * Driver name: free text input * License number: free text input * License type: free text input * Driver phone: free text input * Vehicle capacity: free text input * Driver address: free text input * Plate: free text input * Date: date input * Place: free text input * Control department: free text input * Description: free text input * Human damage: free text input * Asset damage: free text input * Observer: free text input * Compensation note: free text input * Attachment: file upload input * Created date: date input * Resolve date: date input * Decision: free text input * Resolve note: free text input | | 2 | Staff input information |  | | 3 | Staff send command to save new information. | System show message compensation information is updated success.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to save new information. | System show error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Compensation information will be updated to database. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow staff to update punishment information.  Goal:  Staff can update punishment information.  Triggers:  Staff send command to update punishment information.  Preconditions:   * User has to logged in to the system as Staff role * Punishment is existed in database   Post Conditions:   * Success: Punishment information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff navigates to update punishment information page | System list out information of punishment:   * Date (required): date input * Title (required): free text input * Attachment (required): file upload input | | 2 | Staff input information |  | | 3 | Staff send command to save new information. | System show message compensation information is updated success.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to save new information. | System show error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Punishment information will be updated to database. * Punishment date must not exceed current date. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow staff to update accident information.  Goal:  Staff can update accident information.  Triggers:  Staff send command to update accident information.  Preconditions:   * User has to logged in to the system as Staff role * Contract of the accident is existed in database   Post Conditions:   * Success: Punishment information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff navigates to accident information page | System list out information of accident:   * Date (required): date input * Title (required): free text input * Attachment (required): file upload input | | 2 | Staff input information |  | | 3 | Staff send command to save new information. | System show message compensation information is updated success.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to save new information. | System show error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Accident information will be updated to database. * Accident date must not exceed current date. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow staff to view published cards information.  Goal:  Staff can view cards information.  Triggers:  Staff navigates to cards information page.  Preconditions:   * User has to logged in to the system as Staff role * Cards information is existed in database   Post Conditions:   * Success: Cards information is shown * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff navigates to cards information page | System list out information of published cards:   * Card ID: text link to card detail page * Card owner: text * Last access time: text * Status: text   [Exception 1] | | 2 | Staff send command to view card detail information | System show 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 navigates to cards information page | System show no connection error message. | | 2 | Staff send command to view card detail information | System show no connection error message. |   Relationships:  Business Rules:   * Cards information will be loaded from database. * History of access will show 10 results at a time, with pagination links at the bottom. | | | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **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 allow staff to update contract type.  Goal:  Staff can update contract type.  Triggers:  Staff send command to update contract type.  Preconditions:   * User has to logged in to the system as Staff role * Contract type is existed in database   Post Conditions:   * Success: Contract type information will be updated * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff navigates to contract type information page | System list out information of accident:   * Name (required): free text input * Description (required): free text input * Price per year (required): free number input | | 2 | Staff input information |  | | 3 | Staff send command to save new information. | System show message contract type information is updated success.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to save new information. | System show error message to ask staff input missing required fields. |   Relationships: N/A  Business Rules:   * Contract type information will be updated to database. * Contract type price per year must be greater than 0. | | | |

###### <Staff> Remove staff

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Remove staff | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:  Staff  Summary:  This use case allow staff to remove other staff from the system.  Goal:  Staff can remove other staff from the system.  Triggers:  Staff send command to remove other staff from the system.  Preconditions:   * User has to logged in to the system as Staff role   Post Conditions:   * Success: Staff be able to remove other staff from the system * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff navigates to manage members page | System list out information of staffs in system:   * Staff code: text * Name: text * Email: text * Phone number: text | | 2 | Staff select checkbox before the staff to delete |  | | 3 | Staff send command to delete other staff. | System show message staff has been removed successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to delete other staff. | System show error message user cannot remove themselves from the system. |   Relationships: N/A  Business Rules:   * Staff can not remove himself/herself from the system. | | | |

###### <Staff> Add staff

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MIC000 | | | |
| Use Case No. | MIC000 | **Use Case Version** | 2.0 |
| Use Case Name | Add staff | | |
| Author | TrungDQ | | |
| Date | 27/05/2015 | **Priority** | High |
| Actor:  Staff  Summary:  This use case allow staff to add other staff to the system.  Goal:  Staff can add other staff to the system.  Triggers:  Staff send command to add other staff to the system.  Preconditions:   * User has to logged in to the system as Staff role   Post Conditions:   * Success: Staff be able to add other staff to the system * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff navigates to add staff page | System require information of staff:   * Staff code (required): text * Name (required): text * Email (required): text * Phone number: text | | 2 | Staff input information |  | | 3 | Staff send command to add other staff. | System show message staff has been added successfully.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff send command to add other staff. | System show 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

###### Verify card validation

###### Get contract information

###### Update punishment info

##### <System> Overview Use Case

###### Notify contract state / expired

###### Notify compensation request state

#### Checker Mobile Application

##### <Police> Overview Use Case

###### Verify card information

###### Add punishment information

#### Printer Mobile Application

##### <Staff> Overview Use Case

###### Search / filter contract

###### View contract information

###### Print information to NFC card

## Software System Attribute

### Usability

### Reliability

### Availability

### Security

### Maintainability

### Portability

### Performance

## Conceptual Diagram