



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

Table of Contents.....	3
List of Tables	6
List of Figures	10
Definitions, Acronyms, and Abbreviations.....	13
A. Introduction.....	14
1. Project Information.....	14
2. Introduction	14
3. Current Situation.....	14
4. Problem Definition	14
5. Proposed Solution	15
5.1. Feature functions.....	15
5.2. Advantages and disadvantages	16
6. Functional Requirements	16
7. Roles and Responsibility	17
B. Software Project Management Plan	18
1. Problem Definition	18
1.1. Name of this Capstone Project	18
1.2. Problem Abstract.....	18
1.3. Project Overview	18
2. Project organization	21
2.1. Software Process Model.....	21
2.2. Roles and responsibilities	22
2.3. Tools and Techniques	23
3. Project Management Plan	23
3.1. Software development life cycle.....	23
3.2. Phase Detail	26
3.3. Task sheet.....	27
3.4. All Meeting Minutes.....	27
4. Coding Convention	27
C. Software Requirement Specification	29
1. User Requirement Specification.....	29
1.1. Customer requirement.....	29
1.2. Staff requirement	29

1.3. Police requirement	30
1.4. Admin requirement.....	30
2. System Requirement Specification	30
2.1. External Interface Requirement.....	30
2.2. System Overview Use Case	30
2.3. List of Use Case.....	34
3. Software System Attribute	100
3.1. Usability	100
3.2. Reliability.....	100
3.3. Availability	100
3.4. Security.....	100
3.5. Maintainability	100
3.6. Portability.....	100
3.7. Performance	100
4. Conceptual Diagram.....	101
D. Software Design Description.....	104
1. Design Overview	104
2. System Architecture Design.....	104
2.1. Web Application architecture description	104
2.2. Mobile Application architecture description.....	105
3. Component Diagram.....	106
4. Detailed Description	107
4.1. Class Diagram	107
4.2. Class Diagram Explanation.....	109
4.3. Interactive Diagram.....	116
5. Interface.....	129
5.1. Component Interface	129
5.2. Web application Design	131
5.3. Checker Mobile Application Design	146
5.4. Printer Mobile Application Design	150
6. Database Design	153
6.1. Entity relationship diagram	153
6.2. Entity Dictionary.....	155
7. Algorithms	155
7.1. Contract State.....	155

7.2.	Concurrency Control.....	156
7.3.	Notification	159
7.4.	System Scheduler Process	159
7.5.	NFC Card Data Format.....	161
7.6.	Strategy for future expand plan.....	164
E.	System Implementation & Test.....	169
1.	Introduction.....	169
1.1.	Overview.....	169
1.2.	Test Approach.....	169
2.	Database Relationship Diagram.....	169
2.1.	Physical Diagram.....	169
2.2.	Data Dictionary	171
3.	Performance Measures	177
3.1.	Web application page load speed.....	177
3.2.	Mobile application API load speed	180
4.	Test Plan	182
4.1.	Features to be tested.....	182
4.2.	Features not to be tested.....	182
5.	System Testing Test Case.....	182
5.1.	Communication Diagrams.....	182
5.2.	Test cases.....	184
5.3.	Test case results statistics.....	207
F.	Software User's Manual.....	209
1.	Installation Guide.....	209
1.1.	Setting up environment at server side.....	209
1.2.	Web Application Deployment Process.....	209
1.3.	Mobile Application Deployment Process	211
2.	User Guide.....	213
2.1.	Web Application	213
2.2.	Checker Application	257
2.3.	Printer Application	259
G.	Appendix	261

List of Tables

Table 1: Definitions, Acronyms, and Abbreviations	13
Table 2 Roles and Responsibility	17
Table 3 Hardware requirement for continuous integrating server	21
Table 4 Hardware requirement for web development	21
Table 5 Hardware requirement for mobile development.....	21
Table 6 Software requirement.....	21
Table 7 Roles and responsibilities.....	23
Table 8 Tools and Techniques.....	23
Table 9 Software development life cycle	25
Table 10 Use case WG01 - <Guest> Login.....	35
Table 11 Use case WG02 - <Guest> Create new contract request	38
Table 12 Use case WC01 - <Customer> Logout.....	40
Table 13 Use case WC02 - <Customer> Renew contract by user	42
Table 14 Use case WC03 - <Customer> Cancel contract.....	43
Table 15 Use case WC04 - <Customer> View payment history.....	44
Table 16 Use case WC05 - <Customer> View accident history.....	45
Table 17 Use case WC06 - <Customer> View punishment history.....	47
Table 18 Use case WC07 - <Customer> Request compensation.....	49
Table 19 Use case WC08 - <Customer> View compensation history	50
Table 20 Use case WC09 - <Customer> New card request.....	52
Table 21 Use case WC10 - <Customer> View card information	53
Table 22 Use case WC11 - <Customer> View personal information	54
Table 23 Use case WC12 - <Customer> Change password	56
Table 24 Use case WS01 - <Staff> View profile	59
Table 25 Use case WS02 - <Staff> Change password.....	60
Table 26 Use case WS03 - <Staff> Resolve new card request.....	62
Table 27 Use case WS04 - <Staff> Resolve compensation request.....	64
Table 28 Use case WS05 - <Staff> View customer information	66
Table 29 Use case WS06 - <Staff> Edit customer information.....	67
Table 30 Use case WS07 - <Staff> Create new customer.....	69
Table 31 Use case WS08 - <Staff> Create new contract.....	71
Table 32 Use case WS09 - <Staff> Renew contract.....	73
Table 33 Use case WS10 - <Staff> Cancel contract	74
Table 34 Use case WS11 - <Staff> Update contract information.....	76
Table 35 Use case WS12 - <Staff> Update compensation history.....	78
Table 36 Use case WS13 - <Staff> Update punishment history	79
Table 37 Use case WS14 - <Staff> Update accident history	80
Table 38 Use case WS15 - <Staff> View cards information	82
Table 39 Use case WS16 - <Staff> Update contract type information.....	83
Table 40 Use case WA01 - <Admin> Remove staff	85
Table 41 Use case WA02 - <Admin> Add staff.....	86
Table 42 Use case WY01 - <System> Notify schedule	88
Table 43 Use case WP01 - <Payment> Payment.....	90

Table 44 Use case CP01 - <Police> Verify card information	93
Table 45 Use case CP02 - <Police> Add punishment information.....	94
Table 46 Use case PS01 - <Staff> Search / filter contract	96
Table 47 Use case PS02 - <Staff> View contract information.....	98
Table 48 Use case PS03 - <Staff> Print information to NFC card	100
Table 49 Conceptual Diagram Data Dictionary	103
Table 50 Component Dictionary	106
Table 51 Class dictionary	109
Table 52 PaymentEntity Attribute	109
Table 53 PaymentEntity Method	110
Table 54 CardEntity Attribute.....	110
Table 55 CardEntity Method.....	110
Table 56 CardInstanceEntity Attribute	110
Table 57 CardInstanceEntity Method	110
Table 58 CustomerEntity Attribute.....	111
Table 59 CustomerEntity Method.....	111
Table 60 ContractEntity Attribute	112
Table 61 Contract Method	112
Table 62 StaffEntity Attribute	112
Table 63 StaffEntity Method	112
Table 64 CompensationEntity Attribute	113
Table 65 CompensationEntity Method	113
Table 66 PunishmentEntity Attribute.....	113
Table 67 PunishmentEntity Method.....	114
Table 68 AccidentEntity Attribute.....	114
Table 69 AccidentEntity Method.....	114
Table 70 ContractTypeEntity Attribute.....	114
Table 71 ContractTypeEntity Method.....	114
Table 72 NewCardRequestEntity Attribute	115
Table 73 NewCardRequestEntity Method	115
Table 74 CardAccessLogEntity Attribute.....	115
Table 75 CardAccessLogEntity Method.....	115
Table 76 NotificationEntity Attribute	116
Table 77 NotificationEntity Method	116
Table 78 NotificationReadEntity Attribute	116
Table 79 NotificationReadEntity Method	116
Table 80 Web Services Interfaces.....	129
Table 81 Output format description	130
Table 82 Exception description.....	130
Table 83 Staff home page fields.....	131
Table 84 Staff home page buttons/hyperlinks	132
Table 85 <Staff> Manage customer fields.....	133
Table 86 <Staff> Manage customer buttons/hyperlinks	133
Table 87 <Staff> Create customer fields	134
Table 88 <Staff> Create customer buttons/hyperlinks.....	134
Table 89 <Staff> Manage contracts fields.....	135

Table 90 <Staff> Manage contracts buttons/hyperlinks	135
Table 91 <Staff> Create contract fields.....	137
Table 92 <Staff> Create contract buttons/hyperlinks.....	138
Table 93 <Staff> Renew contract fields.....	139
Table 94 <Staff> Renew contract buttons/hyperlinks	139
Table 95 <Staff> Cancel contract fields	140
Table 96 <Staff> Cancel contract buttons/hyperlinks.....	140
Table 97 <Customer> Manage contract fields	141
Table 98 <Customer> Manage contract buttons/hyperlinks.....	141
Table 99 <Customer> Contract detail fields	142
Table 100 <Customer> Contract detail buttons/hyperlinks.....	143
Table 101 <Customer> Cancel contract fields	143
Table 102 <Customer> Cancel contract button/hyperlinks	143
Table 103 <Customer> Renew contract fields.....	144
Table 104 <Customer> Renew contract button/hyperlinks	144
Table 105 <Customer> Select payment method fields.....	145
Table 106 <Customer> Select payment method buttons/hyperlinks	145
Table 107 Scan NFC card screen - Fields	146
Table 108 Scan NFC card screen - Buttons.....	146
Table 109 View card information screen - Buttons.....	148
Table 110 Search contract screen - Fields.....	150
Table 111 Search contract screen - Buttons.....	151
Table 112 View contract screen - Buttons	152
Table 113 Entity dictionary	155
Table 114 Contract State Dictionary.....	156
Table 115 Contract State Flow.....	156
Table 116 Notification use cases.....	159
Table 117 NDEF Message Types	163
Table 118 Data table dictionary	172
Table 119 Data attribute dictionary	177
Table 120 Web application page load speed test result	179
Table 121 Mobile app API load speed client specification.....	180
Table 122 Mobile app API load speed test cases	180
Table 123 Mobile application API load speed test result	181
Table 124 Test case - <Guest> Register new contract.....	187
Table 125 Test case - <Guest> Register new contract payment.....	188
Table 126 Test case - <Staff> View contract.....	189
Table 127 Test case - <Staff> Create contract.....	190
Table 128 Test case - <Staff> Renew contract.....	193
Table 129 Test case - <Staff> Cancel contract	194
Table 130 Test case - <Staff> View compensation list.....	194
Table 131 Test case - <Staff> View compensation detail	195
Table 132 Test case - <Staff> Resolve compensation	196
Table 133 Test case - <Staff> View new card requests	196
Table 134 Test case - <Staff> Resolve new card request.....	197
Table 135 Test case - <Customer> Create contract.....	198

Table 136 Test case - <Customer> Renew contract	200
Table 137 Test case - <Customer> Cancel contract.....	201
Table 138 Test case - <Customer> Send compensation request.....	203
Table 139 Test case - <Customer> Send new card request.....	206
Table 140 Test case results statistics.....	208
Table 141 Hardware requirements	209
Table 142 Software requirements.....	209

List of Figures

Figure 1: Waterfall model	22
Figure 2: Web Application Overview Use Case.....	31
Figure 3: Checker Mobile Application.....	33
Figure 4: Printer Mobile Application.....	33
Figure 5 <Guest> Overview Use Case.....	34
Figure 6 <Guest> Login	34
Figure 7 <Guest> Create new contract request.....	35
Figure 8 <Customer> Overview Use Case.....	39
Figure 9 <Customer> Renew contract	40
Figure 10 <Customer> Cancel contract	42
Figure 11 <Customer> View payment history.....	43
Figure 12 <Customer> View accident history.....	44
Figure 13 <Customer> View punishment history	46
Figure 14 <Customer> Request compensation	47
Figure 15 <Customer> View compensation history	49
Figure 16 <Customer> New card request.....	50
Figure 17 <Customer> View card information.....	52
Figure 18 <Customer> View personal information	53
Figure 19 <Customer> Change password.....	55
Figure 20 <Staff> Overview Use Case.....	57
Figure 21 <Staff> View profile	58
Figure 22 <Staff> Change password	59
Figure 23 <Staff> Resolve new card request.....	61
Figure 24 <Staff> Resolve compensation request.....	62
Figure 25 <Staff> View customer information	64
Figure 26 <Staff> Edit customer information	66
Figure 27 <Staff> Create new customer	67
Figure 28 <Staff> Create contract.....	69
Figure 29 <Staff> Renew contract	71
Figure 30 <Staff> Cancel contract.....	73
Figure 31 <Staff> Update contract information	74
Figure 32 <Staff> Update compensation history	76
Figure 33 <Staff> Update punishment history	78
Figure 34 <Staff> Update accident history.....	79
Figure 35 <Staff> View card information	80
Figure 36 <Staff> Update contract type information	82
Figure 37 <Admin> Overview Use Case	82
Figure 38 <Admin> Remove staff	84
Figure 39 <Admin> Add staff.....	85
Figure 40 <System> Overview Use Case	86
Figure 41 <System> Notify schedule	87
Figure 42 <Payment> Payment	88
Figure 43 <Payment> Payment	89

Figure 44 <Police> Overview Use Case.....	91
Figure 45 <Police> Verify card information	91
Figure 46 <Police> Add punishment information.....	93
Figure 47 <Staff> Overview Use Case.....	95
Figure 48 <Staff> Search / filter contract	95
Figure 49 <Staff> View contract information.....	97
Figure 50 <Staff> Print information to NFC card.....	98
Figure 51 Conceptual diagram.....	102
Figure 52 System architecture design.....	104
Figure 53 Component Diagram.....	106
Figure 54 Class Diagram.....	108
Figure 55 Sequence diagram - <Staff> Create new contract	117
Figure 56 Sequence diagram - <Staff> Renew contract	118
Figure 57 Sequence diagram - <Staff> Cancel contract.....	119
Figure 58 Sequence diagram - <Customer> Cancel contract	120
Figure 59 Sequence diagram - <Customer> Renew contract.....	121
Figure 60 Sequence diagram - <Guest> Register new contract.....	122
Figure 61 Sequence diagram - <Guest> PayPal payment.....	123
Figure 62 <Police> Verify card validation	124
Figure 63 <Police> Add punishment information.....	125
Figure 64 <Staff> Search contract	126
Figure 65 <Staff> View contract information.....	127
Figure 66 <Staff> Print information to NFC card.....	128
Figure 67 Interface - <Staff> Home page	131
Figure 68 Interface - <Staff> Manage customer	132
Figure 69 Interface - <Staff> Create customer.....	133
Figure 70 Interface - <Staff> Manage contract	134
Figure 71 Interface - <Staff> Create contract.....	136
Figure 72 Interface - <Staff> Renew contract	138
Figure 73 Interface - <Staff> Cancel contract.....	139
Figure 74 Interface - <Customer> Manage contract.....	140
Figure 75 Interface - <Customer> Contract detail	142
Figure 76 Interface - <Customer> Cancel contract	143
Figure 77 Interface - <Customer> Renew contract.....	144
Figure 78 Interface - <Customer> Select payment method.....	145
Figure 79 Scan NFC card screen	146
Figure 80 View card information screen	147
Figure 81 Search contract screen	150
Figure 82 View contract screen	151
Figure 83 Print card screen.....	152
Figure 84 Entity relationship diagram.....	154
Figure 85 Contract State Chart.....	155
Figure 86 Concurrency control flow	158
Figure 87 System Scheduler Process.....	161
Figure 88 NDEF message structure	162
Figure 89 Isolated Single Service Deployment Model	164

Figure 90 Entities group by components (Isolated Single Service)	165
Figure 91 Isolated Multiple Service Deployment Model.....	165
Figure 92 Entities group by components (Isolated Multiple Service).....	166
Figure 93 Distributed Multiple Service Deployment Model.....	167
Figure 94 Components group by container (Distributed Multiple Service)	168
Figure 95 Physical diagram.....	170
Figure 96 Web application page load speed test result	179
Figure 97 Mobile application API load speed test result	181
Figure 98 Web Application Communication Diagram.....	183
Figure 99 Mobile Applications Communication Diagram	183

Definitions, Acronyms, and Abbreviations

Name	Definition
MIC	Motor Insurance Card
NFC	Near field communication

Table 1: Definitions, Acronyms, and Abbreviations

A. Introduction

1. Project Information

- Project name: **Insurance Card**
- Project Code: **MIC**
- Product Type: **Website & Android Application**
- Start Date: **May 11th, 2015**
- End Date: **August 24th, 2015**

2. 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.

3. Current Situation

When participating in traffic, vehicle owners are required to have compulsory insurance (according to Article 6, Decree on compulsory insurance for civil liability of motor vehicle owners, Decree No. 103/2008/NĐ-CP by Vietnam Government). 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.

4. 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.
- **No mechanism to renew old contract:** customers have to handy register new contract when the old one is expired, this is inconvenient for customers.
- **Insurance certificate made of paper:** It could be torn, wet, smudged and especially is counterfeited.
- **Claim/compensation process is ineffective** between customer and insurance company.
- **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.

According to Vietnam's laws, motor vehicle owners must have insurance contract with fixed term and fixed fee for each type of vehicle.

5. Proposed Solution

Our proposed solution is to build an insurance NFC card system named "MIC system" to resolve the current situations and compatible with current laws, we also design the system to be scalable so we can deploy this system to a multiple insurance services company in future plan.

MIC system includes a web application and two mobile applications with following functions:

5.1. Feature functions

- Web application:
 - **Register insurance:** user can register a new insurance card with on website using online payment. A staff will contact the user to create contract and sends an insurance NFC card to him/her. If users already have a NFC card, they can use the website to renew current contract.
 - **Check card information:** user can login into the website and check for their card's information.
 - **Request compensation:** user can fill data into the sample fields and sends compensation request to the company.
 - **Make/manage contracts:** staff can make and manage contracts.
 - **Resolve compensation:** staff can receive and resolve compensation requests.
 - **Notify contract state:** system will 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.

5.2. 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. Moreover, it is difficult to be counterfeit than insurance card made of paper.
 - Support police to check valid insurance card easier.
- Disadvantages:
 - Currently 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.

6. Functional Requirements

Function requirements of the system are listed as below:

- **User component:**
 - New contract request
 - Check card information.
 - Renew contract.
 - Request compensation.
 - New card request
 - Cancel contract
- **Staff component**
 - Create new contracts
 - Manage contracts.
 - Resolve compensation requests.
 - Resolve new 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.

7. Roles and Responsibility

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

Table 2 Roles and Responsibility

B. Software Project Management Plan

1. Problem Definition

1.1. Name of this Capstone Project

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

1.2. 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. Insurance company supplies the NFC card when the customer buy insurance. The card contains the information of customer, if the customer joins with many insurance service they just use only one card.

We provide a software to check the validation of card, the expired date of card and insurance services that customer joined. We also provide other advantages 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.

1.3. Project Overview

1.3.1. Current Situation

Below are the problems encountered in this project:

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

1.3.2. 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.

1.3.2.1.Website

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

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

1.3.2.2.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.

1.3.2.3. Checker Mobile Application

This is a simulating application to simulate the work of Card Checker Device for traffic police officers. In practical situation this application will connect to the center database from police department to get information about vehicle. In this project we simulate this device as an application run on an Android smartphone.

Traffic police officer uses this mobile application. This application does 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.

1.3.3. Boundaries of the System

This section supposes that the government laws 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:
 - o Computer system with internet connection.
 - o Smartphone devices with built-in NFC technology.
- The language of this system is Vietnamese
- The complete product includes:
 - o Website application for staffs and users
 - o Printer mobile application for staffs.
 - o Checker mobile application for traffic police officers

1.3.4. Future plans

Current system only can deploy to a company which provides single service: motor insurance card, we call this is the **Isolated Single Service Model**. We design the system to make it easy to scale to 2 bigger models:

- **Isolated Multiple Service Model:** system can be deployed to one company which provide multiple insurance services such as motor insurance, health insurance, assets insurance...
- **Distributed Multiple Service Model:** system can be deployed to multiple companies; each company provides multiple insurance services. The companies all share information through a center system to maintain the synchronization between companies.

1.3.5. Development Environment

1.3.5.1.Hardware requirement

- For continuous integrating server:

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

Table 3 Hardware requirement for continuous integrating server

- For web development:

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

Table 4 Hardware requirement for web development

- For mobile development:

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

Table 5 Hardware requirement for mobile development

1.3.5.2.Software requirement

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

Table 6 Software requirement

2. Project organization

2.1. Software Process Model

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

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

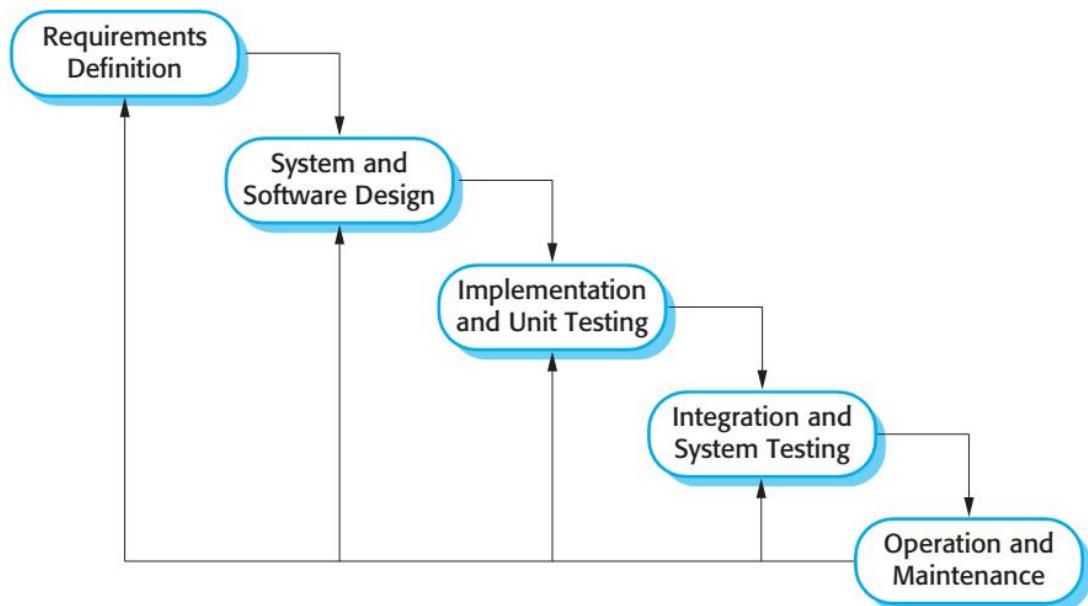


Figure 1: Waterfall model

Reference: Page 30, chapter 2, Software process model, 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.

2.2. Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Kiều Trọng Khánh	Supervisor / Project Manager	<ul style="list-style-type: none"> - Clarify user requirement. - Technical support and business analysis. - Tracking development process. - Review document and product.

2	Đinh Quang Trung	Team leader, BA, Developer, Tester	<ul style="list-style-type: none"> - Tracking process. - Planning project, distribute tasks. - Requirement analysis. - Database design. - Documentation. - GUI Design. - Coding. - Testing. - Deploy product.
3	Nguyễn Hữu Phúc	BA, Developer, Tester	<ul style="list-style-type: none"> - Requirement analysis. - Database design. - Documentation. - GUI Design. - Coding. - Testing.
4	Phùng Quang Minh Trí	BA, Developer, Tester	<ul style="list-style-type: none"> - Requirement analysis. - Database design. - Documentation. - GUI Design. - Coding. - Testing.
5	Nguyễn Chí Kha	BA, Developer, Tester	<ul style="list-style-type: none"> - Requirement analysis. - Database design. - Documentation. - GUI Design. - Coding. - Testing.

Table 7 Roles and responsibilities

2.3. 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	StarUML 5.0, Lucid Chart
Document tool	Microsoft Word 2013

Table 8 Tools and Techniques

3. Project Management Plan

3.1. 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 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	<ul style="list-style-type: none"> - Deploy the system - Create the user's manuals. - Do routine maintenance activities. 	Report Software Manual	No.6 User's	15 man-days	Integration and System Testing are finished.	User's manual may be difficult for user to understand and confuse.
----------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------	-------------	-------------	----------------------------------------------	--------------------------------------------------------------------

Table 9 Software development life cycle

3.2. Phase Detail

3.2.1. Phase 1: Requirements Definition

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

3.2.2. 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.	<ul style="list-style-type: none"> - 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.	<ul style="list-style-type: none"> - Design database for the system. 	Đinh Quang Trung Nguyễn Hữu Phúc Phùng Quang Minh Trí Nguyễn Chí Kha

3.2.3. 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.	<ul style="list-style-type: none"> - Write Unit test cases. - Implement Unit tests. 	Đinh Quang Trung Nguyễn Hữu Phúc Phùng Quang Minh Trí Nguyễn Chí Kha

3.2.4. 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

3.2.5. 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

3.3. Task sheet

Refer to “Task sheet” folder.

3.4. All Meeting Minutes

Refer to “Meeting minutes” folder.

4. 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 four 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.

C. Software Requirement Specification

1. User Requirement Specification

1.1. Customer requirement

Customer is user who uses service of system. The customer can use some following functions:

- View history information include:
 - o View payment history
 - o View accident history
 - o View punishment history
- View compensation history
- View contract information
- View personal information
- Create new contract
- New card request
- Renew contract
- Cancel contract
- Request compensation

1.2. Staff requirement

Staff is people who works directly with system to track the information of customer or manages customer, staff can handle directly some problems if it happens from customer. Staff can use some following functions:

- View profile: they can change password of customer
- View customer information
- View card information
- Resolve new card request
- Resolve compensation requests
- Update contract type information
- Manage customer includes:
 - o View customer information
 - o Edit customer information
 - o Create new customer
- Manage contracts includes:
 - o Create new contract
 - o Update contract information
 - o Renew contract
 - o Cancel contract
 - o Update compensation history
 - o Update punishment history
 - o Update accident history
- Print NFC card for customer

1.3. Police requirement

Police is people who is interactive with system for checking information about customer's NFC card and handling in case the customer violates the traffic rule or make accident. Police can use some following functions:

- Verify card validation
- Get contract information
- Update punishment information

1.4. Admin requirement

Admin is people who manages staff. Administrator can use some following functions:

- Manage staff includes:
 - o Remove staff
 - o Add staff

2. System Requirement Specification

2.1. External Interface Requirement

2.1.1. User interface

- The user interface uses Vietnamese language.
- Use consistent palette of colors between the text and the background.
- The user interface displays best on 1024x768-screen size.

2.1.2. Hardware Interface

- Smartphone with NFC support.

2.1.3. Software Interface

- Web application: work with Firefox (v30 or above), Chromes (v14 or above), Internet Explorer (v10 or above) browse.
- Mobile application: Android operating system (v 4.0 or above).

2.1.4. Communication Protocol

- Use HTTP protocol 1.1 for communication between the web browser and the web server.
- Use HTTP protocol 1.1 for communication between the mobile application and the web service.

2.2. System Overview Use Case

2.2.1. Web Application

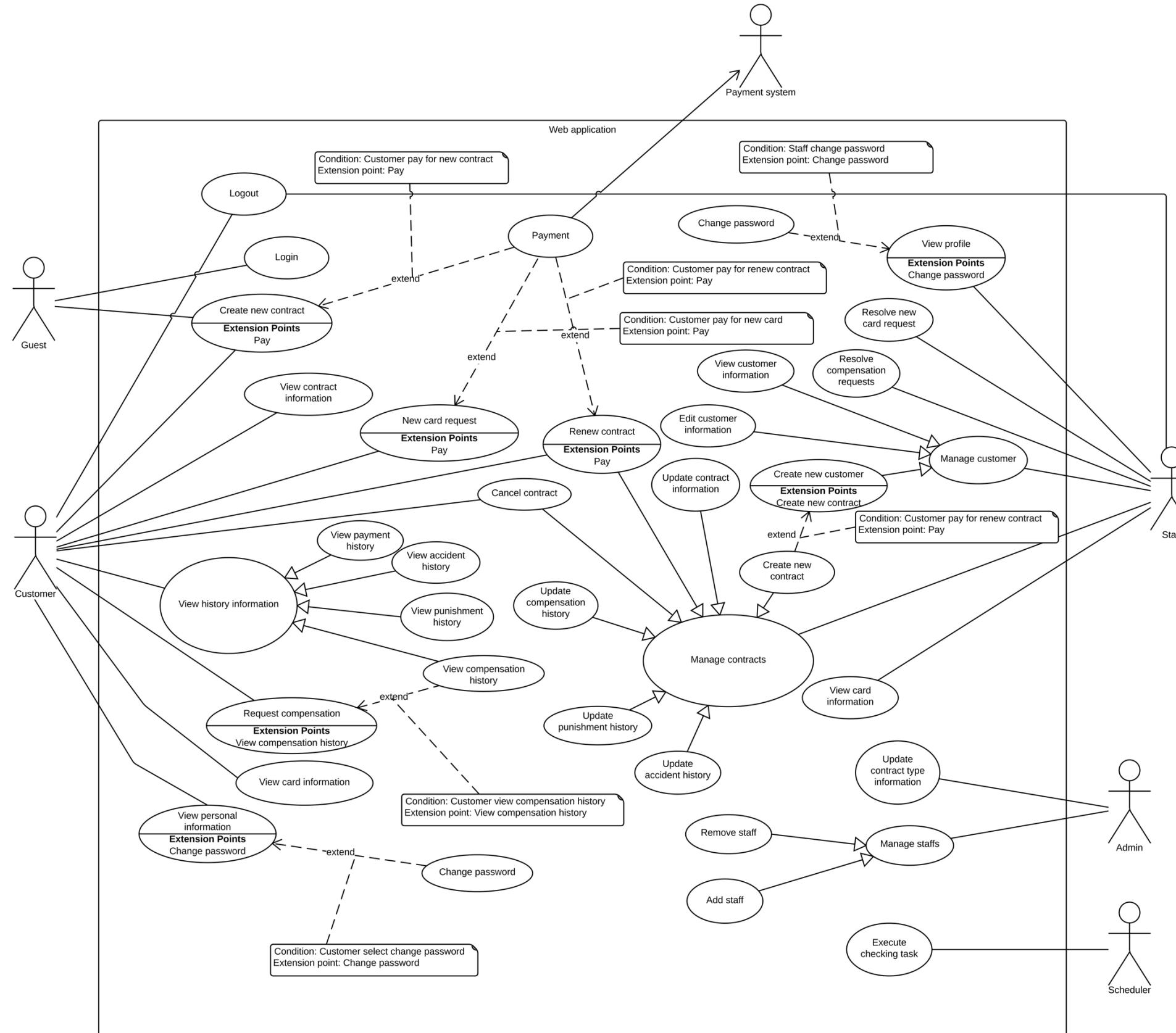


Figure 2: Web Application Overview Use Case

2.2.2. Checker Mobile Application

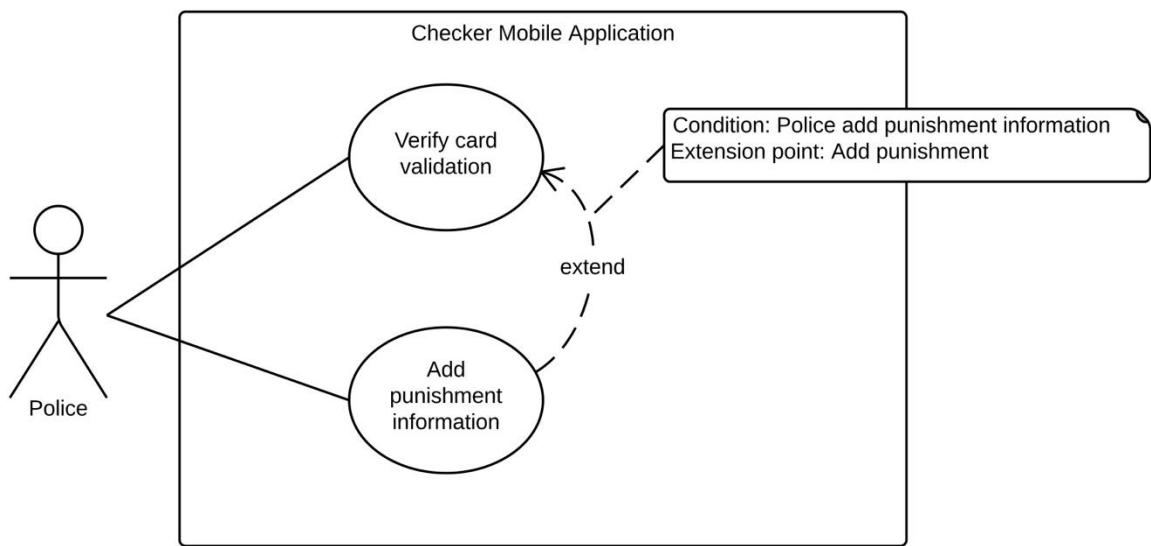


Figure 3: Checker Mobile Application

2.2.3. Printer Mobile Application

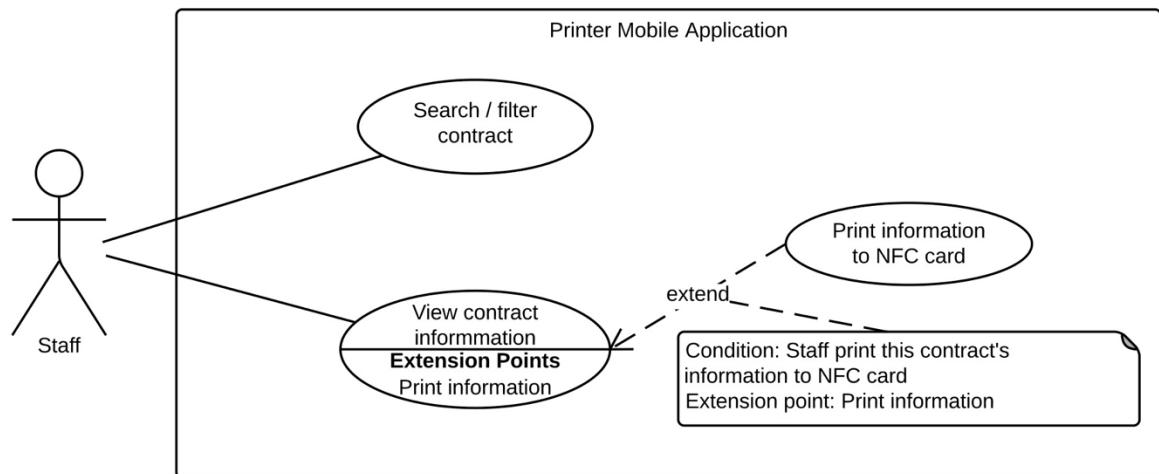


Figure 4: Printer Mobile Application

2.3. List of Use Case

2.3.1. Web Application

2.3.1.1.<Guest> Overview Use Case

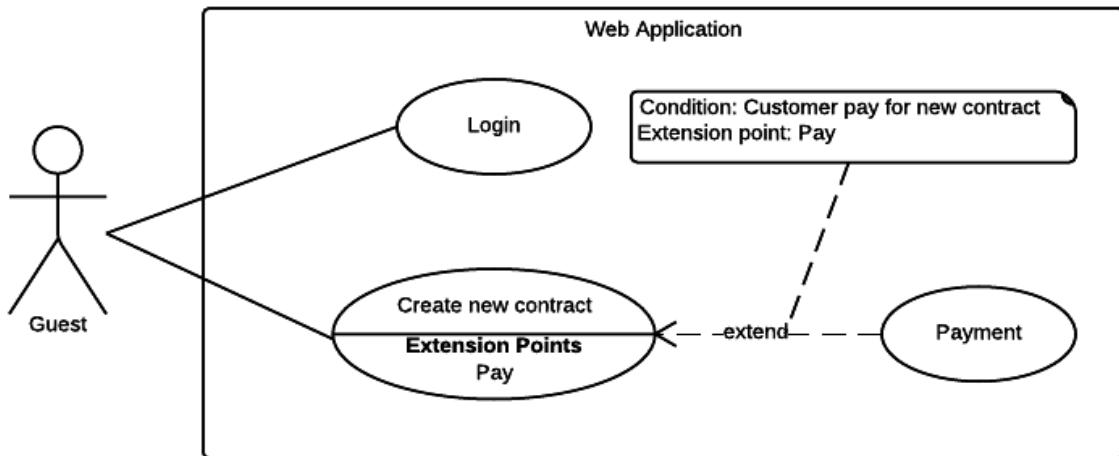


Figure 5 <Guest> Overview Use Case

2.3.1.1.1. <Guest> Login

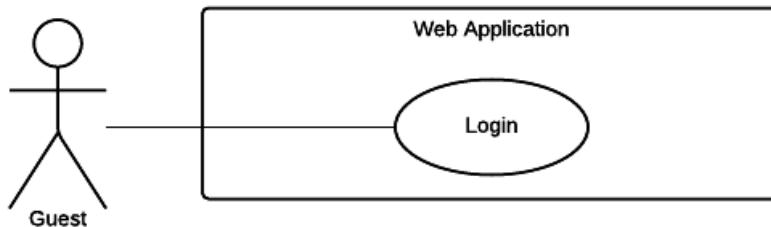


Figure 6 <Guest> Login

USE CASE - WG01			
Use Case No.	WG01	Use Case Version	2.0
Use Case Name	Login		
Author	TrungDQ	Priority	Normal
Date	27/05/2015	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Guest 		
Summary:	<ul style="list-style-type: none"> - This use case allows guest to log in the system. 		
Goal:	<ul style="list-style-type: none"> - Guest can log in the system. 		
Triggers:	<ul style="list-style-type: none"> - Guest sends the login command. 		
Preconditions:	<ul style="list-style-type: none"> - N/A 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Guest login the system. - Fail: Show error message. 		

Main Success Scenario:

Step	Actor Action	System Response
1	Guest goes to login view.	System requires identity information from Guest: - Email or customer code: free text input, required, length 3 – 250 - Password: free text input, required, length 6 – 32
2	Guest inputs information.	
3	Guest sends command to login to system	Guest will login system with their specific role [Alternative 1] [Exception 1]

Alternative Scenario:

Step	Actor Action	System Response
1	Guest enter wrong identity information.	Wrong identity information, System shows error message.

Exceptions:

Step	Actor Action	System Response
1	Missing of required fields	System notify guest to enter their identity information

Relationships: N/A

Business Rules:

- Password are encrypted before being sent to server.
- After login to system, guest will be redirected to specific view based on their role on the system: staff or customer.
 - o If role is “Customer”, the system will display to Customer view.
 - o If role is “Staff”, the system will display to Staff Dashboard view.

Table 10 Use case WG01 - <Guest> Login

2.3.1.1.2. <Guest> Create new contract request

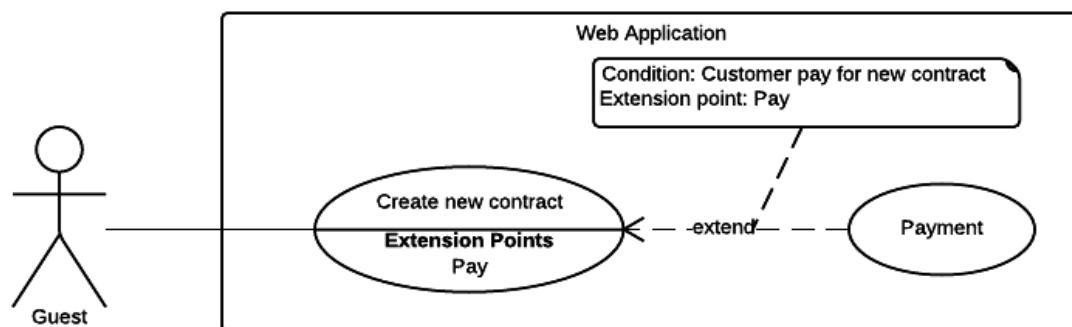


Figure 7 <Guest> Create new contract request

USE CASE – WG02

Use Case No.	WG02	Use Case Version	2.0
Use Case Name	Create new contract request		
Author	TrungDQ		
Date	27/05/2015	Priority	Normal

Actor:

- Guest

Summary:

- This use case allows guest to create new contract request.

Goal:

- Guest can create new contract request.

Triggers:

- Guest sends command to create contract request.

Preconditions:

- N/A

Post Conditions:

- **Success:** New account and new contract will be created for guest.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest goes to new contract view.	<p>System requires information from guest:</p> <p>Personal information</p> <ul style="list-style-type: none"> - Name: free text input, required, length 3 – 80. - Address: free text input, required, length 3 – 250. - Email: free text input, required, length 3 – 250. - Phone number: free text input, required, length 8 – 15. - Personal ID: free text input, length 8 – 15. <p>Contract information (all information below are required)</p> <ul style="list-style-type: none"> - Contract's type: select one of the options. - Start date: date time input, required. - Contract term: text - Contract's fee: text <p>Vehicle information</p> <ul style="list-style-type: none"> - Plate: free text input, required, length 4 – 15. - Brand: free text input, required, length 2 – 20. - Model code: free text input, length 2 – 20. - Vehicle type: free text input, length 2 – 20. - Color: free text input, length 2 – 20. - Engine: free text input, required, length 2 – 20. - Chassis: free text input, required, length 2 – 20.

		<ul style="list-style-type: none"> - Capacity: free text input, required, length 2 – 20. - Year of manufacture: number text input, value from 1900 to current year. - Weight: free text input, value from 1 – 1000, unit: kilogram - Seat capacity: free text input, value from 1 – 100. <p>Security question</p> <ul style="list-style-type: none"> - Answer: free text input, required, length 1 - 10
2	Guest inputs information.	
3	Guest sends command to create new contract request.	System validate information, display contract details and request for confirmation. [Exception 1, 2, 3]
4	Guest sends command to create new contract request.	Add new account and new contract information to the system. Show successful message and ask user to process payment.
5	Guest sends command to process payment	Display new view let user select one of following payment gateways: <ul style="list-style-type: none"> - PayPal payment gateway. - Direct payment. And show guest the fee: Contract's fee: text.
6	If user chooses PayPal gateway and sends confirm command. [Alternative 1]	Forward to PayPal payment view to process the payment.
7	User process the PayPal payment	If payment succeed: Show message created successful. [Exception 4]

Alternative Scenario:

No	Actor Action	System Response
1	If user chooses direct payment method	Show company address map.

Exceptions:

No	Actor Action	System Response
1	Guest sends command to create new contract request	System shows error message to ask user input missing required fields.
2	Guest's email is existed in the system	Show message to notify guest that their email is existed in the system.
3	Guest's vehicle plate is existed in the system	Show message to notify guest that their vehicle is existed in the system.

4	If payment failed	Show message to notify user that payment failed and the renew request has been aborted.
---	-------------------	-----------------------------------------------------------------------------------------

Relationships: Payment

Business Rules:

- New customer account and new contract will be created in the system with inputted information.
- The initial status of contract will be set to “Pending”.
- When customer completed payment process:
 - + if the contract’s start date has come, contract’s status would change from “Pending” to “No Card”.
 - + If start date is not come yet, the contract status is not changed.
- Staff will receive a notification about new contract request, they verify contract’s information and issue a card for this contract, in this case, contract’s status would change from “No Card” to “Ready”.
- System must ensure has no duplicate customer or vehicle.
- An email contains customer code and password will be sent to user, user can use this information to login to the system later.
- Start date must not be earlier than the current date.
- Contract term is specified by the system.
- Contract types are loaded from system, contract type can be managed by system administrator.
- Contract price would be calculated from contract type and contract term.

Table 11 Use case WG02 - <Guest> Create new contract request

2.3.1.2.<Customer> Overview Use Case

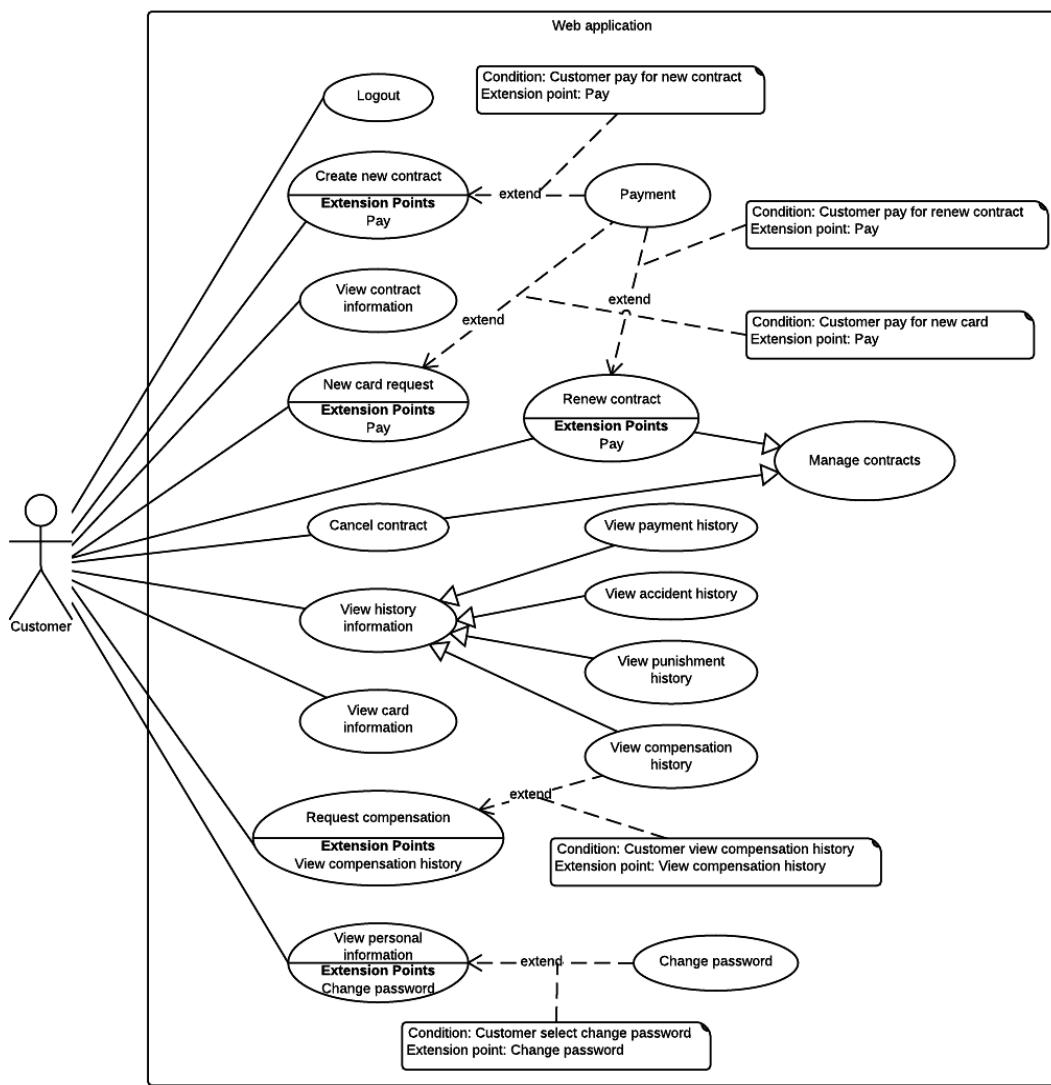


Figure 8 <Customer> Overview Use Case

2.3.1.2.1. <Customer> Logout

USE CASE - WC01			
Use Case No.	WC01	Use Case Version	2.0
Use Case Name	Logout		
Author	TriPQM		
Date	27/05/2015	Priority	Medium
Actor:	<ul style="list-style-type: none"> - Customer, staff. 		
Summary:	<ul style="list-style-type: none"> - This use case allows user to log out the system. 		
Goal:	<ul style="list-style-type: none"> - User can log out the system. 		
Triggers:	<ul style="list-style-type: none"> - User sends the logout command. 		
Preconditions:			

- User logged in the system.

Post Conditions:

- **Success:** User logs out the system.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User sends logout command.	User exits from the system.

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- If user is inactive in 60 minutes, system automatically log user out.

Table 12 Use case WC01 - <Customer> Logout

2.3.1.2.2. <Customer> Renew contract

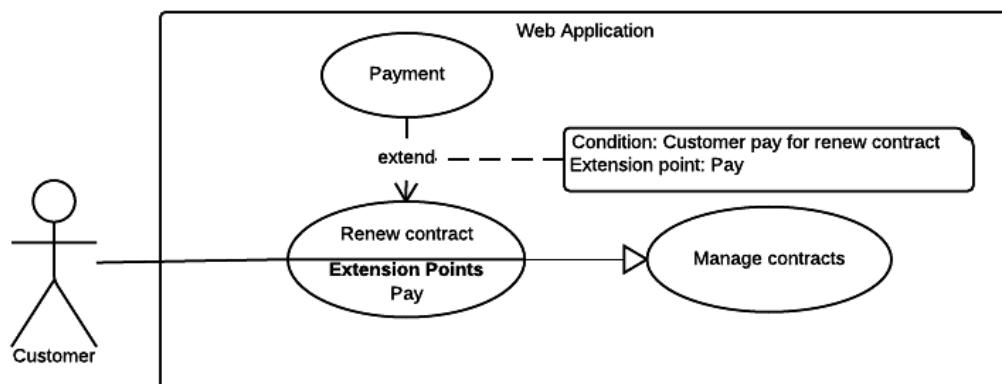


Figure 9 <Customer> Renew contract

USE CASE - WC02			
Use Case No.	WC02	Use Case Version	2.0
Use Case Name	Renew contract by user		
Author	TriPQM		
Date	27/05/2015	Priority	High
Actor:	<ul style="list-style-type: none"> - Customer. 		
Summary:	<ul style="list-style-type: none"> - This use case helps user to renew their contract. 		
Goal:	<ul style="list-style-type: none"> - User can renew their insurance contract. 		
Triggers:	<ul style="list-style-type: none"> - User sends renew contract command. 		
Preconditions:			

- User must login into the system with role Customer.
- Contract's status must be "No Card", "Ready" or "Expired".
- The contract type belongs to this contract is NOT deactivated yet.
- Contract remaining days must NOT exceed the limit in administrator's configuration.

Post Conditions:

- **Success:** Customer's contract is renewed or a renew contract request would be sent to the Staff.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User goes to renew contract view.	Display renew contract information: - Type of contract: text - Start date: text - New expired date: text - Renewal fee: text.
2	User sends renew contract command.	Display new view with following information: - Payment content: text - Renewal fee: text And let user select one of following payment gateways: - PayPal payment gateway. - Direct payment.
3	If user chooses PayPal gateway and sends confirm command. [Alternative 1]	Forward to PayPal payment view to process the payment.
4	User process the PayPal payment	If payment succeed: Update information to the system. Renew user insurance contract. Show message renew successful. [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	If user chooses direct payment method and sends the confirm command.	Show list of company brands address.

Exceptions:

No	Actor Action	System Response
1	If payment failed	Show message to notify user that payment failed and the renew request has been aborted.

Relationships: Extend "Payment".

Business Rules:

- If contract status is "Expired" and user paid the renew fee through PayPal, system automatically change the contract status to "Ready".

- If user paid the renew fee directly, contract's status is "Pending" and Staff will update the payment for that contract and change contract status to "Ready".
- If contract status as "No Card", it will remain in despite of renew process is completed successfully.
- Contract type cannot be changed when renew contract.
- If contract type of contract is not active the customer will not be able renew contract.
- A notification will be sent to staff after the process is completed.

Table 13 Use case WC02 - <Customer> Renew contract by user

2.3.1.2.3. <Customer> Cancel contract

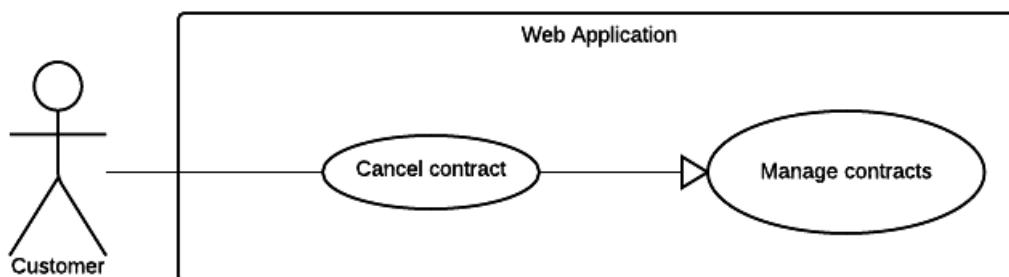


Figure 10 <Customer> Cancel contract

USE CASE - WC03									
Use Case No.	WC03	Use Case Version	2.0						
Use Case Name	Cancel contract								
Author	TriPQM	Priority	High						
Actor:	<ul style="list-style-type: none"> - Customer. 								
Summary:	<ul style="list-style-type: none"> - This use case helps user cancel their contract. 								
Goal:	<ul style="list-style-type: none"> - Customer can cancel the contract. 								
Triggers:	<ul style="list-style-type: none"> - Customer sends cancel contract request. 								
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Customer. - User's contract has not expired. - Customer's contract status must not be "Expired", "Cancelled" or "Request cancel". 								
Post Conditions:	<ul style="list-style-type: none"> - Success: Send to the staff the cancel contract request. - Fail: Show error message. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>User goes to cancel contract view.</td><td> Display new view require user input some information: - Reason to cancel the contract: can be optional selected from these values: </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	User goes to cancel contract view.	Display new view require user input some information: - Reason to cancel the contract: can be optional selected from these values:
Step	Actor Action	System Response							
1	User goes to cancel contract view.	Display new view require user input some information: - Reason to cancel the contract: can be optional selected from these values:							

		<ul style="list-style-type: none"> ○ “Xe cơ giới bị thu hồi đăng ký và biển số theo quy định của pháp luật” ○ “Xe cơ giới hết niên hạn sử dụng theo quy định của pháp luật” ○ “Xe cơ giới bị mất được cơ quan công an xác nhận” ○ “Xe cơ giới hỏng không sử dụng được hoặc bị phá huỷ do tai nạn giao thông được cơ quan công an xác nhận” ○ Other reason: free text input, required, length 1-250.
2	User inputs information	
3	User sends cancel contract request command.	<ul style="list-style-type: none"> - Change contract status. - Send request to the Staff. <p>[Exception 1]</p>

Alternative Scenario: N/A

Exceptions:

No	Actor Action	System Response
1	If user didn't check any reason to cancel contract	Show message to notify user that they have to choose the reason for cancel contract.

Relationships: N/A

Business Rules:

- Cancel contract request will be sent to the system with inputted information.
- System update status of the contract from “Pending”, “No Card” or “Ready” to “Request cancel”.
- A notification will be sent to staff after the process is completed.

Table 14 Use case WC03 - <Customer> Cancel contract

2.3.1.2.4. <Customer> View payment history

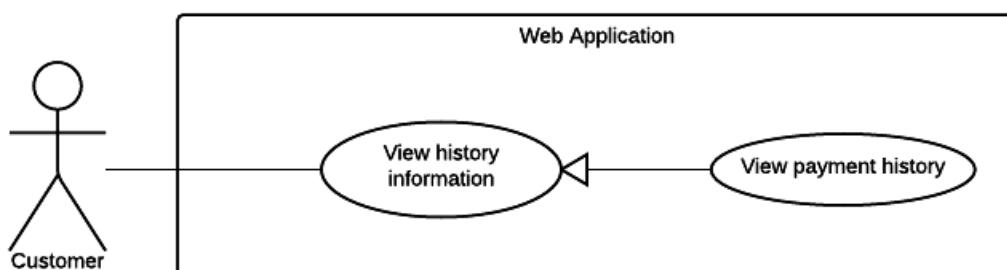


Figure 11 <Customer> View payment history

USE CASE – WC04			
Use Case No.	WC04	Use Case Version	2.0
Use Case Name	View payment history		
Author	TriPQM		
Date	27/05/2015	Priority	Medium
Actor:	<ul style="list-style-type: none"> - Customer. 		

Summary:

- This use case shows user the history of customer's transactions.

Goal:

- User can view list of transactions were made.

Triggers:

- User sends view payment history command.

Preconditions:

- User must login into the system with role Customer.
- There is at least 01 transaction belongs to customer in the system.

Post Conditions:

- **Success:** Show the customer's transactions history to the user.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User sends view payment history command.	<p>Display new view that show user list of transaction history information includes:</p> <ul style="list-style-type: none"> - Transaction code: text - Date of transaction: text - Payment method: text - Service: text - Amount: text - View detail transaction: button <p>[Alternative 1]</p>

Alternative Scenario:

No	Actor Action	System Response
1	If there are no transaction was made.	Show message to notify that there are no transaction was made.

Exceptions: N/A

Relationships: N/A

Business Rules:

- The information is loaded from the system.
- List of transactions will be sorted by date in descending order.
- Pagination must be displayed if number of transactions larger than 10.
- Search bar on the top help user finding transactions faster.

Table 15 Use case WC04 - <Customer> View payment history

2.3.1.2.5. <Customer> View accident history

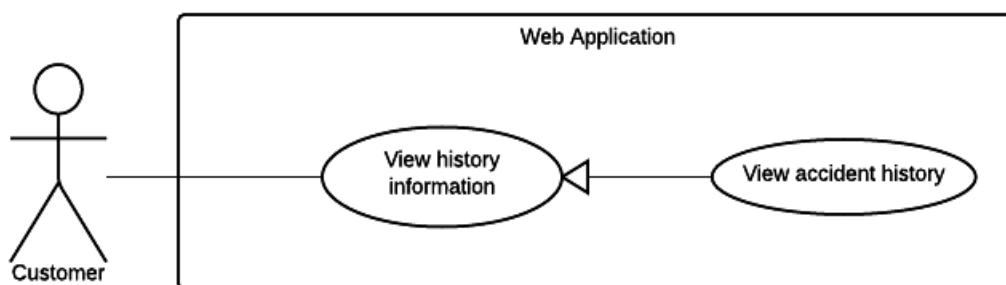


Figure 12 <Customer> View accident history

USE CASE – WC05					
Use Case No.	WC05	Use Case Version	2.0		
Use Case Name	View accident history				
Author	TriPQM				
Date	27/05/2015	Priority	Medium		
Actor:					
- Customer.					
Summary:					
- This use case shows user the history of customer's accidents.					
Goal:					
- User can view list of their accidents history.					
Triggers:					
- User sends view accident history command.					
Preconditions:					
- User must login into the system with role Customer.					
- There is at least 01 accident belongs to customer in the system.					
Post Conditions:					
- Success: Show the customer's accidents history to the user.					
- Fail: Show error message.					
Main Success Scenario:					
Step	Actor Action	System Response			
1	User sends view accident history command	Display new view show list of accidents information includes: - Accident code: text - Content of accident: text - Date of accident: text - Attachment file: link to file. [Alternative 1]			
Alternative Scenario:					
No	Actor Action	System Response			
1	There are no accident.	Show message to notify that there are no accident.			
Exceptions: N/A					
Relationships: N/A					
Business Rules:					
<ul style="list-style-type: none"> - The information is loaded from the system. - List of accidents will be sorted by date in descending order. - Pagination must be displayed if number of accidents larger than 10. - Search bar on the top help user to search accident by title. 					

Table 16 Use case WC05 - <Customer> View accident history

2.3.1.2.6. <Customer> View punishment history

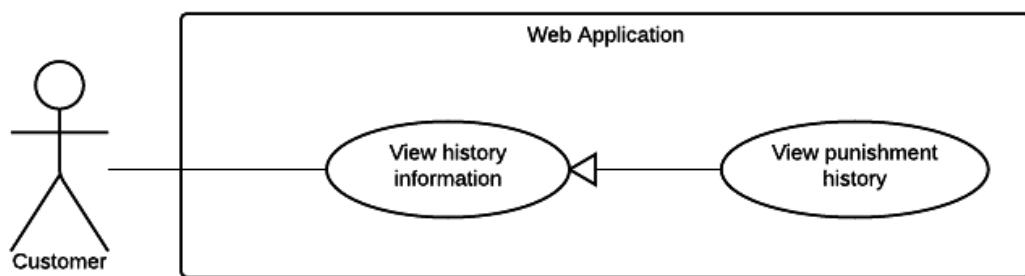


Figure 13 <Customer> View punishment history

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

- Search bar on the top help user finding punishment information faster.
- Pagination must be displayed if number of punishments larger than 10.

Table 17 Use case WC06 - <Customer> View punishment history

2.3.1.2.7. <Customer> Request compensation

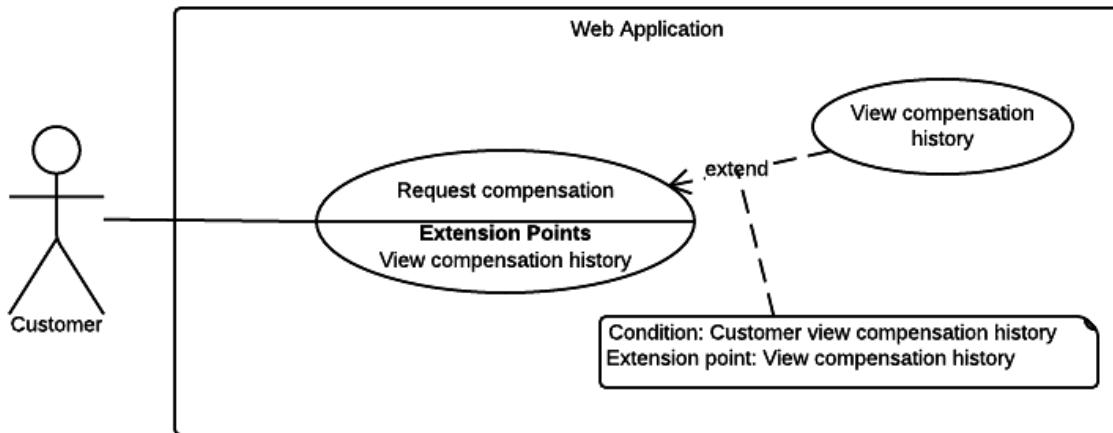


Figure 14 <Customer> Request compensation

USE CASE – WC07									
Use Case No.	WC07	Use Case Version	2.0						
Use Case Name	Request compensation								
Author	TriPQM								
Date	27/05/2015	Priority	High						
Actor:	<ul style="list-style-type: none"> - Customer. 								
Summary:	<ul style="list-style-type: none"> - This use case helps user to request compensation. 								
Goal:	<ul style="list-style-type: none"> - User can request compensation. 								
Triggers:	<ul style="list-style-type: none"> - User sends request compensation command. 								
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Customer. - Contract's status must not be "Pending". - Contract's status must not be "Cancelled" or "Expired" has expired date is more than 30 days with current date. 								
Post Conditions:	<ul style="list-style-type: none"> - Success: Store the compensation request to into the system. - Fail: Show error message. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>User goes to request compensation view.</td><td> Display new view ask user to input required information includes: <ul style="list-style-type: none"> - Driver name: free text input, required, length 3-80. - License number: free text input, required, length 10-15. </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	User goes to request compensation view.	Display new view ask user to input required information includes: <ul style="list-style-type: none"> - Driver name: free text input, required, length 3-80. - License number: free text input, required, length 10-15.
Step	Actor Action	System Response							
1	User goes to request compensation view.	Display new view ask user to input required information includes: <ul style="list-style-type: none"> - Driver name: free text input, required, length 3-80. - License number: free text input, required, length 10-15. 							

		<ul style="list-style-type: none"> - License type: free text input, required, length 1-10. - Driver phone: free text input, required, length 8-15. - Vehicle capacity: free text input, required, length 1-20. - Driver address: free text input, required, length 3-250. - Plate number of accident motor: free text input, required, length 4-15. - Date of accident: date time input, required. - Place of accident: free text input, required, length 3-250. - Control Police Department: free text input, required, length 3-250. - Description: free text input, required, length 1-2000. - Human damage: free text input, required, length 1-2000. - Asset damage: free text input, required, length 1-2000. - Observer: free text input, required, length 3-80. - Compensation note: free text input, required, length 1-2000. - Attachment file: file upload input.
2	User fill required information and attach the minutes of the accident (if any).	
3	User sends request compensation command	Show message to notify that request punishment succeed. [Exception 1, 2]

Alternative Scenario: N/A

Exceptions:

No	Actor Action	System Response
1	User input missed one of requirement information.	Show message to notify user what required information is missing.
2	The inputted information length is out of range.	Show message to notify user what information is out of range.

Relationships: N/A

Business Rules:

- Compensation request will be sent to the system with inputted information.
- Only permit user chooses the accident date before or at the request date.
- New compensation request status will be set to “Đang xử lý”.
- Compensation decision will be set to “Chưa quyết định”.
- A notification will be sent to staff after the process is completed.

Table 18 Use case WC07 - <Customer> Request compensation

2.3.1.2.8. <Customer> View compensation history

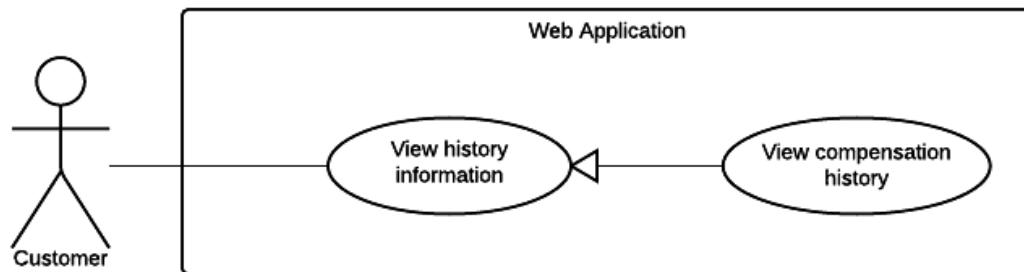


Figure 15 <Customer> View compensation history

USE CASE - WC08												
Use Case No.	WC08	Use Case Version	2.0									
Use Case Name	View compensation history.											
Author	TriPQM											
Date	27/05/2015	Priority	Medium									
Actor:	<ul style="list-style-type: none"> - Customer. 											
Summary:	<ul style="list-style-type: none"> - This use case helps user to view compensation history. 											
Goal:	<ul style="list-style-type: none"> - User can view compensation history. 											
Triggers:	<ul style="list-style-type: none"> - User sends view compensation history command. 											
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Customer. - There is at least 01 compensation belongs to customer in the system. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: Show the compensation history to the user. - Fail: Show error message. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>User sends view compensation history command.</td><td> Display new view show a table of compensations includes: <ul style="list-style-type: none"> - Compensation code: text. - Contract code: text. - Request date: text. - Resolve date: text. </td></tr> <tr> <td>2</td><td>User selects a compensation to view its details.</td><td> Display new view that show the detail of the selected compensation includes: <ul style="list-style-type: none"> - Contract code: text. - Customer name: text. - Driver name: text. - License number: text. </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	User sends view compensation history command.	Display new view show a table of compensations includes: <ul style="list-style-type: none"> - Compensation code: text. - Contract code: text. - Request date: text. - Resolve date: text. 	2	User selects a compensation to view its details.	Display new view that show the detail of the selected compensation includes: <ul style="list-style-type: none"> - Contract code: text. - Customer name: text. - Driver name: text. - License number: text.
Step	Actor Action	System Response										
1	User sends view compensation history command.	Display new view show a table of compensations includes: <ul style="list-style-type: none"> - Compensation code: text. - Contract code: text. - Request date: text. - Resolve date: text. 										
2	User selects a compensation to view its details.	Display new view that show the detail of the selected compensation includes: <ul style="list-style-type: none"> - Contract code: text. - Customer name: text. - Driver name: text. - License number: text. 										

	<ul style="list-style-type: none"> - License type: text. - Driver phone: text. - Vehicle capacity: text. - Driver address: text. - Plate number of accident motor: text. - Date of accident: text. - Place of accident: text. - Control Police Department: text. - Description: text. - Human damage: text. - Asset damage: text. - Observer: text. - Compensation note: text. - Attachment file: link to file. - Created date: text. - Resolve date: text. - Decision: text. - Resolve note: text. <p>[Alternative 1]</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Alternative Scenario:

No	Actor Action	System Response
1	There are no compensation	Show message to notify that there are no compensation.

Exceptions: N/A

Relationships: N/A

Business Rules:

- The information is loaded from the system.
- List of compensation is sorted by resolve date in decreasing order. The unsolved compensations will be the top rows.
- Pagination must be displayed if number of compensations larger than 10.
- Search bar on the top help user to finding compensation faster.

Table 19 Use case WC08 - <Customer> View compensation history

2.3.1.2.9. <Customer> New card request

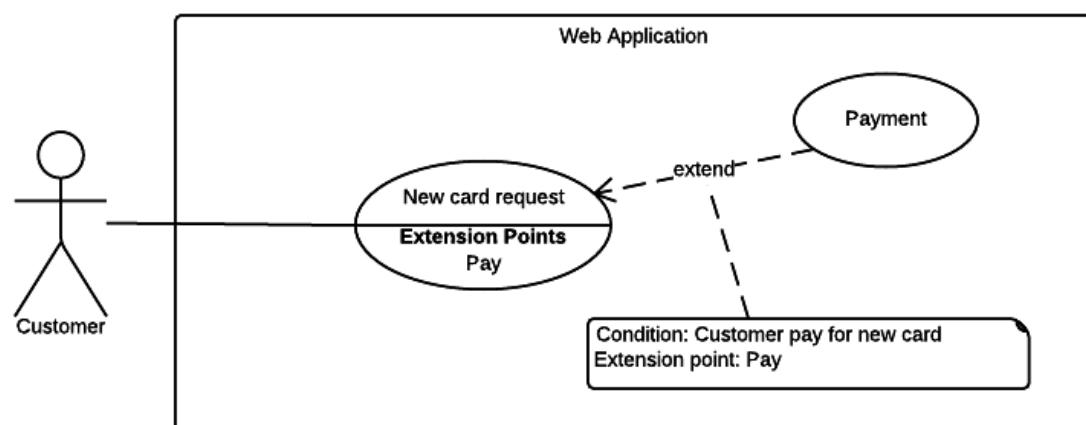


Figure 16 <Customer> New card request

USE CASE - WC09			
Use Case No.	WC09	Use Case Version	2.0
Use Case Name	New card request		
Author	TriPQM		
Date	27/05/2015	Priority	High
Actor:	<ul style="list-style-type: none"> - Customer. 		
Summary:	<ul style="list-style-type: none"> - This use case helps user to request a new card. 		
Goal:	<ul style="list-style-type: none"> - User can request a new card. 		
Triggers:	<ul style="list-style-type: none"> - User sends new card request command. 		
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Customer. - User must has a card in the system. - Contract status must be "Ready" 		
Post Conditions:	<ul style="list-style-type: none"> - Success: New card request will be sent to the system. - Fail: Show error message. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	User sends new card request command.	<p>Display a new view shows to user:</p> <ul style="list-style-type: none"> - A text box to confirm by password: free text input, required, length 6-32. - A text box to enter note: : free text input, length 0-2000. - Payment gateways: can be optional selected from these selections: <ul style="list-style-type: none"> o PayPal payment gateway. o Direct payment. - Check boxes for customer to choose deactivate the old card immediately and delivery card request. - The new card fee: text. - The delivery fee: text. - The total fee: text. 	
2	User enter password and choose the PayPal payment gateway. Then sends confirm command. [Alternative 1]	<p>Forward to PayPal payment process view.</p> <p>[Alternative 2]</p>	
3	User process the PayPal payment.	<p>If payment succeed, Show message to notify that the new card request and payment is succeed.</p> <p>[Exception 1]</p>	

Alternative Scenario:

No	Actor Action	System Response
1	User choose the direct payment method.	Show message to notify that the new card is sent. [Alternative 2]
2	If user enter wrong password	Show message to notify that user has entered wrong password.

Exceptions:

No	Actor Action	System Response
1	If payment failed	Show message to notify that the payment failed. The request is aborted.

Relationships: <Payment system> Payment (WP01)

Business Rules:

- The new card fee and delivery fee is specified by the system.
- At a time, user can only has one new card request for each contract.
- A notification will be sent to staff after the process is completed.

Table 20 Use case WC09 - <Customer> New card request

2.3.1.2.10. <Customer> View card information

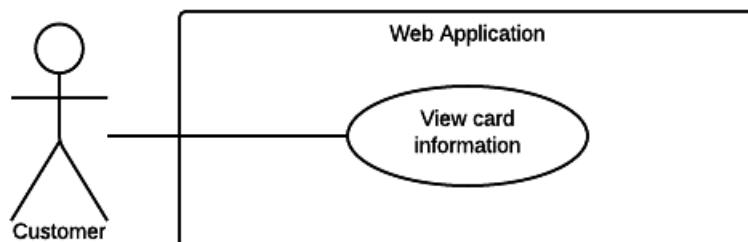


Figure 17 <Customer> View card information

USE CASE - WC10			
Use Case No.	WC10	Use Case Version	2.0
Use Case Name	View card information		
Author	TriPQM		
Date	27/05/2015	Priority	Medium
Actor:			
- Customer.			
Summary:			
- This use case shows user their card information.			
Goal:			
- User can view their card information.			
Triggers:			
- User sends view card information command.			
Preconditions:			
- User must login into the system with role Customer.			
- There is at least 1 card belongs to customer in the system.			
- Card must belong to current logged in customer.			
- Card must NOT be deactivated.			
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	<p>Display new view shows the card information includes:</p> <ul style="list-style-type: none"> - Card ID: text - User name: text - Card status: text - Activated date: text <p>And a table shows list of access history information includes:</p> <ul style="list-style-type: none"> - Access date: text - Access device: text - Request: text - Response: text

Alternative Scenario:

No	Actor Action	System Response
1	There are no card	Show message to notify that there are no card.

Exceptions: N/A

Relationships: N/A

Business Rules:

- The information is loaded from the system.
- List of access history is sorted by access date in descending order.
- Pagination must be displayed if number of access logs larger than 10.
- Filter bar on the top help user to finding access logs faster.

Table 21 Use case WC10 - <Customer> View card information

2.3.1.2.11. <Customer> View personal information

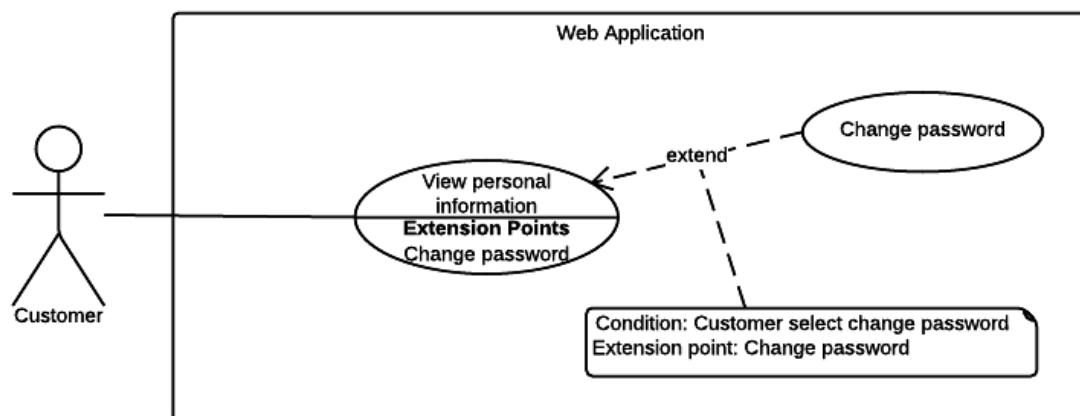


Figure 18 <Customer> View personal information

USE CASE – WC11					
Use Case No.	WC11	Use Case Version	2.0		
Use Case Name	View personal information				
Author	TriPQM				
Date	27/05/2015	Priority	Medium		
Actor:	<ul style="list-style-type: none"> - Customer. 				
Summary:	<ul style="list-style-type: none"> - This use case shows user their personal information. 				
Goal:	<ul style="list-style-type: none"> - User can review their card information. 				
Triggers:	<ul style="list-style-type: none"> - User sends view personal information command. 				
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Customer. 				
Post Conditions:	<ul style="list-style-type: none"> - Success: Show the user's personal information. - Fail: Show error message. 				
Main Success Scenario:					
Step	Actor Action	System Response			
1	User sends view personal information command	Display new view shows the user information includes: <ul style="list-style-type: none"> - User code: text - User name: text - Address: text - Email: text - Phone number: text - Personal ID: text [Exception 1]			
Alternative Scenario:	N/A				
Exceptions:					
No	Actor Action	System Response			
1	User sends view personal information command	User has been logged out because of inactivity too long.			
Relationships:	N/A				
Business Rules:	<ul style="list-style-type: none"> - If user is inactive in 60 minutes, system will automatically log user out. - Customer profile is loaded from system. 				

Table 22 Use case WC11 - <Customer> View personal information

2.3.1.2.12. <Customer> Change password

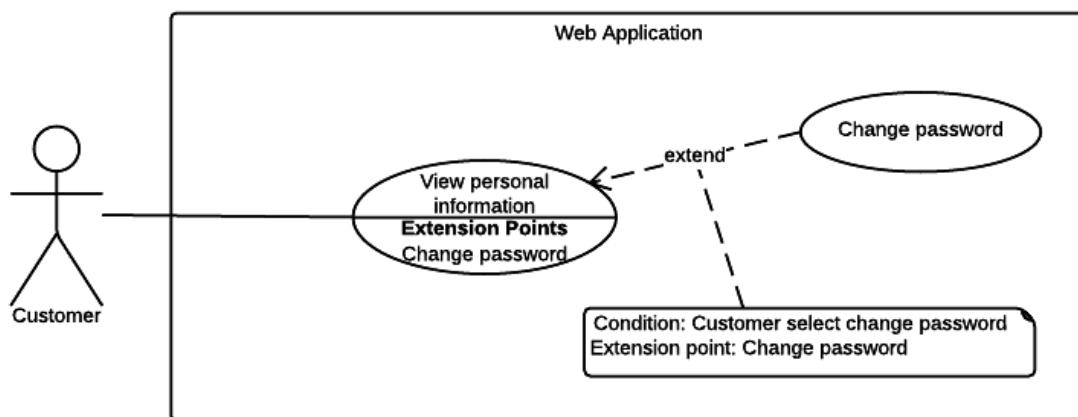


Figure 19 <Customer> Change password

USE CASE - WC12			
Use Case No.	WC12	Use Case Version	2.0
Use Case Name	Change password.		
Author	TriPQM		
Date	27/05/2015	Priority	High
Actor:	<ul style="list-style-type: none"> - Customer. 		
Summary:	<ul style="list-style-type: none"> - This use case helps user to change their password. 		
Goal:	<ul style="list-style-type: none"> - User can change the password. 		
Triggers:	<ul style="list-style-type: none"> - User sends change password command. 		
Preconditions:	<ul style="list-style-type: none"> - User must login into the system. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Update new user's password to the system. - Fail: Show error message. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	User goes to change password view.	Display new view contains textboxes for user to input their current and new password: <ul style="list-style-type: none"> - Current password: free text input, required, length 6-32 - New password: free text input, required, length 6-32 - Confirm new password: free text input, required, length 6-32 	
2	User input their current password, new password and confirm new password.		

3	User sends change password command.	Show message to notify that the password has been changed successfully. [Alternative 1, 2] [Exception 1]
---	-------------------------------------	----------------------------------------------------------------------------------------------------------------

Alternative Scenario:

No	Actor Action	System Response
1	If user inputs the wrong current password	Show message to notify that the current password is wrong.
2	If the new password and the confirm password not matched.	Show message to notify that the confirm password is not matched with the new password.

Exceptions:

No	Actor Action	System Response
1	User sends view personal information command	User has been logged out because of inactivity too long.

Relationships: N/A

Business Rules:

- The new password will be saved into the system if the provided information is correct.
- The new password must different than the old password
- If user is inactive in 60 minutes, system will automatically log user out.

Table 23 Use case WC12 - <Customer> Change password

2.3.1.3.<Staff> Overview Use Case

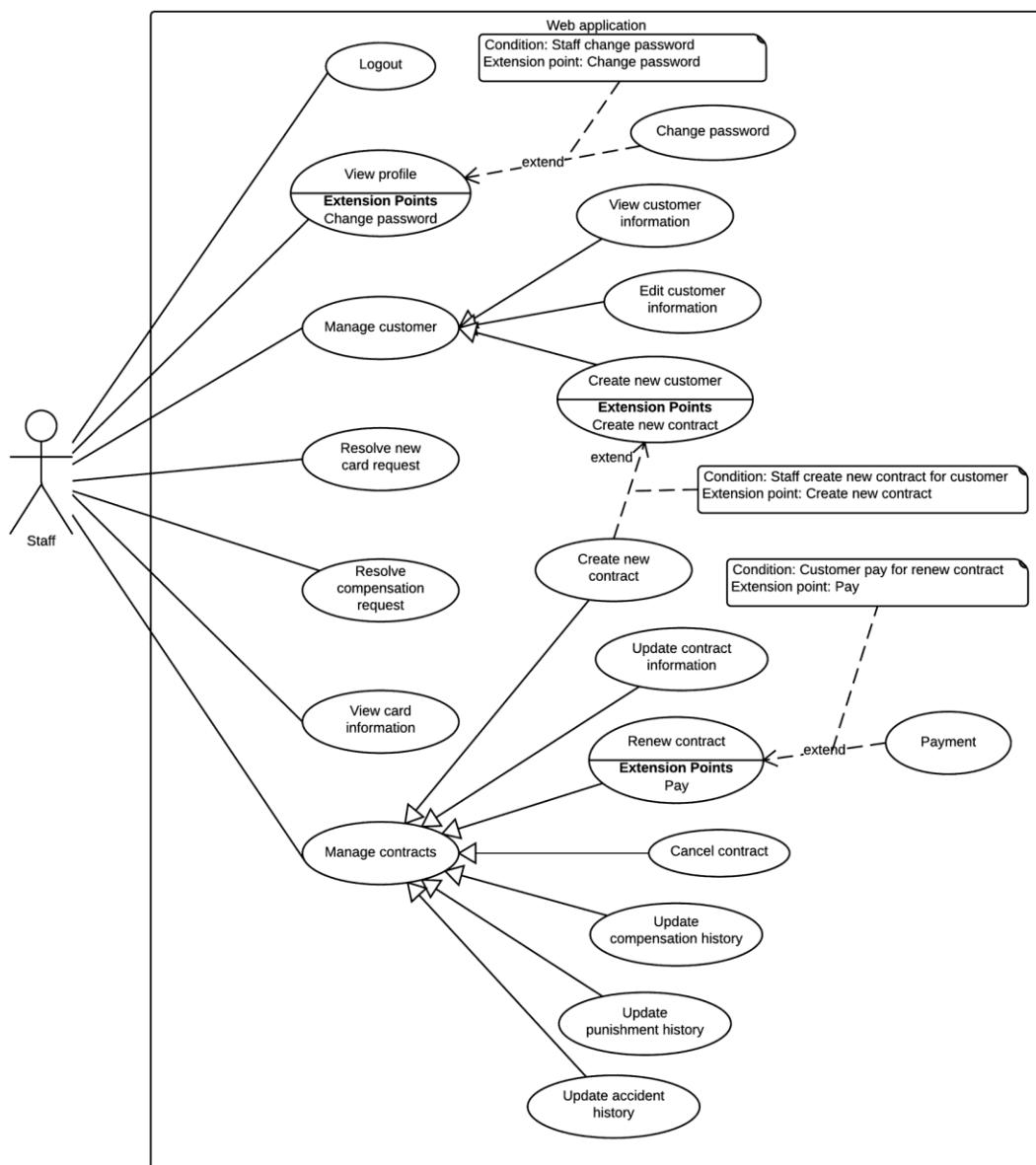


Figure 20 <Staff> Overview Use Case

2.3.1.3.1. <Staff> View profile

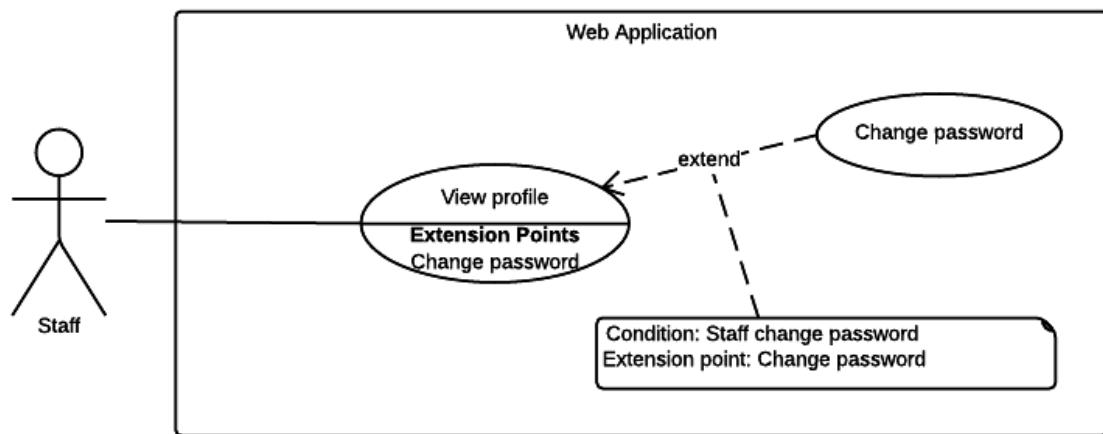


Figure 21 <Staff> View profile

USE CASE – WS01			
Use Case No.	WS01	Use Case Version	2.0
Use Case Name	View profile		
Author	KhaNC		
Date	27/05/2015	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff view their profile. 		
Goal:	<ul style="list-style-type: none"> - Staff can view their detailed profile. 		
Triggers:	<ul style="list-style-type: none"> - Staff goes to view their profile. 		
Preconditions:	<ul style="list-style-type: none"> - Staff must login into the system with roles Staff. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Staff profile information is shown. - Fail: Show error message. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff goes to view their profile.	Staff's detail profile will be shown with following information: <ul style="list-style-type: none"> - Staff's code: text - Staff's full name: text - Staff's email: text - Staff's phone: text 	
Alternative Scenario: N/A			

Exceptions:

No	Actor Action	System Response
1	User sends view personal information command	User has been logged out because of inactivity too long.

Relationships: N/A

Business Rules:

- Staff's profile information is always loaded from the system.
- If user is inactive in 60 minutes, system will automatically log user out.

Table 24 Use case WS01 - <Staff> View profile

2.3.1.3.2. <Staff> Change password

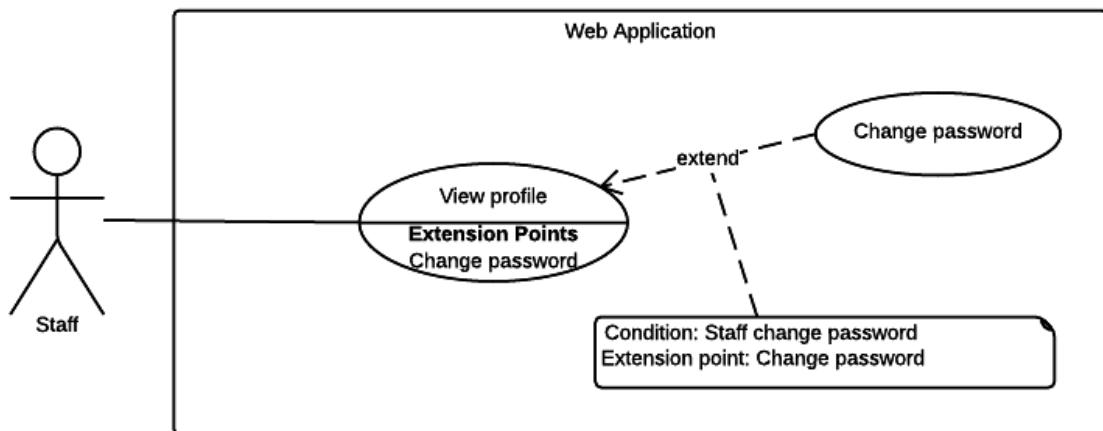


Figure 22 <Staff> Change password

USE CASE - WS02			
Use Case No.	WS02	Use Case Version	2.0
Use Case Name	Change password		
Author	KhaNC		
Date	27/05/2015	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff change their password. 		
Goal:	<ul style="list-style-type: none"> - Staff can change their account password. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends change password command. 		
Preconditions:	<ul style="list-style-type: none"> - Staff must login into the system with roles Staff. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Staff's new password is updated to the system. - Fail: Show error message. 		

Main Success Scenario:

Step	Actor Action	System Response
1	User goes to change password view.	Change password view will be shown with following information: - Current password: free text input, required - New password: free text input, required, length 6 – 32 - Confirm new password: free text input, required, length 6 – 32
2	User enters required information	
3	User sends change password command.	- Validate data [Alternative 1, 2] [Exception 1, 2] - Update staff's new password into the system. - Display message notify user their password has changed successfully.

Alternative Scenario:

No	Actor Action	System Response
1	Current password does not match with staff's password in the system	Show message notify staff entered current password does not match with staff's password in the system
2	Confirm new password does not match with new password	Show message notify staff entered confirm password is not match with their new password

Exceptions:

No	Actor Action	System Response
1	Missing of required fields	Show message to notify staff input missed fields
2	Length of new password not in range	Show message notify staff's new password is out of range

Relationships: N/A

Business Rules:

- In case of successful scenario, staff's new password would be updated to the system.
- The user should be able to know how strong their password is when they input their new password.
- The new password must be different than the old password

Table 25 Use case WS02 - <Staff> Change password

2.3.1.3.3. <Staff> Resolve new card request

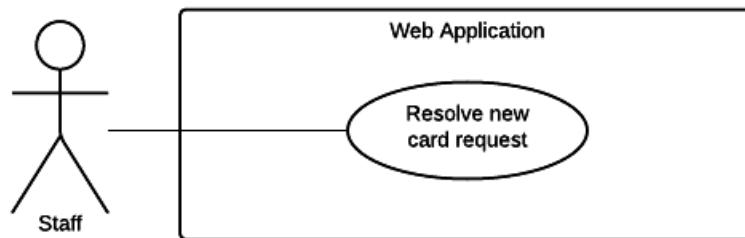


Figure 23 <Staff> Resolve new card request

USE CASE – WS03			
Use Case No.	WS03	Use Case Version	2.1
Use Case Name	Resolve new card request		
Author	KhaNC		
Date	27/05/2015	Priority	Medium
Actor:	<ul style="list-style-type: none"> - Staff. 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff view and solve new card request. 		
Goal:	<ul style="list-style-type: none"> - Staff can view and solve request for new card from customers. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends view and solve new card request command. 		
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Staff. - There is at least 1 unsolved request for new card from customer. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: A customer's new card request is solved, a new card issued and the date request solved updated. - Fail: Request is unsolved. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff goes to view all new card request view.	New card request view will be shown with following information: <ul style="list-style-type: none"> - Date of request: text - Note of customer: text - ID of old card: link to card details. - Name of card owner: link to detailed information of the customer who own this card. - Date a new card be issued for this customer: text - ID of new card: link to card details. 	
2	Staff add new card for this customer	Update new card information to customer's request and sends notification. <ul style="list-style-type: none"> - Date new card issued. 	

		- ID of new card
Alternative Scenario:		
No	Actor Action	System Response
1	There are no unsolved new card request	Show message to notify staff there are no unsolved new card request.

Exceptions: N/A
Relationships: N/A
Business Rules:

- List of new card request is always loaded from the system.
- List of new card request is sorted by resolved date in ascending order.
- The old card will change status from “Ready” to “Deactivated”.
- The new card will have status is “Ready”.
- Search bar on the top help user finding card faster by its id.
- Pagination must be displayed if number of requests larger than 10.
- A solved request must have code of new card and the date this card issued.

Table 26 Use case WS03 - <Staff> Resolve new card request

2.3.1.3.4. <Staff> Resolve compensation request

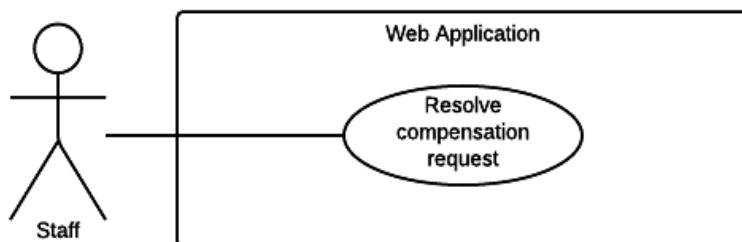


Figure 24 <Staff> Resolve compensation request

USE CASE – WS04			
Use Case No.	WS04	Use Case Version	2.0
Use Case Name	Resolve compensation request		
Author	KhaNC		
Date	27/05/2015	Priority	High
Actor:			
- Staff.			
Summary:			
- This use case allows staff view and solve compensation requests.			
Goal:			
- Staff can view and solve request for compensation from customers.			
Triggers:			
- Staff sends view and solve compensation request command.			
Preconditions:			
- User must login into the system with role Staff.			
- There is at least 1 unsolved request for compensation from customer.			
- The compensation's decision must be negotiated with the customer, approved by the insurance company.			
Post Conditions:			

- **Success:** A customer's compensation request solved, decision for this compensation has been made and the date request solved is updated.
- **Fail:** Request is unsolved.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to view list of request for compensation.	<p>List of compensation request will be shown with following information:</p> <ul style="list-style-type: none"> - Compensation code: link to request detail - Request created date: text - Contract's code: link to contract detail - Contract's owner full name: link to customer detail - Request status: text
2	Staff select an unsolved request to view.	<p>Compensation request detail will be shown with following information:</p> <ul style="list-style-type: none"> - Contract's code: link to contract details. - Contract's owner full name: link to customer details. - Compensation created date: text - Compensation status: text - Driver's full name: text - Driver's phone number: text - Driver's address: text - Driver's license number: text - Driver's license type: text - Accident vehicle plate number: text - Vehicle's capacity or seat capacity: 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 - Address of the observer: text - Detail of compensation request from customer: text - Compensation attachment: link to attachment detail
3	Staff send resolve this compensation request command.	<p>Resolve compensation request view is shown with following labels and fields</p> <ul style="list-style-type: none"> - Resolve date: date time input, required

		- Note for compensation: free text input, length 1 – 2000
4	Staff input resolve compensation request information and select one from two decision	
5	Staff sends save changes command	Update compensation status and notify to customer

Alternative Scenario:

No	Actor Action	System Response
1	There are no compensation request	Show message to notify staff there are no compensation request.

Exceptions: N/A

Relationships: N/A

Business Rules:

- List of compensation request is always loaded from the system.
- List of compensation request is sorted by compensation request resolved date and created date in ascending order.
- With unsolved request, its solved date does not exist.
- Compensation resolved date must be restricted in the limit by configuration of the administrator.
- A compensation request could have one of following decision:
 - o “Chấp nhận bồi thường”
 - o “Từ chối bồi thường”
- A compensation request could have one of following status:
 - o “Đã giải quyết”
 - o “Chưa giải quyết”
- Compensation information input in this screen is the final information after company and customer has done physical business about dealing the compensation fee.
- Search bar on the top help user finding contract or customer faster.
- Pagination must be display if number of requests larger than 10.
- A notification will be sent to request customer after the process is completed.

Table 27 Use case WS04 - <Staff> Resolve compensation request

2.3.1.3.5. <Staff> View customer information

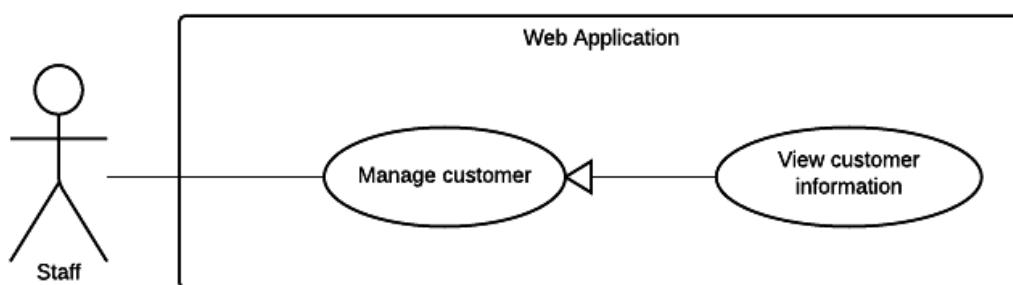


Figure 25 <Staff> View customer information

USE CASE - WS05		
Use Case No.	WS05	Use Case Version
		2.0

Use Case Name	View customer information				
Author	KhaNC				
Date	20/05/2015	Priority	Medium		
Actor:					
- Staff.					
Summary:					
- This use case allows staff view customer details.					
Goal:					
- Staff can view customer's detail information.					
Triggers:					
- Staff sends view customer information command.					
Preconditions:					
- User must login into the system with role Staff.					
Post Conditions:					
- Success: Customer's detail information is shown.					
- Fail: Show error message.					
Main Success Scenario:					
Step	Actor Action	System Response			
1	Staff goes to list of customer.	List of customer will be shown with following information: - Customer's code: link to customer detail - Customer's full name: text - Customer's phone: text - Customer's email address: text			
2	Staff selects customer to view their detail information.	Customer detail will be shown with following information: - Customer's code: text - Customer's full name: link to customer detail - Customer's address: text - Customer's email address: text - Customer's phone: text - Customer's personal ID: text List of customer's contract: - Customer's contract code: link to contract detail. - Customer's contract type: text - Contract's status: text			
Alternative Scenario: N/A					
Exceptions: N/A					
Relationships: N/A					
Business Rules:					
<ul style="list-style-type: none"> - List of customer and their detail information is always loaded from the system. - Search bar on the top help user find customer faster. - Pagination must be display if number of requests larger than 10. - Contract's status is one of 6 below: <ul style="list-style-type: none"> o Pending 					

- No card
- Ready
- Expired
- Request cancel
- Cancelled

Table 28 Use case WS05 - <Staff> View customer information

2.3.1.3.6. <Staff> Edit customer information

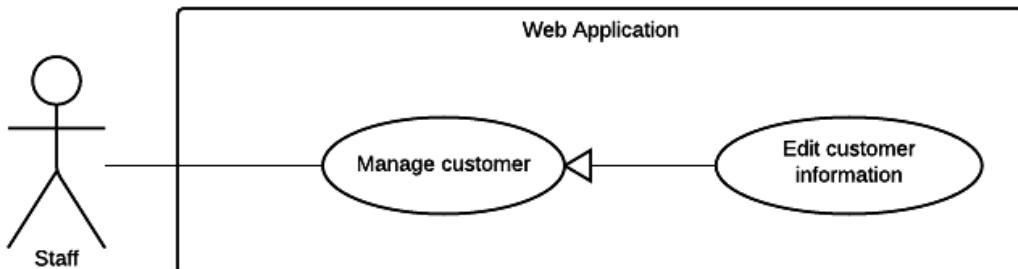


Figure 26 <Staff> Edit customer information

USE CASE – WS06			
Use Case No.	WS06	Use Case Version	2.0
Use Case Name	Edit customer information		
Author	KhaNC		
Date	20/05/2015	Priority	Medium
Actor:	<ul style="list-style-type: none"> - Staff. 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff edit customer's information. 		
Goal:	<ul style="list-style-type: none"> - Customer's information is updated to the system. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends update customer's information command. 		
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Staff. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Customer's information is updated. - Fail: Show error message. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff goes to edit customer information.	Edit customer view is shown with following labels and fields: <ul style="list-style-type: none"> - Customer's full name: free text input, required, length 3 – 80 - Customer's address: free text input, required, length 3 – 250 - Customer's email address: free text input, required, length 3 – 250 	

		<ul style="list-style-type: none"> - Customer's phone number: free text input, required, length 8 – 15 - Customer's personal ID: free text input, length 8 – 15
2	Staff fills out the form.	
3	Staff sends update customer's information command.	<ul style="list-style-type: none"> - Validate data [Exception 1, 2, 3] - Update customer's new information to the system. - Reload customer detail.

Alternative Scenario: N/A

Exceptions:

No	Actor Action	System Response
1	Missing of required fields	Show message notify staff input missed fields
2	Length of field's value is out of range	Show message notify staff which field's value is out of range
3	Entered email address is not a valid email	Show message notify entered email is not valid

Relationships: N/A.

Business Rules:

- Customer's detail information is always loaded from the system.
- In case of success scenarios, customer new information would be updated to the system.
- Reloaded customer detail will display customer updated information.
- An email address must be validated by this regular expression:
 $/^([a-z0-9_\.]+)@([\da-z\-.]+)\.([a-z\.]\{2,6\})$/$
- Customer's current information must be shown in its respective fields.

Table 29 Use case WS06 - <Staff> Edit customer information

2.3.1.3.7. <Staff> Create new customer

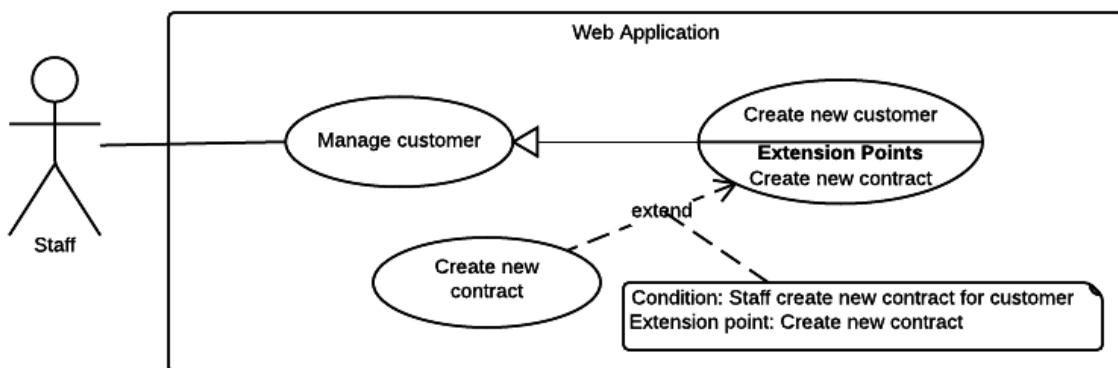


Figure 27 <Staff> Create new customer

USE CASE – WS07					
Use Case No.	WS07	Use Case Version	2.0		
Use Case Name	Create new customer				
Author	KhaNC				
Date	20/05/2015	Priority	Medium		
Actor:	<ul style="list-style-type: none"> - Staff. 				
Summary:	<ul style="list-style-type: none"> - This use case allows staff create new customer. 				
Goal:	<ul style="list-style-type: none"> - A new customer is added to the system. 				
Triggers:	<ul style="list-style-type: none"> - Staff sends create new customer command. 				
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Staff. - This customer is not existed in the system yet. 				
Post Conditions:	<ul style="list-style-type: none"> - Success: New customer is added to the system. - Fail: Show error message. 				
Main Success Scenario:					
Step	Actor Action	System Response			
1	Staff goes to create customer view.	Create customer view is shown with following labels and fields: <ul style="list-style-type: none"> - Customer's full name: free text input, required, length 3 – 80 - Customer's address: free text input, required, length 3 – 250 - Customer's email address: free text input, required, length 3 – 250 - Customer's phone number: free text input, required, length 8 – 15 - Customer's personal ID: free text input, length 8 – 15 			
2	Staff fills out the form.				
3	Staff sends create new customer command.	<ul style="list-style-type: none"> - Validate data [Exception 1, 2, 3, 4] - Add new customer to the system. - Display create customer success message. 			
Alternative Scenario: N/A					
Exceptions:					
No	Actor Action	System Response			
1	Missing of required fields	Show message notify staff input missed fields			
2	Length of field's value is out of range	Show message notify staff which field's value is out of range			
3	Entered email address is not a valid email	Show message notify entered email is not valid			

4	Entered email is existed in the system	Show message notify entered email is existed
Relationships: <Staff> Create new contract (WS08)		
Business Rules:		
<ul style="list-style-type: none"> - In case of success scenarios, a new customer would be added to the system. - An email address must be validated by this regular expression: /^([a-z0-9_\.-]+)@([\da-z\.-]+\.)\.([a-z\.]{2,6})\$/ - System will suggest staff create new contract for the customer has just created. 		

Table 30 Use case WS07 - <Staff> Create new customer

2.3.1.3.8. <Staff> Create contract

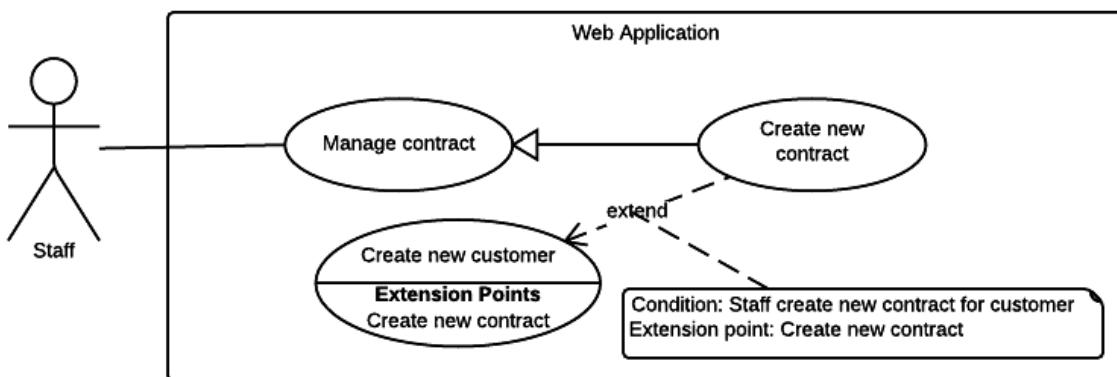


Figure 28 <Staff> Create contract

USE CASE – WS08			
Use Case No.	WS08	Use Case Version	2.0
Use Case Name	Create new contract		
Author	KhaNC		
Date	20/05/2015	Priority	Medium
Actor:			
<ul style="list-style-type: none"> - Staff. 			
Summary:			
<ul style="list-style-type: none"> - This use case allows staff create new contract. 			
Goal:			
<ul style="list-style-type: none"> - A new contract is added to the system. 			
Triggers:			
<ul style="list-style-type: none"> - Staff sends create new contract command. 			
Preconditions:			
<ul style="list-style-type: none"> - User must login into the system with role Staff. - There is at least 1 customer in the system. 			
Post Conditions:			
<ul style="list-style-type: none"> - Success: New contract is added to the system - Fail: Show error message. 			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff goes to create new contract view.	Contract detail is shown with following label and fields:	

		<ul style="list-style-type: none"> - Customer code: free text input, required, length 6 – 10 <p>Contract's information</p> <ul style="list-style-type: none"> - Contract' type: select one of the options. - Start date: date time input, required - Expired date: date time input, required - Contract's fee: text <p>Vehicle's information:</p> <ul style="list-style-type: none"> - Vehicle's plate number: free text input, required, length 4 – 15 - Vehicle's brand: free text input, required, length 2 – 20 - Vehicle's model code: free text input, length 2 – 20 - Vehicle's type: free text input, length 2 – 20 - Vehicle's color: free text input, length 2 – 20 - Vehicle's engine: free text input, required, length 2 – 20 - Vehicle's chassis: free text input, required, length 2 – 20 - Vehicle's capacity: free text input, required, length 2 – 20 - Vehicle's year of manufacture: free text input, length: 4 - Vehicle's empty weight: free text input, length 1 – 4 - Vehicle's seat capacity: free text input, length 1 – 3 <p>Payment information:</p> <ul style="list-style-type: none"> - Paid date: date time input, required
2	Staff fills out the form.	
3	Staff sends create new contract command.	Display contract's information to review and request for confirmation
4	Staff confirms create new contract command.	<ul style="list-style-type: none"> - Validate data [Exception 1, 2, 3] - Add new contract's information to the system. - Notify to staff created contract successfully.

Alternative Scenario: N/A**Exceptions:**

No	Actor Action	System Response
1	Missing of required fields	Show message notify staff input missed fields

2	Length of field's value is out of range	Show message notify staff which field's value is out of range
3	Entered vehicle is existed in the system	Show message notify staff this vehicle is existed.

Relationships: <Staff> Create new customer (WS07)

Business Rules:

- In case of success scenarios, a new contract would be added to the system.
- Staff can search and select a contract owner from available customers.
- List of contract's type must be loaded from the system.
- A new contract created successfully will have the initial status is "No card".
- Contract's start date, expired date, paid date must be restricted in the limit by configuration of the administrator.
- Default contract's term is set up by configuration of the administrator.
- Contract's term must not exceed contract's default term.
- Contract's expired date must not be earlier than contract's start date.
- Contract's fee is calculated by the following formula, with price per year is loaded from the system belongs to selected contract type:

$$\text{contract fee} = \frac{\text{price per year}}{12} \times \text{contract term}$$

- Contract's term will be rounded up to nearest month, for example: 2 months 18 days is 3 months

Table 31 Use case WS08 - <Staff> Create new contract

2.3.1.3.9. <Staff> Renew contract

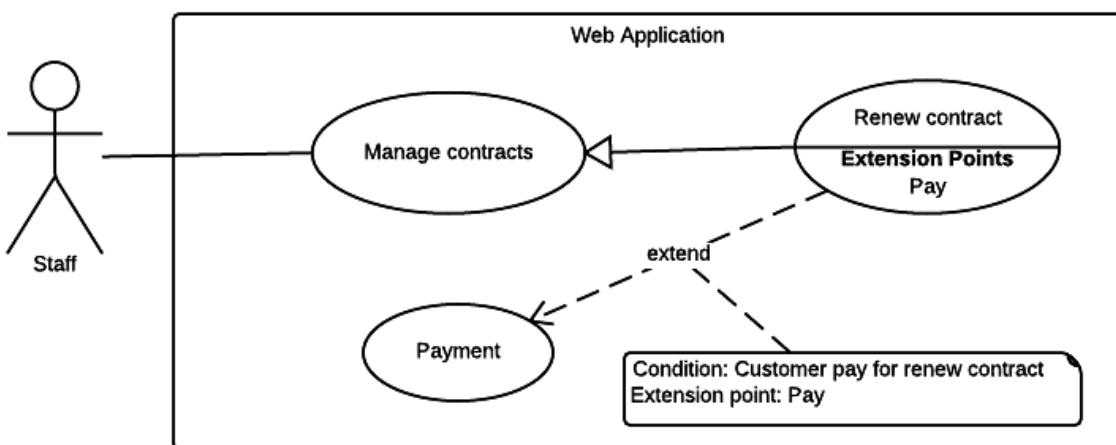


Figure 29 <Staff> Renew contract

USE CASE – WS09			
Use Case No.	WS09	Use Case Version	2.0
Use Case Name	Renew contract		
Author	KhaNC		
Date	27/05/2015	Priority	Medium
Actor:	<ul style="list-style-type: none"> - Staff. 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff renew contract. 		
Goal:			

- Contract's new expired date will be updated to the system.

Triggers:

- Staff sends renew contract command.

Preconditions:

- User must login into the system with role Staff.
- The contract type belongs to this contract is NOT deactivated yet.
- Contract remaining days must NOT exceed the limit in administrator's configuration.
- Contract's status must be "No card", "Ready" or "Expired".

Post Conditions:

- **Success:** Contract is renewed
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to renew contract view.	Renew contract detail is shown with following renew contract information: <ul style="list-style-type: none"> - Contract's code: text - Contract's status: text - Contract's type: text - Contract's status: text - Contract's start date: text - Contract's expired date: text - Contract's new expired date: date time input, required - Renew contract fee: text - Request for new card: free text input - New card fee: text - Delivery new card: free text input - Delivery new card fee: text - Paid date: date time input, required
2	Staff fills out the form.	
3	Staff sends renew this contract command	Display confirm renew contract view.
4	Staff confirms renew this contract command.	<ul style="list-style-type: none"> - Validate data [Exception 1, 2] - Update contract's information to the system. - Reload contract detail.

Alternative Scenario: N/A

Exceptions:

No	Actor Action	System Response
1	Missing of required fields	Show message notify staff input missed fields
2	Length of field's value is out of range	Show message notify staff which field's value is out of range

Relationships: <Payment system> Payment (WP01)

Business Rules:

- In case of success scenarios, a new expired date of contract would be updated to the system.
 - A contract renewed successfully will have status is “Ready”:
 - o If contract’s status is “No card” or “Ready”, there are no changes.
 - o In case of the contract has status is “Expired”, its will be changed to “Ready”.
 - Contract’s new expired date, paid date must be restricted in the limit by configuration of the administrator.
 - Default contract’s term is set up by configuration of the administrator.
 - Contract’s term must not exceed contract’s default term.
 - Contract’s expired date must not be earlier than contract’s start date.
 - Contract’s fee is calculated by the following formula, with price per year is loaded from the system belongs to current contract type:
- $$\text{contract fee} = \frac{\text{price per year}}{12} \times \text{contract term}$$
- Contract’s term will be rounded up to nearest month, for example: 2 months 18 days is 3 months.

Table 32 Use case WS09 - <Staff> Renew contract

2.3.1.3.10. <Staff> Cancel contract

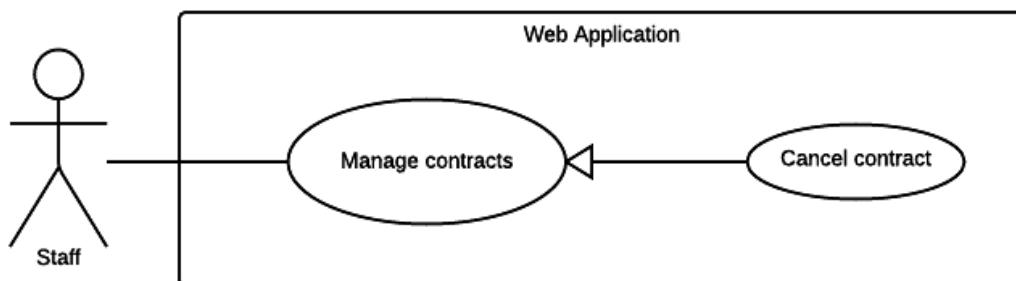


Figure 30 <Staff> Cancel contract

USE CASE - WS10			
Use Case No.	WS10	Use Case Version	2.0
Use Case Name	Cancel contract		
Author	KhaNC		
Date	27/05/2015	Priority	Medium
Actor:	<ul style="list-style-type: none"> - Staff. 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff cancel a contract. 		
Goal:	<ul style="list-style-type: none"> - Contract’s new status is updated to the system. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends cancel contract command. 		
Preconditions:	<ul style="list-style-type: none"> - User must login into the system with role Staff. - Contract’s status must be “Pending”, “No card” or “Ready”. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Contract is cancelled. 		

<ul style="list-style-type: none"> - Fail: Show error message. 		
Main Success Scenario:		
Step	Actor Action	System Response
1	Staff goes to cancel contract view.	<p>Cancel contract detail is shown with following information:</p> <ul style="list-style-type: none"> - Cancel date: date time input, required - Cancel reason: free text input, required, length 1 – 250 - Description: free text input, length 1 – 2000
2	Staff fills out the form.	
3	Staff sends cancel contract command.	Display confirm cancel contract view.
4	Staff confirms cancel contract command.	<ul style="list-style-type: none"> - Validate data [Exception 1, 2] - Update contract's information to the system. - Reload contract detail.
Alternative Scenario: N/A		
Exceptions:		
No	Actor Action	System Response
1	Missing of required fields	Show message notify staff input missed fields
2	Length of field's value is out of range	Show message notify staff which field's value is out of range
Relationships: N/A		
Business Rules:		
<ul style="list-style-type: none"> - When contract cancelled successfully, there are no notification would be sent to the customer who own this contract. - A contract cancelled successfully will change status from “No card”, “Ready” or “Pending” to “Cancelled”. - Contract's cancel date must be restricted in the limit by configuration of the administrator. 		

Table 33 Use case WS10 - <Staff> Cancel contract

2.3.1.3.11. <Staff> Update contract information

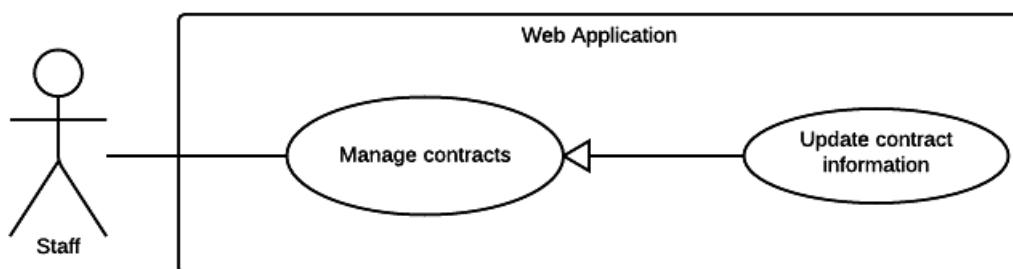


Figure 31 <Staff> Update contract information

USE CASE – WS11

Use Case No.	WS11	Use Case Version	2.0
Use Case Name	Update contract information		
Author	TrungDQ		
Date	27/05/2015	Priority	High
Actor:			
- Staff			
Summary:			
- This use case allows staff to update contract information.			
Goal:			
- Staff can update contract information.			
Triggers:			
- Staff sends command to update contract information.			
Preconditions:			
- User has to logged in to the system as Staff role			
- Contract is existed in the system			
- Contract status must not be "Cancelled"			
Post Conditions:			
- Success: Contract information will be updated			
- Fail: Show error message.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff goes to update contract information.	System list out information of contract: Contract general information - Contract's code: text - Contract's status: text - Contract's type: text - Contract's start date: text - Contract's expired date: text Vehicle information - Plate: free text input, required, length 4 – 15. - Brand: free text input, required, length 2 – 20. - Model code: free text input, length 2 – 20. - Vehicle type: free text input, length 2 – 20. - Color: free text input, length 2 – 20. - Engine: free text input, required, length 2 – 20. - Chassis: free text input, required, length 2 – 20. - Capacity: free text input, required, length 2 – 20. - Year of manufacture: free text input, value from 1900 to 2200. - Weight: free text input, value from 1 – 1000, unit: kilogram - Seat capacity: free text input, value from 1 – 100.	

2	Staff inputs required information.	
3	Staff sends command to save new information.	System shows message notify contract information is updated successfully. [Exception 1, 2]

Alternative Scenario: N/A

Exceptions:

Step	Actor Action	System Response
1	Staff sends command to save new information.	System shows error message to ask staff input missing required fields.
2	Entered vehicle plate is existed in the system	System shows error message to ask staff input another plate number.

Relationships: N/A

Business Rules:

- Contract information will be updated to system.
- Current contract information will be filled to corresponding input fields.

Table 34 Use case WS11 - <Staff> Update contract information

2.3.1.3.12. <Staff> Update compensation history

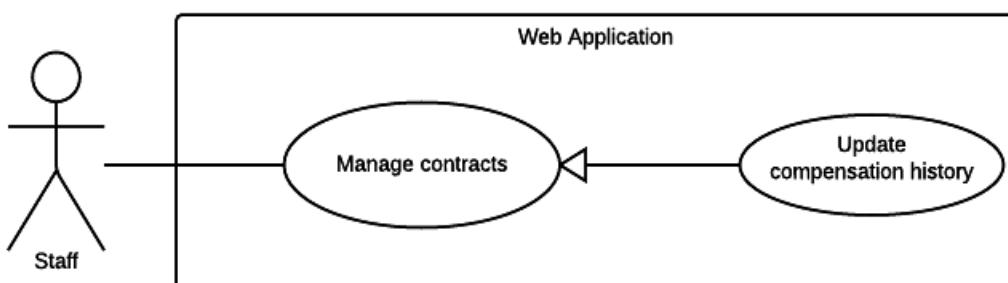


Figure 32 <Staff> Update compensation history

USE CASE – WS12			
Use Case No.	WS12	Use Case Version	2.0
Use Case Name	Update compensation history		
Author	TrungDQ		
Date	27/05/2015	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff to update compensation information. 		
Goal:	<ul style="list-style-type: none"> - Staff can update compensation information. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends command to update compensation information. 		
Preconditions:	<ul style="list-style-type: none"> - User has to logged in to the system as Staff role - Compensation is existed in the system 		

- If contract status is “Expired” or “Cancelled”, must not exceed update contract due date has been set up in the administrator’s configuration.

Post Conditions:

- **Success:** Compensation information will be updated
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to update compensation information.	<p>System lists out information of compensation:</p> <ul style="list-style-type: none"> - Contract code: free text input, required, length 6 – 10 - Created date: date time input, required. - Driver name: free text input, required, length 3 – 80. - Driver phone: free text input, required, length 8 – 15. - Driver address: free text input, required, length 3 – 250. - License number: free text input, required, length 10 – 15. - License type: free text input, required, length 1 – 10. - Plate: free text input, required, length 4 – 15. - Vehicle capacity: free text input, required, length 1 – 20. - Date: date time input, required. - Place: free text input, required, length 4 – 15. - Control department: free text input, required, length 3 – 250. - Description: free text input, required, length 1 – 2000. - Human damage: free text input, required, length 1 – 2000. - Asset damage: free text input, required, length 1 – 2000. - Observer: free text input, required, length 3 – 80. - Compensation note: free text input, required, length 1 – 2000. - Attachment: file upload input - Resolve date: date time input, required. - Decision: free text input, required, length 1 – 250. - Resolve note: free text input, required, length 1 – 2000.
2	Staff inputs information	

	3	Staff sends command to save new information.	System shows message notify compensation information is updated successfully. [Exception 1]
--	---	----------------------------------------------	------------------------------------------------------------------------------------------------

Alternative Scenario: N/A

Exceptions:

Step	Actor Action	System Response
1	Staff sends command to save new information.	System shows error message to ask staff input missing required fields.

Relationships: N/A

Business Rules:

- Compensation information will be updated to system.
- Staff can search and select a contract from available contract in the system.
- Created date must not exceed current date.
- Accident date must not exceed created date.
- If the compensation request has not been resolved yet, it would not have resolve date, decision and resolve note.

Table 35 Use case WS12 - <Staff> Update compensation history

2.3.1.3.13. <Staff> Update punishment history

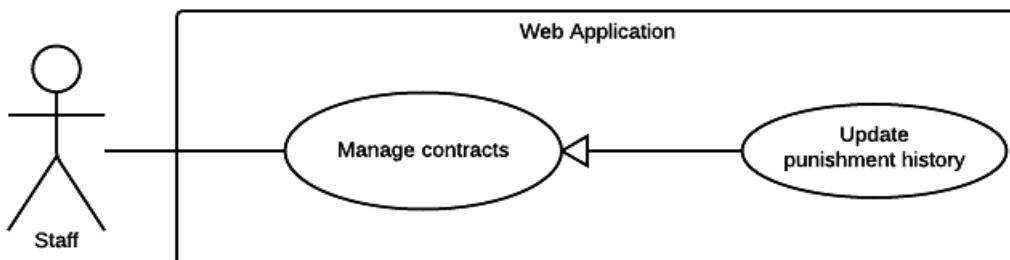


Figure 33 <Staff> Update punishment history

USE CASE – WS13			
Use Case No.	WS13	Use Case Version	2.0
Use Case Name	Update punishment history		
Author	TrungDQ		
Date	27/05/2015	Priority	High
Actor:			
- Staff			
Summary:			
- This use case allows staff to update punishment information.			
Goal:			
- Staff can update punishment information.			
Triggers:			
- Staff sends command to update punishment information.			
Preconditions:			
- User has to logged in to the system as Staff role			
- Punishment is existed in the system			
- If contract status is “Expired” or “Cancelled”, must not exceed update contract due date has been set up in the administrator’s configuration.			

Post Conditions:

- **Success:** Punishment information will be updated
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to update punishment information.	System lists out information of punishment: - Date: date time input, required. - Title: free text input, required, length 1 – 250. - Attachment: file upload input, required.
2	Staff inputs information	
3	Staff sends command to save new information.	System shows message notify punishment information is updated successfully. [Exception 1]

Alternative Scenario: N/A

Exceptions:

Step	Actor Action	System Response
1	Staff sends command to save new information.	System shows error message to ask staff input missing required fields.

Relationships: N/A

Business Rules:

- Punishment information will be updated to system.
- Punishment date must not exceed current date.

Table 36 Use case WS13 - <Staff> Update punishment history

2.3.1.3.14. <Staff> Update accident history

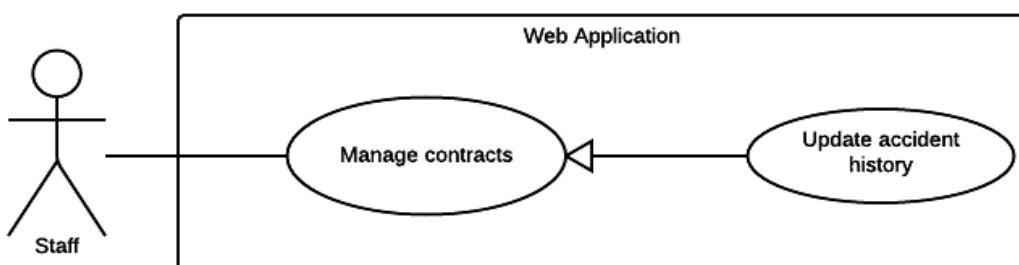


Figure 34 <Staff> Update accident history

USE CASE – WS14

Use Case No.	WS14	Use Case Version	2.0
Use Case Name	Update accident history		
Author	TrungDQ		
Date	27/05/2015	Priority	High
Actor:			
- Staff			
Summary:			
- This use case allows staff to update accident information.			

Goal:

- Staff can update accident information.

Triggers:

- Staff sends command to update accident information.

Preconditions:

- User has to logged in to the system as Staff role
- Accident is existed in the system
- If contract status is “Expired” or “Cancelled”, must not exceed update contract due date has been set up in the administrator’s configuration.

Post Conditions:

- **Success:** Punishment information will be updated
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff goes to accident information.	System list out information of accident: - Date: date time input, required. - Title: free text input, required, length 1 – 250. - Attachment: file upload input, required.
2	Staff inputs information.	
3	Staff sends command to save new information.	System shows message notify compensation information is updated successfully. [Exception 1]

Alternative Scenario: N/A

Exceptions:

Step	Actor Action	System Response
1	Staff sends command to save new information.	System shows error message to ask staff input missing required fields.

Relationships: N/A

Business Rules:

- Accident information will be updated to the system.
- Accident date must not exceed current date.

Table 37 Use case WS14 - <Staff> Update accident history

2.3.1.3.15. <Staff> View cards information

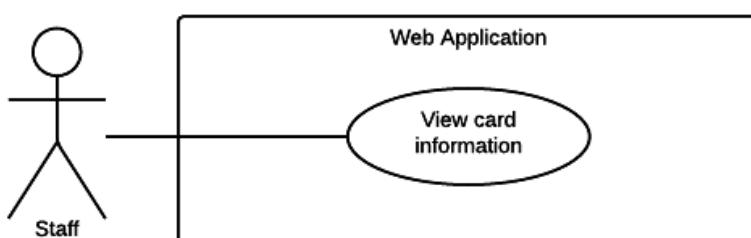


Figure 35 <Staff> View card information

USE CASE - WS15			
Use Case No.	WS15	Use Case Version	2.0

Use Case Name	View cards information				
Author	TrungDQ				
Date	27/05/2015	Priority	High		
Actor:					
- Staff					
Summary:					
- This use case allows staff to view published cards information.					
Goal:					
- Staff can view cards information.					
Triggers:					
- Staff goes to view cards information.					
Preconditions:					
- User has to logged in to the system as Staff role					
- Cards information is existed in the system					
Post Conditions:					
- Success: Cards information is shown					
- Fail: Show error message.					
Main Success Scenario:					
Step	Actor Action	System Response			
1	Staff goes to view cards information	System list out information of published cards: - Card ID: link to card detail - Card owner: link to customer detail - Contract code: link to contract detail - Activated date: text - Status: text [Exception 1]			
2	Staff sends command to view card detail information	System shows card detail information: - Card ID: text - Status: text - Card owner: link to customer detail - Contract code: link to contract detail - Activated date: text Access history: - Access date: text - Device: text - Request service: text - Responde result: text Published history - Contract code: link to contract detail - Activated date: text - Deactivated date: text [Exception 2]			
Alternative Scenario: N/A					
Exceptions:					
Step	Actor Action	System Response			
1	Staff goes to view cards information	System shows no connection error message.			

2	Staff sends command to view card detail information	System shows no connection error message.
---	-----------------------------------------------------	-------------------------------------------

Relationships:

Business Rules:

- Cards information will be loaded from the system.
- History of access will show 10 results at a time, with pagination links at the bottom.

Table 38 Use case WS15 - <Staff> View cards information

2.3.1.4.<Admin> Overview Use Case

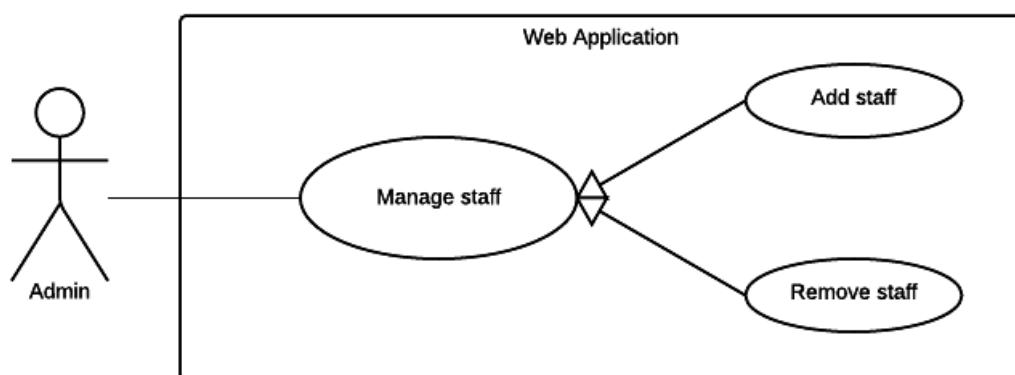


Figure 36 <Admin> Overview Use Case

2.3.1.4.1. <Admin> Update contract type information

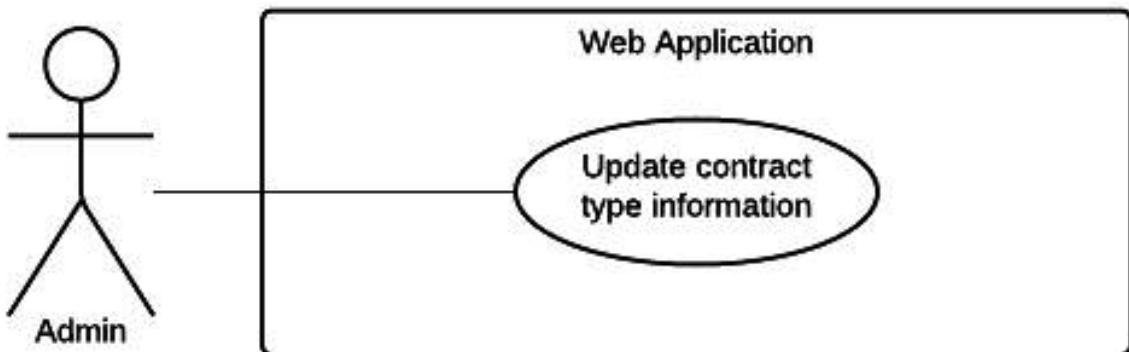


Figure 37 <Staff> Update contract type information

USE CASE – WS16			
Use Case No.	WS16	Use Case Version	2.0
Use Case Name	Update contract type information		
Author	TrungDQ		
Date	27/05/2015	Priority	High
Actor:			
- Admin			
Summary:			
- This use case allows admin to update contract type.			
Goal:			

- Admin can update contract type.

Triggers:

- Admin sends command to update contract type.

Preconditions:

- User has to logged in to the system as Admin role
- Contract type is existed in the system

Post Conditions:

- **Success:** Contract type information will be updated
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin goes to view contract type information.	System list out information of contract type: - Name: free text input, required, length 1 – 250. - Description: free text input, required, length 1 – 2000. - Price per year: free number input, required, value from 0 to 1 billion, unit: VNĐ.
2	Admin inputs information	
3	Admin sends command to save new information.	System shows message notify contract type information is updated successfully. [Exception 1,2]

Alternative Scenario: N/A

Exceptions:

Step	Actor Action	System Response
1	Admin sends command to save new information.	System shows message notify missing required fields.

Relationships: N/A

Business Rules:

- Contract type information will be updated to system.
- Contract type price per year must be greater than 0 and less than 1 billion.
- When a contract type is deactivated, all the contract which having the contract type is still remain working but will not be able to renew.
- When a contract type is deactivated, it will not be shown at the create contract page / contract type options.

Table 39 Use case WS16 - <Staff> Update contract type information

2.3.1.4.2. < Admin> Remove staff

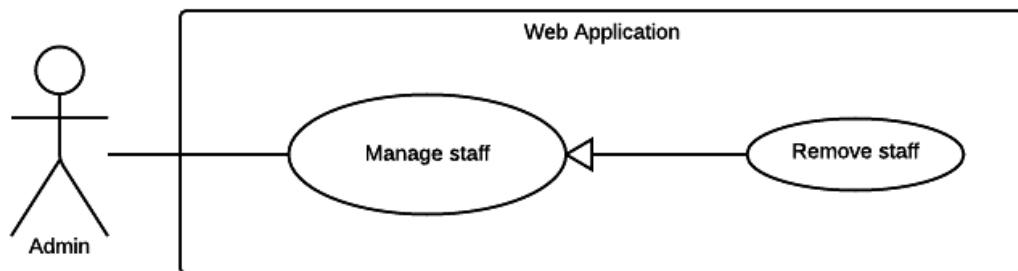


Figure 38 <Admin> Remove staff

USE CASE – WA01					
Use Case No.	WA01	Use Case Version	2.0		
Use Case Name	Remove staff				
Author	TrungDQ				
Date	27/05/2015	Priority	High		
Actor:					
- Admin					
Summary:					
- This use case allows admin to remove other staff from the system.					
Goal:					
- Admin can remove staff from the system.					
Triggers:					
- Admin sends command to remove staff from the system.					
Preconditions:					
- User has to logged in to the system as Admin role					
Post Conditions:					
- Success: Admin be able to remove staff from the system					
- Fail: Show error message.					
Main Success Scenario:					
Step	Actor Action	System Response			
1	Admin goes to manage members	System list out information of staffs in system: - Staff code: text - Name: text - Email: text - Phone number: text			
2	Admin select the staff to delete				
3	Admin sends command to delete staff.	System shows message staff has been removed successfully. [Exception 1]			
Alternative Scenario: N/A					
Exceptions:					
Step	Actor Action	System Response			

1	Admin sends command to delete staff.	System shows error message user cannot remove themselves from the system.
---	--------------------------------------	---------------------------------------------------------------------------

Relationships: N/A

Business Rules:

- Their must be at least 01 staff in the system.
- Staff account will be deactivated.
- All information about staff still remain in the system.

Table 40 Use case WA01 - <Admin> Remove staff

2.3.1.4.3. < Admin> Add staff

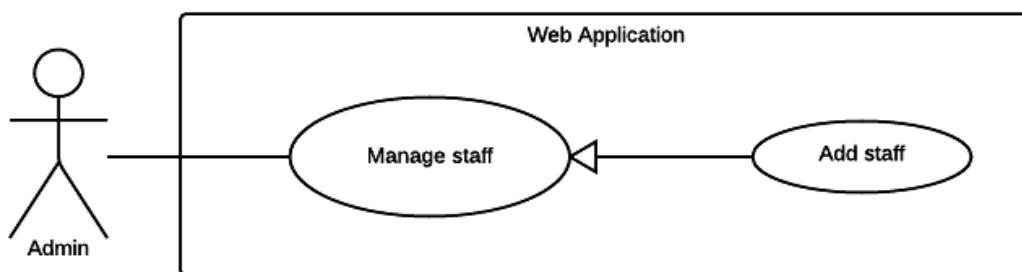


Figure 39 <Admin> Add staff

USE CASE - WA02									
Use Case No.	WA02	Use Case Version	2.0						
Use Case Name	Add staff								
Author	TrungDQ								
Date	27/05/2015	Priority	High						
Actor:	<ul style="list-style-type: none"> - Admin 								
Summary:	<ul style="list-style-type: none"> - This use case allows admin to add new staff to the system. 								
Goal:	<ul style="list-style-type: none"> - Admin can add new staff to the system. 								
Triggers:	<ul style="list-style-type: none"> - Admin sends command to add new staff to the system. 								
Preconditions:	<ul style="list-style-type: none"> - User has to logged in to the system as Admin role 								
Post Conditions:	<ul style="list-style-type: none"> - Success: New staff is added into the system - Fail: Show error message. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Admin goes to add staff page.</td><td> System require information of staff: - Email: text, required, length 3 - 250. </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Admin goes to add staff page.	System require information of staff: - Email: text, required, length 3 - 250.
Step	Actor Action	System Response							
1	Admin goes to add staff page.	System require information of staff: - Email: text, required, length 3 - 250.							

		<ul style="list-style-type: none"> - Password: text, required, length 2 - 32. - Name: text, required, length 3 – 80. - Phone number: text, required, length 8 – 15.
2	Admin inputs information	
3	Admin sends command to add new staff.	System shows message staff has been added successfully. [Exception 1]
Alternative Scenario: N/A		
Exceptions:		
Step	Actor Action	System Response
1	Admin sends command to add other staff.	System shows error message to ask user to enter missing required field.
Relationships: N/A		
Business Rules:		
<ul style="list-style-type: none"> - Staff code is automatic initialized by the system. - Staff email must not be duplicated. 		

Table 41 Use case WA02 - <Admin> Add staff

2.3.1.5.<Scheduler> Overview Use Case

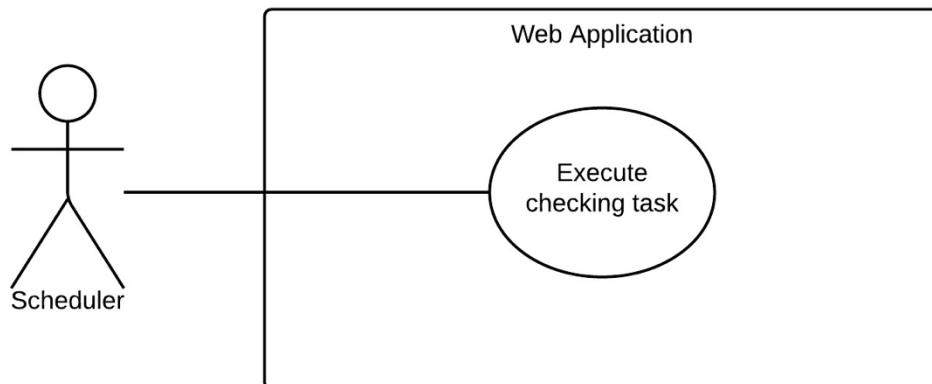


Figure 40 <System> Overview Use Case

2.3.1.5.1. <Scheduler> Execute checking task

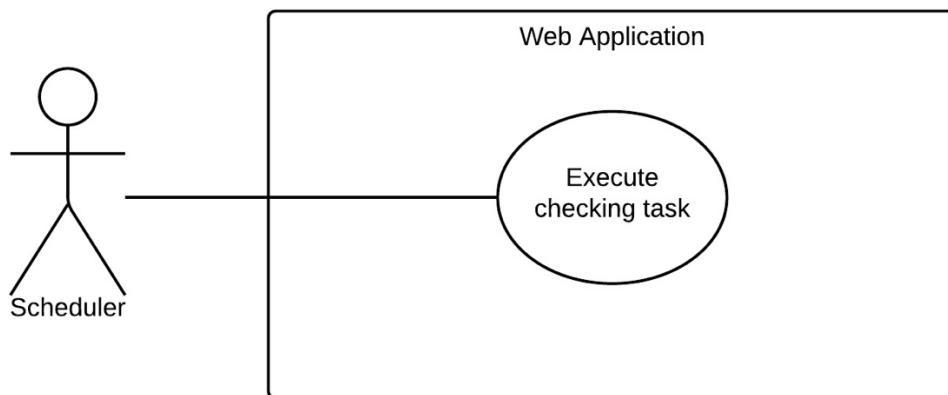


Figure 41 <System> Notify schedule

USE CASE - WY01			
Use Case No.	WY01	Use Case Version	2.0
Use Case Name	Execute checking task		
Author	TrungDQ		
Date	26/05/2015	Priority	Medium
Actor:			
- Scheduler			
Summary:			
- This use case describes how the system run a schedule task to notify to users when necessary.			
Goal:			
- System will able to update status of contract, compensation and send notification to users.			
Triggers:			
- System run a timer task that trigger check event.			
Preconditions:			
- System time is at 00:00 AM.			
- There must be at least one contract in the system.			
Post Conditions:			
- Success: Show the status of contract and notify to user.			
- Fail: System display error message.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	System run timer task to check contract state and sends notify	System response: - List of contract status that need to be change. - List of compensation status that need to be change. - Notifications that need to be sent to users. [Exception 1]	
Exceptions:			
No	Actor Action	System Response	

1	System timer task is interrupted	No notification will be sent. Error detail will be tracked in a log file.
---	----------------------------------	---------------------------------------------------------------------------

Relationships: N/A

Business rules:

- System timer will send check event at 00:00 AM everyday.
- If a contract has expired date remaining is less than a number of days specified by admin, system will send notification to users to ask users to renew their contract. This notification will repeat 2 more times in the specified days if the user do not renew the contract.
- If a contract exceeded expired date, the contract status will be set to "Expired".
- If a contract is not paid within a number of days specified by the system, the contract status will be set to "Cancelled".
- If a pending contract's start date arrived, the contract will be started. And the contract status will be set to "No Card" or "Ready".
- If there is new contract, cancel contract request, new card request or new compensation, notifications will be sent to staffs. And if staff resolved the requests, notifications will be sent to customers.

Table 42 Use case WY01 - <Scheduler> Execute checking task

2.3.1.6.<Payment system> Overview Use Case

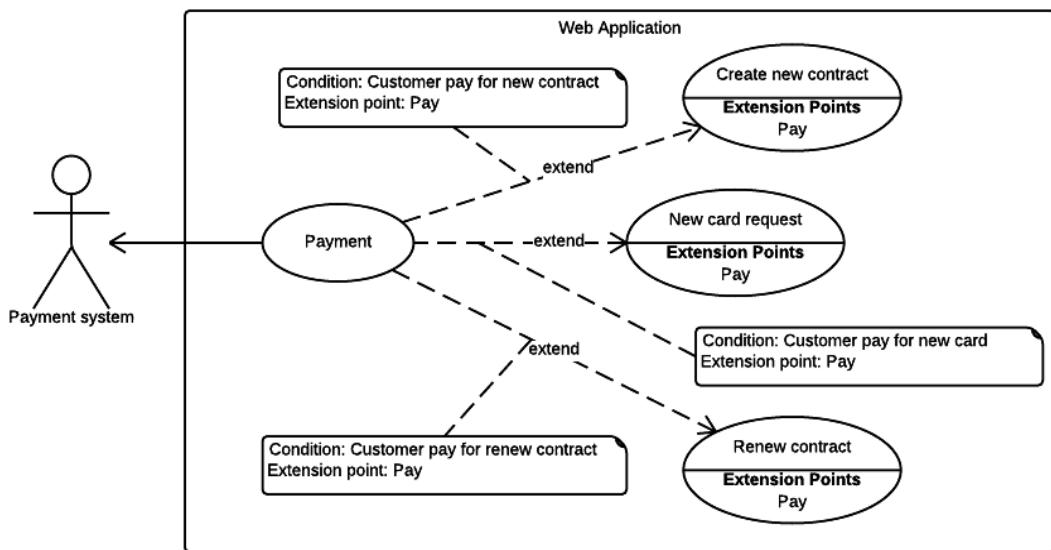


Figure 42 <Payment> Payment

2.3.1.6.1. <Payment system> Payment

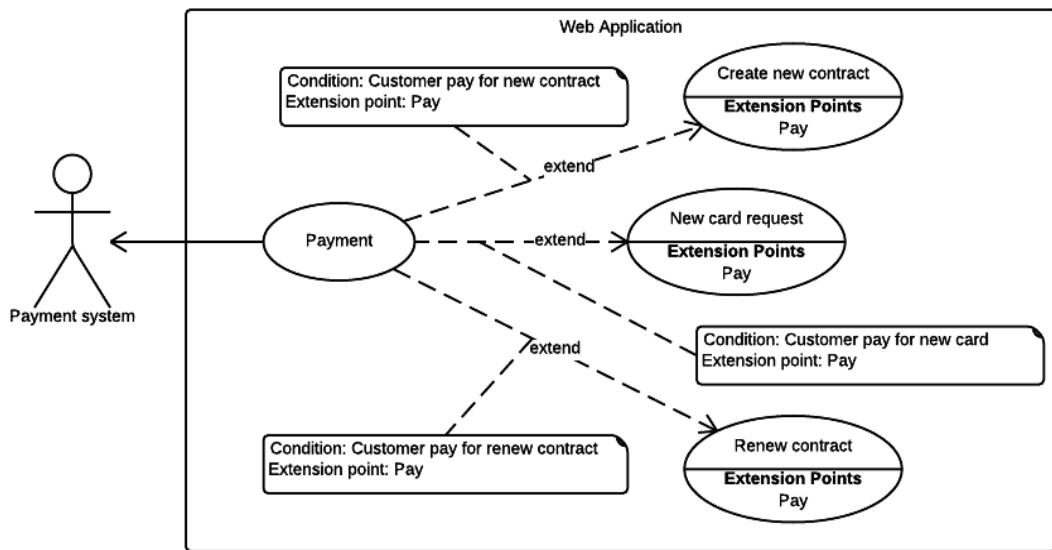


Figure 43 <Payment> Payment

USE CASE – WP01			
Use Case No.	WP01	Use Case Version	2.0
Use Case Name	Payment		
Author	KhaNC		
Date	03/06/2015	Priority	High
Actor:			
- Payment system.			
Summary:			
- This use case allows payment system process the payment transaction.			
Goal:			
- Payment process is completed.			
Triggers:			
- Customer sends pay for charge by payment system command.			
Preconditions:			
- User must login into the system with role Customer.			
Post Conditions:			
- Success: Payment is completed successfully. - Fail: Show error message.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Customer select payment via PayPal.	Summary of customer payment: - Description: text. - Amount: text. - Total: text. Payment view is shown with following labels and fields: - Customer's PayPal account: free text input, required, length 3 – 250	

		- Customer's PayPal password: free text input, required, length 6 – 32
2	Staff fills out the form.	
3	Staff sends pay command.	<ul style="list-style-type: none"> - Validate data [Alternative 1] [Exception 1, 2, 3] - Process payment in PayPal

Alternative Scenario:

No	Actor Action	System Response
1	Identity information is not match.	Show message notify customer input wrong email or password.

Exceptions:

No	Actor Action	System Response
1	Missing of required fields	Show message notify staff input missed fields
2	Length of field's value is out of range	Show message notify staff which field's value is out of range
3	Entered email address is not a valid email	Show message notify entered email is not valid

Relationships:

- <Customer> Create new contract
- <Customer> New card request
- <Customer> Renew contract

Business Rules:

- Use PayPal API to validate customer email and password.
- PayPal will send notification to alert transaction is failed or succeeded.
- An email address must be validated by this regular expression:
 $/^([a-zA-Z_\\.-]+)@([\\da-zA-Z\\.-]+)\\.([a-zA-Z]{2,6})$/$

Table 43 Use case WP01 - <Payment> Payment

2.3.2. Checker Mobile Application

2.3.2.1.<Police> Overview Use Case

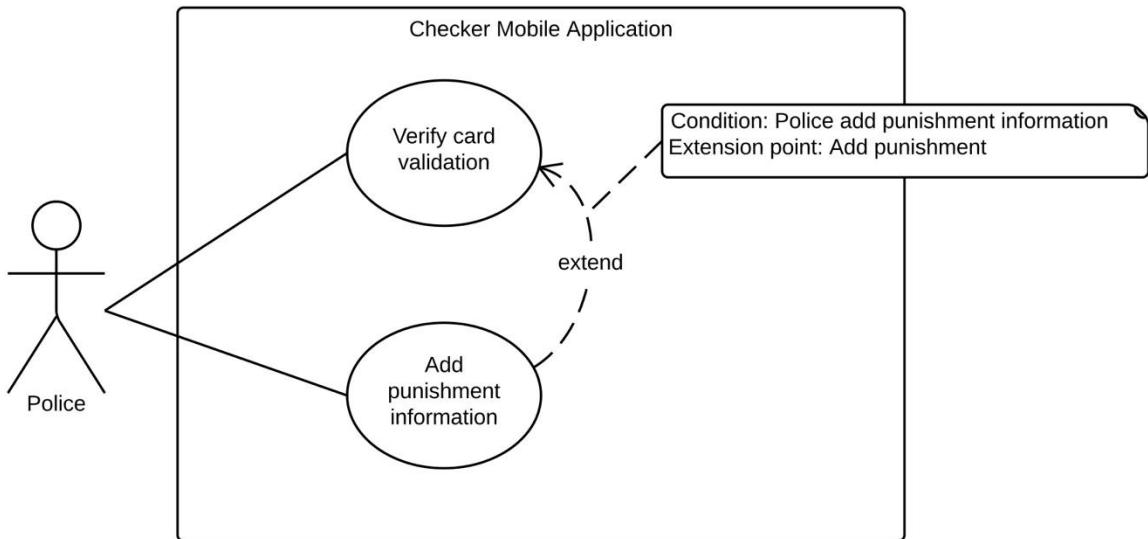


Figure 44 <Police> Overview Use Case

2.3.2.1.1. <Police> Verify card information

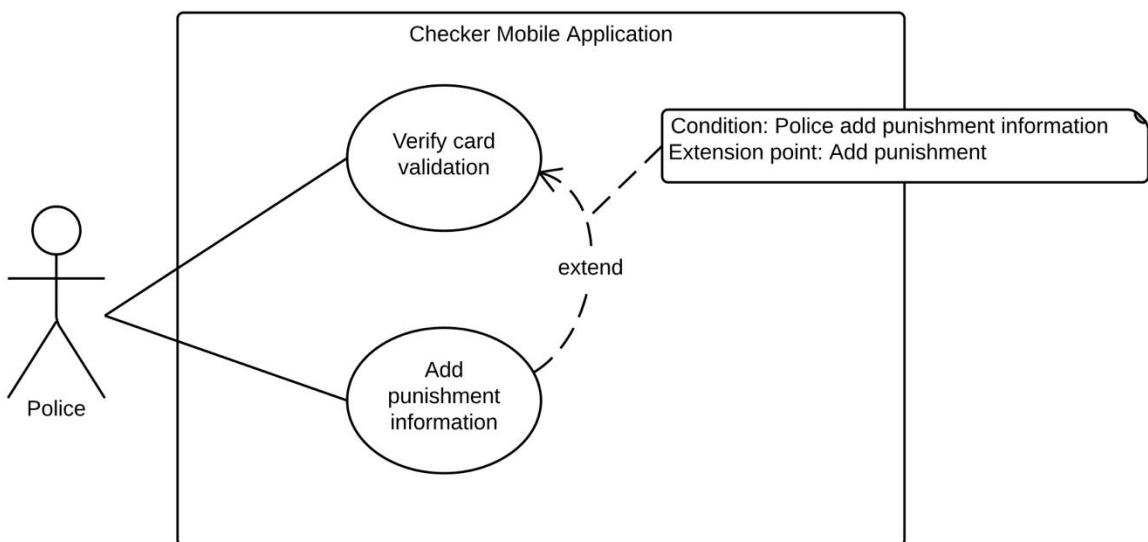


Figure 45 <Police> Verify card information

USE CASE – CP01			
Use Case No.	CP01	Use Case Version	2.0
Use Case Name	Verify card information		
Author	PhucNH		
Date	20/06/2015	Priority	High
Actor:	- Police		
Summary:			

- Traffic Police and Police Department can use the device to view card's information and check if the card is valid or not.

Goal:

- Check if the insurance card is valid or not.

Triggers:

- Police put the NFC card near the device to read card information.

Preconditions:

- Device NFC is turned on
- Device is connected to the internet

Post Conditions:

- **Success:** Show the insurance card information.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Police put the NFC card close to the device.	<p>Show the insurance contract and the motor information:</p> <ul style="list-style-type: none"> - Motor owner: text. - Owner address: text. - Phone number: text. - Plate number: text. - Engine: text. - Chassis: text. - Brand: text. - Model Code: text. - Type: text. - Color: text. - Capacity: text. - Year of manufacture: text. - Weight: text. - Seat capacity: text. - Contract fee: text. - Start date: text - Expired date: text - Card Status: text. <p>[Alternative 1] [Exception 1, 2]</p>

Alternative Scenario:

No	Actor Action	System Response
1	If unable to read the card or the card is invalid	Mobile app shows error to notify police that mobile cannot read this card

Exceptions:

No	Actor Action	System Response
1	Cannot read card	Mobile app notifies that cannot read card
2	Cannot connect to server.	Mobile app shows error connect to server is fail

Relationships: <Police> Add punishment (CP02)

Business Rules:

- Mobile application sends request about card information to server and receives information about contract.
- A card is “Valid” if all of the following conditions are true:
 - o The card ID is in activated status.
 - o Contract has status “Ready” or “Request cancel”
- A card is “Nearly expired” if the card is “Valid” and the contract expired day remaining is less than a number of days specified by the system.
- Highlight the status of the card include:
 - o Valid card
 - o Expired card
 - o Card is nearly expired, show remaining days
- Verify card activity and device ID will be logged to card access log in the system.

Table 44 Use case CP01 - <Police> Verify card information

2.3.2.1.2. <Police> Add punishment information

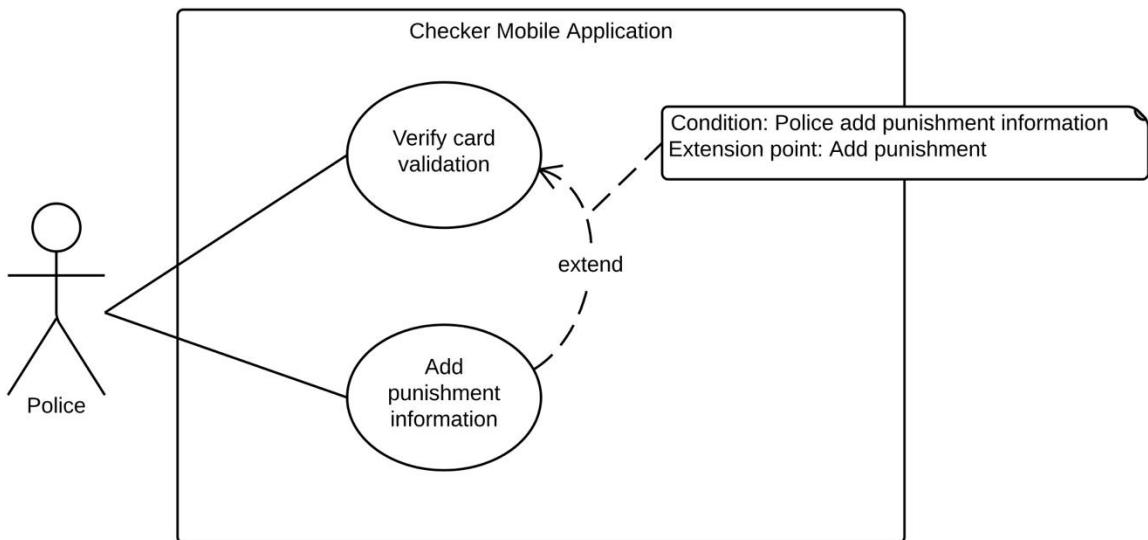


Figure 46 <Police> Add punishment information

USE CASE – CP02			
Use Case No.	CP02	Use Case Version	2.0
Use Case Name	Add punishment information		
Author	PhucNH		
Date	26/05/2015	Priority	High
Actor:	<ul style="list-style-type: none"> - Police 		
Summary:	<ul style="list-style-type: none"> - Traffic Police and Police Department can use the device to add punishment information of the driver. 		
Goal:	<ul style="list-style-type: none"> - Add punishment information of the driver. 		
Triggers:	<ul style="list-style-type: none"> - Device NFC is turned on 		

- Device is connected to the internet		
- Reads card successful.		
Preconditions:		
- The card is verified.		
Post Conditions:		
- Success: Add punishment information of the driver.		
- Fail: Show error message.		
Main Success Scenario:		
Step	Actor Action	System Response
1	Police chooses new punishment	Mobile required police input description about punishment and picture about it. - Description: free text input, length 1 - 200 Picture: file upload input, required
2	Police fill the punishment information and sends add punishment command	Mobile sends punishment from police and notify this action is success [Exception 1, 2]
Exceptions:		
No	Actor Action	System Response
1	Police input missed one of punishment description and picture of punishment record.	Mobile reminds input required information.
2	Cannot connect to server.	Show error message to notify that can not connect to server.
Relationships: <Police> Verify card information (CP01)		
Business Rules:		
<ul style="list-style-type: none"> - Information about punishment will be sent in server - Punishment information is updated to contract of user. - Picture will be resized to 1000 width to minimize the data sent to server and better performance. - Add punishment activity and device ID will be logged to card access log in the system. 		

Table 45 Use case CP02 - <Police> Add punishment information

2.3.3. Printer Mobile Application

2.3.3.1.<Staff> Overview Use Case

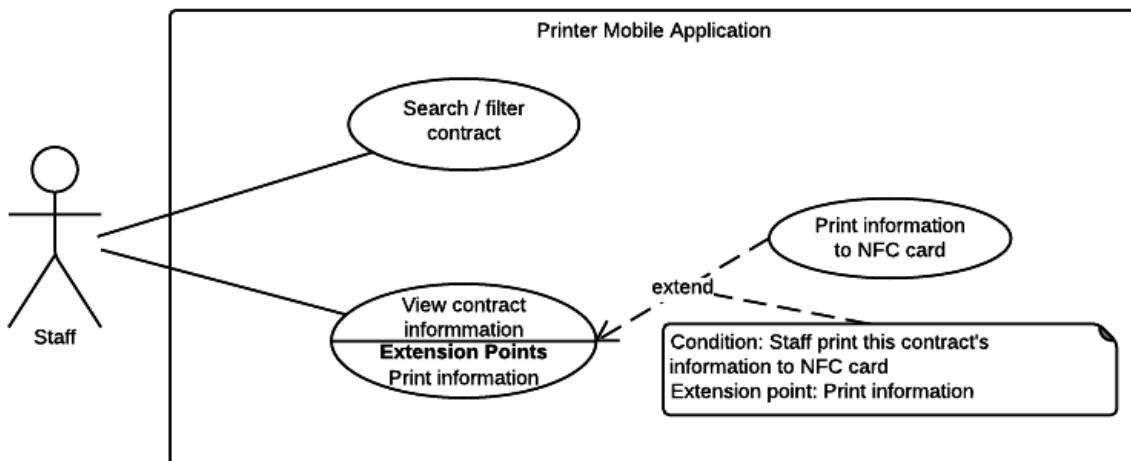


Figure 47 <Staff> Overview Use Case

2.3.3.1.1. <Staff> Search / filter contract

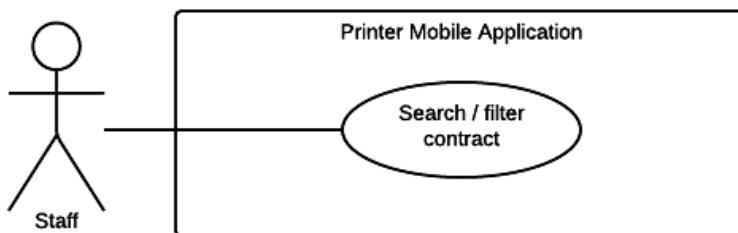


Figure 48 <Staff> Search / filter contract

USE CASE – PS01			
Use Case No.	PS01	Use Case Version	2.0
Use Case Name	Search / filter contract		
Author	PhucNH		
Date	26/05/2015	Priority	Medium
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case helps staff can search or filter the information of contract by contract code or customer name. 		
Goal:	<ul style="list-style-type: none"> - The information of contract is shown following the request search of staff. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends command to search/ filter contract. 		
Preconditions:	<ul style="list-style-type: none"> - Device is connected to the internet - There must be at least 1 contract in the system 		
Post Conditions:	<ul style="list-style-type: none"> - Success: The information of request search is shown. - Fail: System display error message. 		

Main Success Scenario:		
Step	Actor Action	System Response
1	Staff enter keyword	
2	Staff sends command to search/ filter contract Staff chooses contract	System shows list result: Information about request search. System shows information about contract: - Contract code: text - Customer name: text - Status: color [Alternative 1] [Exception 1]
Alternative Scenario:		
No	Actor Action	System Response
1	System cannot finds the result from request	System notifies cannot find the request search.
Exceptions:		
No	Actor Action	System Response
1	The request search is not valid, includes: - Invalid the value of textbox - Fill some especial character	Website show message to notify.
Relationships: N/A		
Business Rules:		
<ul style="list-style-type: none"> - Staff can search by customer name - Staff can search by contract code - Search algorithm: return all contract that have customer name or contract code contains the keyword. - If keyword is empty, system returns 10 latest contracts. - In the result list, contracts having stats “No card” will be listed first. - Each contract in the list has an identify color to display contract status. <ul style="list-style-type: none"> o Gray: Pending o Blue: No card o Green: Ready o Red: Expired o Pink: Request cancel o Black: Cancelled 		

Table 46 Use case PS01 - <Staff> Search / filter contract

2.3.3.1.2. <Staff> View contract information

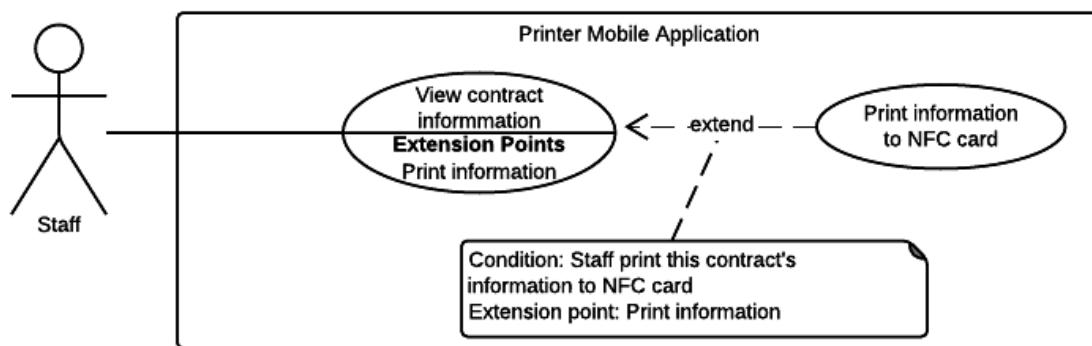


Figure 49 <Staff> View contract information

USE CASE - PS02									
Use Case No.	PS02	Use Case Version	2.0						
Use Case Name	View contract information								
Author	PhucNH								
Date	21/05/2015	Priority	Medium						
Actor:	<ul style="list-style-type: none"> - Staff 								
Summary:	<ul style="list-style-type: none"> - This use case helps staff view contract's information. 								
Goal:	<ul style="list-style-type: none"> - Staff can view the contract's information. 								
Triggers:	<ul style="list-style-type: none"> - Staff send view contract information command. 								
Preconditions:	<ul style="list-style-type: none"> - Device is connected to the internet 								
Post Conditions:	<ul style="list-style-type: none"> - Success: Show contract's information. - Fail: Show error message. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Staff send view contract information command.</td><td> Show information about the contract: <ul style="list-style-type: none"> - Motor owner name: text. - Owner address: text. - Phone number: text - Owner ID/passport number. - Contract code: text - Plate number: text. - Engine: text. - Chassis: text. - Brand: text. - Model Code: text. - Type: text. - Color: text. - Capacity: text. </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Staff send view contract information command.	Show information about the contract: <ul style="list-style-type: none"> - Motor owner name: text. - Owner address: text. - Phone number: text - Owner ID/passport number. - Contract code: text - Plate number: text. - Engine: text. - Chassis: text. - Brand: text. - Model Code: text. - Type: text. - Color: text. - Capacity: text.
Step	Actor Action	System Response							
1	Staff send view contract information command.	Show information about the contract: <ul style="list-style-type: none"> - Motor owner name: text. - Owner address: text. - Phone number: text - Owner ID/passport number. - Contract code: text - Plate number: text. - Engine: text. - Chassis: text. - Brand: text. - Model Code: text. - Type: text. - Color: text. - Capacity: text. 							

		<ul style="list-style-type: none"> - Year of manufacture: text - Weight: text. - Seat capacity: text. - Contract fee: text. - Start date: text - Expired date: text
Exceptions:		
No	Actor Action	System Response
1	Cannot connect to server.	System shows error that cannot connect to server.

Relationships: <Police> Print information to NFC card (PS03)

Business Rules:

- The status and the remaining day (if any) of the contract will be highlighted.
- Contract has an identify color to display contract status.
 - o Gray: Pending
 - o Blue: No card
 - o Green: Ready
 - o Red: Expired
 - o Pink: Request cancel
 - o Black: Cancelled
- Contract information must exactly same with contract information in the system.
- Contract start date and expired date is formatted as dd/MM/yyyy
- If information is too long that cannot fit the screen, there is “...” text at the end of the line.

Table 47 Use case PS02 - <Staff> View contract information

2.3.3.1.3. <Staff> Print information to NFC card

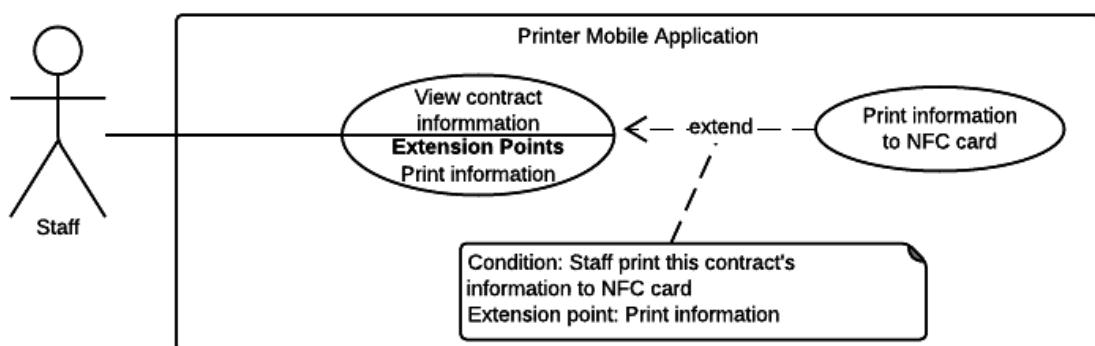


Figure 50 <Staff> Print information to NFC card

USE CASE – PS03			
Use Case No.	PS03	Use Case Version	2.0
Use Case Name	Print information to NFC card		
Author	PhucNH		
Date	26/05/2015	Priority	High

Actor:												
- Staff												
Summary:												
- This use case helps staff can print information of contract into NFC card.												
Goal:												
- The information of NFC can be printed from application on mobile												
Triggers:												
- Staff sends print information into card command.												
Preconditions:												
- Device must be connected to the internet. - View contract information succeed. - Device NFC must be turned on.												
Post Conditions:												
- Success: Print information to NFC card. - Fail: Show error message.												
Main Success Scenario:												
<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Staff put the NFC card nearly the device and sends print information into card command</td><td>Show message to notify print successful. [Alternative 1,2,3] [Exception 1]</td></tr> </tbody> </table>	Step	Actor Action	System Response	1	Staff put the NFC card nearly the device and sends print information into card command	Show message to notify print successful. [Alternative 1,2,3] [Exception 1]						
Step	Actor Action	System Response										
1	Staff put the NFC card nearly the device and sends print information into card command	Show message to notify print successful. [Alternative 1,2,3] [Exception 1]										
Alternative Scenario:												
<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>If the contract is cancelled, expired or not paid.</td><td>Show message to notify that can't print card for that contract.</td></tr> <tr> <td>2</td><td>The NFC card is already issued for another contract.</td><td>Show message to notify that the NFC card is already issued for another contract.</td></tr> <tr> <td>3</td><td>Can't find NFC card</td><td>Show message to notify that can not find NFC card.</td></tr> </tbody> </table>	No	Actor Action	System Response	1	If the contract is cancelled, expired or not paid.	Show message to notify that can't print card for that contract.	2	The NFC card is already issued for another contract.	Show message to notify that the NFC card is already issued for another contract.	3	Can't find NFC card	Show message to notify that can not find NFC card.
No	Actor Action	System Response										
1	If the contract is cancelled, expired or not paid.	Show message to notify that can't print card for that contract.										
2	The NFC card is already issued for another contract.	Show message to notify that the NFC card is already issued for another contract.										
3	Can't find NFC card	Show message to notify that can not find NFC card.										
Exceptions:												
<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Can not connect to server.</td><td>Show error message to notify that can not connect to server.</td></tr> </tbody> </table>	No	Actor Action	System Response	1	Can not connect to server.	Show error message to notify that can not connect to server.						
No	Actor Action	System Response										
1	Can not connect to server.	Show error message to notify that can not connect to server.										
Relationships: <Staff> View contract information (PS02)												
Business Rules:												
<ul style="list-style-type: none"> - If the card is already existed in system, it can only be printed when the card is recycled by staff before. - If there is a new card request for current contract, the new card request status will be set to done. - Following contracts will not be able to print card: <ul style="list-style-type: none"> o Pending (no payment) o Cancelled - If the contract has no card before, the contract status will be changed from "No card" to "Ready". - If the contract has a card before, the old card will be deactivated. - Card access log and history will be logged separately for new card and old card. 												

Table 48 Use case PS03 - <Staff> Print information to NFC card

3. Software System Attribute

3.1. Usability

3.1.1. Graphic User Interface

All the texts, labels and alerts will be written in Vietnamese.

3.1.2. Usability

- The system usability is easy to use that will need less than 3 days of training for company staffs to use the system.
- Police officers need less than 1 hours of training to use the mobile app.

3.1.3. Installation

- User can follow installation and manual guide for installation. If there are any problems, user can contact developer for help.

3.2. Reliability

- System notification success rate is less than 2 failed notifications per 1000 sent.
- System email notification delivery success rate is less than 2 failed deliveries per 1000 sent.
- Scheduler task runs at 00:00 everyday with 100% execution rate.
- Web service API response success rate is less than 2 failed requests per 10,000 requests.

3.3. Availability

- N/A

3.4. Security

- All input data are validated before saving to database.
- All privacy information such as password is encrypted to ensure security.
- Users are authenticated/authorized for all users when they log in to the system.

3.5. Maintainability

- The system is separated into modules.

3.6. Portability

- Admin, staff and customer can use application on every OS supported web browser.
- Police officers and staffs can use mobile application on every Android smartphone that have version greater than 4.0.

3.7. Performance

- Requests from web application are responded in less than 10 seconds at 8 Mbps bandwidth speed.
- Mobile checker application should return card information in less than 1 minute at 8 Mbps bandwidth speed.
- Mobile printer application should write data to NFC card successfully in less than 1 minute at 8 Mbps bandwidth speed.

4. Conceptual Diagram

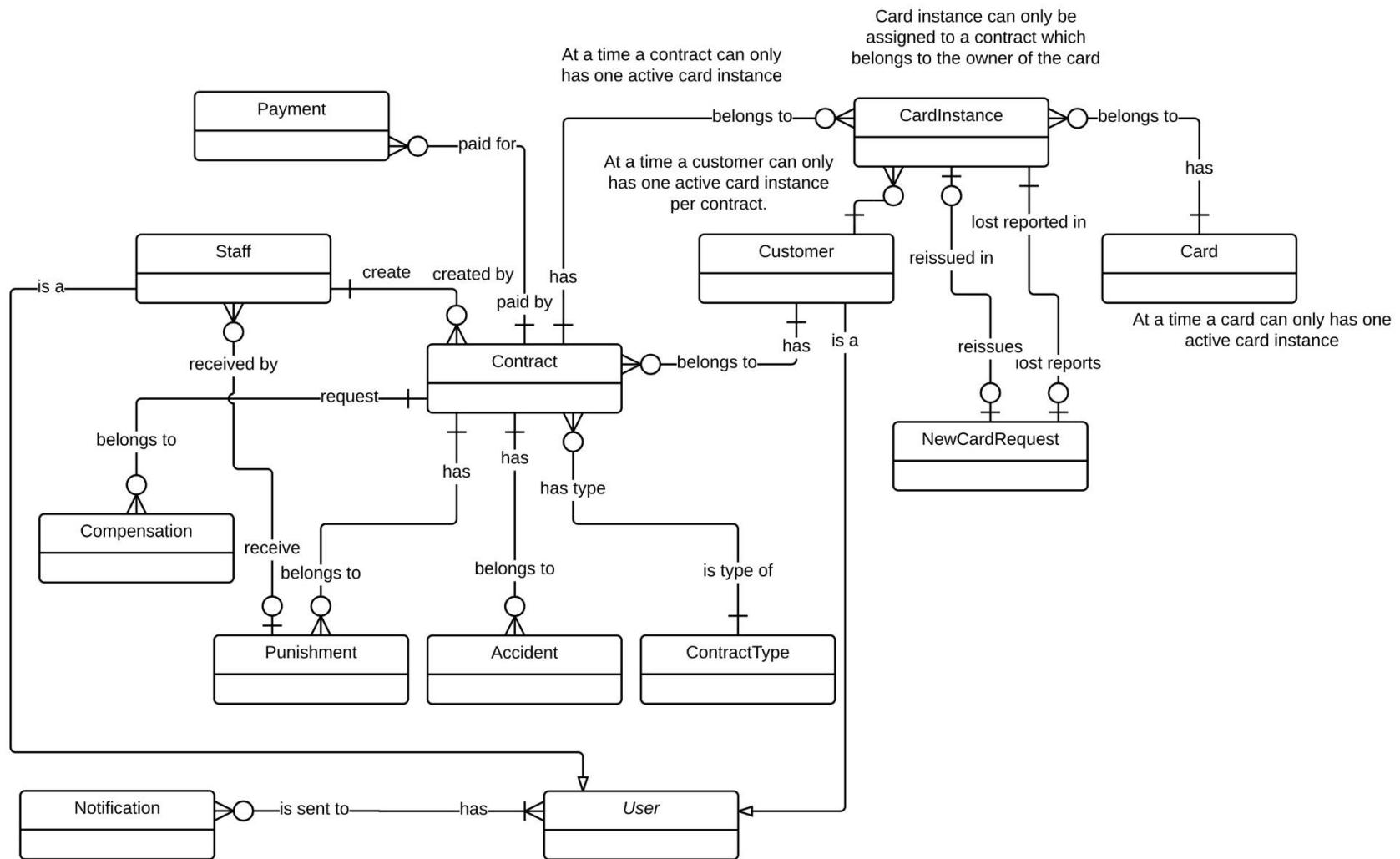


Figure 51 Conceptual diagram

Data Dictionary

Entity Data dictionary: describe all content of all entities	
Entity Name	Description
User	Abstract entity describes a user in system
Customer	Contain the customer information.
Contract	Contain the contract information.
Card	Contain the card information
CardInstance	Represent a card assigned to a contract
Payment	Contain the payment information.
Staff	Contain the staff information.
Compensation	Contain the compensation information.
Punishment	Contain the punishment information.
Accident	Contain the accident information.
ContractType	Contain the contract type information.
NewCardRequest	Contain the new card request information.
Notification	Contain the notification information

Table 49 Conceptual Diagram Data Dictionary

D. Software Design Description

1. Design Overview

- This document describes the technical and user interface design of MIC system. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
- The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
- The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
- The database design describes the relationships between entities and details of each entity.
- Document overview:
 - o Section 2: gives an overall description of the system architecture design.
 - o Section 3: gives component diagrams that describe the connection and integration of the system.
 - o Section 4: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
 - o Section 5: describe a fully attributed Entity Relationship Diagram.

2. System Architecture Design

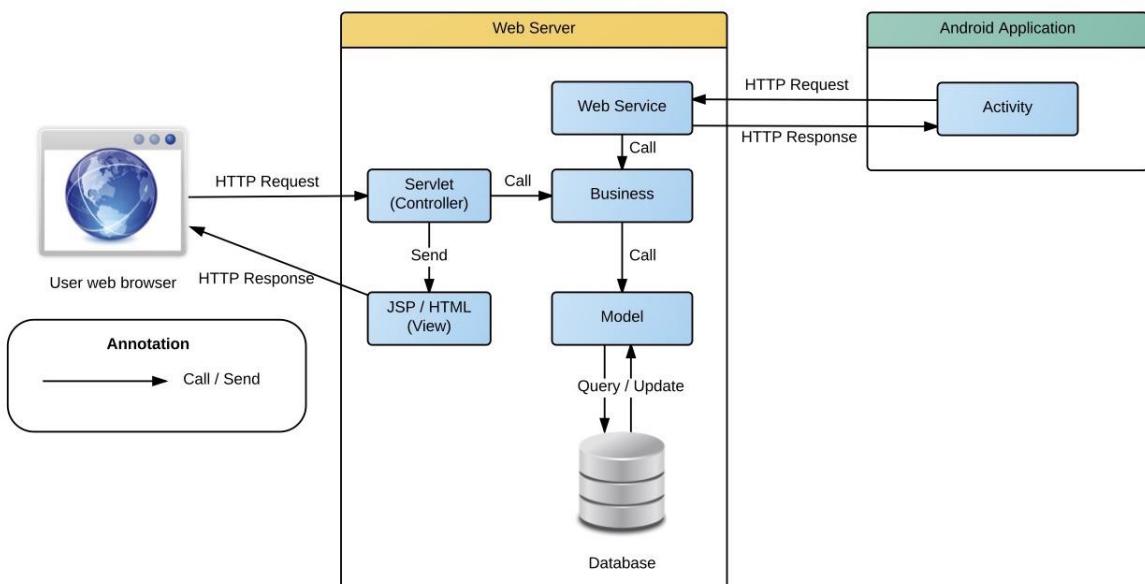


Figure 52 System architecture design

This diagram is referenced and modified from an original concept from: Chapter 6 Architecture Design, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

2.1. Web Application architecture description

In Web Application, the system is developed under J2EE MVC architecture style. We choose this architecture for Web application because of following advantages:

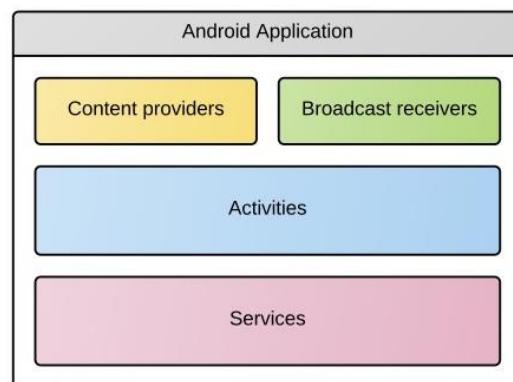
- Web app contains a Web service (public API for mobile app), with MVC architecture, we can separate business code with Controller and View, so we can use the business code in web service without repeat the code.
- Current system only supports motor insurance card, with MVC architecture, we can organize the code better for maintainability, extensibility, reusability so we can expand the scope to other kind of insurance services such as Health insurance or Asset insurance...
- In scope of 4-member team, MVC architecture make it easier to split the big project into small modules and make it easier to assign each module for members in our team.

This project follows MVC architecture with following components:

- **Servlet (Controller)** is the parts of the application that acts like event handler to handles user interaction. Typically, controller read data from a request and calls appropriate Business's method then selects view to return to user.
- **JSP/HTML (View)** is the parts of the application that handles the display of the data. The selection of View is under control of Controller.
- **Business** is the parts of the application that do business processing to solve domain problems.
- **Model** is the parts of the application that acts like a data transfer object between the system and database.
- **Web Service** is the parts of the application that acts like event handler for web and mobile communication via REST method.

2.2. Mobile Application architecture description

The application is developed as an Android native application. In general, the application architecture conforms to Android architecture.



Reference: [Android Developer Guide - Application Fundamentals](#)

<http://developer.android.com/guide/components/fundamentals.html>

This project follows Android application architecture with following component:

- **Activity** is the basic core of an android application that handles user input, create thread to run asynchronous tasks, send request and receive data from server via web services ...

3. Component Diagram

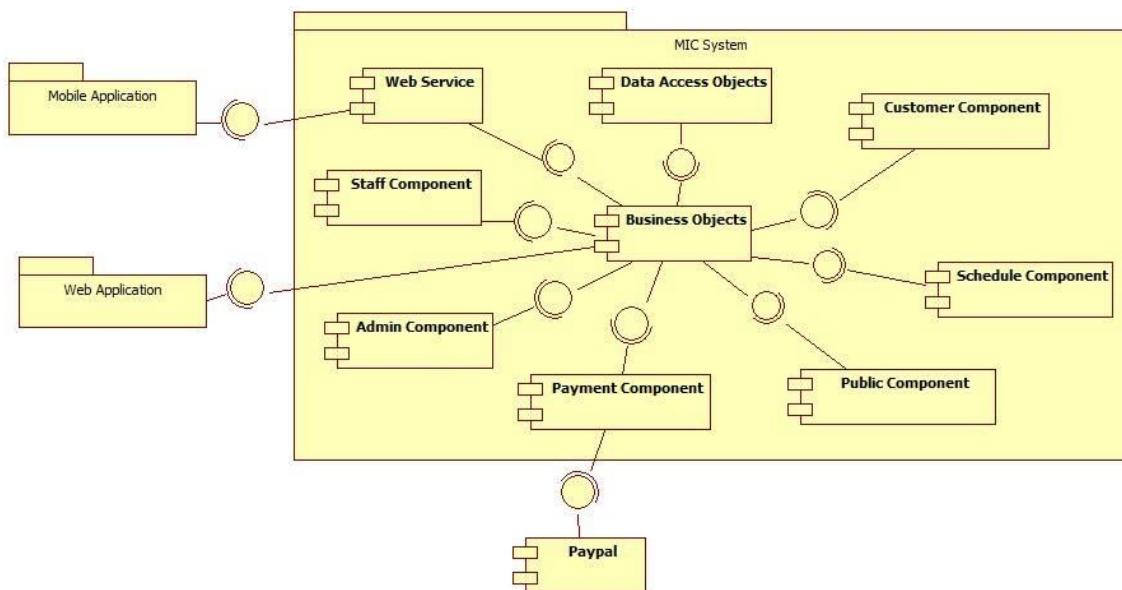


Figure 53 Component Diagram

Component Dictionary: Describes components	
Web Application	Web application package: View, Controller
Mobile Application	Mobile application package
PayPal	Handle payment process with PayPal API
Payment Component	Component to handle payment process
Web Service	Provide API for mobile applications to interact with the system.
Staff Component	Component to handle staff activities in the system
Customer Component	Component to handle customer activities in the system
Public Component	Component to handle guest activities in the system
Admin Component	Component to handle admin activities in the system
Schedule Component	Component to handle scheduler in the system
Business Objects	Common objects to handle domain business operations for each components
Data Access Objects	Component to handle interaction between the system and database

Table 50 Component Dictionary

4. Detailed Description

4.1. Class Diagram

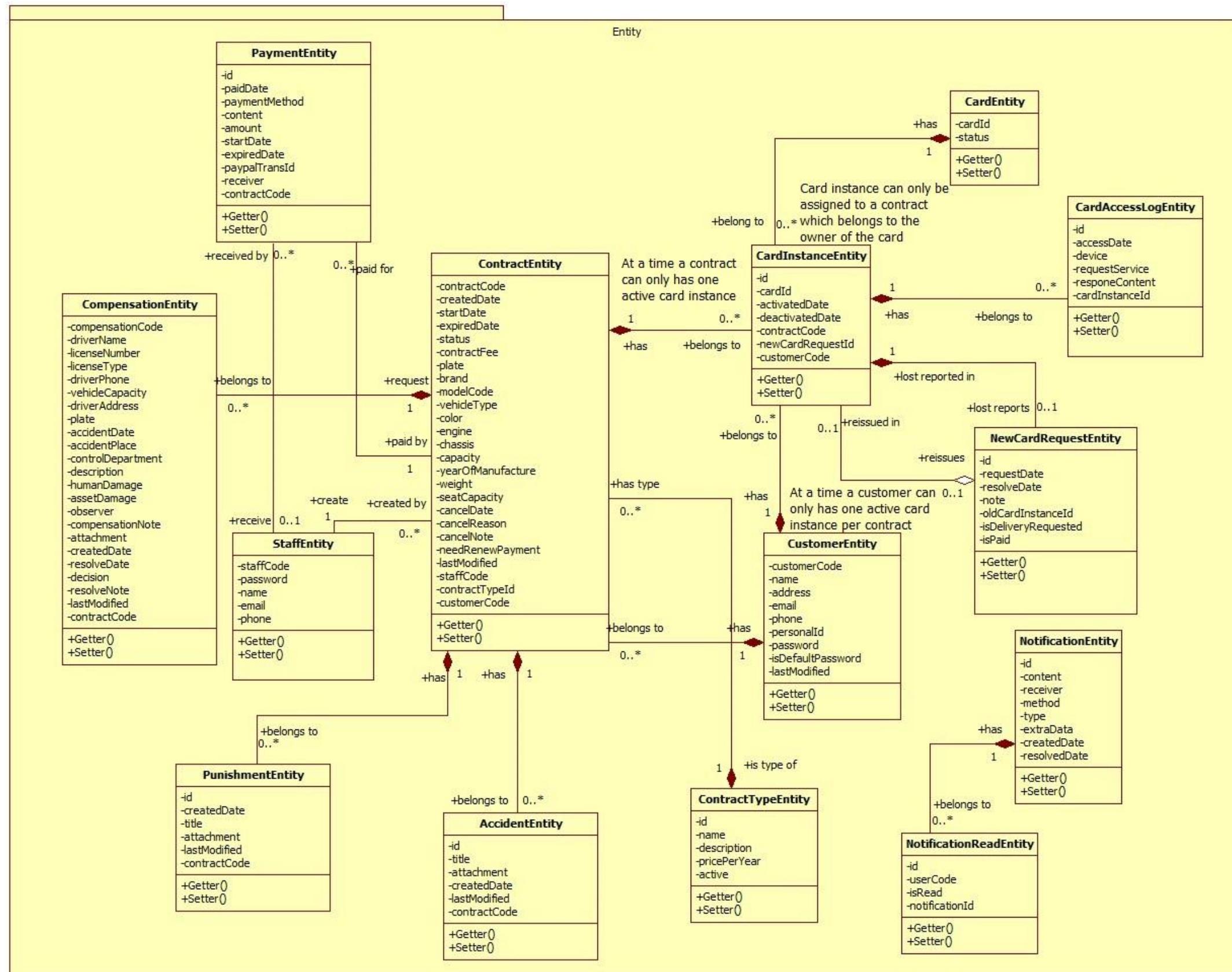


Figure 54 Class Diagram

Class dictionary: describe Class		
Class Name	Mapping column with Conceptual diagram	Description
PaymentEntity	Payment	Contain the payment information.
CardEntity	Card	Contain the card information.
CardInstanceEntity	CardInstance	Contain the card instance information
CustomerEntity	Customer	Contain the customer information.
ContractEntity	Contract	Contain the contract information.
StaffEntity	Staff	Contain the staff information.
CompensationEntity	Compensation	Contain the compensation information.
PunishmentEntity	Punishment	Contain the punishment information.
AccidentEntity	Accident	Contain the accident information.
ContractTypeEntity	ContractType	Contain the contract type information.
NewCardRequestEntity	NewCardRequest	Contain the new card request information.
CardAccessLogEntity	N/A	Not exist in conceptual diagram. But needed in class diagram to contain the card access log information.
NotificationEntity	N/A	Not exist in conceptual diagram. But needed in class diagram to contain the notification information.
NotificationReadEntity	N/A	Not exist in conceptual diagram. But needed in class diagram to know what notifications is read.

Table 51 Class dictionary

4.2. Class Diagram Explanation

4.2.1. PaymentEntity

Attribute	Type	Visibility	Description
Id	int	Private	Unique identifier of an payment
paidDate	Timestamp	Private	The day process the payment
paymentMethod	String	Private	The payment method
content	String	Private	The content of the payment
amount	Double	Private	Total amount of the payment
receiver	String	Private	The payment receiver
paypalTransId	String	Private	The PayPal transaction ID
contractCode	String	Private	The contract code.

Table 52 PaymentEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 53 PaymentEntity Method

4.2.2. CardEntity

Attribute	Type	Visibility	Description
CardId	String	Private	Unique identifier of a card
Status	Integer	Private	Status of the card

Table 54 CardEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 55 CardEntity Method

4.2.3. CardInstanceEntity

Attribute	Type	Visibility	Description
Id	Integer	Private	Unique identifier of a card instance
cardId	String	Private	Card id of the instance
activatedDate	Timestamp	Private	The date the card was activated.
deactivatedDate	Timestamp	Private	The date the card was deactivated.
contractCode	String	Private	The contract code
newCardRequestId	String	Private	The new card request ID.

Table 56 CardInstanceEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 57 CardInstanceEntity Method

4.2.4. CustomerEntity

Attribute	Type	Visibility	Description
customerCode	String	Private	Unique identifier of an customer
name	String	Private	Customer's name.
address	String	Private	Customer's address.
email	String	Private	Customer's email.
phone	String	Private	Customer's phone number.
personalId	String	Private	The customer's personal ID.
password	String	Private	Customer's password.
customerCode	String	Private	The customer's code.

Table 58 CustomerEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 59 CustomerEntity Method

4.2.5. ContractEntity

Attribute	Type	Visibility	Description
contractCode	String	Private	Unique identifier of a contract
startDate	Timestamp	Private	The start date of the contract.
expiredDate	Timestamp	Private	The expired date of the contract.
status	String	Private	The status of the contract.
contactFee	Double	Private	The contract fee.
plate	String	Private	The plate number of the motor.
brand	String	Private	The brand of the motor.
modelCode	String	Private	The model code of the motor.
vehicleType	String	Private	The vehicle type.
color	String	Private	The motor's color.
engine	String	Private	The motor's engine.
chassis	String	Private	The motor's chassis.
capacity	String	Private	The motor's capacity.
yearOfManufacture	int	Private	The year of manufacture.

weight	int	Private	The motor's weight.
seatCapacity	int	Private	The seat capacity.
cancelDate	Timestamp	Private	The cancel date of the contract.
cancelReason	String	Private	The cancel reason.
cancelNote	String	Private	The cancel note.
contractTypeId	int	Private	The contract type ID.
customerCode	String	Private	The customer code.

Table 60 ContractEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 61 Contract Method

4.2.6. StaffEntity

Attribute	Type	Visibility	Description
staffCode	String	Private	Unique identifier of a staff
password	String	Private	The staff's password
name	String	Private	The staff's name.
email	String	Private	The staff's email.
phone	String	Private	The staff's phone number.

Table 62 StaffEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 63 StaffEntity Method

4.2.7. CompensationEntity

Attribute	Type	Visibility	Description
compensationCode	String	Private	Unique identifier of a compensation
driverName	String	Private	The driver name.
licenseNumber	String	Private	The license number of the motor.
licenseType	String	Private	The license type of the motor.

driverPhone	String	Private	The driver's phone number.
vehicleCapacity	String	Private	The vehicle capacity.
driverAddress	String	Private	The driver's address.
plate	String	Private	The plate number.
accidentDate	Timestamp	Private	The accident date.
accidentPlace	String	Private	The accident place.
controlDepartment	String	Private	The police control department.
description	String	Private	The compensation's description.
humanDamage	String	Private	The human damages.
assetDamage	String	Private	The asset damages.
observer	String	Private	The observer.
compensationNote	String	Private	The compensation note.
attachment	String	Private	The attachment.
createdDate	Timestamp	Private	The created date of the compensation.
resolveDate	Timestamp	Private	The resolve date.
decision	String	Private	The decision.
resolveNote	String	Private	The resolve note.
contractCode	String	private	The contract code.

Table 64 CompensationEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 65 CompensationEntity Method

4.2.8. PunishmentEntity

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of a punishment
createdDate	Timestamp	Private	The created date.
title	String	Private	The title of the punishment.
attachment	String	Private	The attachment.
contractCode	String	Private	The contract code.

Table 66 PunishmentEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 67 PunishmentEntity Method

4.2.9. AccidentEntity

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of an accident
title	String	Private	The title of the accident.
attachment	String	Private	The attachment.
createdDate	Timestamp	Private	The created date.
contractCode	String	Private	The contract code.

Table 68 AccidentEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 69 AccidentEntity Method

4.2.10. ContractTypeEntity

Attribute	Type	Visibility	Description
id	int	Private	unique identifier of a contract type
name	String	Private	The contract's name.
description	String	Private	The description.
pricePerYear	Double	Private	The price per year of the contract.

Table 70 ContractTypeEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 71 ContractTypeEntity Method

4.2.11. NewCardRequestEntity

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of a new card request
requestDate	Timestamp	Private	The request date.

resolveDate	Timestamp	Private	The resolve date.
note	String	Private	The new card request note.
oldCardId	String	Private	The ID of the old card.
customerCode	String	Private	The customer code.

Table 72 NewCardRequestEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 73 NewCardRequestEntity Method

4.2.12. CardAccessLogEntity

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of an card access log
accessDate	Timestamp	Private	The access date.
device	String	Private	The device name.
requestService	String	Private	The request service.
responseContent	String	Private	The response content.

Table 74 CardAccessLogEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 75 CardAccessLogEntity Method

4.2.13. NotificationEntity

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of the notification.
content	String	Private	Content of the notification.
receiver	String	Private	A regex pattern that match with staffCode or customerCode.

method	int	Private	Notification method such as email or web notification.
type	int	Private	Type of the action were notified.
extraData	String	Private	The extra data such as contractCode, compensationCode,etc.
createdDate	Timestamp	Private	The created date.
resolvedDate	Timestamp	Private	The resolved date.

Table 76 NotificationEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 77 NotificationEntity Method

4.2.14. NotificationReadEntity

Attribute	Type	Visibility	Description
id	int	Private	Unique identifier of the notification read.
userCode	String	Private	Customer code or staff code.
isRead	int	Private	Check if the notification is read.
notificationId	int	Private	Notification's Id.

Table 78 NotificationReadEntity Attribute

Method	Return Type	Visibility	Description
Getter	Attribute type	Public	Get attribute value.
Setter	Void	Public	Set value of attribute.

Table 79 NotificationReadEntity Method

4.3. Interactive Diagram

4.3.1. Web Application

4.3.1.1. Staff

4.3.1.1. Create new contract

Summary: this diagram show process of staff creates new contract

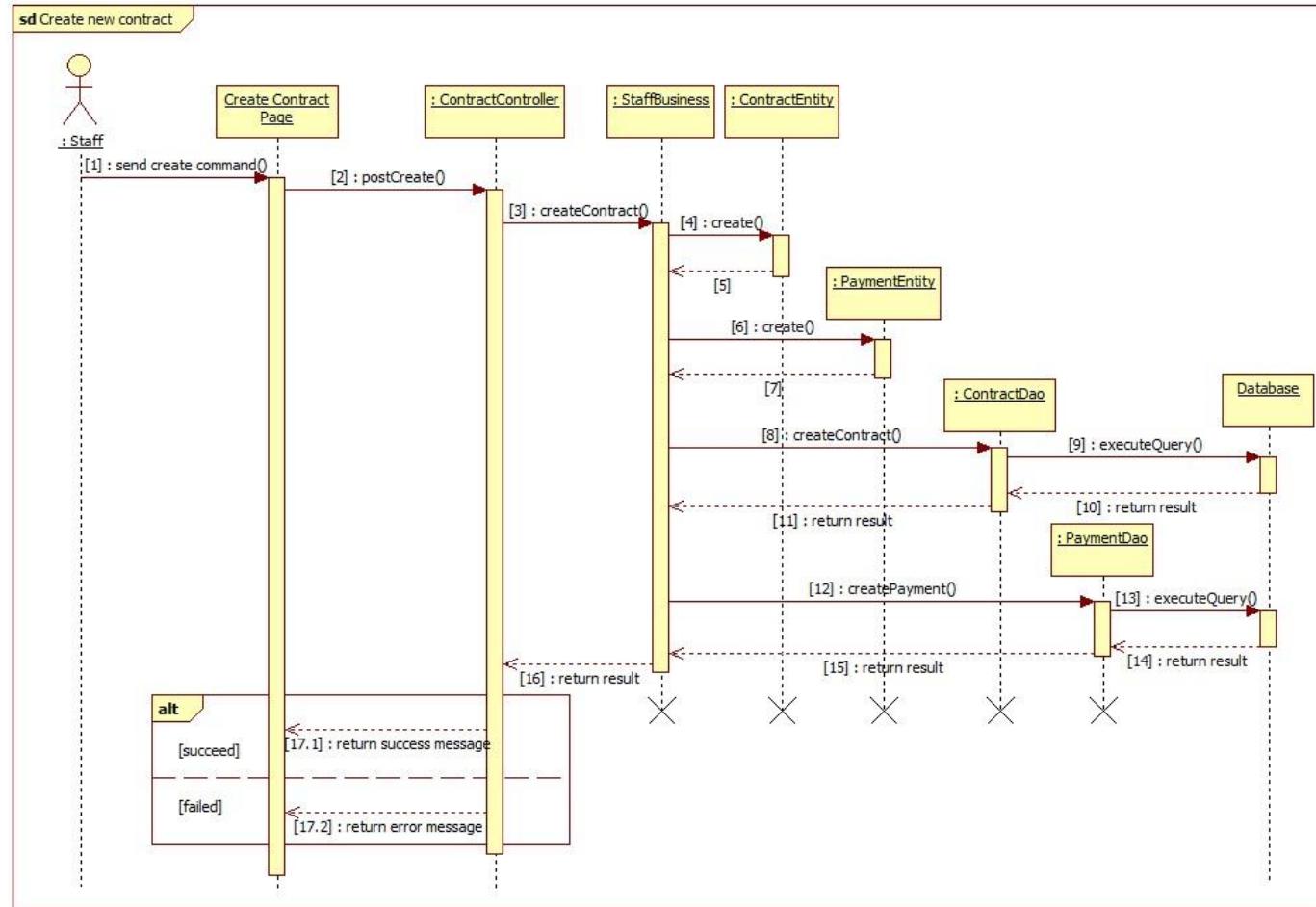


Figure 55 Sequence diagram - <Staff> Create new contract

4.3.1.1.2. Renew contract

Summary: this diagram show process of staff renews contract

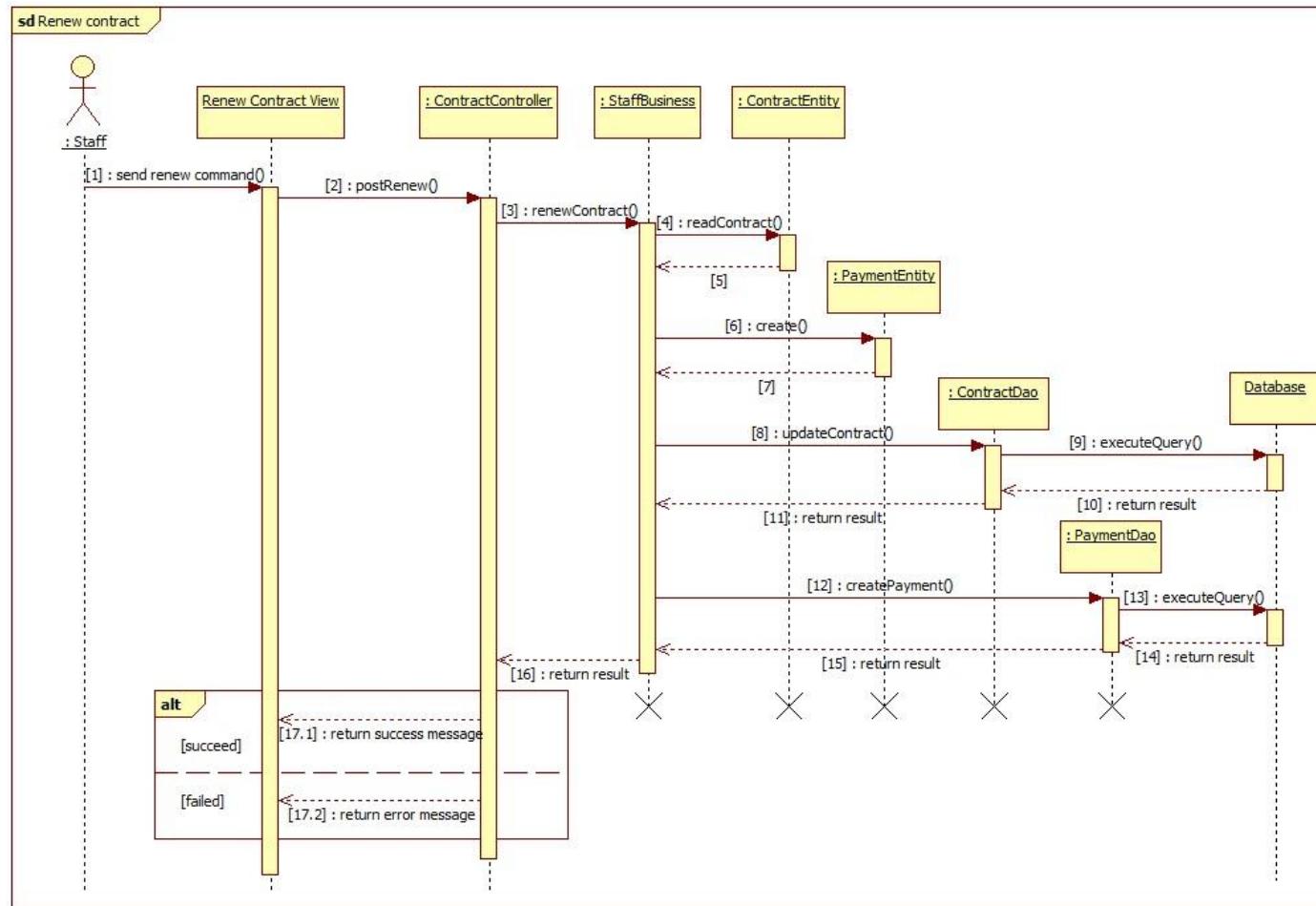


Figure 56 Sequence diagram - <Staff> Renew contract

4.3.1.1.3. Cancel contract

Summary: this diagram show process of staff cancels contract

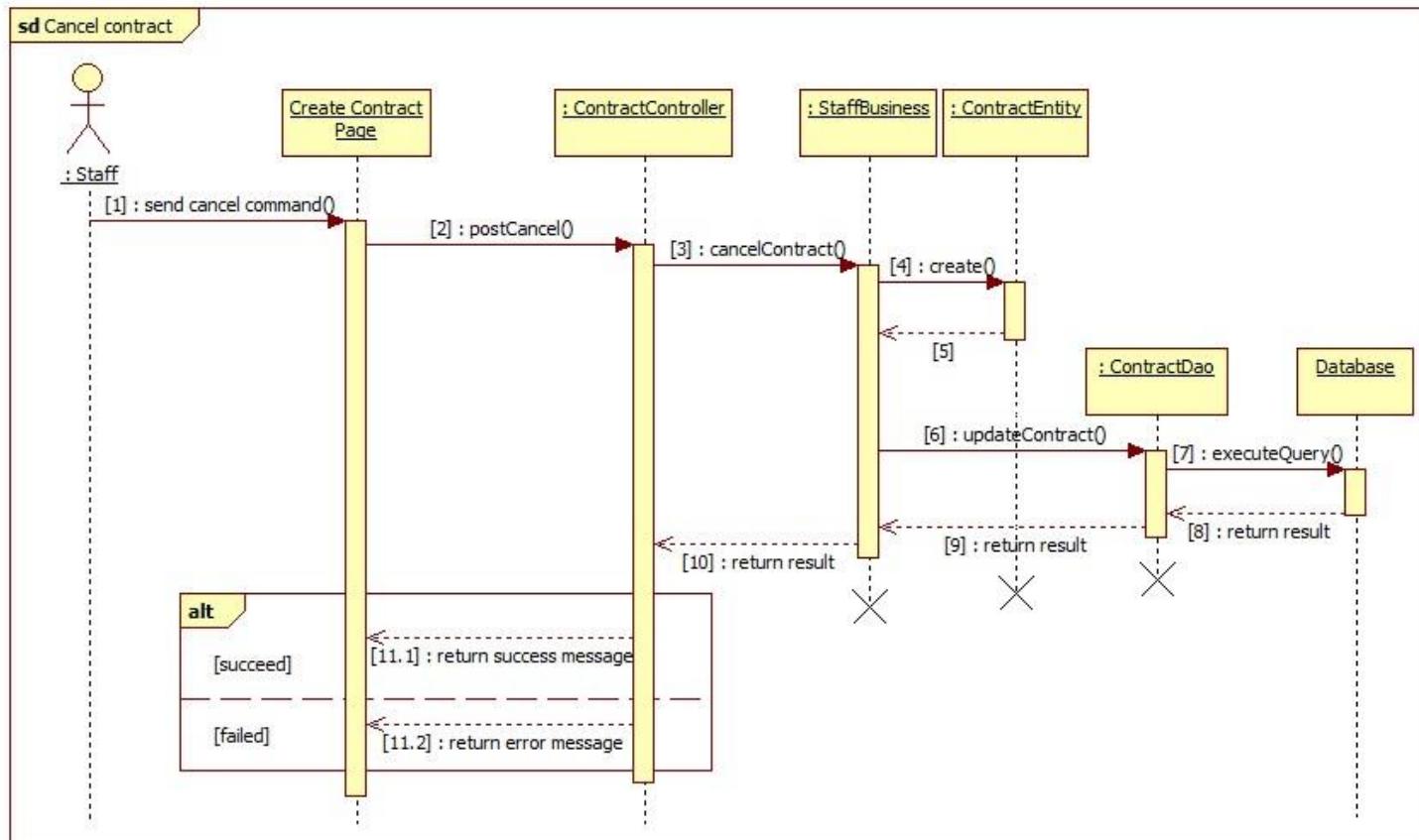


Figure 57 Sequence diagram - <Staff> Cancel contract

4.3.1.2.Customer

4.3.1.2.1. Cancel contract

Summary: this diagram show process of customer cancels contract

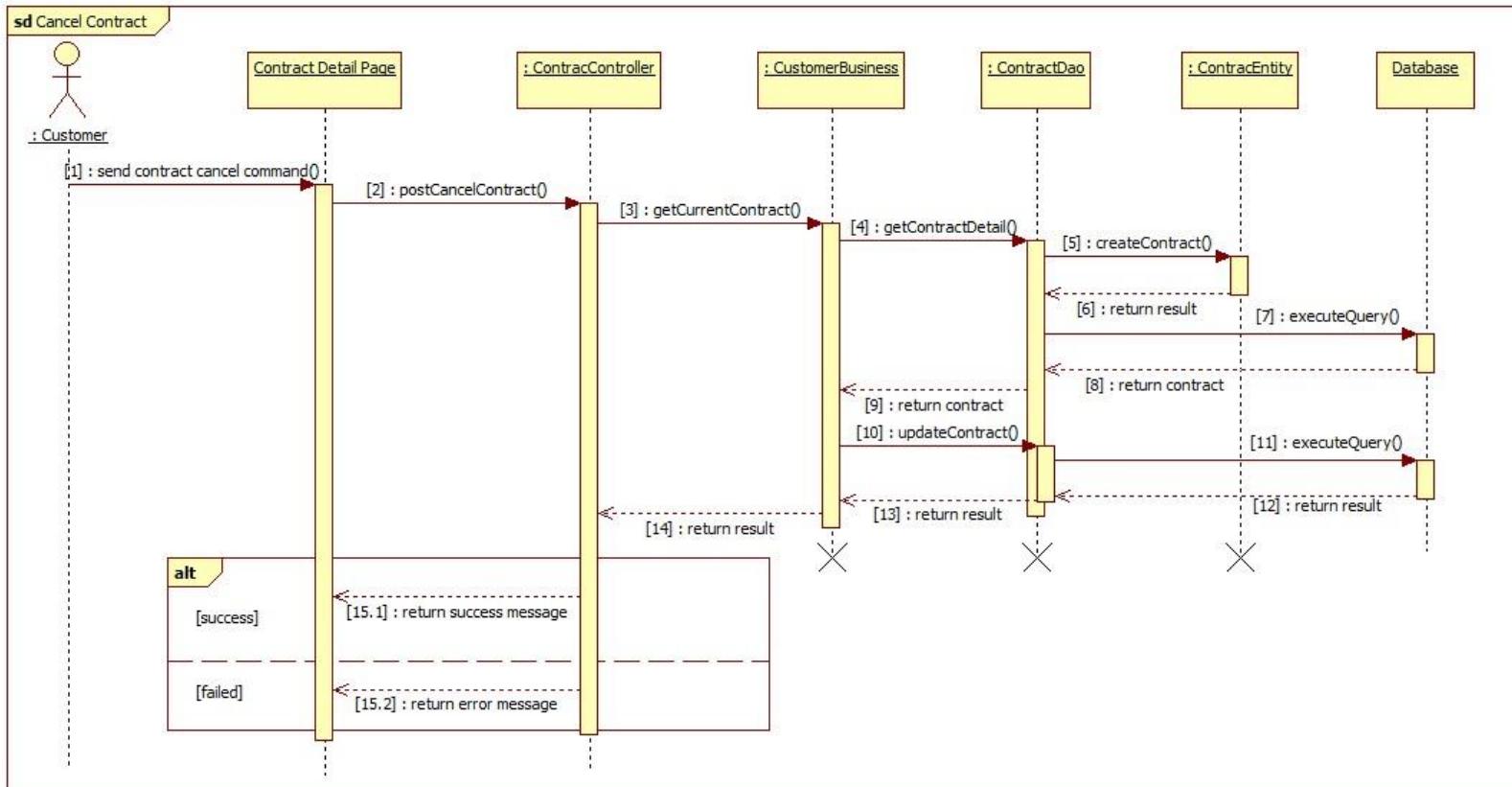


Figure 58 Sequence diagram - <Customer> Cancel contract

4.3.1.2.2. Renew contract

Summary: this diagram show process of customer renews contract

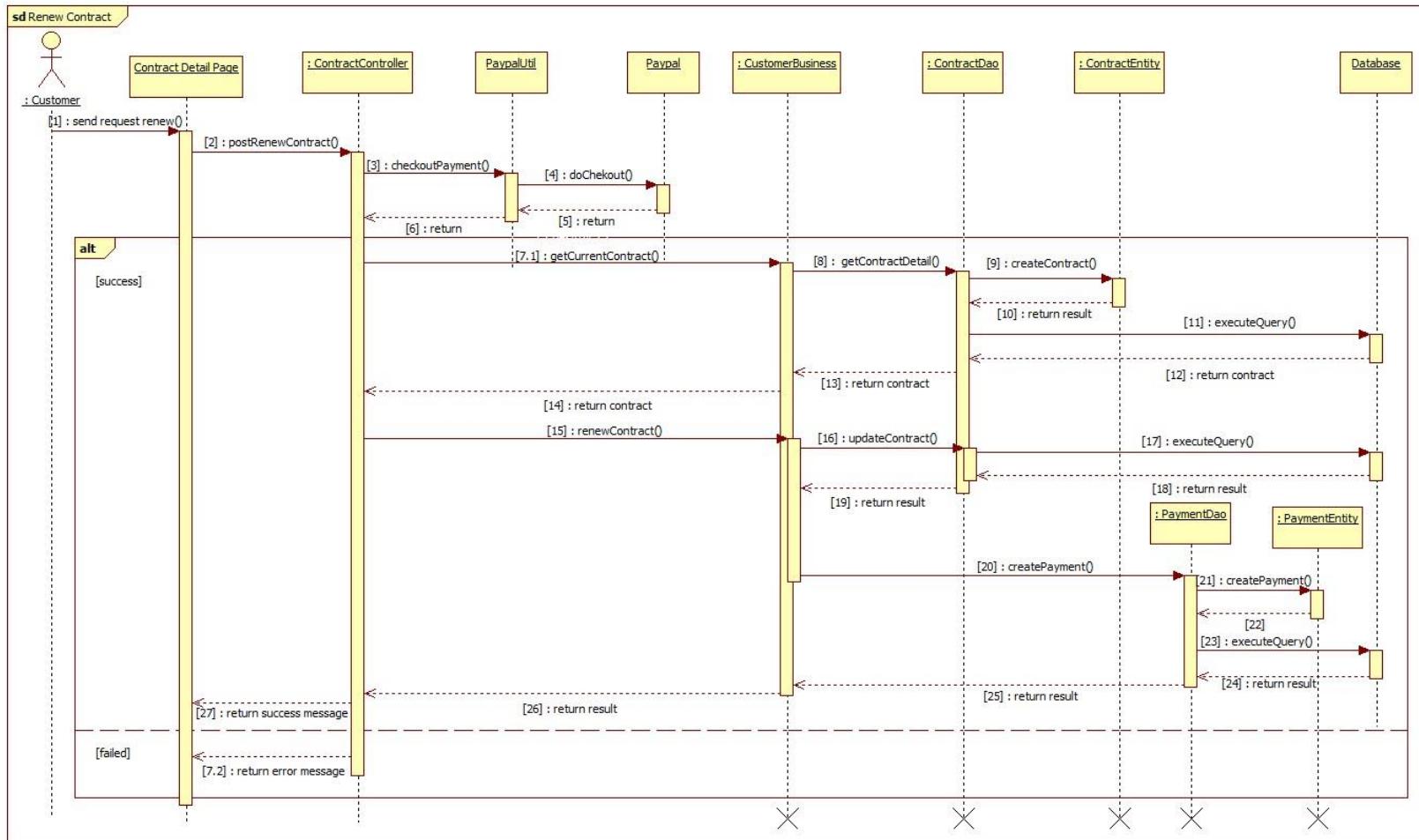


Figure 59 Sequence diagram - <Customer> Renew contract

4.3.1.3.Guest

4.3.1.3.1. Register new contract

Summary: this diagram show process of guest registers new contract

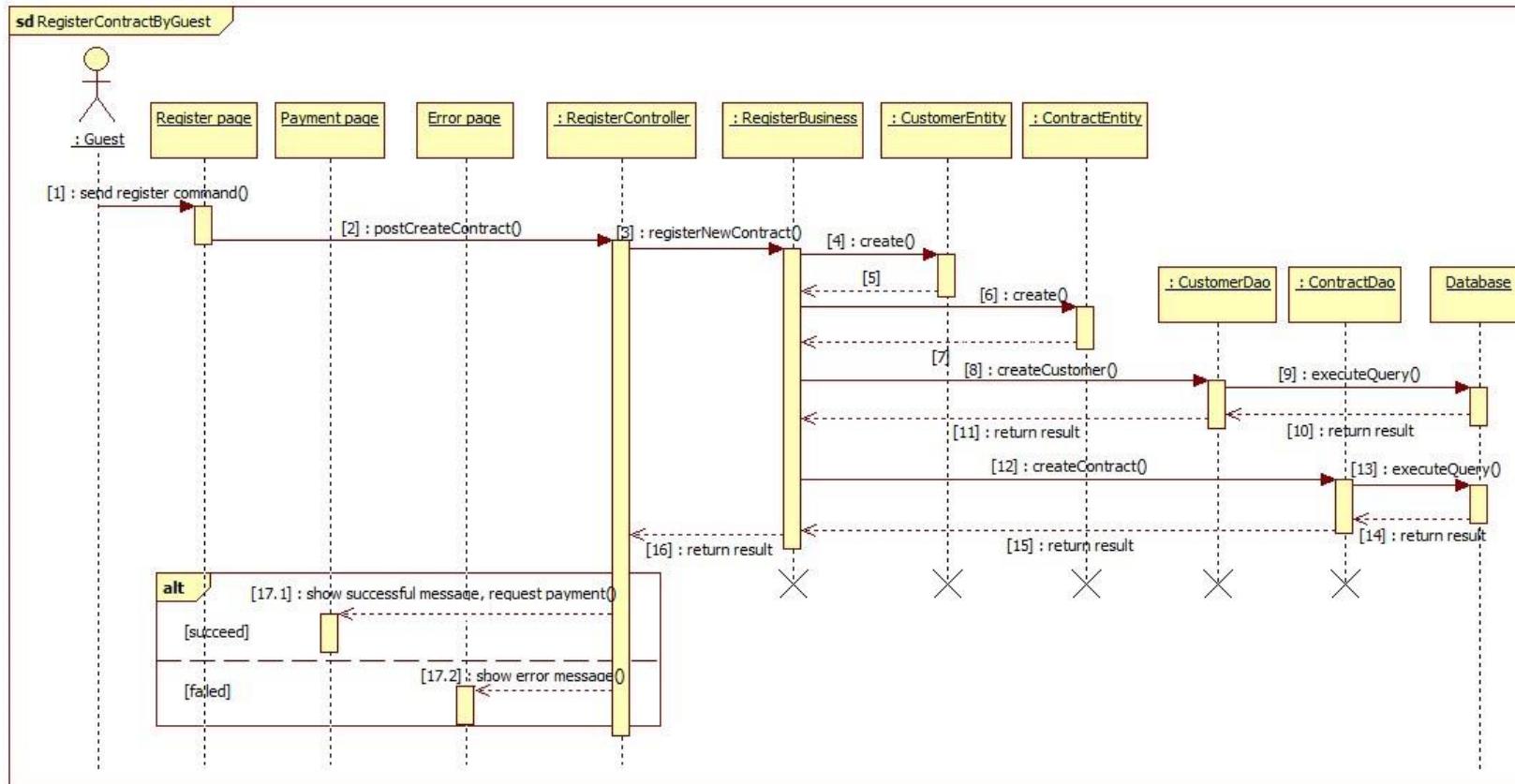


Figure 60 Sequence diagram - <Guest> Register new contract

4.3.1.3.2. PayPal payment

Summary: this diagram show process of guest does payment

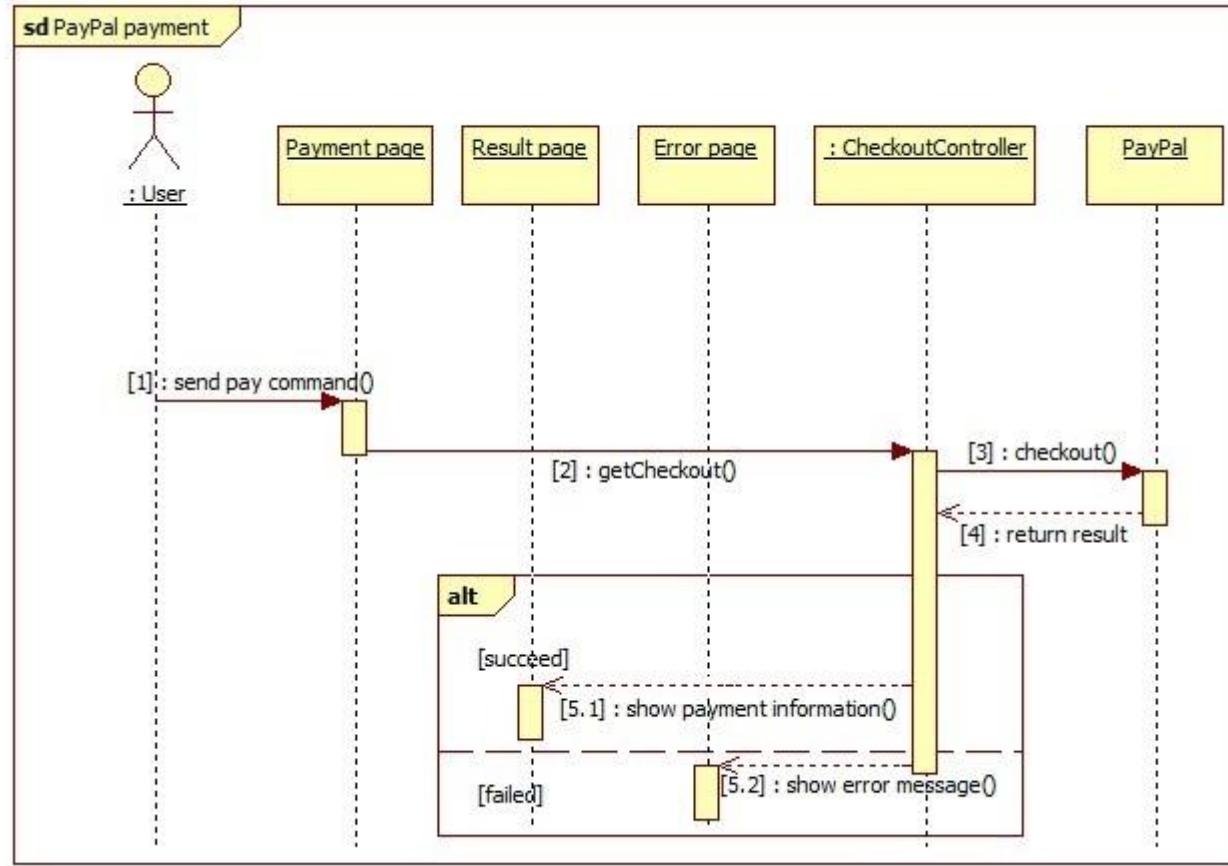


Figure 61 Sequence diagram - <Guest> PayPal payment

4.3.2. Mobile Application

4.3.2.1. Checker Mobile Application

4.3.2.1.1. <Police> Verify card validation

Summary: this diagram show process of police checks card validation.

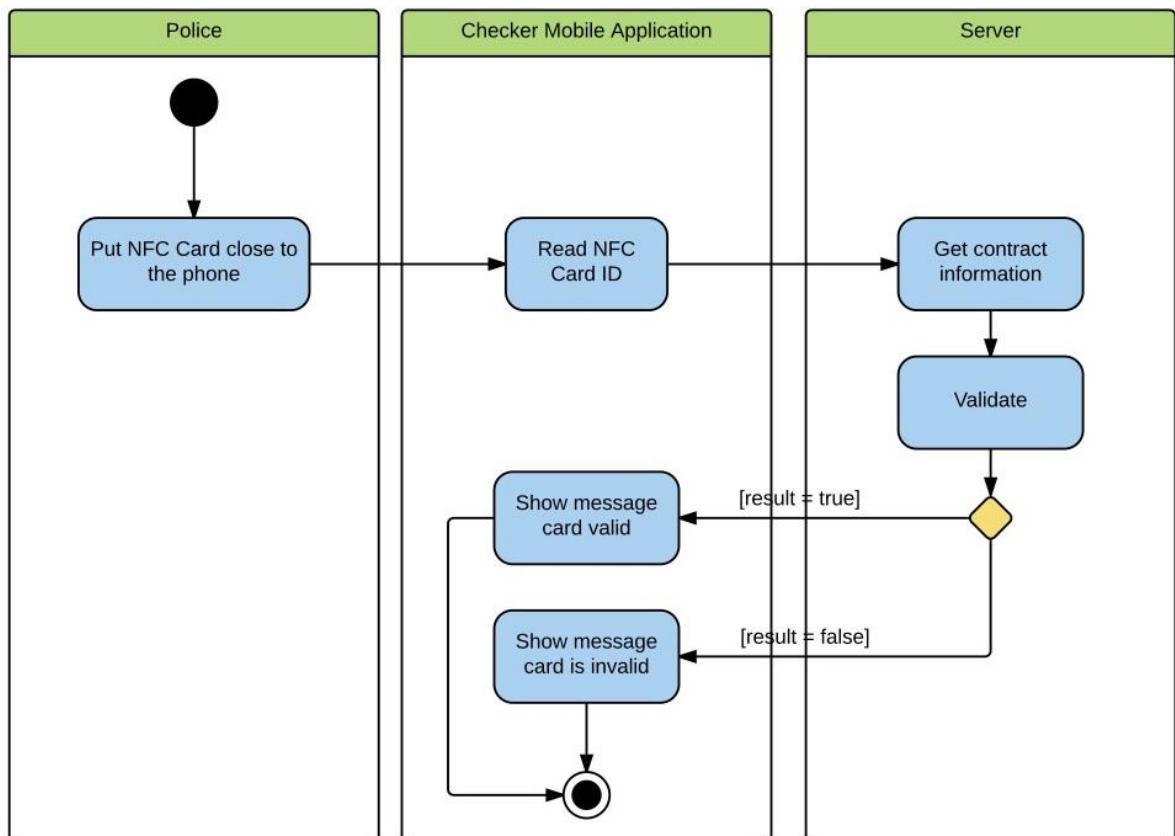


Figure 62 <Police> Verify card validation

4.3.2.1.2. <Police> Add punishment information

Summary: this diagram show process of police adds punishment information.

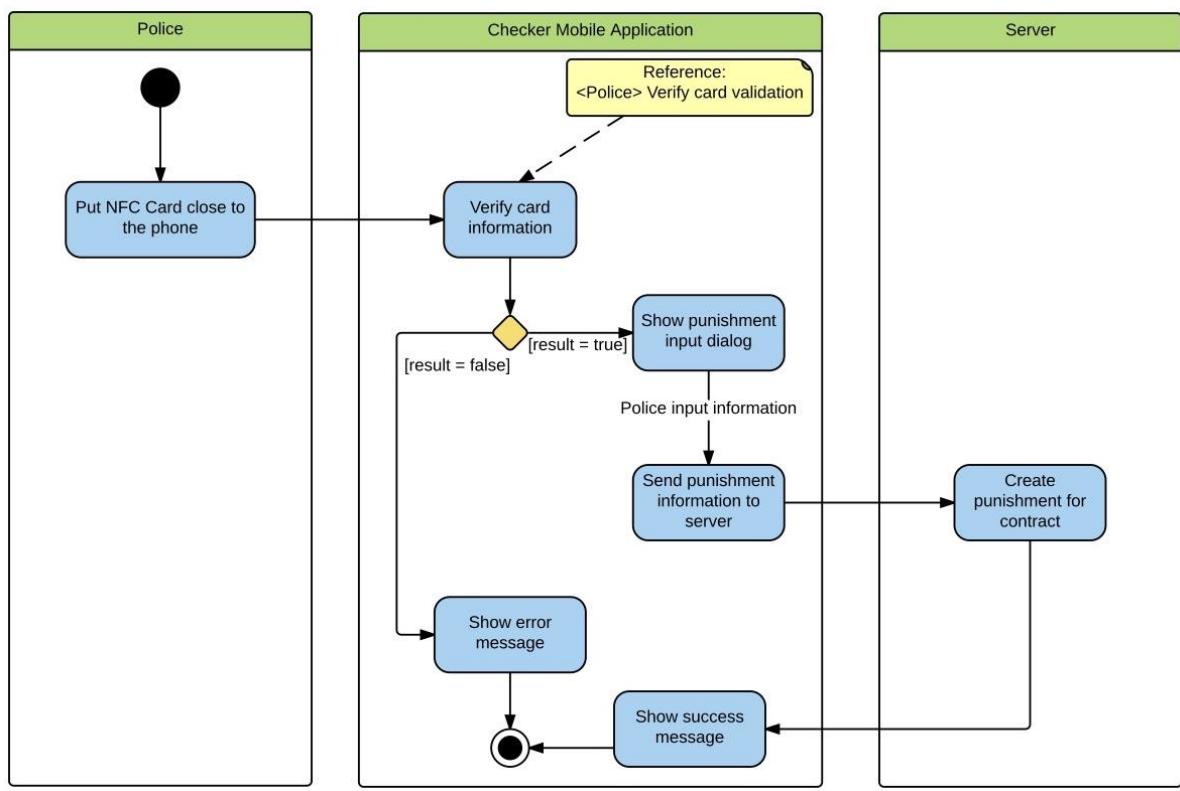


Figure 63 <Police> Add punishment information

4.3.2.2. Printer Mobile Application

4.3.2.2.1. <Staff> Search contract

Summary: this diagram shows process of staff searchs for contracts.

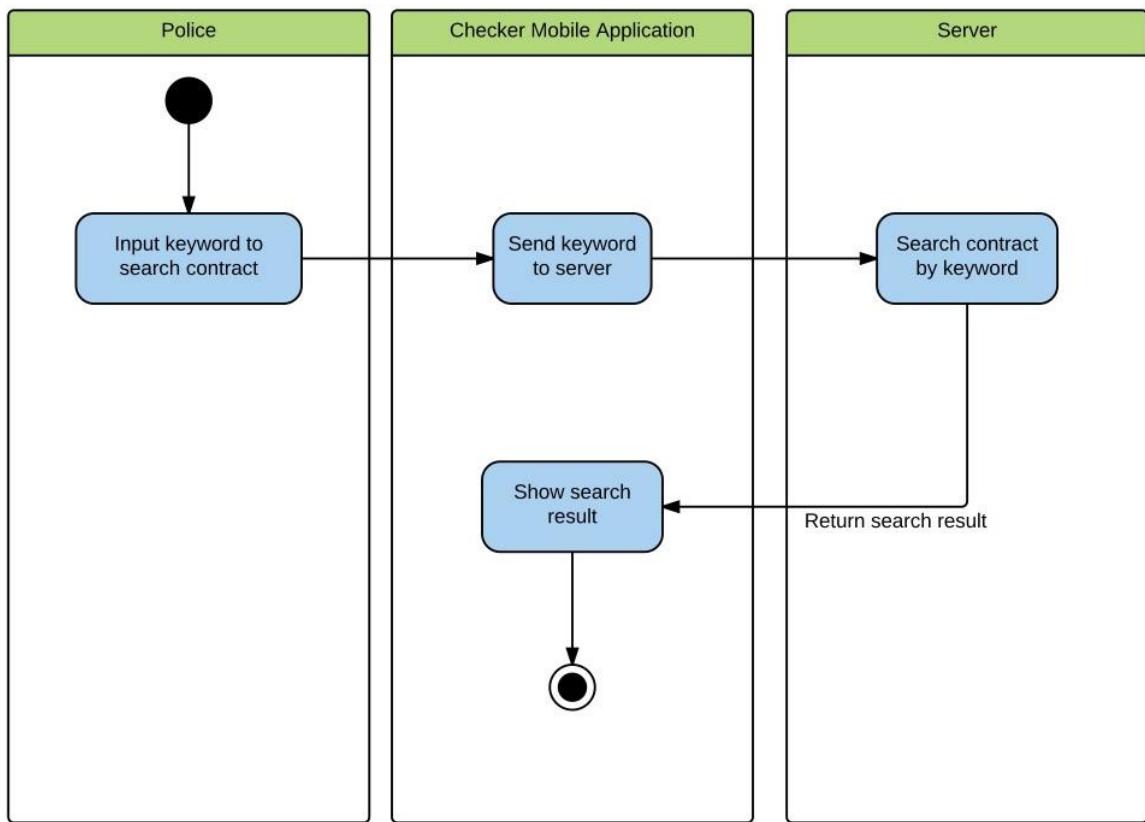


Figure 64 <Staff> Search contract

4.3.2.2.2. <Staff> View contract information

Summary: this diagram shows process of staff views contract information.

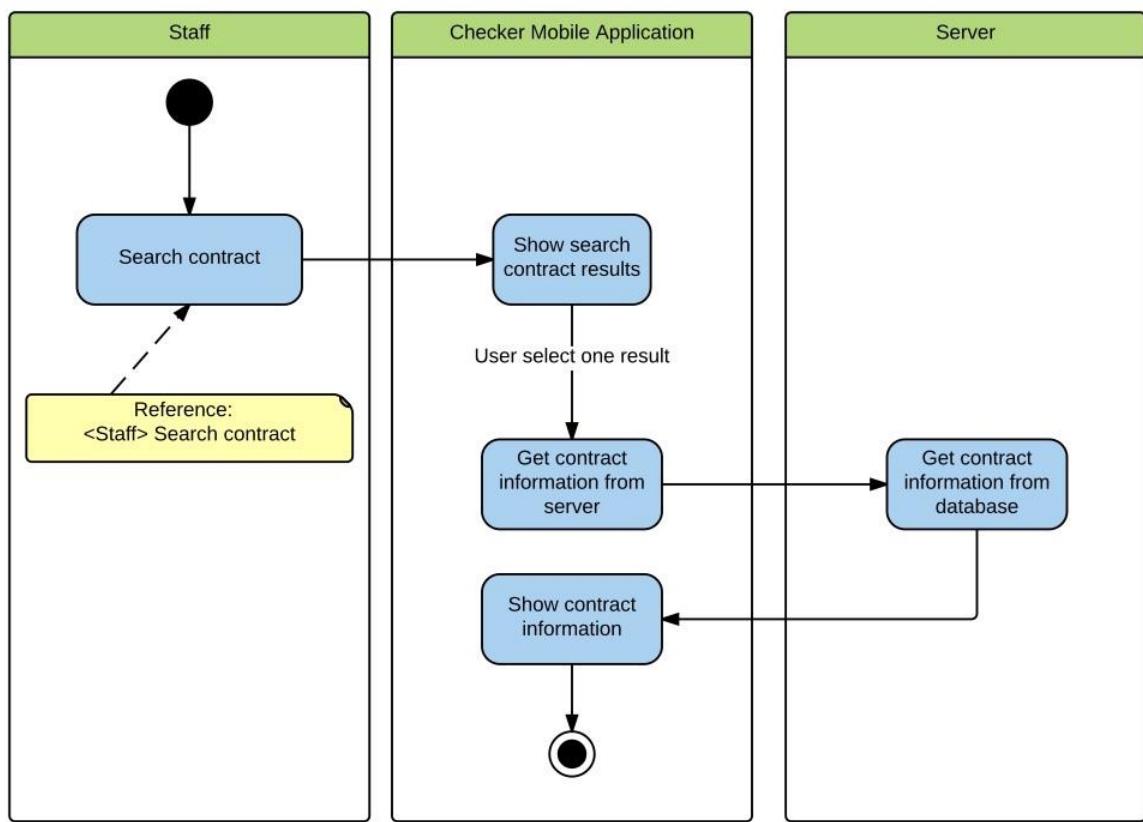


Figure 65 <Staff> View contract information

4.3.2.2.3. <Staff> Print information to NFC Card

Summary: this diagram shows the process of staff printing information to an NFC card.

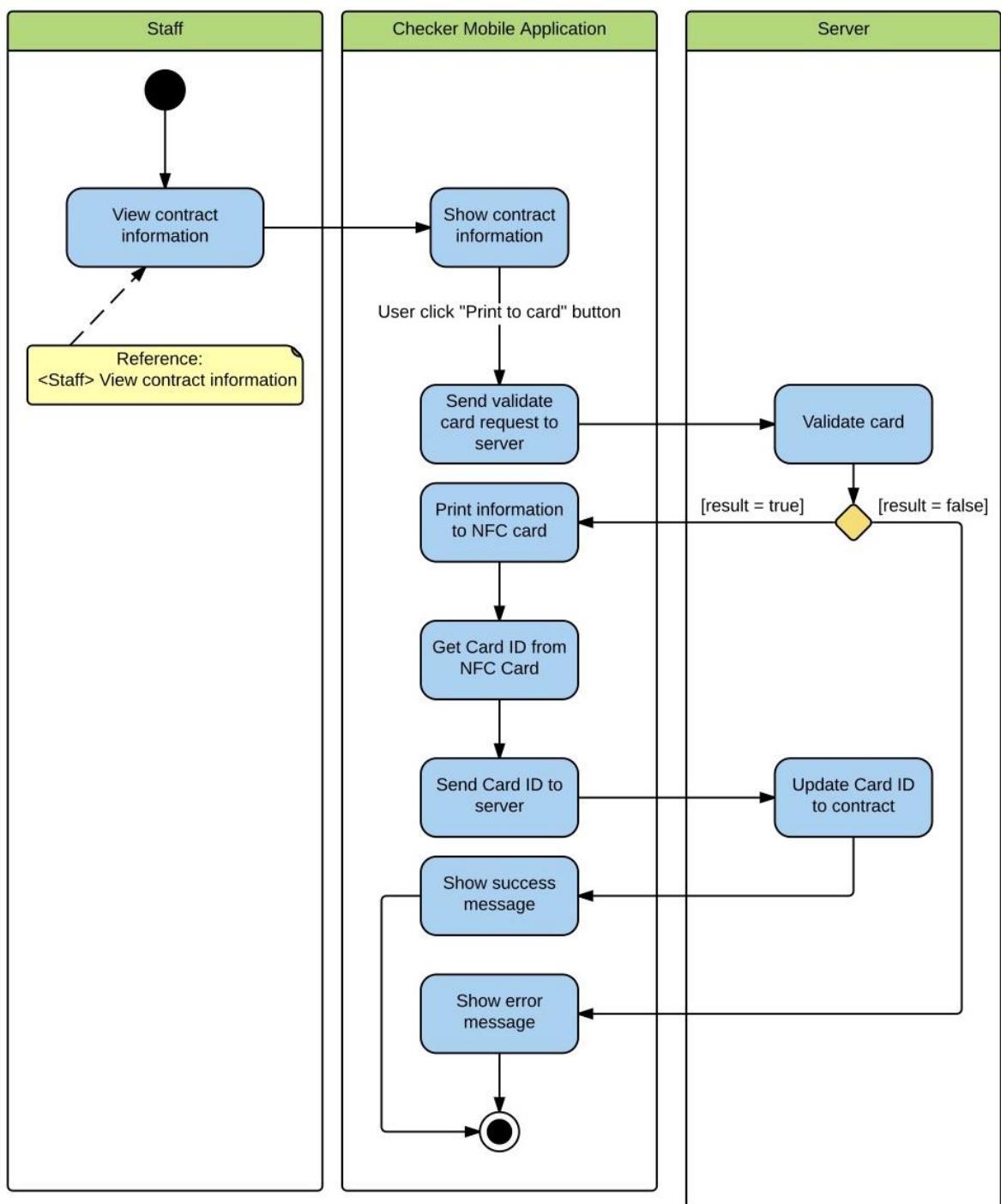


Figure 66 <Staff> Print information to NFC card

5. Interface

5.1. Component Interface

5.1.1. Web Service Interface

Signature	Description	Input	Output	Output Format	Exception
public ResponseObject getCheckConnection(R r)	Check server status	Request object r	Json Boolean the status of server	Boolean	JsonProcessingException
public ResponseObject getContracts(R r)	Search contracts by code or customer name	Request object r contains: Keyword: String	Json string List of ContractSearchResultDto	ContractEntity CustomerEntity	JsonProcessingException
public ResponseObject getUpdateCardID(R r)	Update card ID to a contract	Request object r contains: cardID: String contractCode: String	Json CardEntity object	CardEntity	JsonProcessingException NoResultException
public ResponseObject getCheckCard(R r)	Check the validation of the card	Request object r contains: cardID: String	Json CardEntity object	CardEntity	JsonProcessingException NoResultException
public ResponseObject getUpdatePunishment(R r)	Update punishment information to a contract	Request object r contains: contractCode: String punishmentInfo: PunishmentInfoObject	Json Boolean the update result	Boolean	JsonProcessingException

Table 80 Web Services Interfaces

Entity	Description
ContractEntity	contractCode startDate expiredDate status contractFee plate brand modelCode vehicleType color engine chassis capacity yearOfManufacture weight seatCapacity cancelDate cancelReason cancelNote staffCode contractTypeId customerCode needRenewPayment lastModified createdDate
CustomerEntity	customerCode name address email phone personalId password isDefaultPassword lastModified
CardEntity	cardId activatedDate deactivatedDate contractCode newCardRequestId

Table 81 Output format description

Exception	Description
JsonProcessingException	encountered when processing (parsing, generating) JSON content that are not pure I/O problems
NoResultException	Thrown by the persistence provider when getSingleResult() is executed on a query and there is no result to return

Table 82 Exception description

5.2. Web application Design

5.2.1. Staff Interface Design

5.2.1.1. Staff home page

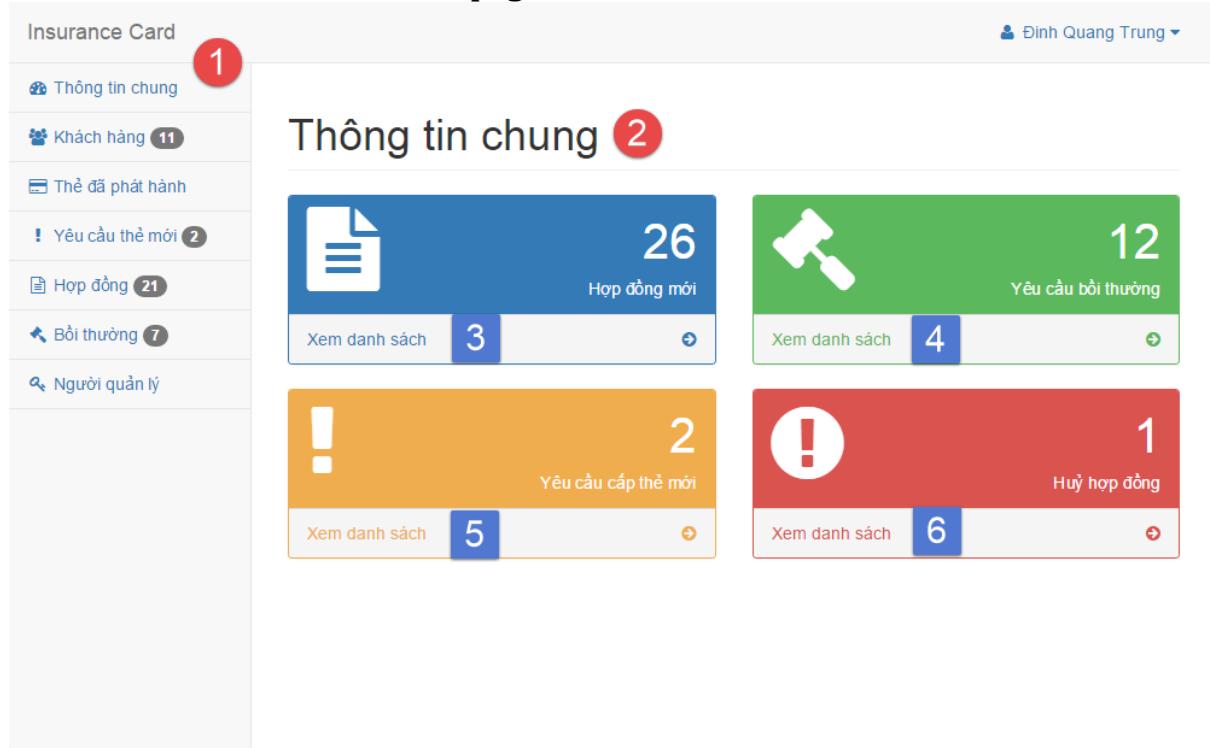


Figure 67 Interface - <Staff> Home page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Menu	Navigation bar	Yes	Yes	Menu bar	N/A	N/A
2	Title	Title of the page	Yes	Yes	Label	N/A	N/A

Table 83 Staff home page fields

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	viewContract	List all contract	N/A	Transfer to list contract page
4	viewCompensation	List all compensation	N/A	Transfer to list compensation page
5	viewNewCardReq	List all new card request	N/A	Transfer to list new card request page

6	viewCancelContract	List all cancel contract request	N/A	Transfer to list contract page
----------	------------------------------------	----------------------------------	-----	--------------------------------

Table 84 Staff home page buttons/hyperlinks

5.2.1.2. Manage customer

#	Mã KH	Tên	Số điện thoại	Mã hợp đồng	Mã thẻ
1	KH000B	Trần Vương	0992837423	Add later	Add later
2	KH000A	Trương Văn Khải	0984283338	Add later	Add later
3	KH0009	Trần Văn Linh	0929384445	Add later	Add later
4	KH0008	Nguyễn Văn Trung	0909231848	Add later	Add later
5	KH0007	Mai Trần Hậu	0982384556	Add later	Add later
6	KH0006	Lý Phương	0928374185	Add later	Add later
7	KH0005	Nguyễn Hữu Cảnh	0988263748	Add later	Add later
8	KH0004	Cao Văn Trường	0935823754	Add later	Add later
9	KH0003	Nguyễn Thị Trang	0987302918	Add later	Add later
10	KH0002	Trần Quang Đại	0903012842	Add later	Add later

Figure 68 Interface - <Staff> Manage customer

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Menu	Navigation bar	Yes	Yes	Menu bar	N/A	N/A
2	Title	Title of the page	Yes	Yes	Label	N/A	N/A
3	Number	Number of available customer	Yes	No	Label	N/A	N/A
4	txtSearchValue	Fill search keyword	No	Yes	Textbox	String	N/A
5	Table	Table of available customer	Yes	Yes	Table	N/A	N/A

Table 85 <Staff> Manage customer fields

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	btnCreate	Create new customer	N/A	Transfer to create new customer page
7	btnSearch	Search by input keyword	N/A	Transfer to search result page
8	customerDetail	View customer detail	N/A	Transfer to customer detail page

Table 86 <Staff> Manage customer buttons/hyperlinks

5.2.1.3.Create customer

The screenshot shows the 'Thêm khách hàng mới' (Add new customer) interface. On the left is a sidebar with various links: 'Thông tin chung' (1), 'Khách hàng' (11), 'Thẻ đã phát hành', 'Yêu cầu thẻ mới' (2), 'Hợp đồng' (21), 'Bồi thường' (7), and 'Người quản lý'. The main content area has a title 'Thêm khách hàng mới' (2). Below it, there are several input fields with asterisks indicating they are required: 'Tên khách hàng' (3), 'Địa chỉ' (4), 'Email' (5), 'Số điện thoại' (6), and 'Số CMND / Hộ chiếu' (7). At the bottom right is a green button labeled '→ Thêm khách hàng' (8).

Figure 69 Interface - <Staff> Create customer

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Leng th
1	Menu	Navigation bar	Yes	Yes	Menu bar	N/A	N/A

2	Title	Title of the page	Yes	Yes	Label	N/A	N/A
3	txtName	Fill customer name	No	Yes	Textbox	String	3 – 80
4	txtAddress	Fill address	No	Yes	Textbox	String	3 – 250
5	txtEmail	Fill email	No	Yes	Textbox	String	3 – 250
6	txtPhone	Fill phone	No	Yes	Textbox	Integer	8 – 15
7	txtPersonal ID	Fill personal ID	No	No	Textbox	Integer	8 – 15

Table 87 <Staff> Create customer fields

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
8	btnCreate	Create new customer	N/A	Transfer to create new customer successful page

Table 88 <Staff> Create customer buttons/hyperlinks

5.2.1.4. Manage contract

#	Mã hợp đồng	Tên khách hàng	Ngày bắt đầu	Ngày kết thúc	Trạng thái
1	HDUA79	Nguyễn Hữu Cảnh	08/05/2009	08/05/2015	Expired
2	HD76FF	Nguyễn Thị Trang	18/02/2012	18/02/2016	Ready
3	HD6679	Trương Văn Khải	18/02/2005	18/02/2015	Cancelled
4	HD453C	Lý Phương	25/04/2016	25/04/2019	Ready
5	HD33FD	Cao Văn Trường	09/04/2013	09/04/2016	Request cancel
6	HD0ES6	Trần Quang Đại	15/03/2014	15/03/2015	No Card
7	HD0A33	Phùng Tri	05/12/2014	05/12/2015	Pending
8	HD0031	Trương Văn Khải	13/06/2015	13/06/2016	No card
9	HD0030	Trương Văn Khải	13/06/2015	13/06/2016	No card
10	HD002Y	Trương Văn Khải	13/06/2015	13/06/2016	No card

Figure 70 Interface - <Staff> Manage contract

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Menu	Navigation bar	Yes	Yes	Menu bar	N/A	N/A
2	Title	Title of the page	Yes	Yes	Label	N/A	N/A
3	Number	Number of available contract	Yes	No	Label	N/A	N/A
4	txtSearchValue	Fill search keyword	No	Yes	Textbox	String	N/A
5	Table	Table of available customer	Yes	Yes	Table	N/A	N/A

Table 89 <Staff> Manage contracts fields

Button/Hyperlinks

No	Function	Description	Validation	Outcome
6	btnCreate	Create new contract	N/A	Transfer to create new contract page
7	btnSearch	Search by input keyword	N/A	Transfer to search result page
8	contractDetail	View contract detail	N/A	Transfer to contract detail page
9	customerDetail	View customer detail	N/A	Transfer to customer detail page

Table 90 <Staff> Manage contracts buttons/hyperlinks

5.2.1.5.Create contract

Insurance Card

Thông tin chung 1

Khách hàng 11

Thẻ đã phát hành

Yêu cầu thẻ mới 2

Hợp đồng 21

Bồi thường 7

Người quản lý

Đinh Quang Trung

Thêm hợp đồng mới 2

Các ô có dấu * là bắt buộc

Thông tin khách hàng

Khách hàng * 3

Chọn 20

Thông tin về dịch vụ bảo hiểm

Loại hình bảo hiểm * 4

Vui lòng chọn loại hợp đồng

Thời điểm có hiệu lực * 5

15/06/2015

Thời điểm hết hiệu lực * 6

15/06/2016

Phí bảo hiểm (VND) * 7

Thông tin về xe cơ giới

Biển số đăng ký * 8

Ví dụ: 78Y9-15383

Nhãn hiệu * 9

Ví dụ: Honda

Số máy * 10

Số khung * 11

Màu sơn 13

Ví dụ: Honda

Dung tích * 12

Loại xe 14

Số loại 15

Ví dụ: Air Blade

Năm sản xuất 16

Tự trọng 17

Số người được chở 18

Thông tin thanh toán

Ngày nộp phí * 19

15/06/2015

Thêm hợp đồng 21

Figure 71 Interface - <Staff> Create contract

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Leng th
1	Menu	Navigation bar	Yes	Yes	Menu bar	N/A	N/A

2	Title	Title of the page	Yes	Yes	Label	N/A	N/A
3	txtCustomerCode	Fill customer code	No	Yes	Textbox	String	6
4	ddlContractType	Select contract type	No	Yes	Select	N/A	N/A
5	txtStartDate	Select start date	No	Yes	Date input form	N/A	N/A
6	txtExpiredDate	Select expired date	No	Yes	Date input form	N/A	N/A
7	txtContractFee	Contract fee	Yes	Yes	Label	N/A	N/A
8	txtPlate	Fill vehicle plate	No	Yes	Textbox	String	4 – 15
9	txtBrand	Fill vehicle brand	No	Yes	Textbox	String	2 – 20
10	txtEngine	Fill vehicle engine	No	Yes	Textbox	String	2 – 20
11	txtChassis	Fill vehicle chassis	No	Yes	Textbox	String	2 – 20
12	txtCapacity	Fill vehicle capacity	No	Yes	Textbox	String	2 – 20
13	txtColor	Fill vehicle color	No	No	Textbox	String	2 – 20
14	txtType	Fill vehicle type	No	No	Textbox	String	2 – 20
15	txtModelCode	Fill vehicle model code	No	No	Textbox	String	2 – 20
16	txtYearOfMan	Fill vehicle year of manufacture	No	No	Textbox	String	4
17	txtWeight	Fill vehicle empty weight	No	No	Textbox	String	1 – 4
18	txtSeatCapacity	Fill vehicle seat capacity	No	No	Textbox	String	1 – 3
19	txtPaidDate	Select paid contract fee date	No	Yes	Date input form	N/A	N/A

Table 91 <Staff> Create contract fields

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
20	btnSelect	Select customer	N/A	Transfer to select customer modal
21	btnCreate	Create new contract	N/A	Transfer to create new contract successful page

Table 92 <Staff> Create contract buttons/hyperlinks

5.2.1.6.Renew contract

The screenshot shows a user interface for renewing a insurance contract. The fields and their values are:

- 1. Gia hạn hợp đồng
- 2. Thời điểm có hiệu lực: 08/05/2015 lúc 00:00:00
- 3. Gia hạn đến*: 15/06/2016
- 4. Phí gia hạn*: 75.000 đ
- 5. Ngày nộp phí*: 15/06/2015
- 6. A blue button labeled 'Gia hạn hợp đồng'.
- 7. A blue button labeled 'Đóng'.

Figure 72 Interface - <Staff> Renew contract

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Leng th
1	Title	Title of the modal	Yes	Yes	Label	N/A	N/A

2	Start date	Contract's start date	Yes	Yes	Label	N/A	N/A
3	txtExpir edDate	Select expired date	No	Yes	Date input form	N/A	N/A
4	txtAmo unt	Renew contract fee	Yes	Yes	Label	N/A	N/A
5	txtPaid Date	Select paid contract fee date	No	Yes	Date input form	N/A	N/A

Table 93 <Staff> Renew contract fields

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	btnRenew	Renew contract	N/A	Transfer to renew contract successful page
7	btnClose	Close modal	N/A	Transfer to contract detail page

Table 94 <Staff> Renew contract buttons/hyperlinks

5.2.1.7.Cancel contract

The screenshot shows a user interface for canceling a contract. At the top, the title "Hủy hợp đồng" is displayed. Below it, there is a field labeled "Ngày hủy hợp đồng *". A date input box shows "15/06/2015". To the right of this box is a red circle containing the number "2". Below the date field is a text input field for "Lý do hủy hợp đồng *". To its right is a red circle containing the number "3". Further down is a larger text area for "Ghi chú", which is currently empty. To its right is a red circle containing the number "4". At the bottom of the interface are two buttons: a blue button labeled "5" and a red button labeled "6". The red button contains the text "✓ Đóng ý hủy" and the blue button contains the text "Đóng".

Figure 73 Interface - <Staff> Cancel contract

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Title	Title of the modal	Yes	Yes	Label	N/A	N/A
2	txtCancelDate	Select cancel contract date	No	Yes	Date input form	N/A	N/A
3	txtCancelReason	Fill cancel contract reason	No	Yes	Textbox	String	1 – 250
4	txtCancelNote	Fill note about cancel contract	No	No	Text area	String	1 - 2000

Table 95 <Staff> Cancel contract fields

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	btnCancel	Cancel contract	N/A	Transfer to cancel contract successful page
6	btnClose	Close modal	N/A	Transfer to contract detail page

Table 96 <Staff> Cancel contract buttons/hyperlinks

5.2.2. Customer Interface Design

5.2.2.1. Manage contract

#	Mã hợp đồng	Ngày bắt đầu	Thời hạn	Trạng thái
1	HD00RU	2015-06-10 20:41:37.0	2016-06-10 20:41:37.0	Ready
2	HD0A33	2001-02-02 00:00:00.0	2002-02-02 00:00:00.0	Request cancel
3	HD0E66	2015-03-15 00:00:00.0	2016-03-15 00:00:00.0	Cancelled
4	HD33FD	2015-06-10 00:00:00.0	2015-06-10 23:12:52.0	Ready
5	HD453C	2015-04-25 00:00:00.0	2016-04-25 00:00:00.0	Request cancel
6	HD6679	2015-02-18 00:00:00.0	2016-02-18 00:00:00.0	Ready
7	HD76FF	2012-02-18 00:00:00.0	2016-02-18 00:00:00.0	Cancelled

Figure 74 Interface - <Customer> Manage contract

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	lblMenu	Navigation bar	Yes	Yes	MenuBar	N/A	N/A
2	lblTitle	Title of the page	Yes	Yes	Label	N/A	N/A
3	lblAmountContract	Number of available customer	Yes	No	Label	N/A	N/A
4	lblContractStatus	Status of contract	Yes	No	Label	N/A	N/A
5	txtSearch	Fill search keyword	No	Yes	TextBox	String	N/A
6	contractTable	Table of available contract	Yes	Yes	Table	N/A	N/A

Table 97 <Customer> Manage contract fields

Buttons / Hyperlinks

No	Function	Description	Validation	Outcome
7	btnSearch	Search by input keyword	N/A	Transfer to search result page
8	btnCreate	Create new customer	N/A	Transfer to create new customer page
9	linkContractDetail	View contract detail	N/A	Transfer to contract detail page

Table 98 <Customer> Manage contract buttons/hyperlinks

5.2.2.2.Contract detail

The screenshot shows a web-based application interface for managing insurance contracts. On the left, there is a vertical navigation menu with the following items:

- Thông tin cá nhân 1
- Hợp đồng 2
- Thông tin thẻ
- Yêu cầu thẻ mới
- Lịch sử giao dịch

The main content area displays a contract titled "Hợp Đồng HD33FD" 2. At the top right are two buttons: "Gia Hạn" 6 and "Hủy Hợp Đồng" 7.

Below the title, there are three tabs: "Thông Tin Chung" (selected), "Lịch sử bồi thường", and "Lịch sử vi phạm luật GT" 3.

The main content area contains several tables of information:

- Mã hợp đồng:** HD33FD
- Ngày tham gia:** 10/06/2015
- Ngày hết hạn:** 10/06/2015
- Quyền lợi bảo hiểm:** Trên 50 CC có bảo hiểm
- Tình trạng hợp đồng:** Ready 4
- Thông tin xe:**

Biển số đăng ký	54F8-24904	Màu sơn	Xanh Trắng
Loại xe	2 Bánh	Số máy	G45H-3256544
Nhãn hiệu	Exiter	Năm sản xuất	2013
Dung tích	125	Số khung	C-434355

Figure 75 Interface - <Customer> Contract detail

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	lblMenu	Navigation bar	Yes	Yes	Menu bar	N/A	N/A
2	lblTitle	Contract code	Yes	Yes	Label	N/A	N/A
3	txtTab	Information about contract	Yes	No	Tab	N/A	N/A
4	lblContractStatus	Status of contract	Yes	No	Label	N/A	N/A
5	txtInformation	Information about contract	No	Yes	Textbox	String	N/A

Table 99 <Customer> Contract detail fields

Buttons / Hyperlinks

No	Function	Description	Validation	Outcome
6	btnRenew	Renew contract	N/A	Show popup to choose the reason cancel
7	btnCancel	Cancel contract	N/A	Show popup to provide new time for contract and the

				way to pay for renew contract
--	--	--	--	-------------------------------

Table 100 <Customer> Contract detail buttons/hyperlinks

5.2.2.3.Cancel contract

Hủy Hợp Đồng 1

Quý khách vui lòng cung cấp lý do hoặc trường hợp hủy hợp đồng!

Xe cơ giới bị thu hồi đăng ký 2
 Xe cơ giới hết niên hạn
 Lý do khác

3 4

Xác Nhận Hủy Bỏ

Figure 76 Interface - <Customer> Cancel contract

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	lblTitle	Title	Yes	Yes	Label	N/A	N/A

Table 101 <Customer> Cancel contract fields

Buttons / Hyperlinks

No	Function	Description	Validation	Outcome
2	chkOptionCancel	Choose reason why customer cancel contract	N/A	N/A
3	btnConfirm	Confirm to cancel contract	If customer doesn't input the reason in case they choose another reason, the system will notify.	Reload contract detail with information about contract is canceled
4	btnLeave	Leave request cancel	N/A	Return to contract detail

Table 102 <Customer> Cancel contract button/hyperlinks

5.2.2.4.Renew contract

The screenshot shows a user interface for renewing a contract. At the top, it says 'Gia hạn hợp đồng 1'. Below that are three data entries:

- Thời điểm bắt đầu: 10/06/2015
- Gia hạn đến: 10/06/2016 2
- Phí thanh toán: 85.000 đ

At the bottom right are two buttons:

- 3 Gia hạn hợp đồng
- 4 Đóng

Figure 77 Interface - <Customer> Renew contract

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Date Type	Length
1	lblTitle	Title	Yes	Yes	Label	N/A	N/A
2	txtInformation	Information about renew contract	Yes	Yes	Label	N/A	N/A

Table 103 <Customer> Renew contract fields

Buttons / Hyperlinks

No	Function	Description	Validation	Outcome
3	btnConfirm	Confirm to renew contract	N/A	Transfer to checkout page
4	btnLeave	Leave request renew	N/A	Return to contract detail

Table 104 <Customer> Renew contract button/hyperlinks

5.2.2.5.Select payment method

THANH TOÁN 1
Xin lựa chọn phương thức thanh toán sau:

Nội dung thanh toán: Renew contract from 10/06/2015
to 10/06/2016 2

Tổng tiền phải trả: 85,000 (VND)



Figure 78 Interface - <Customer> Select payment method

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	lblTitle	Title	Yes	Yes	Label	N/A	N/A
2	txtInformation	Information about renew contract	Yes	Yes	Label	N/A	N/A

Table 105 <Customer> Select payment method fields

Buttons / Hyperlinks

No	Function	Description	Validation	Outcome
3	btnPaypal	Payment by paypal	N/A	Transfer to paypal page
4	btnDirect	Payment direct at company	N/A	Show address of list company

Table 106 <Customer> Select payment method buttons/hyperlinks

5.3. Checker Mobile Application Design

5.3.1. Scan NFC card screen

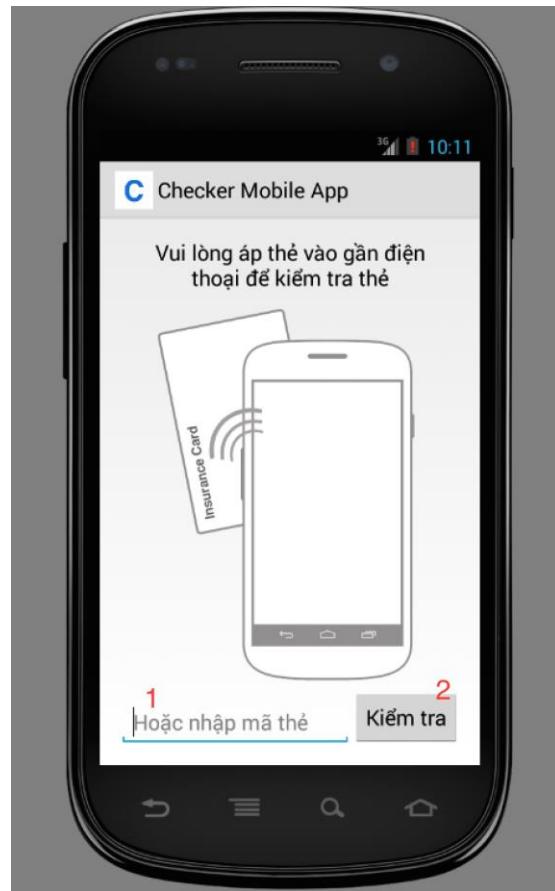


Figure 79 Scan NFC card screen

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	txtCardID	ID of the card	No	No	Textbox	String	20

Table 107 Scan NFC card screen - Fields

Buttons

No	Function	Description	Validation	Outcome
2	btnCheckCard	Verify card validation	N/A	Go to view card information screen

Table 108 Scan NFC card screen - Buttons

5.3.2. View card information screen

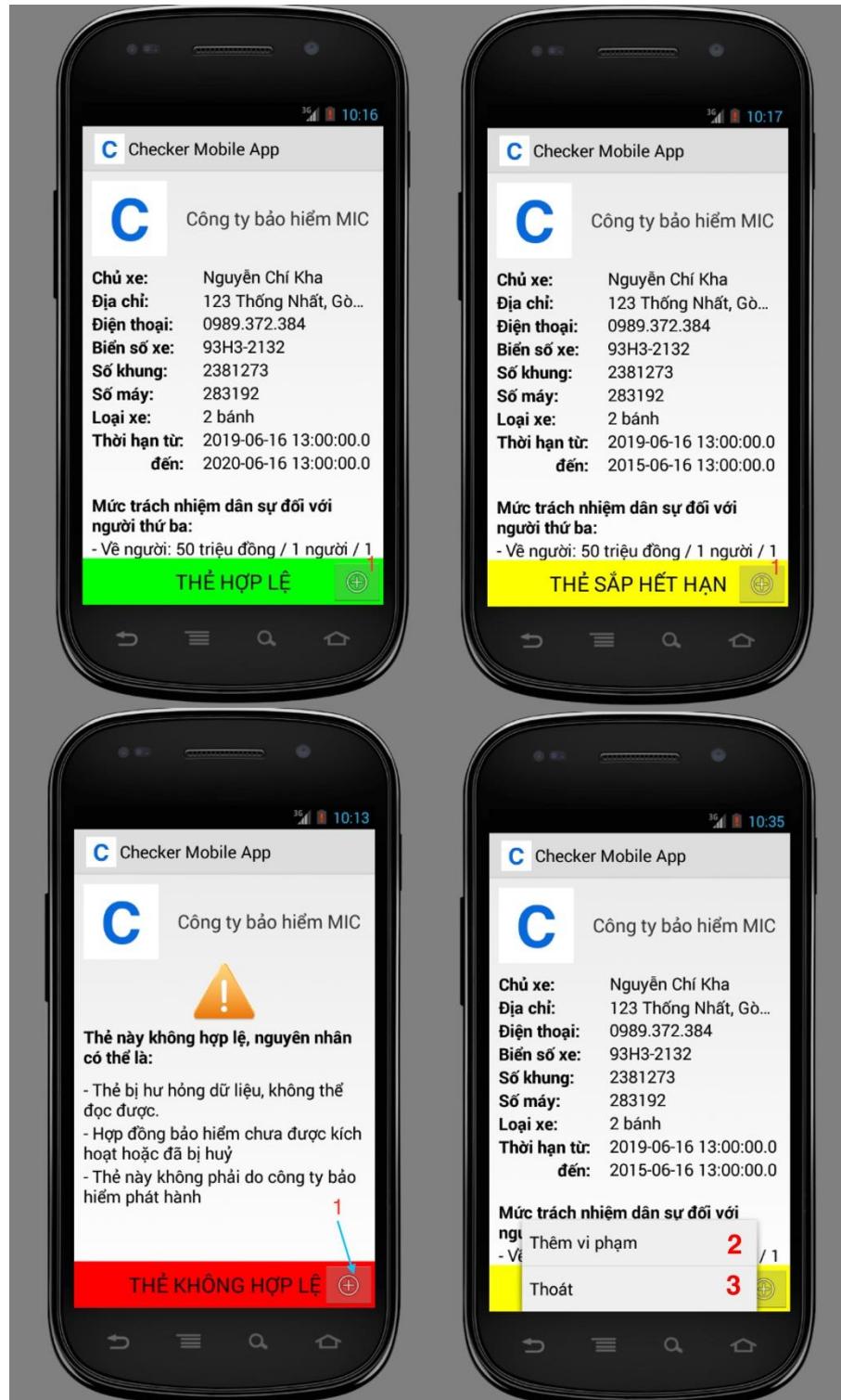


Figure 80 View card information screen

Buttons

No	Function	Description	Validation	Outcome
1	btnMenu	Show menu which contains “Add punishment” button and “Exit” button	N/A	Open menu
2	mnuAddPunishment	Add punishment information	N/A	Open add punishment dialog
3	mnuExit	Exit the current page	N/A	Return to Scan screen

Table 109 View card information screen - Buttons

5.3.3. Add punishment screen



Figure 81 Add punishment screen

Fields

No	Field Name	Description	Rea d only	Mandator y	Control Type	Data Type	Lengt h

1	txtDescription	Description of punishment	No	No	Textbox	String	2-200
2	flePicture	Picture of punishment	No	Yes	Button	Image	Width: 100px

Table 110 Add punishment screen - Fields

Buttons

No	Function	Description	Validation	Outcome
1	btnSend	Send punishment information to server	N/A	Show information sent success, back to home screen.

Table 111 Add punishment screen - Buttons

5.4. Printer Mobile Application Design

5.4.1. Search contract screen

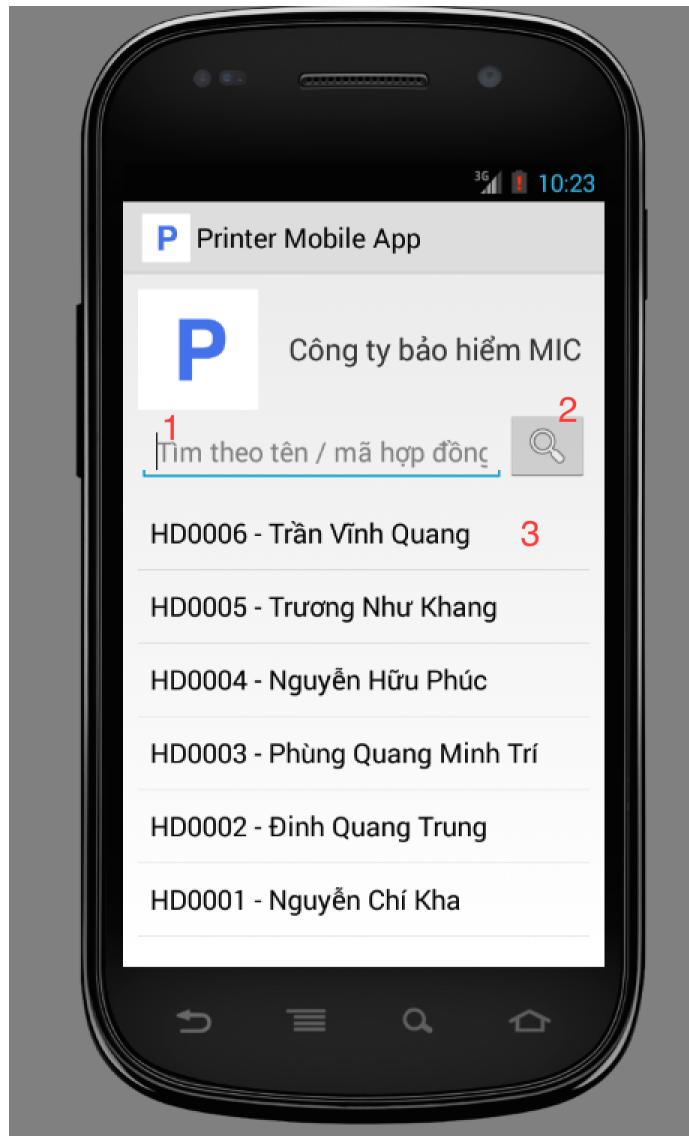


Figure 82 Search contract screen

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	txtSearch	Search box where user input search keyword	No	No	Textbox	String	0 - 200

Table 112 Search contract screen - Fields

Buttons

No	Function	Description	Validation	Outcome
----	----------	-------------	------------	---------

2	btnSearch	Search contract with keyword	N/A	The list will be updated with search results
3	lstContracts	Click to a row from the list to view detail information of that contract	N/A	Show view contract screen

Table 113 Search contract screen - Buttons

5.4.2. View contract screen



Figure 83 View contract screen

Buttons

No	Function	Description	Validation	Outcome
----	----------	-------------	------------	---------

1	btnPrint	Click to print information to NFC card	Phone have to have NFC support	Open print card screen
---	----------	----------------------------------------	--------------------------------	------------------------

Table 114 View contract screen - Buttons

5.4.3. Print card screen



Figure 84 Print card screen

6. Database Design

6.1. Entity relationship diagram

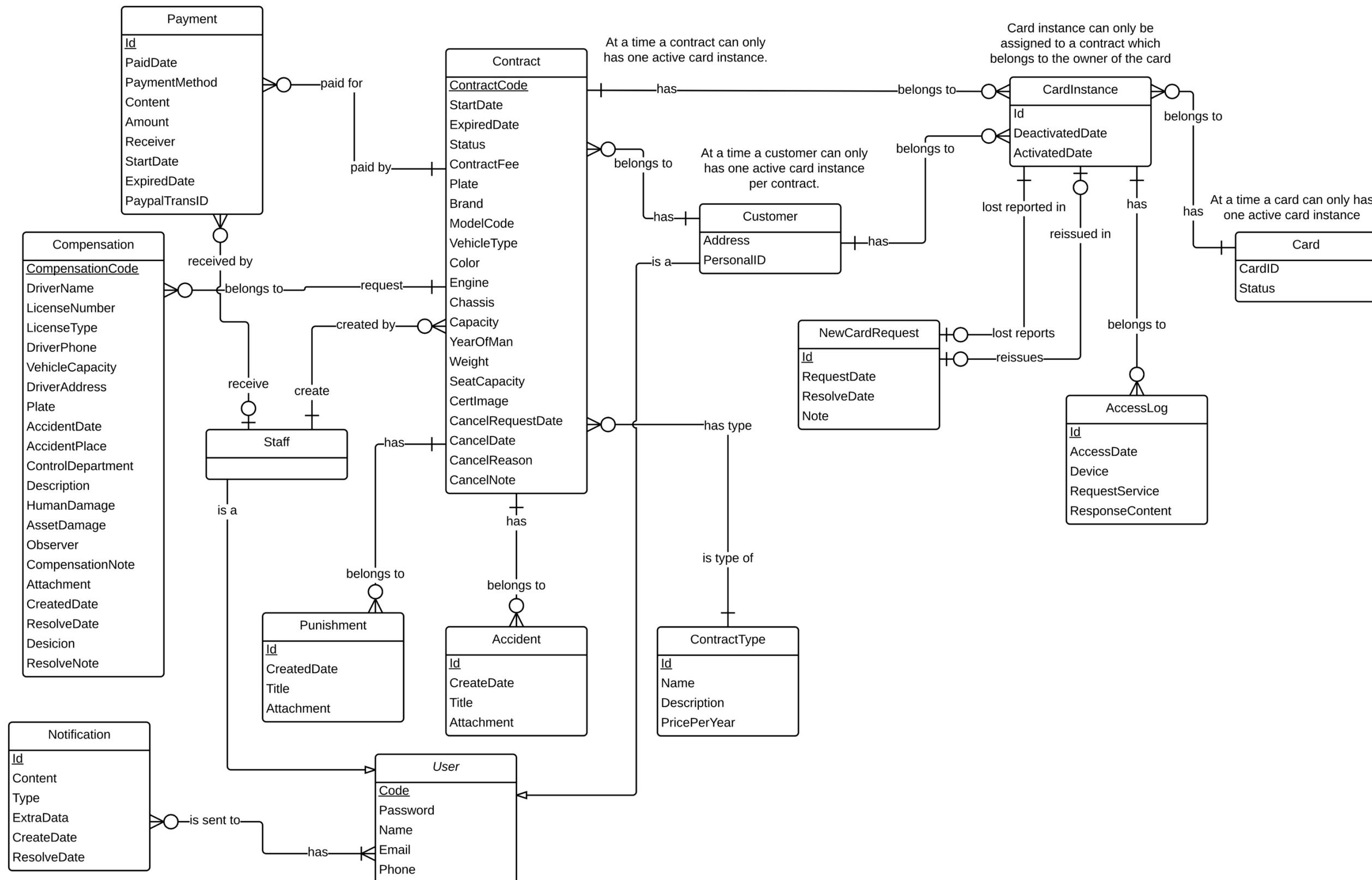


Figure 85 Entity relationship diagram

6.2. Entity Dictionary

Entity Data Dictionary: describe content of all entities	
Entity name	Description
User	Abstract entity describes a user in system
Customer	Contain the customer information.
Contract	Contain the contract information.
Card	Contain the card information
CardInstance	Represent a card assigned to a contract
Payment	Contain the payment information.
Staff	Contain the staff information.
Compensation	Contain the compensation information.
Punishment	Contain the punishment information.
Accident	Contain the accident information.
ContractType	Contain the contract type information.
NewCardRequest	Contain the new card request information.
CardAccessLog	Contain the new card access log.
Notification	Contain the notification information

Table 115 Entity dictionary

7. Algorithms

7.1. Contract State

The contracts in MIC system is complex and can be managed differently during the operation. The state chart bellow describes all the state of a contract.

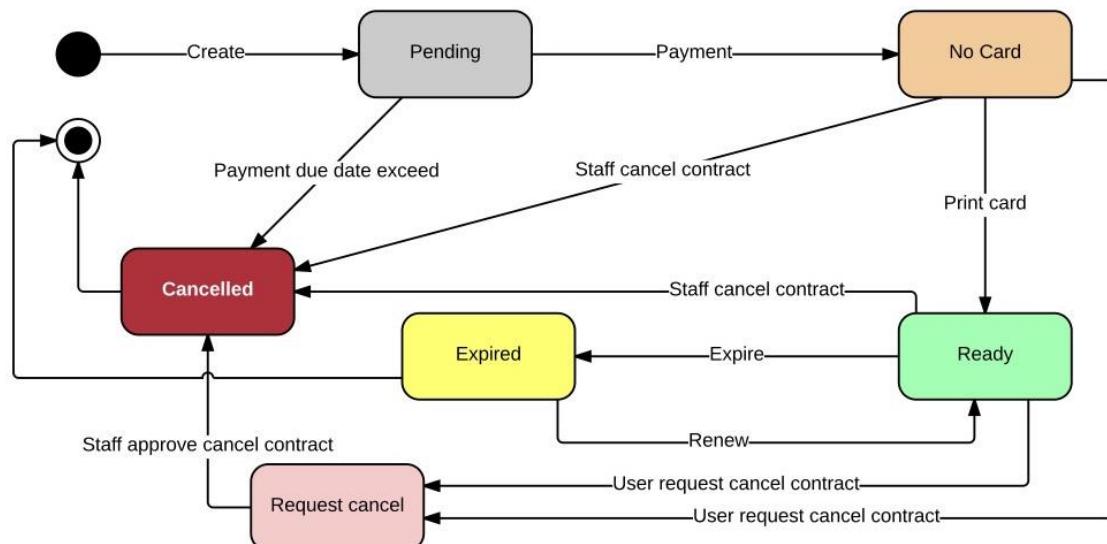


Figure 86 Contract State Chart

State	Description
-------	-------------

Pending	The contract is created and do not have payment or the start date is not come yet
No card	The contract had have payment and contract start date is arrived but have not assigned to a card
Ready	The contract is assigned with a card and ready to use
Expired	The contract due date is exceed and no longer valuable
Request cancel	The contract is requested to cancel by customer
Cancelled	The contract is cancelled and no longer valuable

Table 116 Contract State Dictionary

Stimulus	Description	State before	State after
Create	Customer create new contract	N/A	Pending
Payment	Customer pay for the contract via PayPal or direct payment and contract start date is arrived.	Pending	No card
Payment due date exceed	When the payment due date is exceed	Pending	Cancelled
Print card	Staff use mobile app to print the card for customer, the card ID is saved to the system	No card	Ready
Expire	When the contract due date is exceed, system will change the contract status	Ready	Expired
Renew	Customer renew contract via web application or direct payment	Expired	Ready
Staff cancel contract	Staff cancel contract via web application	No card, Ready	Cancelled
Customer cancel contract	Customer cancel contract via web application	No card, Ready	Request cancel
Staff approve cancel contract	Staff approve cancel request from customer via web application	Request cancel	Cancelled

Table 117 Contract State Flow

7.2. Concurrency Control

7.2.1. Definition

Concurrency control is a method to handle data which can be accessed and modified from many sources at the same time to prevent conflict and data loss.

7.2.2. Define problem

In web application, one contract might be edited by stakeholders at the same time, this situation may cause an inconsistent state of the contract. We have to find a solution for this problem.

7.2.3. Solution

We defined a list of entity that might be modified by many sources at a same time:

- **Contract:** can be edited/renew by stakeholders at a same time. This includes the information assigned with the contract: Payment, Compensation, Accident and Punishment information.

We use timestamp concurrency control method to solve the concurrency problem. Contract information have a “last modified” value, this value represents the last time the contract is modified. When update the contract information, a validation will run to check if the last modified is changed or not. If it is changed there will be error message to users.

7.2.4. Flow chart

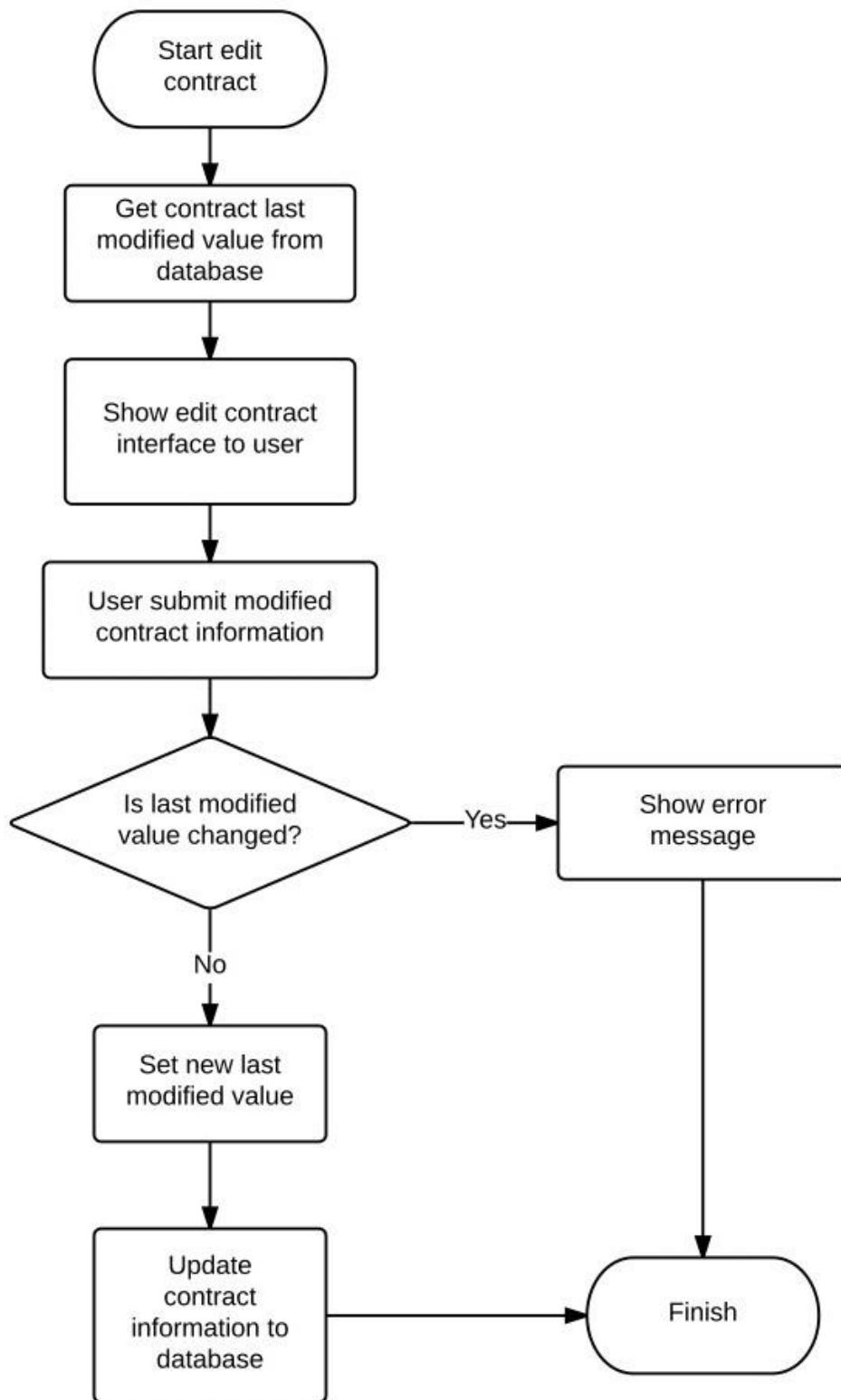


Figure 87 Concurrency control flow

7.3. Notification

7.3.1. Definition

Notification is a feature of MIC Web application to notify stakeholders when important events occurs.

7.3.2. Notification Methods

- Send email to receiver
- Show notification icon on web interface

7.3.3. Notification use cases

Type	Trigger	Receiver	Notify method
1	Customer creates new contract	All staff	Web
2	Customer sends compensation request	All staff	Web
3	Customer sends new card request	All staff	Web
4.1	Contract is nearly expired (send first)	Customer	Web, Email
4.2	Contract is nearly expired (send 2nd time)	Customer	Web, Email
4.3	Contract is nearly expired (send 3rd time)	Customer	Web, Email
5	Contract is expired	Customer	Web, Email
6	Customer sends request cancel	All staff	Web
7	Contract is cancelled because payment due date exceed	Customer	Web, Email
8	Paid Pending Contract start date come	Customer	Web
9	Compensation resolved	Customer	Web, Email

Table 118 Notification use cases

- For staffs, every notification has "status" to show that if the notification not resolved or when the notification is resolved and by whom. Except Type 1 notification has no status.
- All the notification can be handedly turned off by clicking Mark as read button
- When a notification is sent to all the staff, each staff has their own "Read" status. That's mean if staff A mark the status as read but staff B still see it as not read.

7.4. System Scheduler Process

7.4.1. Definition

System scheduler is a component of the Web application, this component is responsible for checking the changes from web application and updates information day by day.

7.4.2. Define problem

In Web application, the we need a system scheduler that runs every day at 00:00 to check the status of contracts, send notification via web, emails, changes the contract status if the due date is exceed... etc.

7.4.3. Solution

We create a scheduler in operating system that run on the same server of the Web application, scheduler will automatically run at specific time that system administrator defines when deploy the system. The scheduler will access to database to check the status and update information to database.

The checking process is described as follow:

1. Get all the contract from the system.
2. With each contract, check following information:
 - a. Contract status
 - b. Contract expired date
 - c. Contract renew due date
 - d. Contract payment due date
 - e. Compensation requests
 - f. New card requests
 - g. Cancel request
3. Update the contract state follow the business rules.

7.4.4. Flow chart

The following flow chart diagram describes the process of the System scheduler every time it runs.

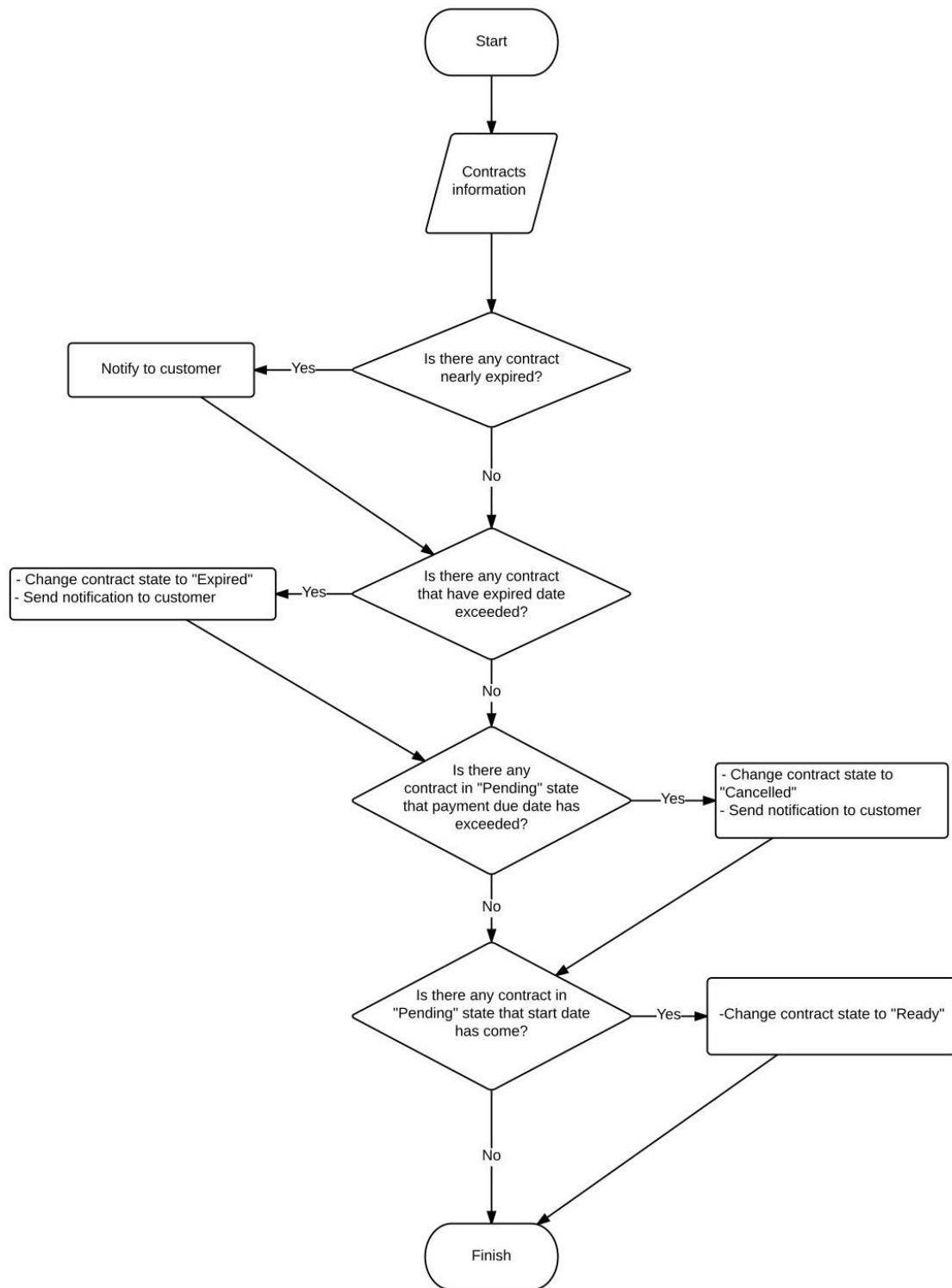


Figure 88 System Scheduler Process

7.5. NFC Card Data Format

7.5.1. Definition

According to Android Developer Forum definition, Near Field Communication (NFC) is a set of short-range wireless technologies, typically

requiring a distance of 4cm or less to initiate a connection. NFC allows you to share small payloads of data between an NFC tag and an Android-powered device, or between two Android-powered devices. NFC Card Data format is a type of format to store data in the card's storage.

7.5.2. Define problem

The data stored in the tag can be written in a variety of formats, in Printer application and Checker application we use a NFC Forum standard called NDEF (NFC Data Exchange Format) which is highly recommended from Android developer community.

According to *NFC Forum™ NDEF 1.0 Specification (2006-07-24)*, NDEF Messages is stored in tag in this structure:

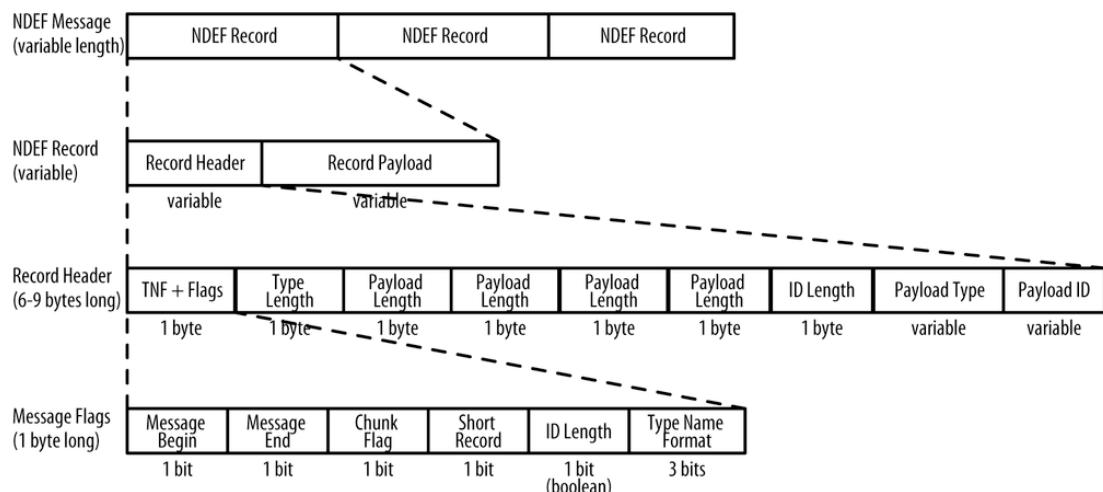


Figure 89 NDEF message structure

Reference: Brian Jepson, Don Coleman, Tom Igoe. (January 2014). *Beginning NFC, Chapter 4 - Introduction to NDEF, NDEF Structure.* O'Reilly Media, Inc.

NDEF supports followings field types of message:

Type Name Format	Description
Empty	Empty record that has no type or payload
Well-known	One of several pre-defined types laid out in the NFC Forum RTD specification.
MIME media-type	An Internet media type as defined in RFC 2046.
Absolute URI	A URI as defined in RFC 3986.
External	A user-defined value, based on rules in the NFC Forum Record Type Definition specification.
Unknown	Type is unknown. Type length must be 0.

Unchanged	Only for middle and terminating records of chunked payloads. Type length must be 0.
Reserved	Reserved by the NFC Forum for future use.

Table 119 NDEF Message Types

Reference: NFC Specification, Appendix A. NFC Specification Codes.

To prevent malicious users to override data on the NFC card or using fake card, we need to find a solution to protect data written on the card.

7.5.3. Solution

For security reason, we decided to not to write any contract information data to the tag but use only the card low-level ID to verify the card.

Card low-level ID, as known as Manufacture ID (IDm) is an 8-byte number that is used by the NFC Forum Device to address a Type 3 NFC Tag.

Reference: *NFC Forum™ Type 3 Tag Operation Specification 1.1, 2011-06-28.*

To make the card ID more friendly to users, we decided to convert the IDm to a series of 2-digit hex number which turn the 8-byte number into a string follow this regex format:

[0-9A-F]{16}

Example: 561F3ADF328D9C4B.

To make sure the card can only be read by our Checker application, we write an Android Application Record (AAR) with the package name of the Checker Mobile application.

In this project we have to use NFC card Type 4, which supports IDm and NDEF data format. Other type of NFC card might result different IDm.

Bellow is the flow to write an NFC tag for customer:

1. Printer application get contract information
2. Staff confirm the contract information is correct
3. Printer read card ID
4. If the card is not existing in system, update the card ID to the contract
5. Write AAR record to the tag and finish.

Bellow is the flow to read an NFC tag for police officer:

1. Checker application read card ID from the card
2. Checker application send card ID to system to verify the card
3. Show result to police officer

7.6. Strategy for future expand plan

MIC system is currently an application to deploy to a single company with single contract type. Our vision for the future plan is to deploy MIC system to multiple companies and with multiple types of contract (motor insurance, health insurance, asset insurance...).

7.6.1. Isolated Single Service

Isolated Single Service Deployment Model

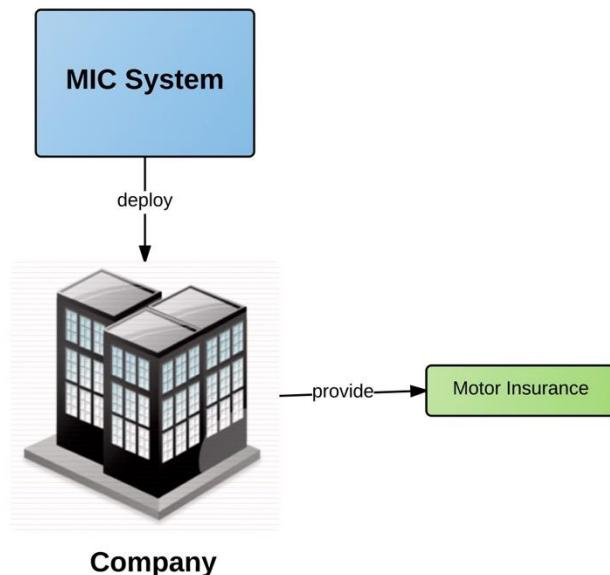


Figure 90 Isolated Single Service Deployment Model

Based on the original requirement, this is the model of MIC system which support manage motor insurance contract. The MIC System will be deployed to a company which provide a single service: motor insurance service.

As a part of the implement process, to obtains the scalability of the system, this is what we designed the components:

* Note: some entities and relations are omitted to simplify the diagram

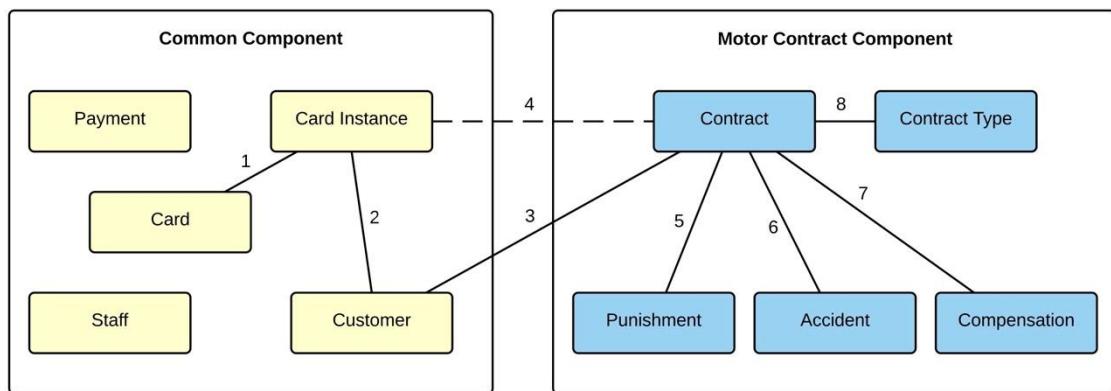


Figure 91 Entities group by components (Isolated Single Service)

From the diagram, we can see the Motor Contract Component is separated with the Common Component which contains common information such as Customer, Card, Payment... This have benefits that if we want to add a new contract type (for example heath insurance contract), the cost of modify the old system is minimize to nearly zero.

Next, we will show the process to expand this system to the new system that supports multiple service.

7.6.2. Isolated Multiple Service

Isolated Multiple Service Deployment Model

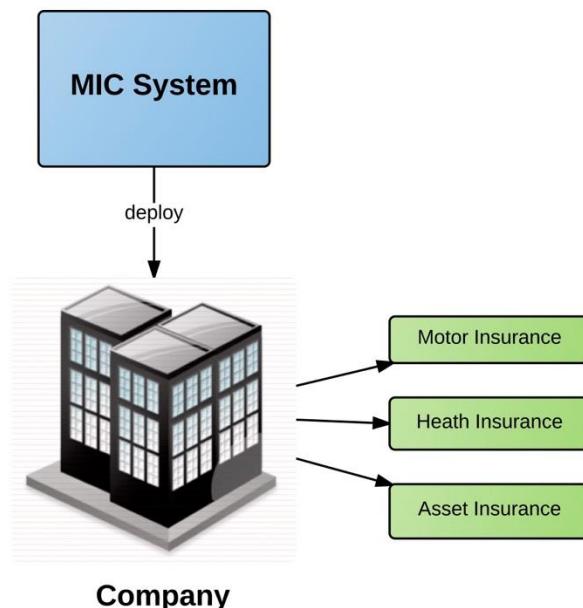


Figure 92 Isolated Multiple Service Deployment Model

This is an expanded MIC system where a company can use MIC System to manage multiple contract type for example motor insurance, health insurance and asset insurance.

Based on the original model, we can implement this model by simply add new component to the current system, this is the planned component and relationships between entities:

* Note: some entities and relations are omitted to simplify the diagram

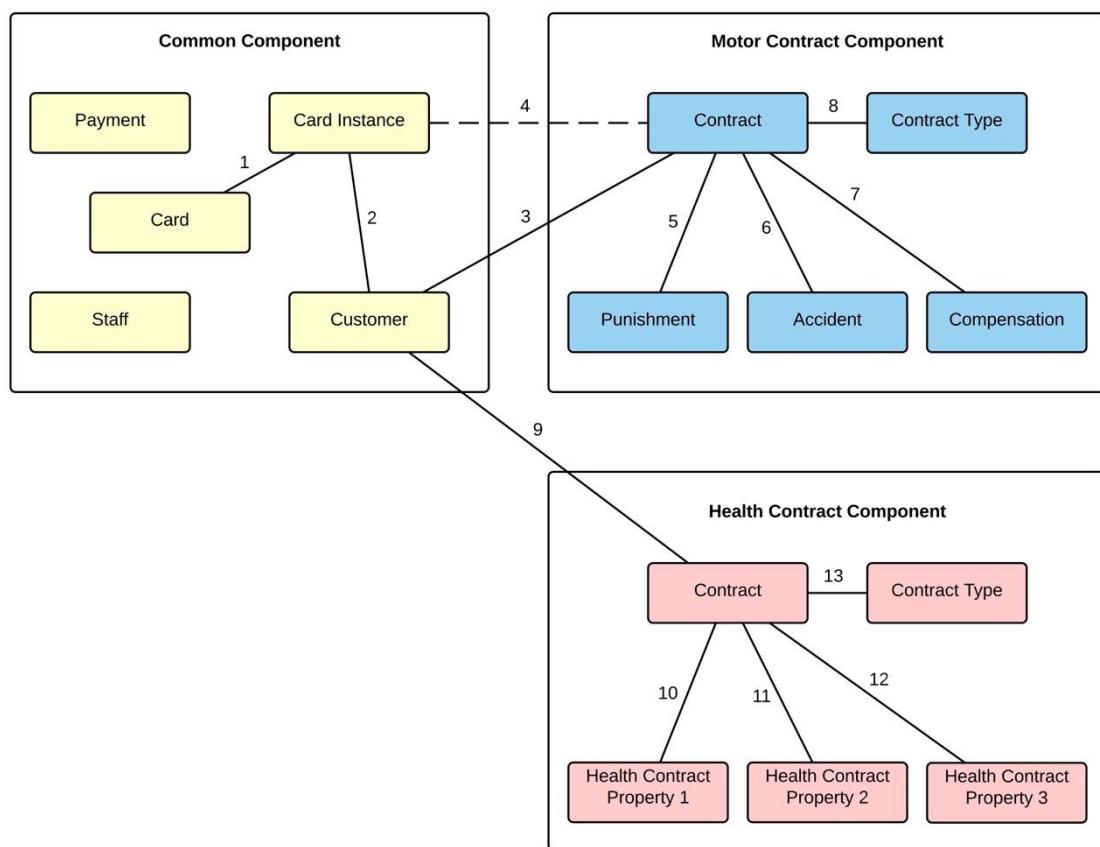


Figure 93 Entities group by components (Isolated Multiple Service)

7.6.3. Distributed Multiple Service

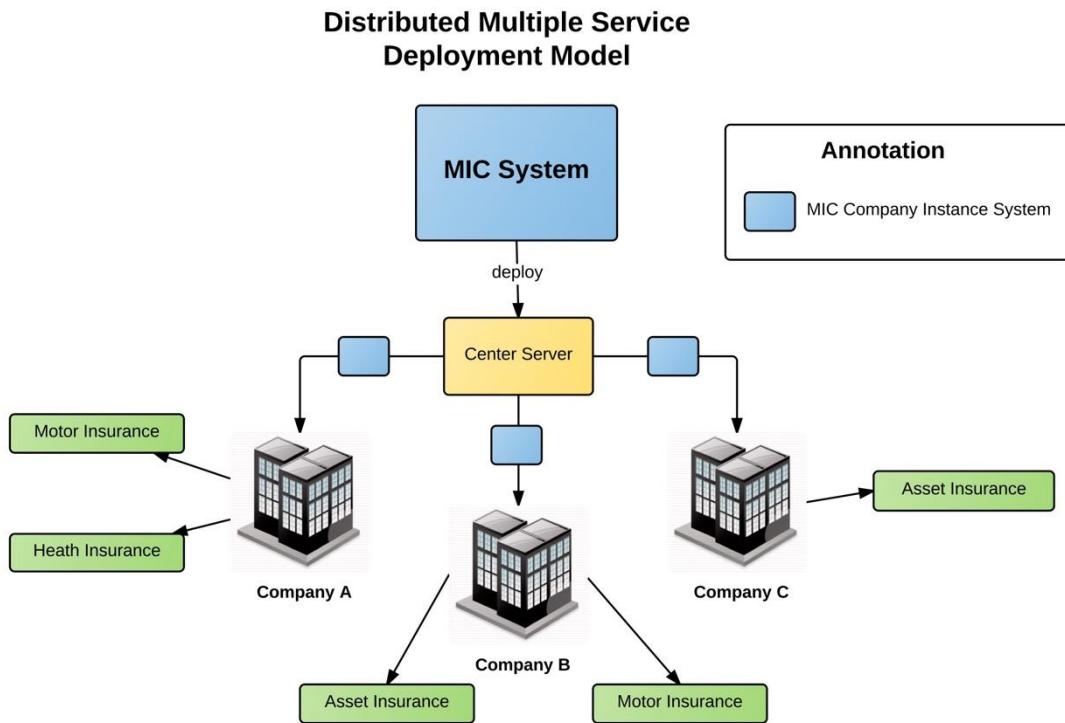


Figure 94 Distributed Multiple Service Deployment Model

This is the final model of MIC system from our vision. In this model, we deploy MIC System in a Center Server which operates multiple insurance company, each company can provide different types of contract (motor insurance, health insurance, asset insurance...). All the company use a unify card system and customer information. By this model, a customer can have multiple contracts from different companies using only one NFC card to use for all the services.

This is the expected deployment component we estimated:

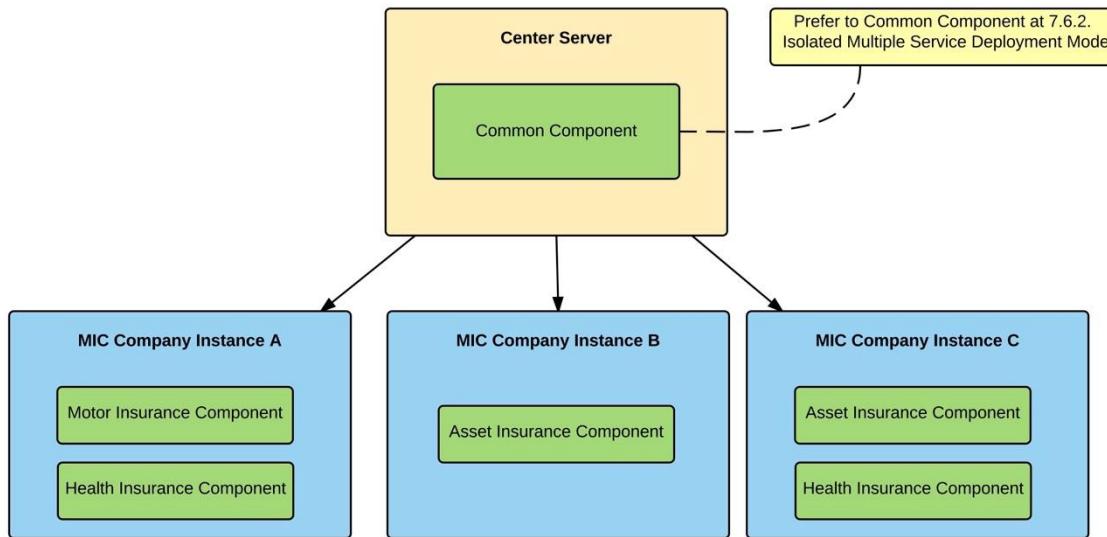


Figure 95 Components group by container (Distributed Multiple Service)

The Distributed Multiple Service plan is still need to be investigate a lot to clarify the components and relationships between entities as well as the architecture design to maintain the best performance and less impact to the current system, we also need to identify the arise problems when switching from isolated model to distributed model.

E. System Implementation & Test

1. Introduction

1.1. Overview

This section describes the approach and methodologies used by group to plan, organize and manage the testing of MIC system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

1.2. Test Approach

- Goal: Test all features in the whole MIC system based on the core flow.
- Method: black-box testing
- Technique: check list

The testing for this project will consists of Integration System test level. Testing the program which was integrated and as a complete system to ensure that the software requirements have been met.

- Integration testing would be performed by all member of team and approved by team leader.
- System testing is focused on assessing the system's reliability. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

2. Database Relationship Diagram

2.1. Physical Diagram

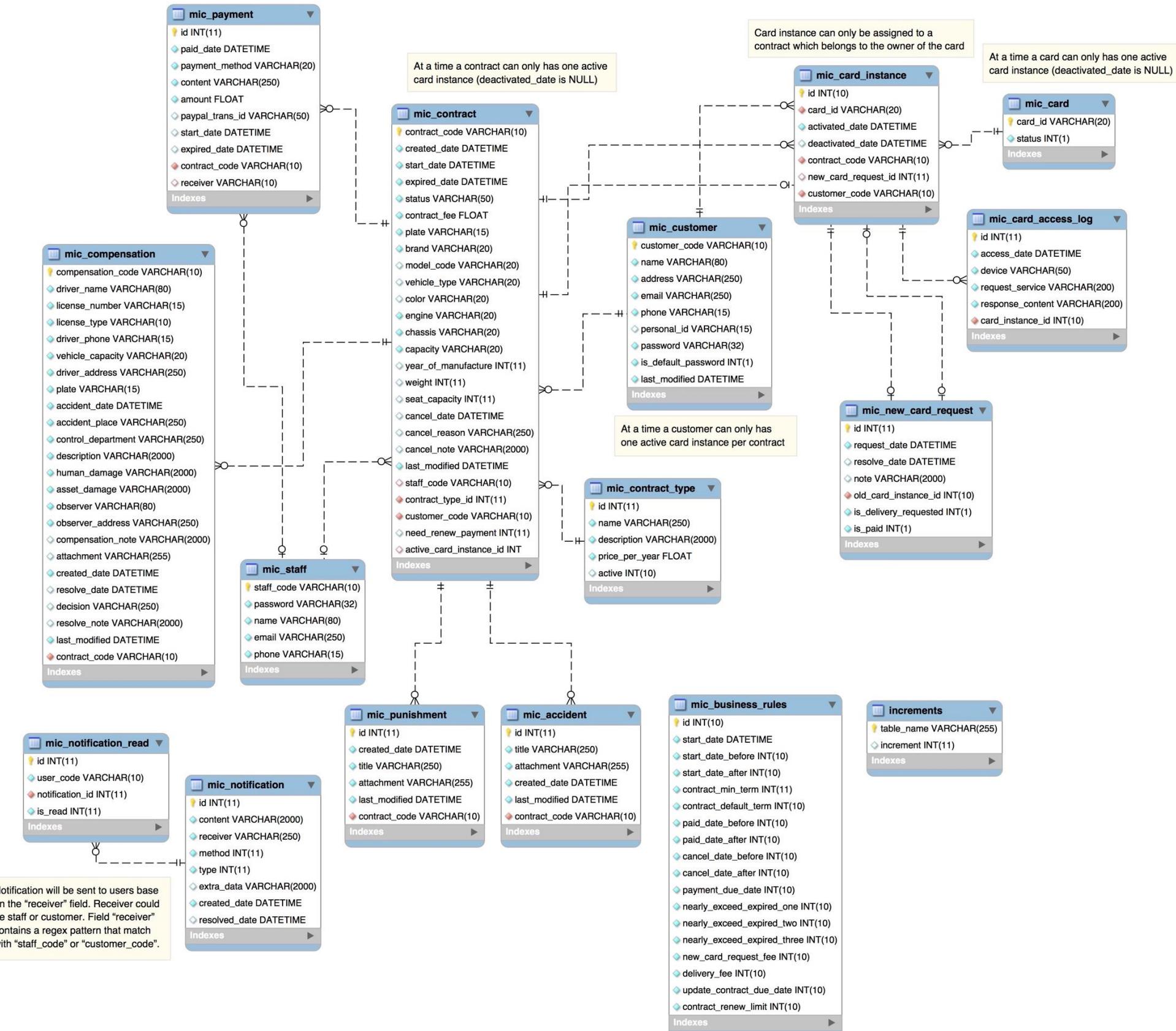


Figure 96 Physical diagram

2.2. Data Dictionary

Data dictionary: describe content of all tables		
Table Name	Mapping with Conceptual diagram	Description
increments	N/A	Not exist in conceptual diagram. Needed in physical diagram to generate codes with prefixes sequentially.
mic_business_rules	N/A	Not exist in conceptual diagram. Needed in physical diagram to describe all configuration based on company's business rules.
mic_contract_type	ContractType	Describe all contract type in the system.
mic_contract	Contract	Describe all contract detail information in the system.
mic_accident	Accident	Describe all accident information belongs to contract in the system.
mic_compensation	Compensation	Describe all compensation detailed information belongs to contract in the system.
mic_punishment	Punishment	Describe all punishment information belongs to contract in the system.
mic_card	Card	Describe all card, which has been issued in the system.
mic_card_instance	CardInstance	Describe all card instance.
mic_card_access_log	N/A	Not exist in conceptual diagram. Needed in physical diagram to describe all history access to card.
mic_new_card_request	NewCardRequest	Describe all new card request in the system.
mic_customer	Customer	Describe all customer profile available in the system.
mic_notification	N/A	Not exist in conceptual diagram. Needed in physical diagram to describe all notification detail in the system.
mic_notification_read	N/A	Not exist in conceptual diagram. Needed in physical diagram to support manage all notification in the system.
mic_payment	Payment	Describe all payment information in the system.

mic_staff	Staff	Describe all staff profile available in the system.
------------------	-------	-----------------------------------------------------

Table 120 Data table dictionary

Table Name	Attributes	Description	Domain	Null
increments	table_name {PK}	Unique identifier of increments	varchar(255)	No
	increment			Yes
mic_business_rules	id {PK}	Unique identifier of business rules	int(10)	No
	start_date_before	Duration contract start before current date	int(10)	No
	start_date_after	Duration contract start after current date	int(10)	No
	contract_min_term	Minimum contract term	int(11)	No
	contract_default_term	Default contract term	int(10)	No
	paid_date_before	Duration pay for contract before current date	int(10)	No
	paid_date_after	Duration pay for contract after current date	int(10)	No
	cancel_date_before	Duration cancel contract before current date	int(10)	No
	cancel_date_after	Duration cancel contract after current date	int(10)	No
	payment_due_date	Duration pay for contract from create date	int(10)	No
	nearly_exceed_expired_one	Duration first time notify contract is going to expired	int(10)	No
	nearly_exceed_expired_two	Duration second time notify contract is going to expired	int(10)	No
	nearly_exceed_expired_three	Duration third time notify contract is going to expired	int(10)	No
	new_card_request_fee	The amount has to paid for new card	int(10)	No
	delivery_fee	The amount has to paid for delivery new card	int(10)	No

mic_contract_type	id {PK}	Unique identifier of contract type, auto increment.	int(11)	No
	name	Contract type name.	varchar(250)	No
	description	Contract type description.	varchar(2000)	No
	price_per_year	Contract type price per year.	float	No
mic_contract	contract_code {PK}	Unique identifier of contract.	varchar(10)	No
	created_date	Contract created date	datetime	No
	start_date	Contract start date	datetime	No
	expired_date	Contract expired date	datetime	No
	status	Contract status	varchar(50)	No
	contract_fee	Contract fee	float	No
	plate	Vehicle plate	varchar(15)	No
	brand	Vehicle brand	varchar(20)	No
	model_code	Vehicle model code	varchar(20)	Yes
	vehicle_type	Vehicle type	varchar(20)	Yes
	color	Vehicle color	varchar(20)	Yes
	engine	Vehicle engine number	varchar(20)	No
	chassis	Vehicle chassis number	varchar(20)	No
	capacity	Vehicle capacity	varchar(20)	No
	year_of_manufacture	Vehicle year of manufacture	int(11)	Yes
	weight	Vehicle empty weight	int(11)	Yes
	seat_capacity	Vehicle seat capacity	int(11)	Yes
	cancel_date	Contract cancel date	datetime	Yes
	cancel_reason	Contract cancel reason	varchar(250)	Yes
	cancel_note	Contract cancel note	varchar(2000)	Yes
mic_accident	last_modified	Contract last modified	datetime	No
	staff_code	Code of the staff who created this contract	varchar(10)	
	contract_type_id	Contract type id	int(11)	No
	customer_code	Code of the customer who own this contract	varchar(10)	No
	active_card_instance_id	Active card instance of the contract	int(11)	Yes

mic_compensation	compensation_code {PK}	Unique identifier of compensation	varchar(10)PK	No
	driver_name	Compensation driver name	varchar(80)	Yes
	license_number	Compensation driver license number	varchar(15)	Yes
	license_type	Compensation driver license type	varchar(10)	Yes
	driver_phone	Compensation driver phone	varchar(15)	Yes
	vehicle_capacity	Compensation driver vehicle capacity	varchar(20)	Yes
	driver_address	Compensation driver address	varchar(250)	Yes
	plate	Compensation driver vehicle plate	varchar(15)	Yes
	accident_date	Compensation accident date	datetime	Yes
	accident_place	Compensation accident place	varchar(250)	Yes
	control_department	Department which control this accident	varchar(250)	Yes
	description	Compensation description	varchar(2000)	Yes
	human_damage	Accident human damage	varchar(2000)	Yes
	asset_damage	Accident asset damage	varchar(2000)	Yes
	observer	Observer name	varchar(80)	Yes
	observer_address	Observer address	varchar(250)	Yes
	compensation_note	Compensation note	varchar(2000)	Yes
	attachment	Compensation attachment	varchar(255)	Yes
	created_date	Compensation created date	datetime	No
	resolve_date	Compensation resolve date	datetime	Yes
	decision	Compensation decision	varchar(250)	Yes
	resolve_note	Compensation resolve note	varchar(2000)	Yes
	last_modified	Compensation last modified time	datetime	No
	contract_code	Code of the contract has this compensation	varchar(10)	No
mic_punishment	id {PK}	Unique identifier of punishment, auto increment.	int(11)	No

	title	Punishment title	varchar(250)	No
	attachment	Punishment attachment	varchar(255)	No
	created_date	Punishment created date	datetime	No
	last_modified	Punishment last modified	datetime	No
	contract_code	Code of the contract has this punishment	varchar(10)	No
mic_card	card_id	Unique identifier of card	varchar(20)	No
	status	Status of the card	int(1)	No
mic_card_instance	Id {PK}	Unique identifier of card instance.	int(11)	No
	card_id	Card ID of the instance	varchar(20)	No
	activated_date	Card activated date	datetime	No
	deactivated_date	Card deactivated date	datetime	Yes
	contract_code	Code of contract has this card	varchar(10)	No
	new_card_request_id	New card request id	int(11)	Yes
	customer_code	Customer code, owner of the card	varchar(10)	No
mic_card_access_log	id {PK}	Unique identifier of card access log, auto increment.	int(11)	No
	access_date	Access card date	datetime	No
	device	Access card device	varchar(50)	No
	request_service	Service requested	varchar(200)	No
	response_content	Content responded	varchar(200)	No
	card_id	Card ID	varchar(20)	No
mic_new_card_request	id {PK}	Unique identifier of new card request, auto increment.	int(11)	No
	request_date	Send request date	datetime	No
	resolve_date	Resolve request date	datetime	Yes
	note	Resolve request note	varchar(2000)	Yes
	old_card_id	ID of old card	varchar(20)	No
mic_customer	customer_code{PK}	Unique identifier of customer	varchar(10)	No
	name	Customer name	varchar(80)	No
	address	Customer address	varchar(250)	No
	email	Customer email	varchar(250)	No
	phone	Customer phone	varchar(15)	No
	personal_id	Customer personal ID	varchar(15)	Yes
	password	Customer password	varchar(32)	No
	is_default_password	Value to check customer password is default or not.	int(1)	No

	last_modified	Customer last modified time	datetime	No
mic_notification	id {PK}	Unique identifier of notification, auto increment.	int(11)	No
	content	Notification content	varchar(2000)	No
	receiver	Notification receiver	varchar(250)	No
	method	Notification method	int(11)	No
	type	Notification type	int(11)	No
	extra_data	Notification extra data	varchar(2000)	Yes
	created_date	Notification created date	datetime	No
	resolved_date	Notification resolved date	datetime	Yes
mic_notification_read	id {PK}	Unique identifier of notification read, auto increment.	int(11)	No
	user_code	Code of user read this notification	varchar(10)	No
	notification_id	Notification ID	int(11)	No
	is_read	Value to check notification is read or not yet.	int(11)	No
mic_payment	id {PK}	Unique identifier of payment, auto increment.	int(11)	No
	paid_date	The date the payment is paid	datetime	No
	payment_method	Payment method	varchar(20)	No
	content	Payment content	varchar(250)	No
	amount	Payment amount	float	No
	paypal_trans_id	PayPal transaction ID	varchar(50)	Yes
	start_date	Contract start date while payment content is renew	datetime	Yes
	expired_date	Contract expired date while payment content is renew	datetime	Yes
	contract_code	Code of contract has this payment	varchar(10)	No
	receiver	Payment receiver	varchar(10)	Yes
mic_staff	staff_code {PK}	Unique identifier of staff	varchar(10)	No
	password	Staff password	varchar(32)	No
	name	Staff name	varchar(80)	No
	email	Staff email	varchar(250)	No
	phone	Staff phone	varchar(15)	No

Table 121 Data attribute dictionary

3. Performance Measures

3.1. Web application page load speed

3.1.1. Definition

This section tests the general page load speed from all the page of the web application.

3.1.2. Test environment

Server:

- **Operating system:** Ubuntu 14.04 Server 32 bit
- **RAM:** 512 MB
- **Storage:** 40 GB
- **Processor:** 2.6 GHz Intel Core i5
- **Network:** 1 Mbps

3.1.3. Test cases

Using browser from client side and go to each page of the web application. Record the page load time using Chrome Dev Tools at client side.

Test pages:

Public	home
	register
	checkout
	login
Staff	dashboard
	contract: <ul style="list-style-type: none"> - list contract - detail contract - create contract
	compensation: <ul style="list-style-type: none"> - list all compensation - list compensation by contract - detail compensation - create compensation - edit compensation
	punishment <ul style="list-style-type: none"> - list punishment by contract - create punishment - edit punishment
	accident <ul style="list-style-type: none"> - list accident by contract - create accident - edit accident
	card <ul style="list-style-type: none"> - list card - detail card

	<ul style="list-style-type: none"> - list new card request
	customer <ul style="list-style-type: none"> - list customer - customer detail - create customer - edit customer information
Customer	dashboard
	contract <ul style="list-style-type: none"> - list contract - detail contract - create new contract
	compensation: <ul style="list-style-type: none"> - list compensation by contract - detail compensation - create compensation
	punishment <ul style="list-style-type: none"> - list punishment by contract - create punishment
	accident <ul style="list-style-type: none"> - list accident by contract - create accident
	card <ul style="list-style-type: none"> - list card - detail card - list new card request
	profile information

3.1.4. Test result

The test is run 10 times, each time we go through all the pages listed above and log down the average page load time.

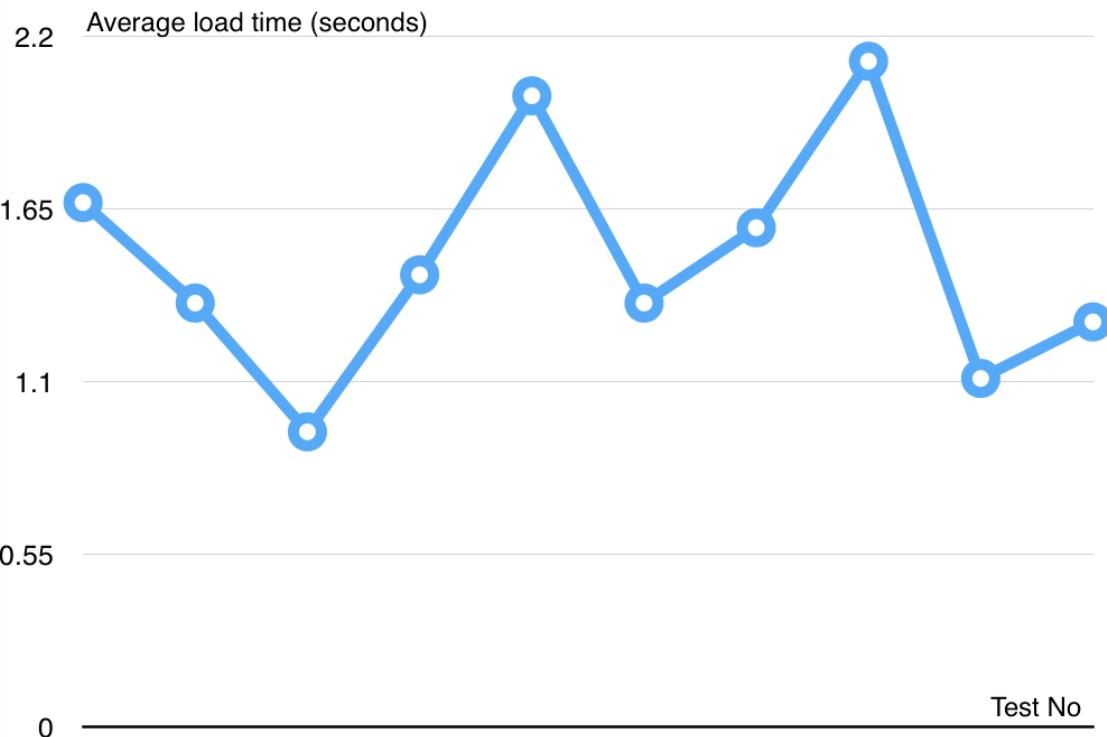


Figure 97 Web application page load speed test result

Test No.	Average page load time (second)	Execute date
1	1.67	6 July 2015
2	1.35	6 July 2015
3	0.94	6 July 2015
4	1.44	6 July 2015
5	2.01	6 July 2015
6	1.35	6 July 2015
7	1.59	6 July 2015
8	2.12	6 July 2015
9	1.11	6 July 2015
10	1.29	6 July 2015
	Average: 1.49 (seconds)	

Table 122 Web application page load speed test result

3.2. Mobile application API load speed

3.2.1. Definition

This section tests the load speed of mobile apps when connection to server through API.

3.2.2. Test environment

Server:

- **Operating system:** Ubuntu 14.04 Server 32 bit
- **RAM:** 512 MB
- **Storage:** 40 GB
- **Processor:** 2.6 GHz Intel Core i5
- **Network:** 1 Mbps

Client:

Hardware Requirements	Minimum	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 123 Mobile app API load speed client specification

3.2.3. Test cases

Using mobile application to send the entire API request to web server.

List of APIs:

No.	API	Description
1	getCheckConnection	Check server status
2	getContracts	Get contract information
3	getUpdateCardID	Update Card ID for contract
4	getCheckCard	Check card validation
5	getUpdatePunishment	Update punishment information for contract

Table 124 Mobile app API load speed test cases

3.2.4. Test result

The test is run 10 times, each time we run all the test cases listed above and log down the average API response time.

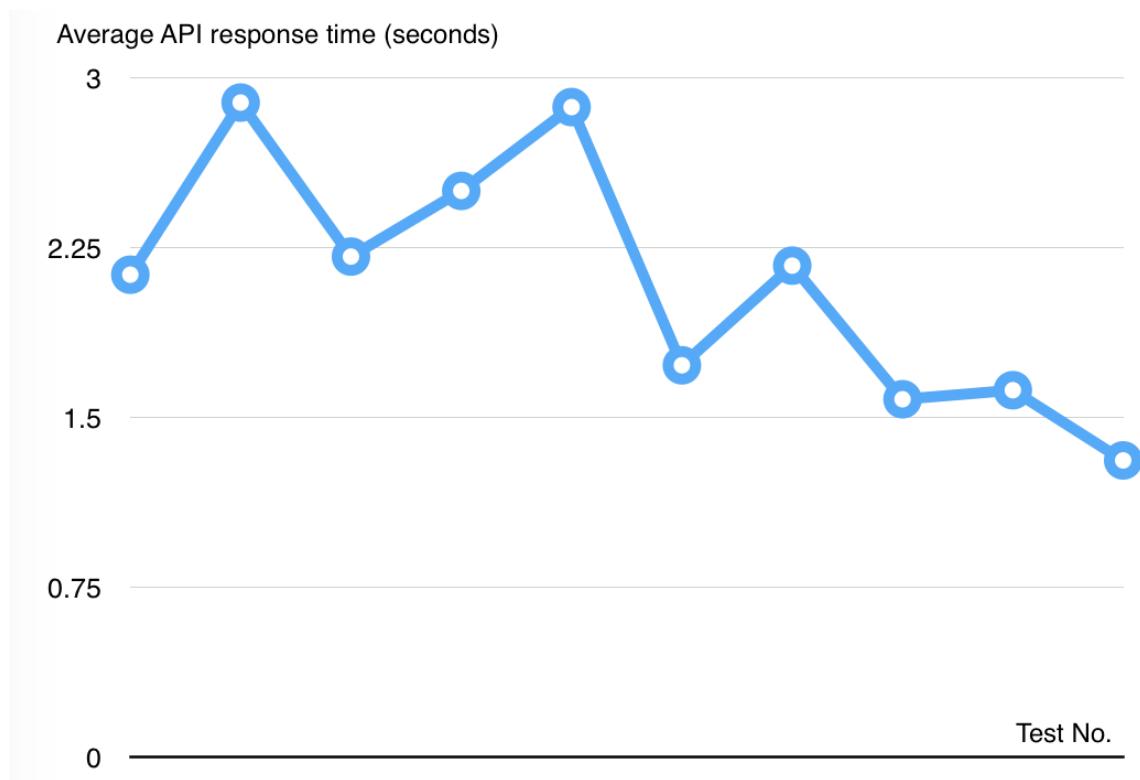


Figure 98 Mobile application API load speed test result

Test No.	Average page load time (second)	Execute date
1	2.13	6 July 2015
2	2.89	6 July 2015
3	2.21	6 July 2015
4	2.5	6 July 2015
5	2.87	6 July 2015
6	1.73	6 July 2015
7	2.17	6 July 2015
8	1.58	6 July 2015
9	1.62	6 July 2015
10	1.31	6 July 2015
Average: 2.10 (seconds)		

Table 125 Mobile application API load speed test result

4. Test Plan

The overall purpose of testing is to ensure MIC system meets its entire technical, functional and business requirement. The purpose of this document is to describe the overall test plan and strategy for testing the MIC system. The approach described in this document provides the framework for all testing related to this application. Individual test cases will be written for each version of the application that is released. This document will also be updated as required for each release.

4.1. Features to be tested

- Staff: view contract information, view compensation information, resolve new card request, resolve compensation request, create new contract, renew contract, cancel contract.
- Customer: view contract information, view compensation information, renew contract, cancel contract, request compensation, new card request, change password.
- Guest: Create new contract request
- System: notify schedule
- Mobile application: verify card, add punishment, print card

4.2. Features not to be tested

- Admin: add staff, remove staff
- Login, logout, change profile
- View information: card, punishment, accident, customer, staff, payment.

5. System Testing Test Case

5.1. Communication Diagrams

Test cases are created from bellow communication diagram:

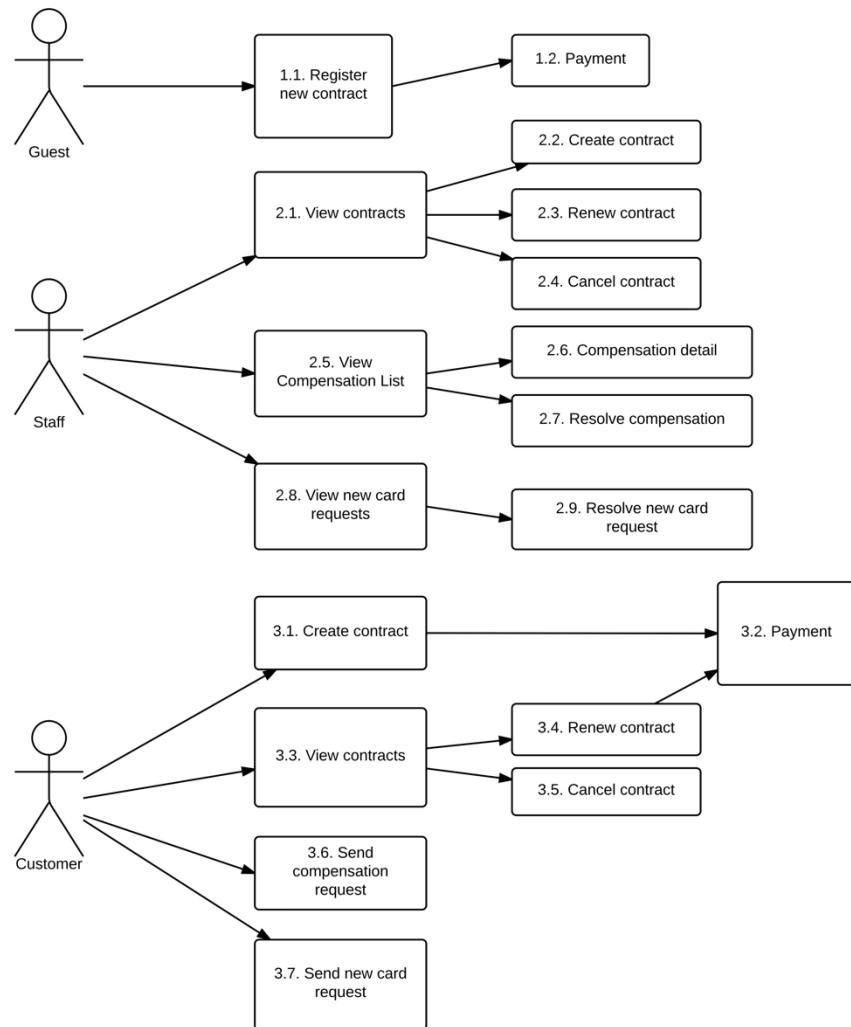


Figure 99 Web Application Communication Diagram

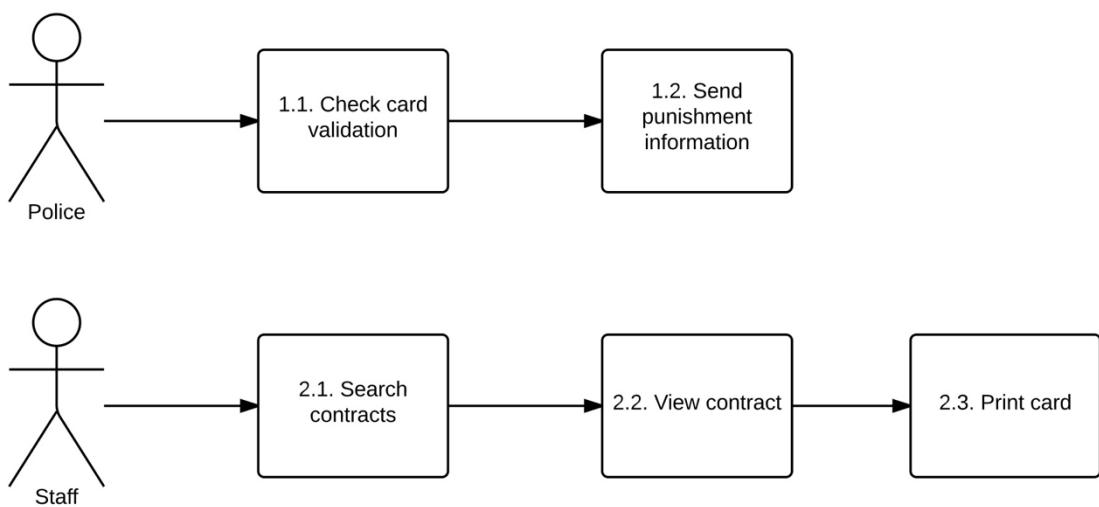


Figure 100 Mobile Applications Communication Diagram

5.2. Test cases

5.2.1. <Guest> Register new contract

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date

RNC1	Test guest register new contract with valid information	<p>1. Go to the home page</p> <p>2. Enter personal information:</p> <ul style="list-style-type: none"> - Họ tên: Nguyễn Văn A - Email: baohiem@yahoo.com - Số điện thoại: 0909099099 - Số CMND/Hộ chiếu: 272185738 - Địa chỉ: 123 Chợ Lớn, Bến Thành, HCM. - Ngày bắt đầu: 13/7/2015 - Hình thức bảo hiểm: Xe trên 50cc có BH cho người trên xe <p>3. Click "Tiếp theo" button.</p> <p>4. Click tab "2.Thông tin xe".</p> <p>5. Enter vehicle information:</p> <ul style="list-style-type: none"> - Biển số đăng ký: 60-Y6 6666 - Nhãn hiệu: Honda - Số khung: 1001001 - Số máy: 1001001 - Dung tích: 110cc - Số loại: Air blade - Loại xe: Hai bánh - Màu sơn: Đỏ - Năm sản xuất: 2000 - Tự trọng: 100 <p>6. Click "TIẾP THEO" button.</p> <p>7. Click "TẠO HỢP ĐỒNG VÀ THANH TOÁN" button.</p>	Create contract successful, send an email contains customer code and password to customer. Request payment		Pass	7/13/15
------	---------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------	--	------	---------

RNC2	Test guest register contract with an exist plate number	<p>1. Register new contract with information like test case "RNC1". And excute payment process like test case "RNCP2".</p> <p>2. Register another contract with information:</p> <ul style="list-style-type: none"> - Họ tên: Nguyễn Văn B - Email: baohiem1@yahoo.com - Số điện thoại: 0909888888 - Số CMND/Hộ chiếu: 123485738 - Địa chỉ: 123 Chợ Nhỏ, Bến Thành, HCM. - Ngày bắt đầu: 13/7/2015 - Hình thức bảo hiểm: Xe trên 50cc có BH cho người trên xe - Biển số đăng ký: 60-Y6 6666 - Nhãn hiệu: Honda - Số khung: 1231231 - Số máy: 1231231 - Dung tích: 110cc - Số loại: Air blade - Loại xe: Hai bánh - Màu sơn: Đỏ - Năm sản xuất: 2000 - Tự trọng: 100 - Số người: 2 	Show error message: "Đang có hợp đồng hiệu lực với xe có biển số này"	RNC1,RNCP2.	Pass	7/13/15
------	---------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------	-------------	------	---------

RNC3	Test guest register contract with an exist email	1. Register new contract with information like test case "RNC1". 2. Go to the home page 3. Enter personal information: - Họ tên: Nguyễn Văn B - Email: baokiem@yahoo.com - Số điện thoại: 0909123123 - Số CMND/Hộ chiếu: 123456789 - Địa chỉ: 123 Chợ Bà Chiểu, Bình Thạnh, HCM. - Ngày bắt đầu: 13/7/2015 - Hình thức bảo hiểm: Xe trên 50cc có BH cho người trên xe 4. Click "Tiếp theo" button.	Show error message: "Email đã được đăng ký bởi người dùng khác"	RNC1	Pass	7/13/15
------	--------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------	------	------	---------

Table 126 Test case - <Guest> Register new contract

5.2.2. <Guest> Register new contract payment

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
RNCP1	Test guest choose direct payment for register new contract	1. Guest registered contract succeed like test case "RNC1". 2. Guest choose direct payment.	Show company address map	RNC1	Pass	7/13/15
RNCP2	Test Paypal payment successful for	1. Guest registered contract succeed like test case "RNC1". 2. Guest choose PayPal payment.	Show success message: "Thanh toán thành công"	RNC1	Pass	7/13/15

	register new contract.	3. Guest log in and pay with PayPal web page.				
RNCP3	Test guest cancel transaction while execute Paypal payment for register new contract.	1. Guest registered contract succeed like test case "RNC1". 2. Guest choose PayPal payment. 3. Guest cancel transaction	Show message to notify that guest has canceled the transaction: "Bạn đã hủy thanh toán. Xin vui lòng Đăng nhập để thanh toán lại hoặc đến thanh toán trực tiếp"	RNC1	Pass	7/13/15

Table 127 Test case - <Guest> Register new contract payment

5.2.3. <Staff> View Contracts

ID	Test case description	Test case procedure	Expected output	Inter-test case dependence	Result	Test date
WC1	View contract detail	1. Create a contract with code HD0001 2. Click contract code in the grid to view its detail	Display contract detailed information page	CC1 - Create Contract	Pass	7/13/15
WC2	Search contract with customer name	1. Create 3 customer with names "Dang Van Ngu", "Duong Tu Ky" and "Nguyen Dinh Bao" alternatively 2. Create 2 contract HD0001 and HD0002 belongs to "Dang Van Ngu" and "Nguyen Dinh Bao" in turn 3. Input "Nguyen Dinh Bao" to search box 4. Click "Tìm kiếm" button	Display contract HD0002 in the grid view	Create Customer CC1 - Create Contract	Pass	7/13/15
WC3	Search contract with no keyword	1. Create 3 customer with names "Dang Van Ngu", "Duong Tu Ky" and "Nguyen Dinh Bao" alternatively	A message notify there are no value for this search	Create Customer CC1 - Create Contract	Pass	7/13/15

		2. Create 2 contract HD0001 and HD0002 belongs to "Dang Van Ngu" and "Nguyen Dinh Bao" in turn 3. Click "Tìm kiếm" button				
--	--	------------------------------------------------------------------------------------------------------------------------------	--	--	--	--

Table 128 Test case - <Staff> View contract

5.2.4. <Staff> Create Contract

ID	Test case description	Test case procedure	Expected output	Inter-test case dependence	Result	Test date
CC1	Create contract success	1. Create customer "Nguyen Van A" has customer code is KH0001 2. Click create new contract, input form is displayed 3. Input all default value according to "Default_value_for_forms.xlsx" file 4. Set customer code is KH0001, contract start date to 12 July 2015 5. Click "Tạo hợp đồng" button	A contract with status is "No card" created successfully	Create Customer	Pass	7/13/15
CC2	Create contract success but not start yet	1. Create customer "Nguyen Van A" has customer code is KH0001 2. Click create new contract, input form is displayed 3. Input all default value according to "Default_value_for_forms.xlsx" file 4. Set customer code is KH0001, contract start date to 21 July 2015 5. Click "Tạo hợp đồng" button	A contract with status is "Pending" created successfully	Create Customer	Pass	7/13/15

CC3	Create contract fail because of existed vehicle plate number	1. Create customer "Nguyen Van A" has customer code is KH0001, customer "Duong Thien Hoa" has customer code is KH0002 2. Create new contract with start date is 12 July 2015, contract status is Ready, vehicle plate number is 59Y9-38482 3. Click create new contract, input form is displayed 4. Input all default value according to "Default_value_for_forms.xlsx" file 5. Set vehicle plate number is 59Y9-38482 6. Click "Tạo hợp đồng" button	A message notify this vehicle plate number is existed in the system	Create Customer CC1 - Create Contract	Pass	7/13/15
CC4	Create contract fail because of there are no customer	1. Create customer "Nguyen Van A" has customer code is KH0001 2. Click create new contract, input form is displayed 3. Input all default value according to "Default_value_for_forms.xlsx" file, customer code is KH0002 4. Set contract start date to 12 July 2015 5. Click "Tạo hợp đồng" button	A message notify this customer is not exist in the system	Create Customer	Pass	7/13/15

Table 129 Test case - <Staff> Create contract

5.2.5. <Staff> Renew Contract

ID	Test case description	Test case procedure	Expected output	Inter-test case dependence	Result	Test date
----	-----------------------	---------------------	-----------------	----------------------------	--------	-----------

RC1	Renew contract success	<ol style="list-style-type: none"> 1. Create new contract has start date is 10 July 2015, expired date is 1 September 2015. Contract status is "No card" 2. Click view this contract detail 3. Click "Gia hạn" button, renew contract form displayed 4. Fill out the form: <ul style="list-style-type: none"> - Gia hạn đến: 01/09/2016 - Ngày nộp phí: 13/07/2015 5. Click "Gia hạn hợp đồng" button 	<p>Renew contract successfully, contract status is "No card" A new payment is created with start date is 01/09/2015, expired date is 01/09/2016</p>	CC1 - Create Contract WC1 - View Contracts	Pass	7/13/15
RC2	Renew contract success with request for new card and delivery it	<ol style="list-style-type: none"> 1. Create new contract has start date is 9 July 2015, expired date is 13 August 2015. Contract status is "No card" 2. Print a card with code "12-34-56-78-91" for this contract. Contract new status is "Ready" 3. Click view this contract detail 4. Click "Gia hạn" button, renew contract form displayed button 5. Fill out the form: <ul style="list-style-type: none"> - Gia hạn đến: 13/08/2016 - Ngày nộp phí: 13/07/2015 6. Check "Cấp thẻ mới", "Vận chuyển thẻ" boxes 7. Click "Gia hạn hợp đồng" button 	<p>Renew contract successfully, its status is "No card". A new payment is created with start date is 13/08/2015, expired date is 13/08/2016 The card with code "12-34-56-78-91" is deactivated</p>	CC1 - Create Contract Print Card WC1 - View Contracts	Pass	7/13/15

RC3	Renew contract success	<ol style="list-style-type: none"> 1. Create new contract has start date is 6 July 2015, expired date is 12 July 2015. Contract status is "Expired" 2. Click view this contract detail 3. Click "Gia hạn" button, renew contract form displayed 4. Fill out the form: <ul style="list-style-type: none"> - Gia hạn đến: 13/08/2016 - Ngày nộp phí: 13/07/2015 5. Click submit button 	Renew contract successfully, contract status is "No card" A new payment is created with start date is 13/07/2015, expired date is 13/07/2016	CC1 - Create Contract WC1 - View Contracts	Pass	7/13/15
RC4	Renew contract fail because of contract term is out of bound	<ol style="list-style-type: none"> 1. Create new contract has start date is 6 July 2015, expired date is 12 August 2015. Contract status is "No card" 2. Click view this contract detail 3. Click "Gia hạn" button, renew contract form displayed 4. Fill out the form: <ul style="list-style-type: none"> - Gia hạn đến: 13/12/2016 - Ngày nộp phí: 13/07/2015 5. Click "Gia hạn hợp đồng" button 	A message notify contract term can not exceed the limit	CC1 - Create Contract WC1 - View Contracts	Pass	7/13/15
RC5	Renew contract success with request for new card fail because this contract have no card	<ol style="list-style-type: none"> 1. Create new contract has start date is 6 July 2015, expired date is 12 July 2015. Contract status is "Expired" 2. Click view this contract detail 3. Click "Gia hạn" button, renew contract form displayed 4. Fill out the form: <ul style="list-style-type: none"> - Gia hạn đến: 13/08/2016 - Ngày nộp phí: 13/07/2015 5. Check "Cấp thẻ mới" box 6. Click "Gia hạn hợp đồng" button 	A message notify this contract have no card	CC1 - Create Contract	Pass	7/13/15

Table 130 Test case - <Staff> Renew contract

5.2.6. <Staff> Cancel Contract

ID	Test case description	Test case procedure	Expected output	Inter-test case dependence	Result	Test date
CC1	Cancel contract having "Ready" status	1. Create new contract has start date is 11 July 2015, expired date is 11 July 2016. Contract status is "No card" 2. Print a card with code "12-34-56-78-AZ" for this contract. Contract new status is "Ready" 3. Click view this contract detail 4. Click "Hủy hợp đồng" button, cancel contract form displayed 5. Fill out the form: - Ngày hủy hợp đồng: 13/07/2015 - Lý do hủy hợp đồng: Mất xe 6. Click "Đồng ý hủy" button	Cancel contract successfully, its status is "Cancelled". The card with code "12-34-56-78-AZ" is deactivated	CC1 - Create Contract PC1 - Print Card WC1 - View Contracts	Pass	7/13/15
CC2	Cancel contract having "Request cancel" status	1. Create new contract has start date is 11 July 2015, expired date is 11 July 2016. Contract status is "Request cancel" 2. Click view this contract detail 3. Click handle cancel contract request button, cancel contract form displayed 4. Quyết định của công ty: Hủy hợp đồng này 5. Click "Xác nhận" button	Cancel contract successfully, its status is "Cancelled".	CC1 - Create Contract WC1 - View Contracts	Pass	7/13/15

CC3	Cancel contract fail because of cancel reason is empty	<ol style="list-style-type: none"> 1. Create new contract has start date is 11 July 2015, expired date is 11 July 2016. Contract status is "No card" 2. Click view this contract detail 3. Click "Hủy hợp đồng" button, cancel contract form displayed 4. Click "Đồng ý hủy" button 	A message notify cancel reason is can not empty, request Staff to fill it out	CC1 - Create Contract WC1 - View Contracts	Pass	7/13/15
------------	--------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------	-----------------------------------------------	------	---------

Table 131 Test case - <Staff> Cancel contract

5.2.7. <Staff> View Compensation List

ID	Test case description	Test case procedure	Expected output	Inter-test case dependence	Result	Test date
WCL1	View compensation list	<ol style="list-style-type: none"> 1. Create new contract with code is "HD0001" 2. Create new compensation for "HD0001" contract 3. Click view list of compensation 	A grid view display list of compensation	CC1 - Create Contract Create Compensation	Pass	7/11/15
WCL2	Search compensation with contract code	<ol style="list-style-type: none"> 1. Create 2 new contract with code is "HD0001", "HD0002" 2. Create 4 new compensation, 2 for each contract 3. Input "HD0001" to search box 4. Click "Tìm kiếm" button 	A grid view display 2 compensation belongs to the contract has code "HD0001"	CC1 - Create Contract Create Compensation	Pass	7/12/15
WCL3	Search contract with no keyword	<ol style="list-style-type: none"> 1. Create 2 new contract with code is "HD0001", "HD0002" 2. Create 4 new compensation, 2 for each contract 3. Click "Tìm kiếm" button 	A message notify there are no value for this search	CC1 - Create Contract Create Compensation	Pass	7/13/15

Table 132 Test case - <Staff> View compensation list

5.2.8. <Staff> Compensation Detail

ID	Test case description	Test case procedure	Expected output	Inter-test case dependence	Result	Test date
CD1	View an unsolved compensation detail	1. Create new contract 2. Create new compensation for this contract 3. Click compensation code in list of compensation	Display compensation detail page with an alert about resolve this compensation	CC1 - Create Contract Create Compensation	Pass	7/13/15
CD2	View an solved compensation detail	1. Create new contract 2. Create new compensation for this contract 3. Resolve this compensation 4. Click compensation code to view its detail	Display compensation detail page with solved compensation information	CC1 - Create Contract Create Compensation RC1 - Resolve Compensation	Pass	7/13/15

Table 133 Test case - <Staff> View compensation detail

5.2.9. <Staff> Resolve Compensation

ID	Test case description	Test case procedure	Expected output	Inter-test case dependence	Result	Test date
RC1	Resolve compensation success	1. Create new contract 2. Create new compensation for this contract 3. Click compensation code in list of compensation 4. Click "Giải quyết yêu cầu" button, solve form is displayed 5. Fill out the form:	Display compensation detail page with solved compensation information	CC1- Create Contract Create Compensation	Pass	7/13/15

	<ul style="list-style-type: none"> - Ngày giải quyết yêu cầu: 13/07/2015 - Quyết định của công ty: "Từ chối bồi thường" 6. Click "Quyết định" button 			
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--

Table 134 Test case - <Staff> Resolve compensation

5.2.10. <Staff> View new card requests

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
VNCR1	Test list new card request in ascending order of request date.	1. Create 5 new card request with 5 different request date. 2. Click "Yêu cầu thẻ mới" bar.	Display the list new card requests in ascending order of request date.	SNCR1	Pass	7/13/15
VNCR2	Test unresolved request will be on top.	1. Create 5 new card request with the same request date. But 2 of them is resolved. 2. Click "Yêu cầu thẻ mới" bar.	Display 3 unresolved new card requests on top and 2 resolved new card request on bottom.	SNCR1	Pass	7/13/15

Table 135 Test case - <Staff> View new card requests

5.2.11. <Staff> Resolve new card request

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
RNCR1	Test resolve new card request, print a new card	1. Contract HD0001 is Ready and having an activated card. 2. Create new card request for contract HD001 like test case SNCR1. 3. Print card for Ready contract "HD0001" like test case PC2.	Printer app show message "Write success", contract card change to new card ID, old card is deactivated	SNCR1,PC2.	Pass	7/13/15

Table 136 Test case - <Staff> Resolve new card request

5.2.12. <Customer> Create New Contract

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
CNC1	Create new contract success	1. Click "Hợp đồng mới" button. 2. Input all default value according to Default form value.xlsx file" (Create new contract form) 3. Click "Thêm hợp đồng" 4. Click "Xác nhận" 5. Click "Thanh toán ngay bây giờ" 6. Login to paypla an pay fee	Customer can see their new contract with status "No card" and some information about compensation, punishment, accident and card	Default form value.xlsx file	Pass	7/13/15
CNC2	Create new contract fail	1. Click "Hợp đồng mới" button. 2. Click "Thêm hợp đồng"	System show message error and required input value	Default form value.xlsx file	Pass	7/13/15

CNC3	Create new contract success	1. Click "Hợp đồng mới" button. 2. Input all default value according to Default form value.xlsx file" (Create new contract form) 3. Click "Thêm hợp đồng" 4. Click "Xác nhận" 5. Click "Thanh toán ngay bây giờ" 6. Click "Trực tiếp"	System shows map with detail address of company where customer can pay direct for their contract. Customer can see their new contract with status "Pending" and some information about compensation, punishment, accident and card	Default form value.xlsx file	Pass	7/13/15
CNC4	Create new contract fail	1. Click "Hợp đồng mới" button. 2. Input all default value according to Default form value.xlsx file" (Create new contract form) 3. Thời điểm có hiệu lực input "07/04/2015" 3. Click "Thêm hợp đồng"	System show message error "Thời điểm có hiệu lực phải sau ngày hiện tại"	Default form value.xlsx file	Pass	7/13/15

Table 137 Test case - <Customer> Create contract

5.2.13. <Customer> Renew Contract

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
RC1	Renew contract success	1. Create new contract with contract code "HD0001" 2. Prepare a contract with startdate: 7/25/2014 and expired date: 7/25/2015 and status is "Ready" 3. Click "Gia hạn" button	System show page with message "Gia hạn thành công" and expired is updated "7/26/2016"	Create new contract	Pass	7/13/15

		4. Click "Gia hạn hợp đồng" 5. Choose "Paypal" and login paypal to pay fee 6. Click "Trở về trang chi tiết hợp đồng"				
RC2	Renew contract success	1. Create new contract with contract code "HD0001" 2. Prepare a contract with startdate: 7/11/2014 and expireddate: 7/11/2015 and status is "Expired" 3. Click "Gia hạn" button 4. Click "Gia hạn hợp đồng" 5. Choose "Paypal" and login paypal to pay fee 6. Click "Trở về trang chi tiết hợp đồng"	System show page with message "Gia hạn thành công" and expired is updated "7/13/2016" and status change to "Ready"	Create new contract	Pass	7/13/15
RC3	Renew contract fail	1. Login with role customer 1. Create new contract with contract code "HD0001" with 2. Prepare a contract with status Expired 3. Open new browser and login with role Staff and goes into contract-detail at contract code "HD0001" 4. Click "Gia hạn hợp đồng" at browser staff 5. Do action like RC2	System will show error message "Thông tin hợp đồng đã bị sửa đổi trước đó bởi một người khác, vui lòng thực hiện lại thao tác"	Create new contract	Pass	7/13/15

		6 Click "Gia hạn hợp đồng" at browser customer				
--	--	------------------------------------------------	--	--	--	--

Table 138 Test case - <Customer> Renew contract

5.2.14. <Customer> Cancel Contract

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
CCC1	Cancel contract success	1. Create new contract with contract code "HD0001" 2. Prepare a contract with status "Ready" check reason cancel 3. Click "Hủy hợp đồng" button 4. Click "Xe cơ giới bị mất được cơ quan công an xác nhận" 5. Click "Xác nhận" button	System will reload page and change status into "Request cancel" and show detail of request cancel, show "Hủy yêu cầu button"	Create new contract	Pass	7/13/15
CCC2	Cancel contract fail	1. Login with role customer 2. Create new contract with contract code "HD0001" with 3. Prepare a contract with status Ready 4. Open new browser and login with role Staff and goes into contract-detail at contract code "HD0001" 5. Click "Hủy hợp đồng" at browser staff 6. Do action like CC1	System will show error message "Thông tin hợp đồng đã bị sửa đổi trước đó bởi một người khác, vui lòng thực hiện lại thao tác"	Create new contract	Pass	7/13/15

		6. Click "Hủy hợp đồng" at browser customer				
CCC3	Cancel contract contract fail	1. Create new contract with contract code "HD0001" 2. Prepare a contract with status "Ready" 3. Click "Hủy hợp đồng" button 4. Click "Xác nhận" button	System will show error message "Vui lòng chọn lí do hủy hợp đồng trước khi xác nhận. Cảm ơn!"	Create new contract	Pass	7/13/15
CCC4	Cancel contract contract fail	1. Create new contract with contract code "HD0001" 2. Prepare a contract with status "Ready" 3. Click "Lý do khác" radio check 4. Click "Xác nhận" button	System will show error message "Vui lòng chọn lí do hủy hợp đồng trước khi xác nhận. Cảm ơn!"	Create new contract	Pass	7/13/15
CCC5	Cancel contract contract fail	1. Create new contract with contract code "HD0001" 2. Prepare a contract with status "Ready" 3. Click "Lý do khác" radio check 4. Input blank value into textarea 4. Click "Xác nhận" button	System will show error message "Vui lòng chọn lí do hủy hợp đồng trước khi xác nhận. Cảm ơn!"	Create new contract	Pass	7/13/15

Table 139 Test case - <Customer> Cancel contract

5.2.15. <Customer> Send Compensation Request

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date

SCR1	Send compensation success	<ol style="list-style-type: none"> 1. Create contract with contract code "HD0001" 2. Click on "Hợp đồng" menu 3. Search Contract with code "HD0001" 4 Click link "HD0001" 5. Click tab "Lịch sử bồi thường" 6. Click "Yêu cầu bồi thương" 7. Input all default value according to Default form value.xlsx file" (Create new compensation) 8. Click "Gởi yêu cầu" 9. Click "Trở về trang chi tiết hợp đồng" 	System show compensation detail about request compensation they just send	Create new contract Default form value.xlsx file	Pass	7/13/15
SCR2	Send compensation fail	<ol style="list-style-type: none"> 1. Create contract with contract code "HD0001" 2. Click on "Hợp đồng" menu 3. Search Contract with code "HD0001" 4 Click link "HD0001" 5. Click tab "Lịch sử bồi thường" 6. Click "Yêu cầu bồi thương" 7. Input all default value according to Default form value.xlsx file" (Create new compensation) 8. Ngày xảy ra tai nạn input "09/13/2015" 8. Click "Gởi yêu cầu" 	System show error message "Ngày xảy ra tai nạn phải trước ngày hiện tại"	Create new contract Default form value.xlsx file	Pass	7/13/15

SCR3	Send compensation fail	<ol style="list-style-type: none"> 1. Create contract with contract code "HD0001" 2. Click on "Hợp đồng" menu 3. Search Contract with code "HD0001" 4 Click link "HD0001" 5. Click tab "Lịch sử bồi thường" 6. Click "Yêu cầu bồi thương" 8. Click "Gởi yêu cầu" 	System show error message "Vui lòng nhập giá trị" and required input into some field have "*" symbol	Create new contract Default form value.xlsx file	Pass	7/13/15
------	------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------	-----------------------------------------------------	------	---------

Table 140 Test case - <Customer> Send compensation request

5.2.16. <Customer> Send new card request

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
SNCR1	Test send new card request with direct payment.	<ol style="list-style-type: none"> 1. Contract HD0001 having an activated card. No new card request before. 2. In "Yêu Cầu Thẻ Mới" page for contract HD0001, input fields: <ul style="list-style-type: none"> - Mật khẩu: correct password - Ghi chú: "Làm lại thẻ". - Choose: "Thanh toán trực tiếp(tại công ty)" - Uncheck "Hủy bỏ thẻ cũ" 3. Click "Xác nhận" button. 	Show success message: "Yêu cầu thẻ mới thành công. Vui lòng đến trực tiếp để thanh toán". Create new card request record.		Pass	7/13/15
SNCR2	Test send new card and deactivate old card request	<ol style="list-style-type: none"> 1. Contract HD0001 having an activated card. No new card request before. 2. In "Yêu Cầu Thẻ Mới" page for contract HD0001, input fields: <ul style="list-style-type: none"> - Mật khẩu: correct password 	Show success message: "Yêu cầu thẻ mới thành công. Vui lòng đến trực tiếp để thanh toán".		Pass	7/13/15

	with direct payment.	<ul style="list-style-type: none"> - Ghi chú: "Làm lại thẻ". - Choose: "Thanh toán trực tiếp(tại công ty)" - Check "Hủy bỏ thẻ cũ" 3. Click "Xác nhận" button. 	Create new card request record and the old card been deactivated.			
SNCR3	Test send new card request with PayPal payment.	<ol style="list-style-type: none"> 1. Contract HD0001 having an activated card. No new card request before. 2. In "Yêu Cầu Thẻ Mới" page for contract HD0001, input fields: <ul style="list-style-type: none"> - Mật khẩu: correct password - Ghi chú: "Làm lại thẻ". - Choose: "Thanh toán qua PayPal" - Uncheck "Hủy bỏ thẻ cũ" - Check "Nhận thẻ tại công ty" 3. Click "Xác nhận" button. 4. User log in and pay with PayPal web page. 	Show success message: "Thanh toán thành công". Create new card request record.		Pass	7/13/15
SNCR4	Test send new card and deactivate old card request with PayPal payment.	<ol style="list-style-type: none"> 1. Contract HD0001 having an activated card. No new card request before. 2. In "Yêu Cầu Thẻ Mới" page for contract HD0001, input fields: <ul style="list-style-type: none"> - Mật khẩu: correct password - Ghi chú: "Làm lại thẻ". - Choose: "Thanh toán qua PayPal" - Check "Hủy bỏ thẻ cũ" - Check "Nhận thẻ tại công ty" 3. Click "Xác nhận" button. 4. User log in and pay with PayPal web page. 	Show success message: "Thanh toán thành công". Create new card request record and the old card been deactivated.		Pass	7/13/15
SNCR5	Test send new card and delivery request with	<ol style="list-style-type: none"> 1. Contract HD0001 having an activated card. No new card request before. 2. In "Yêu Cầu Thẻ Mới" page for contract HD0001, input fields: 	Show success message: "Thanh toán thành công". Create new card		Pass	7/13/15

	PayPal payment	<ul style="list-style-type: none"> - Mật khẩu: correct password - Ghi chú: "Làm lại thẻ". - Choose: "Thanh toán qua PayPal" - Uncheck "Hủy bỏ thẻ cũ" - Uncheck "Nhận thẻ tại công ty" 3. Click "Xác nhận" button. 4. User log in and pay with PayPal web page. 	request record with delivery requested.			
SNCR6	Test send new card, deactivate old card and delivery request with PayPal payment	<ul style="list-style-type: none"> 1. Contract HD0001 having an activated card. No new card request before. 2. In "Yêu Cầu Thẻ Mới" page for contract HD0001, input fields: <ul style="list-style-type: none"> - Mật khẩu: correct password - Ghi chú: "Làm lại thẻ". - Choose: "Thanh toán qua PayPal" - Check "Hủy bỏ thẻ cũ" - Uncheck "Nhận thẻ tại công ty" 3. Click "Xác nhận" button. 4. User log in and pay with PayPal web page. 	Show success message: "Thanh toán thành công". Create new card request record with delivery requested and the old card been deactivated.		Pass	7/13/15
SNCR7	Test send new card request with PayPal payment but user cancel payment process	<ul style="list-style-type: none"> 1. Contract HD0001 having an activated card. No new card request before. 2. In "Yêu Cầu Thẻ Mới" page for contract HD0001, input fields: <ul style="list-style-type: none"> - Mật khẩu: correct password - Ghi chú: "Làm lại thẻ". - Choose: "Thanh toán qua PayPal" - Uncheck "Hủy bỏ thẻ cũ" - Check "Nhận thẻ tại công ty" 3. Click "Xác nhận" button. 4. User cancel PayPal payment process. 	Show error message: "Bạn đã hủy thanh toán. Xin vui lòng thực hiện lại hoặc đến thanh toán trực tiếp"		Pass	7/13/15

SNCR8	Test send new card request for a card already had another new card request before.	1. Contract HD0001 having an activated card with a new card request before. 2. Send new card request for contract HD0001 like test case SNCR1.	Show error message:"Bạn đã yêu cầu thẻ mới trước đó. Vui lòng chờ xử lý"	SNCR1	Pass	7/13/15
SNCR9	Test send new card request for a contract have no card.	1. Contract HD0001 not having a card. 2. Send new card request for contract HD0001 like test case SNCR1.	Show error message:"Hợp đồng chưa có thẻ bảo hiểm. Xin vui lòng đợi phát hành"	SNCR1	Pass	7/13/15

Table 141 Test case - <Customer> Send new card request

5.3. Test case results statistics

Test case	Runs	Pass rate (%)
RNC1	50	98
RNC2	50	100
RNC3	50	98
RNCP1	50	100
RNCP2	50	100
RNCP3	50	100
WC1	50	98
WC2	50	100
WC3	50	100
CC1	50	100
CC2	50	100
CC3	50	100
CC4	50	100
WCL1	50	98
WCL2	50	100
WCL3	50	100
CD1	50	100
CD2	50	98
RC1	50	100
VNCR1	50	100
VNCR2	50	100
RNCR1	50	100
CNC1	50	100
CNC2	50	98

CNC3	50	100
CNC4	50	100
RC1	50	100
RC2	50	100
RC3	50	100
CCC1	50	100
CCC2	50	100
CCC3	50	100
CCC4	50	98
CCC5	50	100
SCR1	50	100
SCR2	50	100
SCR3	50	98
SNCR1	50	100
SNCR2	50	100
SNCR3	50	100
SNCR4	50	100
SNCR5	50	100
SNCR6	50	100
SNCR7	50	98
SNCR8	50	100
SNCR9	50	100

Table 142 Test case results statistics

F. Software User's Manual

1. Installation Guide

1.1. Setting up environment at server side

Bellows are requirements for hardware and software environment to run MIC system in 10 years. The specifications are based on the dependencies requirements and performance test result from previous section of this document.

1.1.1. Hardware requirements

Hardware	Specification
Internet Connection	8 Mbps
Computer Processor	Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz
Computer Memory	3GB of RAM or more
Hard Disk Drive	40GB or more

Table 143 Hardware requirements

1.1.2. Software requirements

Software	Application name / version
Operating System	Ubuntu Server 14.04.2 LTS
Java	1.7.0_79
Web Server	Apache Tomcat 7.0.52
Database	MySQL 5.6

Table 144 Software requirements

1.2. Web Application Deployment Process

1.2.1. Check environment

Check Ubuntu version: 14.04.2 LTS

```
root@MIC:~# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 14.04.2 LTS
Release: 14.04
Codename: trusty
```

Check Java version: 1.7.0_79

```
root@MIC:~# java -version
java version "1.7.0_79"
OpenJDK Runtime Environment (IcedTea 2.5.6) (7u79-2.5.6-
0ubuntu1.14.04.1)
OpenJDK 64-Bit Server VM (build 24.79-b02, mixed mode)
```

Check MySQL version: 5.6

```
root@MIC:~# mysql --version
mysql Ver 14.14 Distrib 5.6.19, for debian-linux-gnu
(x86_64) using EditLine wrapper
```

Check Apache Tomcat version: 7.0.52

```
root@MIC:~# /usr/share/tomcat7/bin/version.sh
Using CATALINA_BASE:      /usr/share/tomcat7
Using CATALINA_HOME:      /usr/share/tomcat7
Using CATALINA_TMPDIR:   /usr/share/tomcat7/temp
Using JRE_HOME:           /usr/lib/jvm/java-7-openjdk-amd64
Using CLASSPATH:
/usr/share/tomcat7/bin/bootstrap.jar:/usr/share/tomcat7/bin/
tomcat-juli.jar
Server version: Apache Tomcat/7.0.52 (Ubuntu)
Server built:   Jun 19 2015 08:54:46
Server number:  7.0.52.0
OS Name:       Linux
OS Version:    3.13.0-57-generic
Architecture:  amd64
JVM Version:   1.7.0_79-b14
JVM Vendor:    Oracle Corporation
```

1.2.2. Import database

Using file **database.sql** located under **Deployment** directory from this document.

```
root@MIC:~# mysql -u root -p mic_data < database.sql
```

1.2.3. Build war artifact

Under **Source code/MICWeb** directory, edit file **pom.xml** and set correct MySQL username and password at **<profiles>** tag:

```
<profiles>
  <profile>
    <id>production</id>
    <properties>
      <db.username>mysql username here</db.username>
      <db.password>mysql password here</db.password>
    </properties>
  </profile>
</profiles>
```

Under **Source code/MICWeb** directory, run bellow command to build war artifact:

```
mvn clean install -pProduction
```

A **war** artifact file will be created under **Source code/MICWeb/target** directory, we will use this file for deployment.

1.2.4. Deploy war artifact

Using file **war** built from previous step.

```
root@MIC:~# mv MIC.war /var/lib/tomcat7/webapps/ROOT.war
```

Web application is now available at <http://server-ip-address>.

1.3. Mobile Application Deployment Process

At mobile device, go to Setting/Security, uncheck check box **Unknown sources**.



Using apk files from **Deployment** directory under this document. Click to install application.

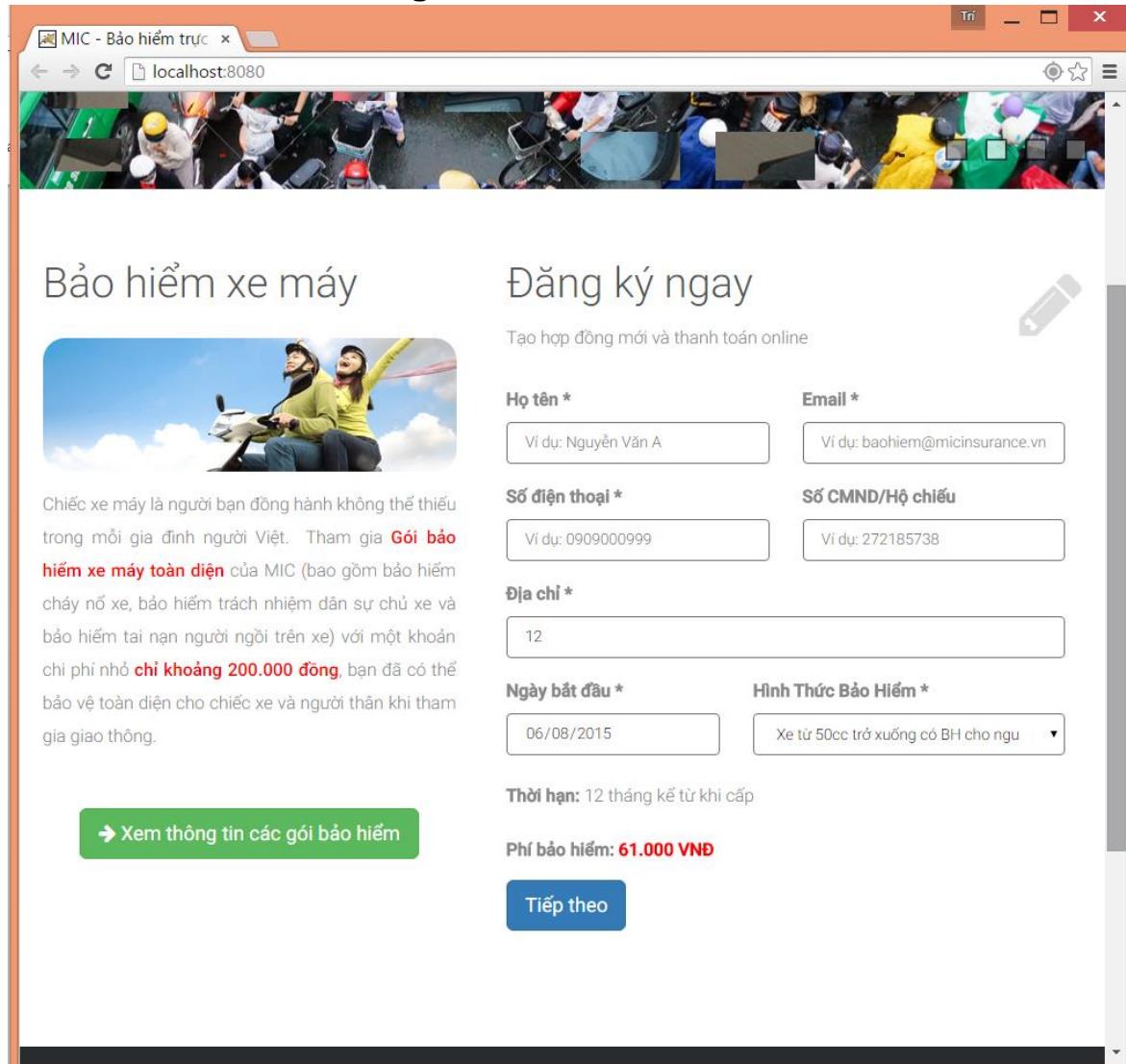


2. User Guide

2.1. Web Application

2.1.1. Guest

2.1.1.1. Guest register contract



The screenshot shows a web browser window titled "MIC - Bảo hiểm trực tuyến" with the URL "localhost:8080". The page features a banner image of people on motorcycles. The main content area has two sections: "Bảo hiểm xe máy" on the left and "Đăng ký ngay" on the right. The "Đăng ký ngay" section contains fields for "Họ tên *", "Email *", "Số điện thoại *", "Số CMND/Hộ chiếu", "Địa chỉ *", "Ngày bắt đầu *", "Hình Thức Bảo Hiểm *", and "Thời hạn: 12 tháng kể từ khi cấp". A green button labeled "Xem thông tin các gói bảo hiểm" is on the left, and a blue button labeled "Tiếp theo" is at the bottom right. Below the form, there is a note about motorcycle insurance being essential for Vietnamese families.

Bảo hiểm xe máy

Đăng ký ngay

Tạo hợp đồng mới và thanh toán online

Họ tên *
Ví dụ: Nguyễn Văn A

Email *
Ví dụ: baohiem@micinsurance.vn

Số điện thoại *
Ví dụ: 0909000999

Số CMND/Hộ chiếu
Ví dụ: 272185738

Địa chỉ *
12

Ngày bắt đầu *
06/08/2015

Hình Thức Bảo Hiểm *
Xe từ 50cc trở xuống có BH cho ngu

Thời hạn: 12 tháng kể từ khi cấp

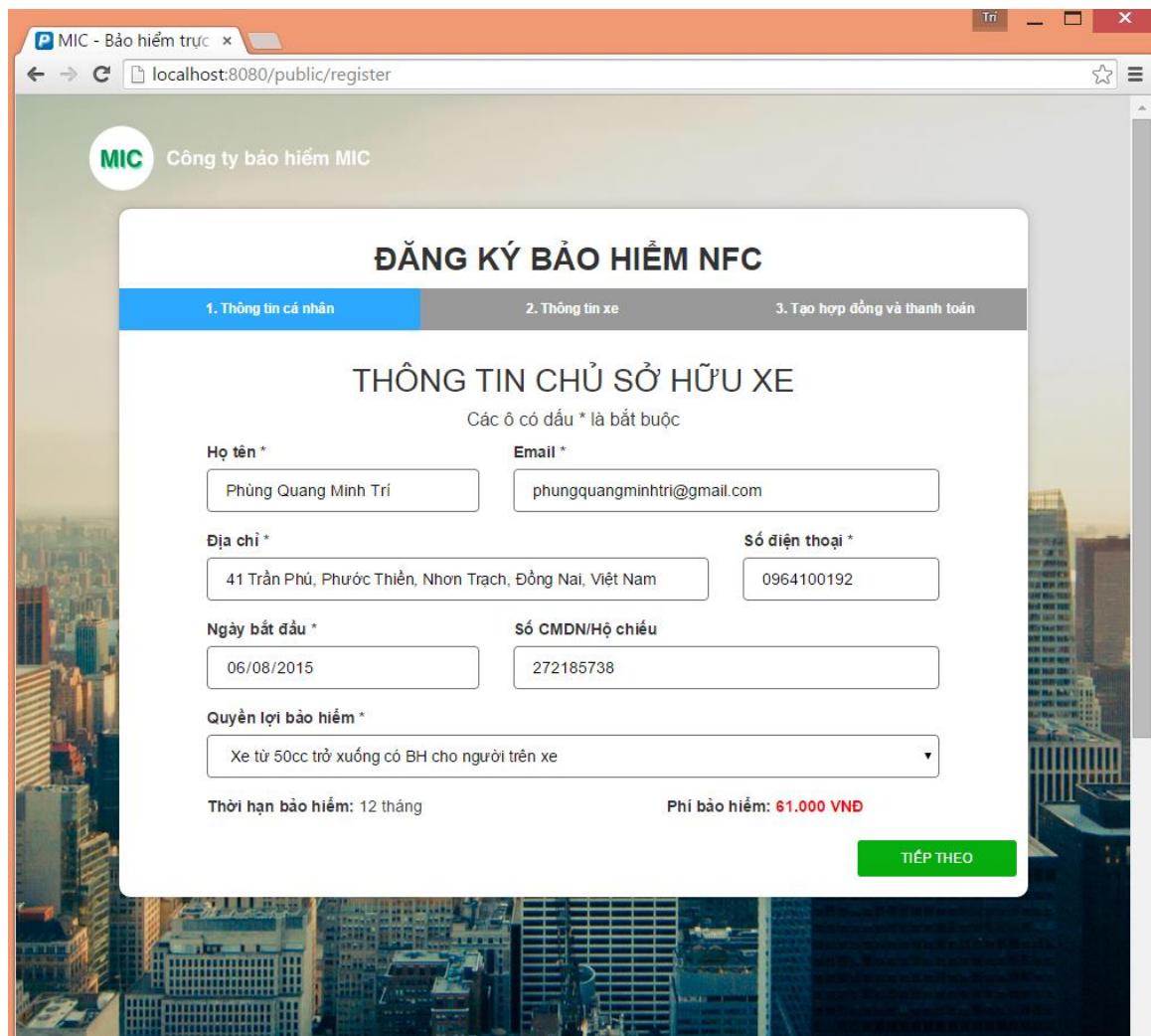
Phí bảo hiểm: 61.000 VNĐ

Xem thông tin các gói bảo hiểm

Tiếp theo

Chiếc xe máy là người bạn đồng hành không thể thiếu trong mỗi gia đình người Việt. Tham gia **Gói bảo hiểm xe máy toàn diện** của MIC (bao gồm bảo hiểm cháy nổ xe, bảo hiểm trách nhiệm dân sự chủ xe và bảo hiểm tai nạn người ngồi trên xe) với một khoản chi phí nhỏ **chỉ khoảng 200.000 đồng**, bạn đã có thể bảo vệ toàn diện cho chiếc xe và người thân khi tham gia giao thông.

Step	Description
1	Fill in fields: “Họ tên”: Customer's name “Email”: Customer's email “Số điện thoại”: Phone number “Số CMND/Hộ chiếu”: Personal ID/Passport “Địa chỉ”: Address “Ngày bắt đầu”: Contract's start date “Hình thức bảo hiểm”: Type of the contract
2	Click “Tiếp theo” button



The screenshot shows a web browser window for 'MIC - Bảo hiểm trực tuyến' at 'localhost:8080/public/register'. The main title is 'ĐĂNG KÝ BẢO HIỂM NFC'. The navigation bar has three tabs: '1. Thông tin cá nhân' (selected), '2. Thông tin xe', and '3. Tạo hợp đồng và thanh toán'. Below the tabs is a section titled 'THÔNG TIN CHỦ SỞ HỮU XE'. It contains several input fields: 'Họ tên *' (Phùng Quang Minh Trí), 'Email *' (phungquangminhtri@gmail.com), 'Địa chỉ *' (41 Trần Phú, Phước Thiền, Nhơn Trạch, Đồng Nai, Việt Nam), 'Số điện thoại *' (0964100192), 'Ngày bắt đầu *' (06/08/2015), 'Số CMDN/Hộ chiếu' (272185738), and 'Quyền lợi bảo hiểm *' (Xe từ 50cc trở xuống có BH cho người trên xe). Below these fields are two status messages: 'Thời hạn bảo hiểm: 12 tháng' and 'Phi bảo hiểm: 61.000 VND'. A green 'TIẾP THEO' button is at the bottom right.

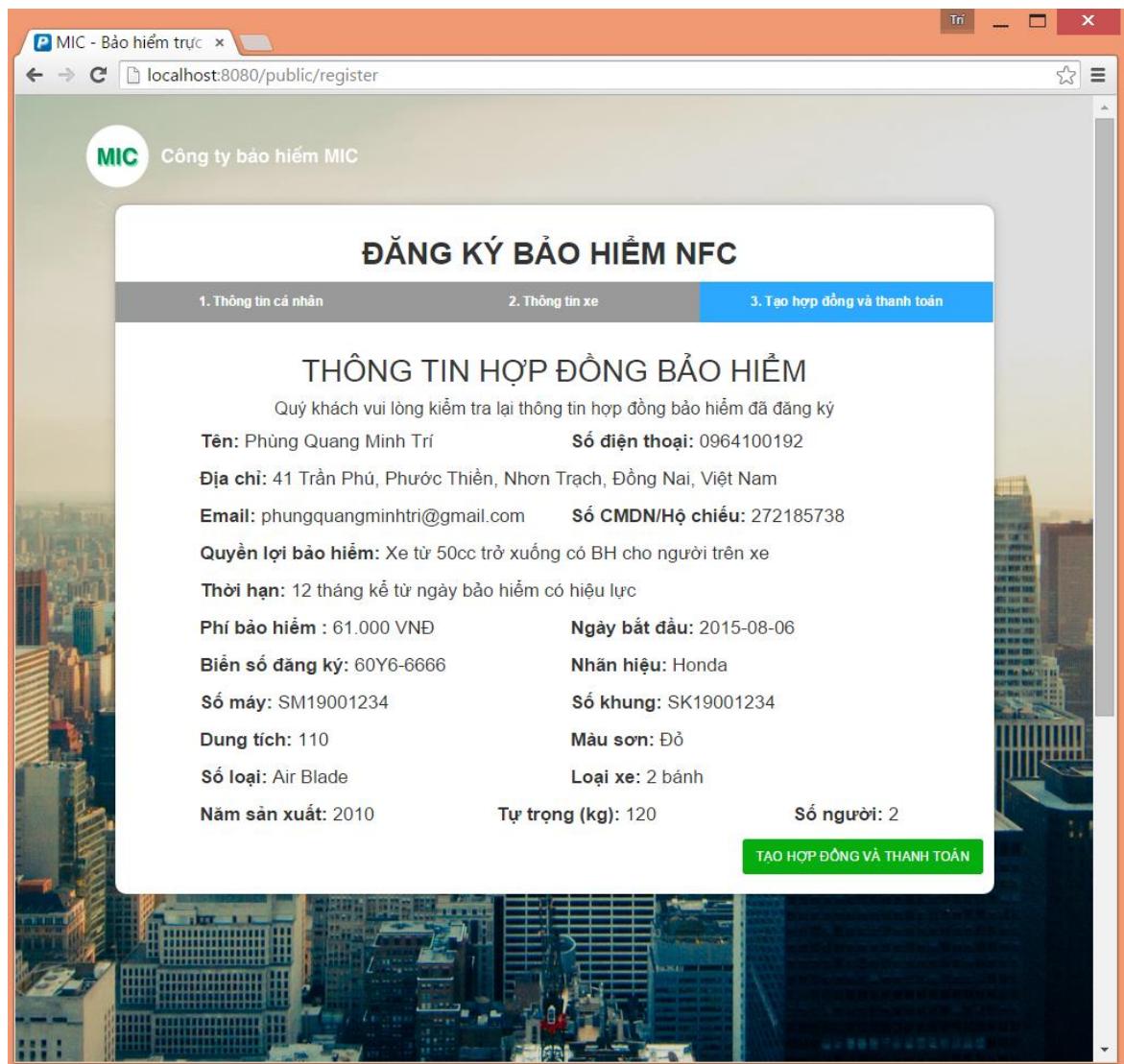
Step	Description
3	Check and edit the previous fields if needed.
4	Click “Tiếp theo” button

The screenshot shows a web browser window for 'MIC - Bảo hiểm trực tuyến' at 'localhost:8080/public/register'. The main title is 'ĐĂNG KÝ BẢO HIỂM NFC'. Below it, a sub-section title is 'THÔNG TIN XE'. A note says 'Quý khách vui lòng nhập thông tin dựa trên đăng ký xe' and 'Các ô có dấu * là bắt buộc'. The form contains several input fields grouped into three columns:

Biển số đăng ký *	Nhãn hiệu *	Dung tích *
Ví dụ: 54-Z6 6666	Ví dụ: Honda, Yamaha, Suzuki...	Ví dụ: 110cc
Số khung *	Số máy *	Màu sơn
Ví dụ: 1033612	Ví dụ: 1033612	Ví dụ: Đỏ
Số loại	Loại xe	Số người
Ví dụ: Air Blade, Future, Wave...	Ví dụ: hai bánh, ba bánh	Ví dụ: 2
Năm sản xuất	Tự trọng (kg)	
Ví dụ: 2000	Ví dụ: 100	

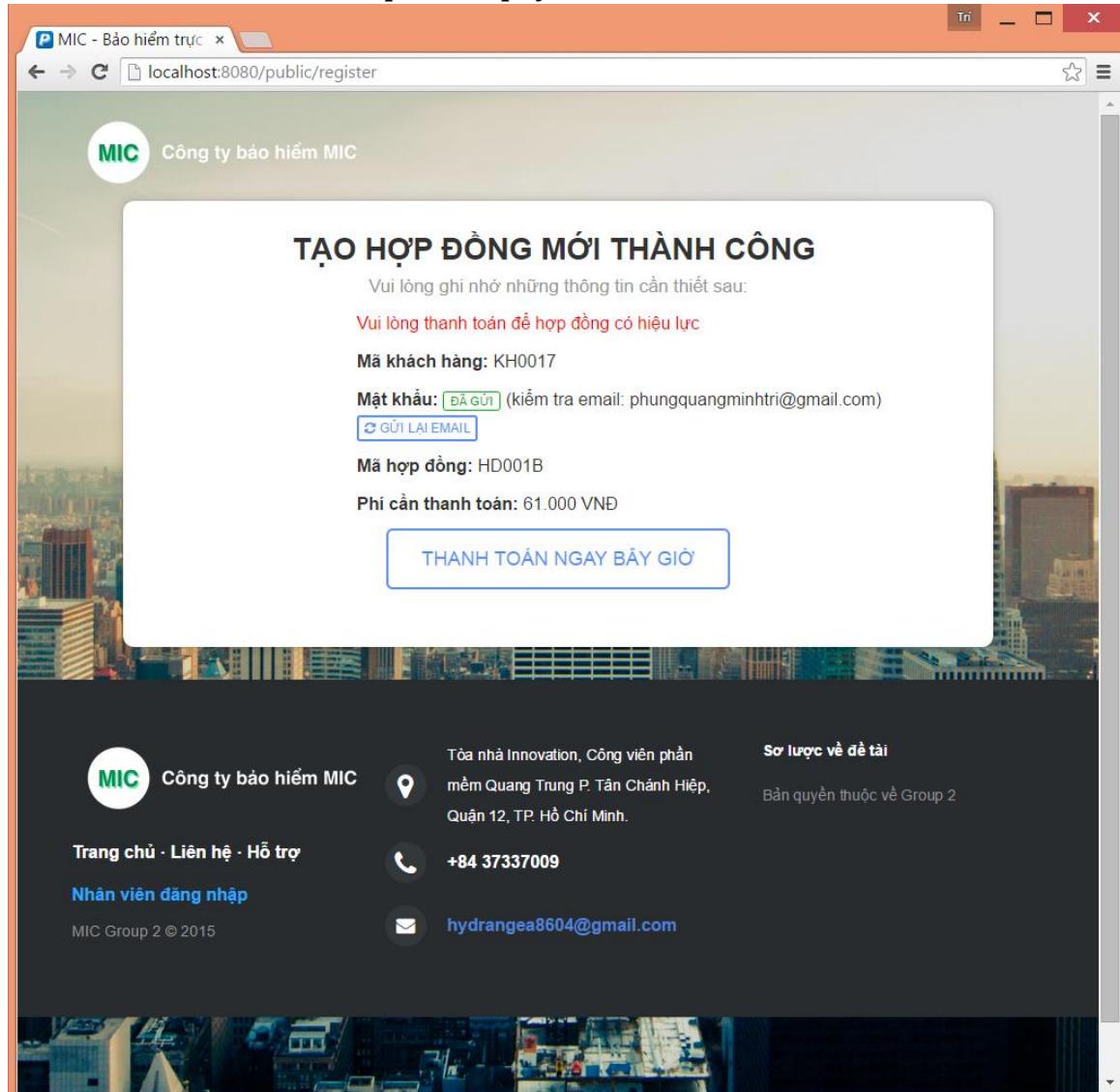
A green 'TIẾP THEO' button is located at the bottom right of the form.

Step	Description
5	Fill in motor information: "Biển số đăng ký": Plate number "Nhãn hiệu": Brand "Số khung": Chassis "Số máy": Engine "Dung tích": Capacity "Số loại": Model "Loại xe": Type "Màu sơn": Color "Năm sản xuất": Year of manufacturer "Tự trọng": Weight "Số người": Seat capacity
6	Click "Tiếp theo" button



Step	Description
7	Confirm the inputted information and click “TẠO HỢP ĐỒNG VÀ THANH TOÁN” button.

2.1.1.2.Guest process payment



Step	Description
1	Click "THANH TOÁN NGAY BÂY GIỜ" button.

The screenshot shows a web browser window with the title "MIC - Bảo hiểm trực tuyến" and the URL "localhost:8080/public/checkout". The main content is a payment form titled "THANH TOÁN". It displays the following information:

- Nội dung thanh toán: Đăng ký hợp đồng mới HD001B
- Tổng tiền phải trả: 61.000 (VND)
- Xin lựa chọn phương thức thanh toán sau:

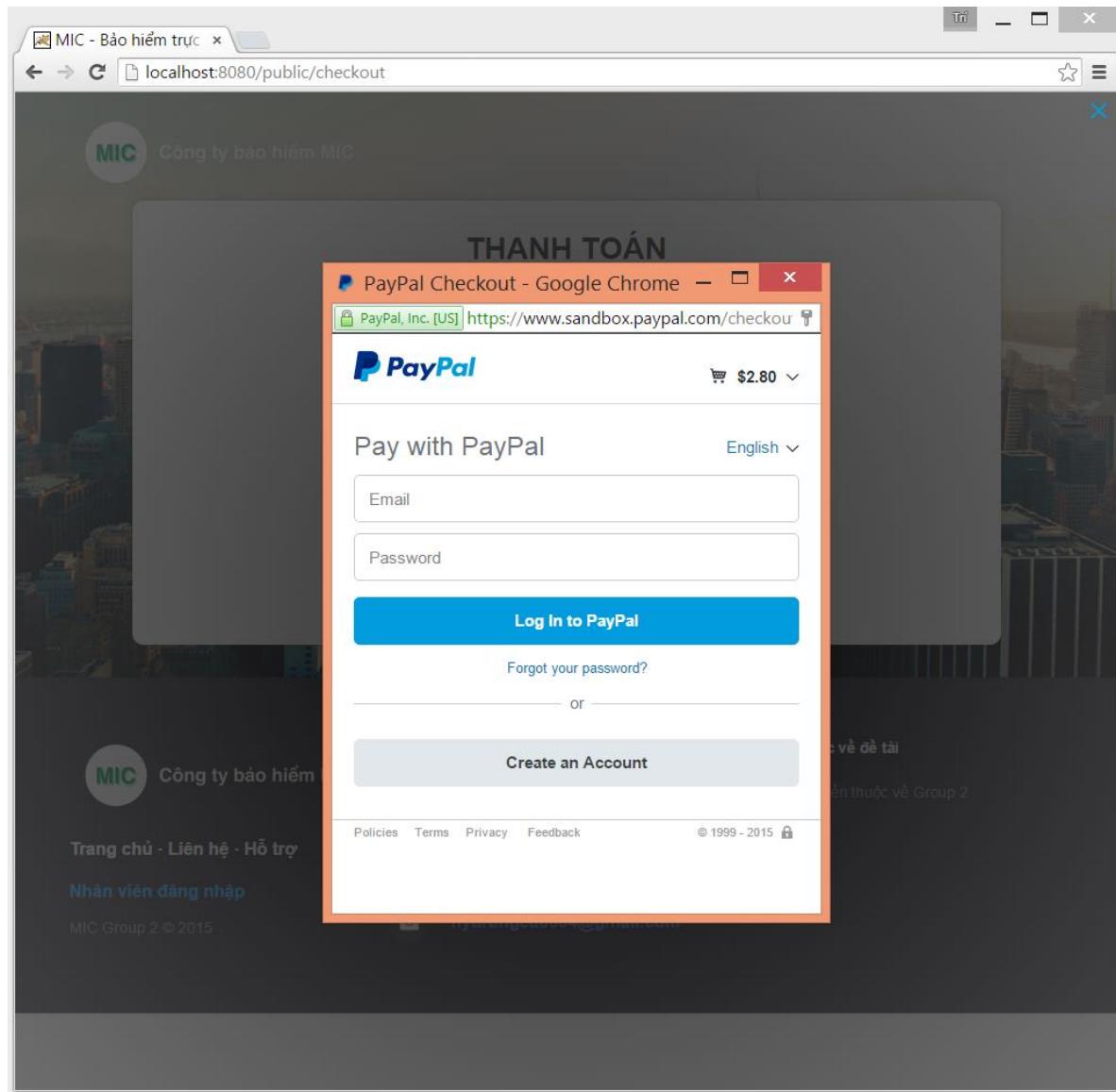
Two payment method options are shown:

- PAYPAL (selected, indicated by a blue border)
- TRỰC TIẾP (indicated by a grey circle)

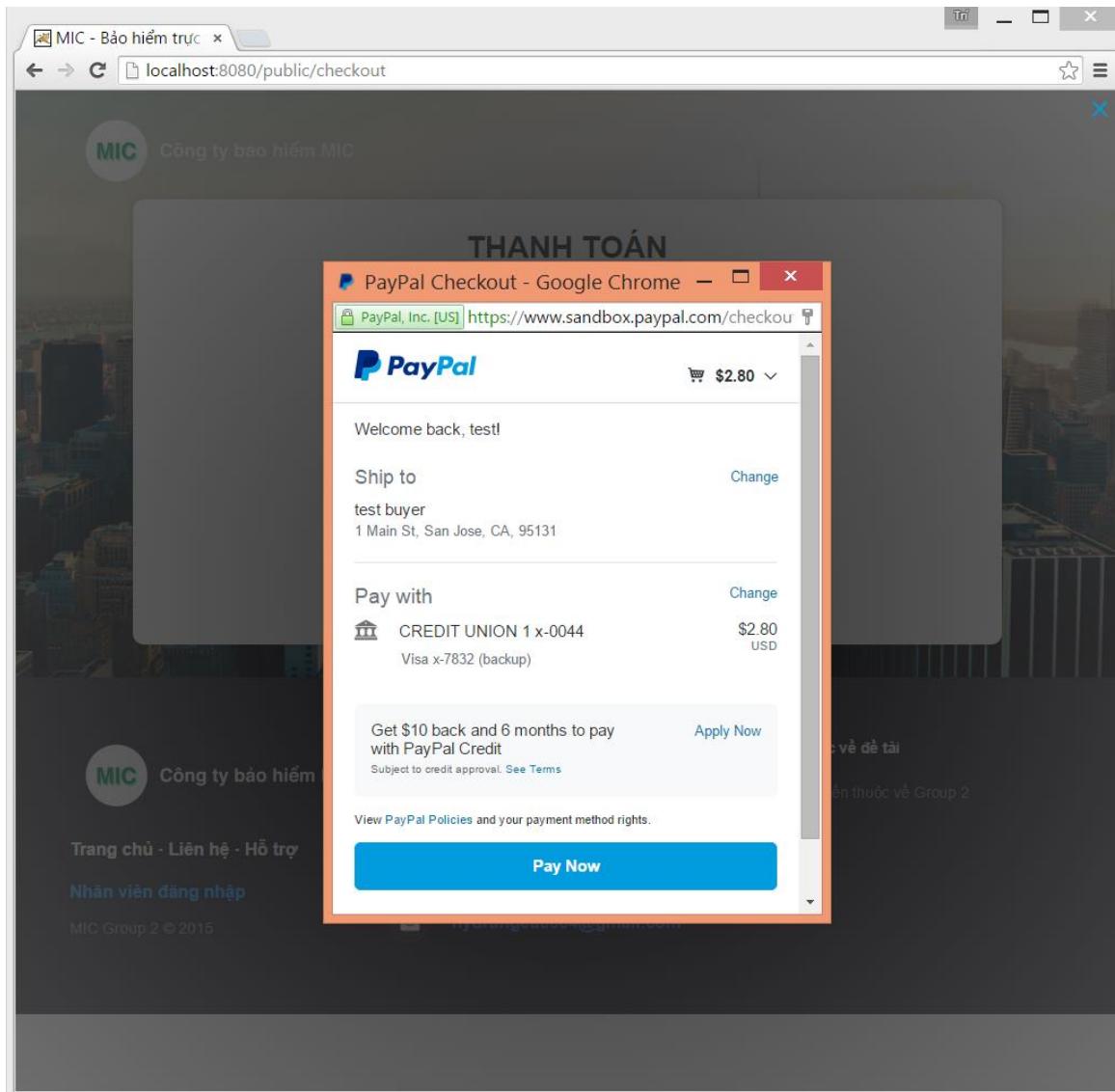
The footer of the page contains the MIC logo and contact information:

- Công ty bảo hiểm MIC
- Tòa nhà Innovation, Công viên phần mềm Quang Trung P. Tân Chánh Hiệp, Quận 12, TP. Hồ Chí Minh.
- Sơ lược về đài tài
- Bản quyền thuộc về Group 2
- Trang chủ · Liên hệ · Hỗ trợ
- +84 37337009
- Nhân viên đăng nhập
- hydrangea8604@gmail.com
- MIC Group 2 © 2015

Step	Description
2	Click "PayPal" button



Step	Description
3	Enter your PayPal email and password
4	Click "Log In to PayPal" button



Step	Description
5	Click “Pay Now” button.

2.1.2. Customer

2.1.2.1. Customer create new contract

The screenshot shows a web application interface for managing insurance contracts. The left sidebar has a navigation menu with 'Khách hàng' (Customer) at the top, followed by 'Thông tin cá nhân', 'Hợp đồng' (selected), 'Thông tin thẻ', 'Yêu cầu thẻ mới', and 'Lịch sử giao dịch'. The main content area is titled 'Hợp đồng' and displays a table with two rows of contract information. The columns are '#', 'Mã hợp đồng', 'Ngày bắt đầu', 'Ngày kết thúc', and 'Trạng thái'. Row 1: HD0002, 27/04/2015, 27/04/2016, Sẵn sàng. Row 2: HD0001, 09/08/2014, 09/08/2016, Sẵn sàng. There are buttons for 'Tìm kiếm theo mã hợp đồng' and 'Tìm kiếm' at the top right, and a green button '+ Hợp đồng mới' on the far right.

Step	Description
1	Click on “Hợp đồng mới” button

The screenshot shows a 'Thêm hợp đồng mới' (Add new contract) form. The left sidebar is identical to the previous screenshot. The main form has two sections: 'Thông tin về dịch vụ bảo hiểm' and 'Thông tin về xe cơ giới'. In the first section, there are dropdowns for 'Loại hình bảo hiểm' (Xe trên 50cc có BH cho người trên xe), a date input for 'Thời điểm có hiệu lực' (08/10/2015), and a note 'Thời điểm hết hiệu lực: 12 tháng kể từ ngày cấp'. Below these, the 'Phi bảo hiểm:' field is highlighted in red with the value '86.000 VND'. In the second section, there are several input fields: 'Biển số đăng ký' (635Y9-13322), 'Nhãn hiệu' (Honda), 'Số máy' (122134), 'Số khung' (35652234), 'Dung tích' (110cc), 'Màu sơn' (xanh), 'Loại xe' (Ví dụ: Hai bánh), 'Số loại' (Air Blade), 'Năm sản xuất' (2006), 'Tự trọng (kg)' (100), and 'Số người được chở' (2). A green button '+ Thêm hợp đồng' is located at the bottom right of the form area.

Step	Description
2	Fill information into blank fields for example:

	Choose “Loại hình bảo hiểm” : Xe trên 50cc có BH cho người trên xe “Thời điểm có hiệu lực” : 10/08/2015 “Biển số đăng ký” : 635Y9-13322 “Nhãn hiệu” : Honda “Số máy” : 122134 “Số khung” : 35652234 “Dung tích” : 110cc “Màu sơn” : xanh “Loại xe” : “Số loại” : Air Blade “Năm sản xuất” : 2006 “Tự trọng (kg)” : 100 “Số người được chở” :
3	Click on “Thêm hợp đồng” button

Vui lòng kiểm tra lại các thông tin hợp đồng
nhấn nút Xác nhận để hoàn tất tạo hợp đồng

Thông tin về dịch vụ bảo hiểm

Loại hình bảo hiểm : Xe trên 50cc có BH cho người trên xe

Thời điểm có hiệu lực : 10/08/2015

Thời điểm hết hiệu lực : 12 tháng kể từ ngày cấp

Phí bảo hiểm (VNĐ) : **86.000**

Thông tin về xe cơ giới

Biển số đăng ký : **635Y9-13322**

Nhãn hiệu : Honda

Số máy : 122134

Số khung : 35652234

Dung tích : 110cc

Màu sơn : xanh

Loại xe : Không có

Số loại : Air Blade

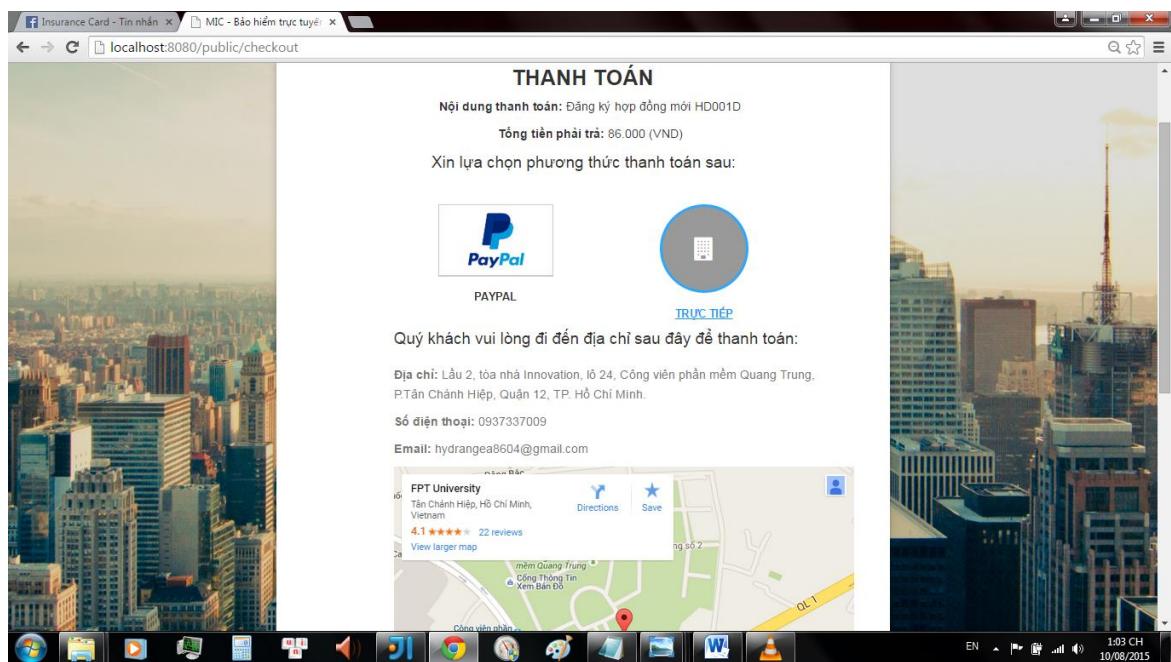
Năm sản xuất : 2006

Tự trọng (kg): 100

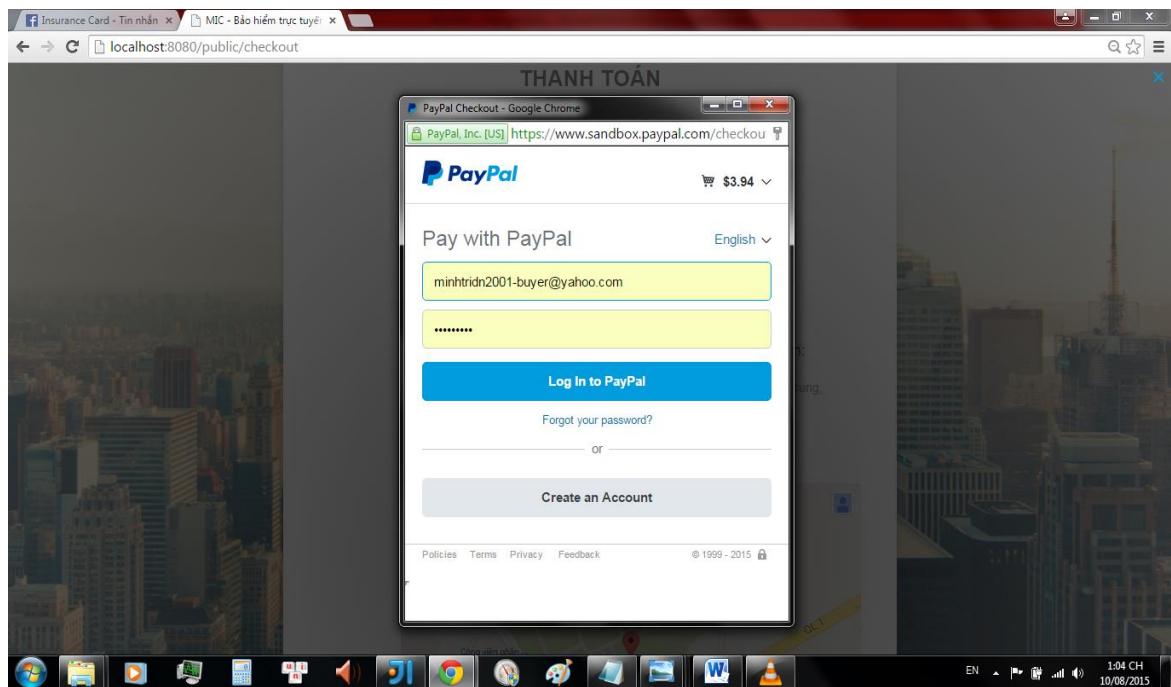
Số người được chở: 2

→ Xác nhận

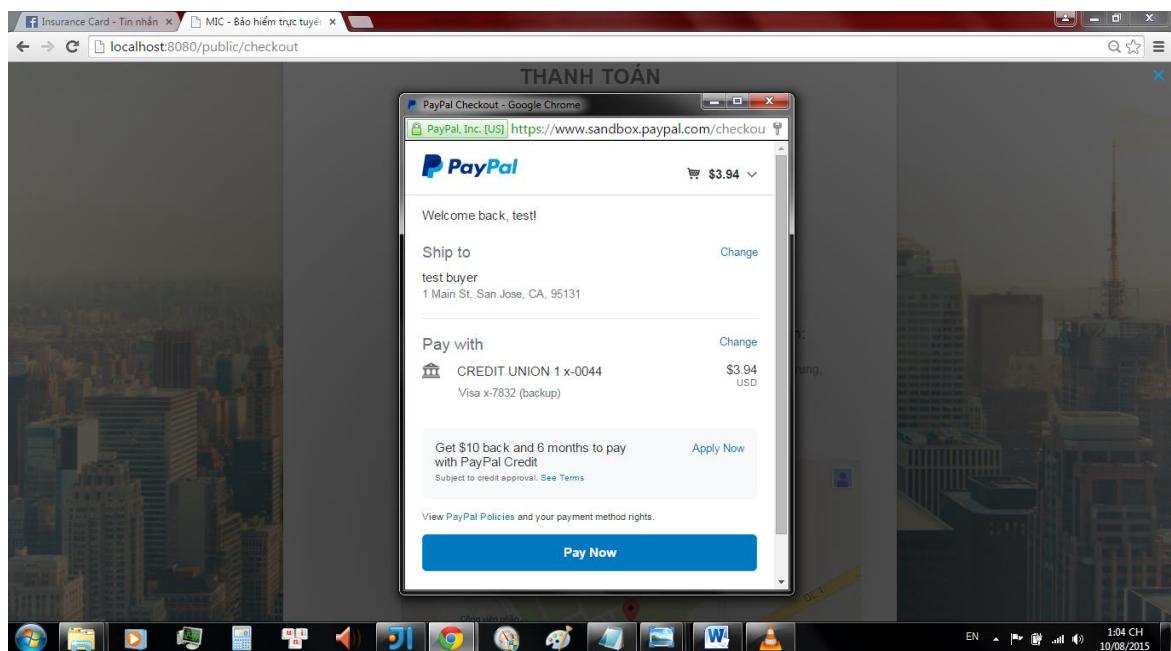
Step	Description
4	Click on “Xác nhận” button



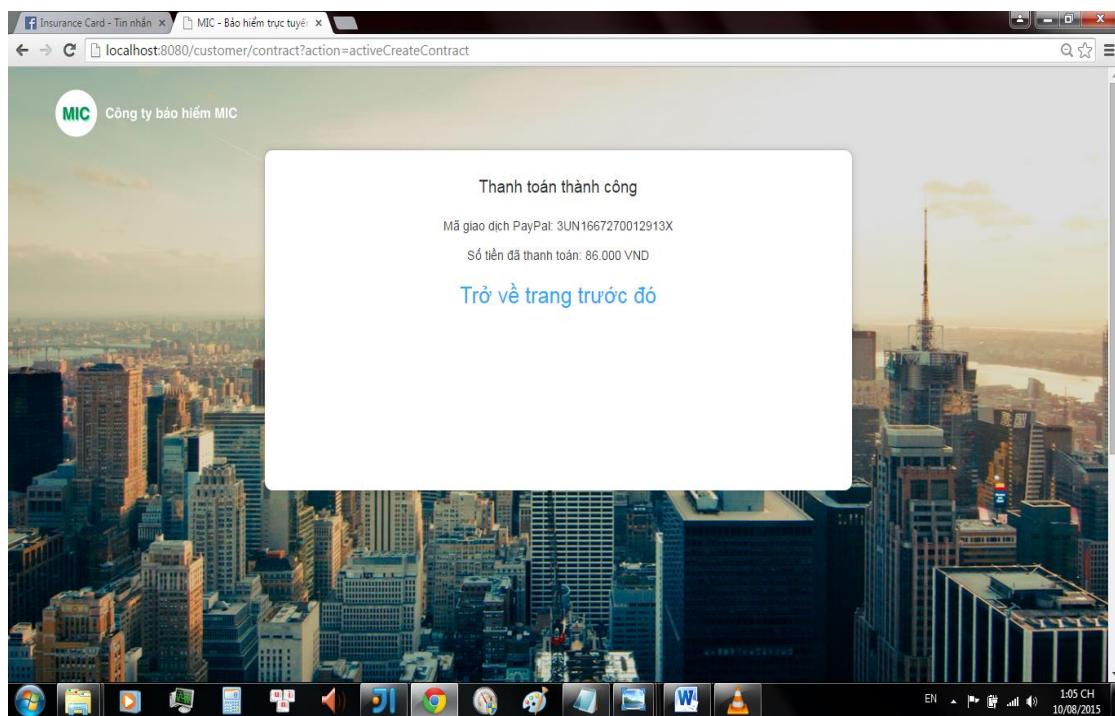
Step	Description
5	<ul style="list-style-type: none"> Click "PayPal" link image button <p>If customer want to pay direct at company they can view address of company on map by clicking "Trực Tiếp" link image button</p>



Step	Description
6	Click "Login In to PayPal" button



Step	Description
7	Click "Pay Now" button



Step	Description
8	Click on "Trở về trang trước đó"

The screenshot shows a web-based insurance management system. On the left, a sidebar for 'Khách hàng' (Customer) shows 'Đinh Quang Trung'. The main content area displays 'Hợp Đồng HD001D'. At the top right is a red button labeled 'Hủy Hợp Đồng' (Cancel Contract). Below the title are tabs: 'Thông tin chung' (General Information), 'Lịch sử bồi thường' (Claim History), 'Lịch sử vi phạm luật GT' (Traffic Violation History), and 'Lịch sử tai nạn' (Accident History). The 'Thông tin hợp đồng' (Contract Information) section contains tables for basic information like 'Mã hợp đồng' (Contract ID: HD001D), 'Ngày tham gia lúc' (Join Date: 10/08/2015), 'Ngày hết hạn' (Expiry Date: 10/08/2016), and coverage details like 'Quyền lợi bảo hiểm' (Insurance Benefits: Xe trên 50cc có BH cho người trên xe) and 'Tình trạng hợp đồng' (Contract Status: Còn hạn: 366 ngày). The 'Thông tin xe' (Vehicle Information) section shows details for a Honda Air Blade motorcycle. The bottom status bar shows system icons and the date/time: 10/08/2015.

2.1.2.2.Customer cancel contract

This screenshot shows a list of contracts for a specific customer. The sidebar remains the same. The main area shows 'Có 2 hợp đồng' (2 contracts) listed in a table:

#	Mã hợp đồng	Ngày bắt đầu	Ngày kết thúc	Trạng thái
1	HD0002	27/04/2015	27/04/2016	Sẵn sàng
2	HD0001	26/03/2015	26/03/2016	Sẵn sàng

A green button '+ Hợp đồng mới' (New Contract) is visible at the top right. The bottom status bar shows the date/time: 09/08/2015.

Step	Description
1	Click on contract code 'HD002' link

Hợp Đồng HD0002

Thông tin chung | Lịch sử bồi thường | Lịch sử vi phạm luật GT | Lịch sử tai nạn | Yêu cầu thẻ mới

Thông tin hợp đồng

Mã hợp đồng	HD0002	Quyền lợi bảo hiểm	Xe dưới 50cc có BH cho người trên xe
Ngày tham gia lúc	27/04/2015	Tình trạng hợp đồng	Còn hạn: 262 ngày
Ngày hết hạn	27/04/2016	Trạng thái	Sẵn sàng

Thông tin xe

Biển số đăng ký	56E4-5953	Nhàm hiệu	Yamaha
Số khung	251263	Số máy	845624
Dung tích	150	Màu sơn	Xanh
Số loại	H66	Loại xe	Hai bánh

Step	Description
2	Click “Hủy Hợp Đồng” button

Hủy Hợp Đồng

Quý khách vui lòng cung cấp lý do hoặc trường hợp hủy hợp đồng

- Xe cơ giới bị thu hồi đăng ký và biển số theo quy định của pháp luật
- Xe cơ giới hết niên hạn sử dụng theo quy định của pháp luật
- Xe cơ giới bị mất được cơ quan công an xác nhận
- Xe cơ giới hỏng không sử dụng được hoặc bị phá hủy do tai nạn giao thông được cơ quan công an xác nhận
- Lý do khác

Xác Nhận | Đóng

Step	Description
3	<ul style="list-style-type: none"> ○ Choose “Xe cơ giới hỏng không sử dụng được hoặc bị phá hủy do tai nạn giao thông được cơ quan công an xác nhận” : cancel reason
4	<ul style="list-style-type: none"> ○ Click on “Xác nhận” button

If customer wants to cancel request cancel contract they can click on “Hủy yêu cầu” button.

2.1.2.3.Customer renew contract

#	Mã hợp đồng	Ngày bắt đầu	Ngày kết thúc	Trạng thái
1	HD0002	27/04/2015	27/04/2016	Sản xuất
2	HD0001	09/08/2014	09/08/2015	Hết hạn

Step	Description
1	Click on “HD001” link

FPT University – Capstone Project Summer 2015 – Group 2 – Insurance Card

Hợp Đồng HD0001

Thông tin chung		Lịch sử bồi thường	Lịch sử vi phạm luật GT	Lịch sử tai nạn
Thông tin hợp đồng				
Mã hợp đồng	HD0001	Quyền lợi bảo hiểm	Xe trên 50cc không có BH cho người trên xe	
Ngày tham gia lúc	09/08/2014	Tình trạng hợp đồng	Còn hạn: 0 ngày	
Ngày hết hạn	09/08/2015	Trạng thái	Hết hạn	
Thông tin xe				
Biển số đăng ký	93H3-1232	Nhãn hiệu	Honda	
Số khung	124232	Số máy	626323	
Dung tích	110	Màu sơn	Đỏ	

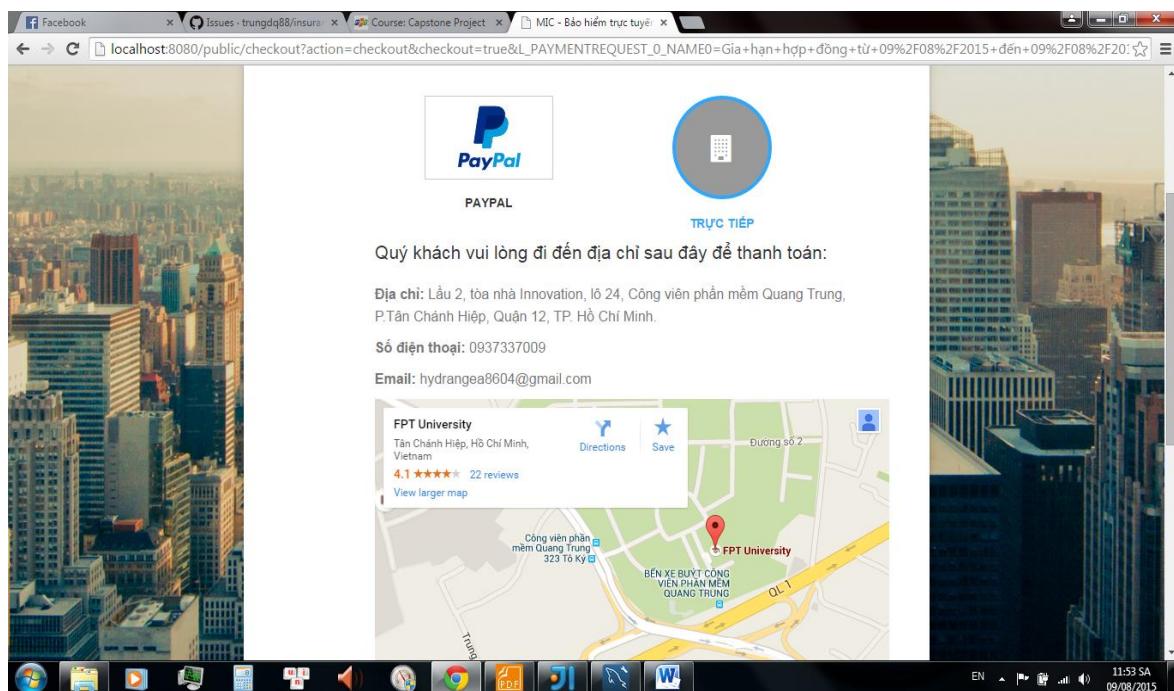
Step	Description
2	Click “Gia Hạn” button

Gia hạn hợp đồng

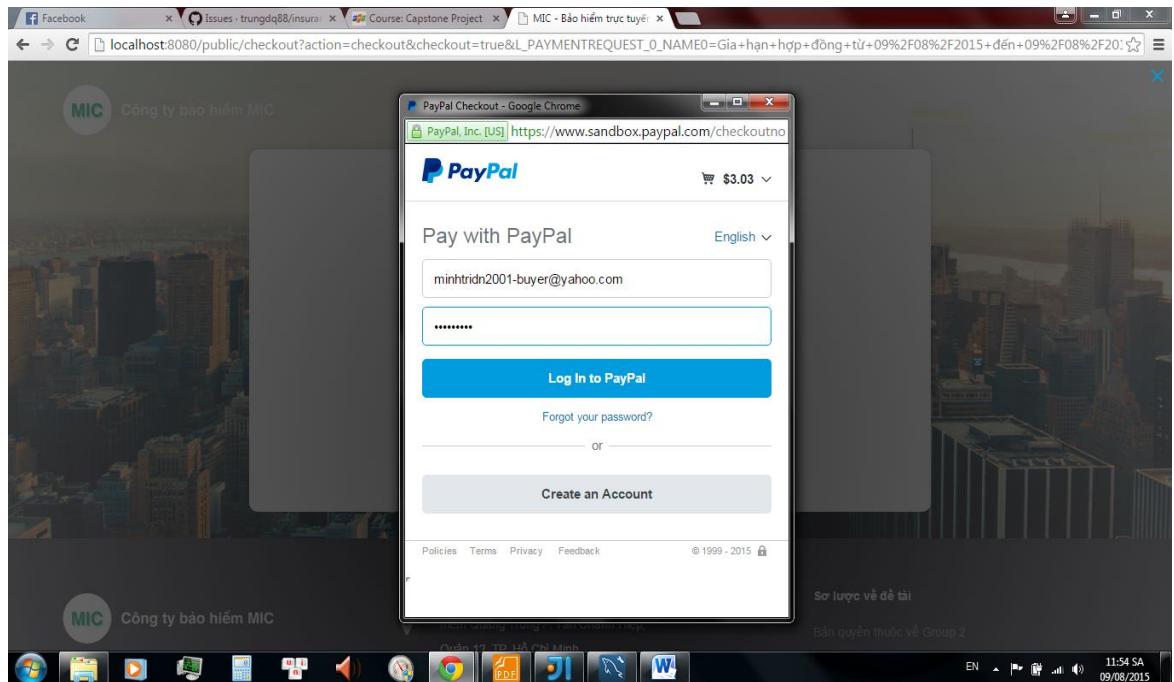
Loại hợp đồng	Xe trên 50cc không có BH cho người trên xe
Thời điểm bắt đầu	09/08/2014
Thời điểm kết thúc	09/08/2015
Gia hạn đến	09/08/2016
Phi thanh toán	66.000 VND

Step	Description
3	Click “Gia hạn hợp đồng” button

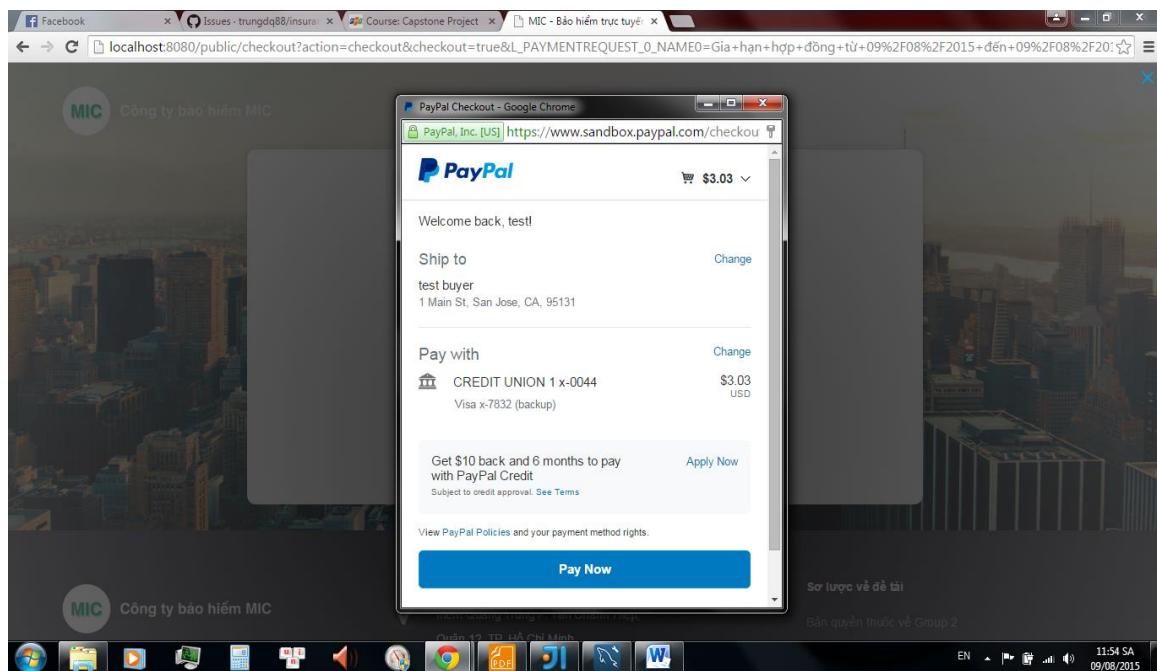
FPT University – Capstone Project Summer 2015 – Group 2 – Insurance Card



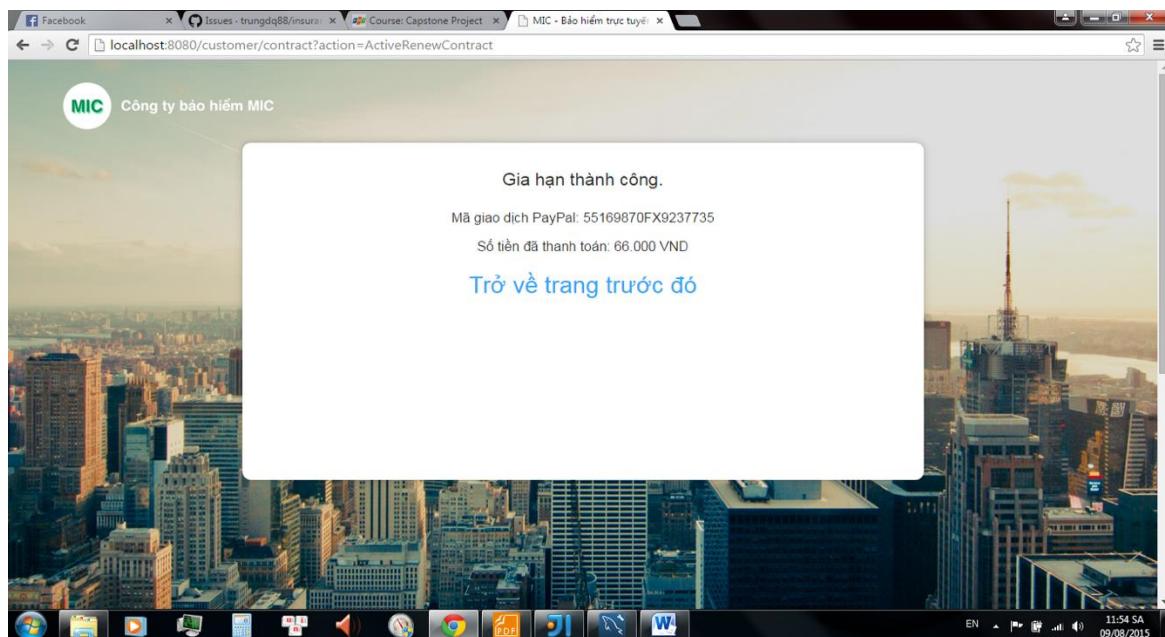
Step	Description
4	<ul style="list-style-type: none"> Click “PayPal” link image button <p>If customer want to pay direct at company they can view address of company on map by clicking “Trực Tiếp” link image button</p>



Step	Description
5	Click “Login In to PayPal” button



Step	Description
6	Click “Pay Now” button



Step	Description
7	Click on “Trở về trang trước đó” link

View result after customer renew contract. Contract status was changed to “Sẵn sàng” and the expired date was increased one year.

The screenshot shows a web-based insurance management system. On the left, a sidebar menu for 'Đinh Quang Trung' includes options like 'Thông tin cá nhân', 'Hợp đồng' (selected), 'Thông tin thẻ', 'Yêu cầu thẻ mới', and 'Lịch sử giao dịch'. The main content area displays 'Hợp Đồng HD0001'. At the top right are buttons for 'Hủy Hợp Đồng' and 'Yêu cầu thẻ mới'. Below this are tabs for 'Thông tin chung', 'Lịch sử bồi thường', 'Lịch sử vi phạm luật GT', and 'Lịch sử tai nạn'. The 'Thông tin hợp đồng' section contains tables for policy details (Mã hợp đồng: HD0001, Ngày tham gia lúc: 09/08/2014, Ngày hết hạn: 09/08/2016) and vehicle information (Quyền lợi bảo hiểm: Xe trên 50cc không có BH cho người trên xe, Tình trạng hợp đồng: Còn hạn: 366 ngày, Trạng thái: Sẵn sàng). The 'Thông tin xe' section shows details for a Honda motorcycle (Biển số đăng ký: 93H3-1232, Số khung: 124232, Số máy: 626323, Dung tích: 110, Màu sơn: Đỏ). The bottom of the screen shows a taskbar with various icons and a system tray indicating the date as 09/08/2015 and time as 11:55 SA.

2.1.2.4.Customer create compensation

This screenshot shows the same insurance management system interface. The sidebar and top navigation bar are identical. The main content area now displays the 'Yêu cầu bồi thường' tab. It shows a message 'Có 0 vụ bồi thường' and a table with columns: #, Mã yêu cầu, Mã hợp đồng, Ngày yêu cầu, and Ngày giải quyết. A note at the bottom states 'Không có đèn bù nào'. The bottom of the screen shows a taskbar and system tray with the date 10/08/2015 and time 10:56 SA.

Step	Description
1	Click on “Yêu cầu bồi thường” button

Yêu cầu bồi thường cho hợp đồng HD0001

Thông tin hợp đồng

Mã hợp đồng: HD0001
Tên khách hàng: Đinh Quang Trung
Tải mẫu văn bản yêu cầu bồi thường

Hay điền trực tiếp thông tin yêu cầu bồi thường dưới đây

Lưu ý quan trọng: Người kê khai phải kê khai đầy đủ và trung thực các nội dung dưới đây. Doanh nghiệp bảo hiểm có thể từ chối một phần số tiền bồi thường nếu nhận được nội dung kê khai thiếu trung thực

Họ tên lái xe *	Đinh Quang Trung	Điện thoại *	123123123
Địa chỉ *	Phường Tân Chánh Hiệp, Q 12, TP HCM	Hạng *	Ví dụ: A1
Giấy phép lái xe số *	Ví dụ: 740118000357	Trọng tải/số chỗ ngồi *	Ví dụ: 7 (tấn/chỗ)
Biển số xe gây tai nạn *	Ví dụ: 78Y9-15383		

Step	Description
2	Click on “Tải mẫu văn bản yêu cầu bồi thường” link

If customer wants to fill compensation information direct on website they can fill data into blank field.

Họ tên lái xe *: Đinh Quang Trung
Điện thoại *: 01228847857
Địa chỉ *: Phường Tân Chánh Hiệp, Q 12, TP HCM
Giấy phép lái xe số *: 59832589373924
Hạng *: A2
Biển số xe gây tai nạn *: 736Y-23222
Trọng tải/số chỗ ngồi *: 2 (tấn/chỗ)
Ngày xảy ra tai nạn *: 08/10/2015
Nơi xảy ra tai nạn *: An Sương, District 12, Ho Chi Minh, Vietnam
CQCA giải quyết *: CSGT bình triều
Diễn biến tai nạn *: xe chạy quá tốc độ mất thắng
Thiệt hại về người *: không có
Thiệt hại về tài sản *: hai xe máy hư hỏng nặng
Người làm chứng *: Đặng Thu Trinh
Địa chỉ liên hệ *: An Phú, Ho Chi Minh City, Ho Chi Minh, Vietnam
Yêu cầu bồi thường:

Step	Description
------	-------------

3	<p>Fill information into blank fields for example</p> <ul style="list-style-type: none"> • “Họ tên lái xe” : Đinh Quang Trung • “Điện thoại” : 01228847857 • “Địa chỉ” : Phường Tân Chánh Hiệp, Q 12, TP HCM • “Giấy phép lái xe số” : 598325893739242 • “Hạng” : A2 • “Bến số xe gây tai nạn” : 736Y-23222 • “Ngày xảy ra tai nạn” : 10/08/2015 • “Trọng tải/số chỗ ngồi” : 2 • “Nơi xảy ra tai nạn” : An Sương, District 12, Ho Chi Minh, Vietnam • “CQCA giải quyết” : CSGT bình triệu • “Diễn biến tai nạn” : xe chạy quá tốc độ mất thăng • “Thiệt hại về người” : không có • “Thiệt hại về tài sản” : hai xe máy hư hỏng nặng • “Người làm chứng” : Đặng Thu Trinh • “Địa chỉ liên hệ” : An Phú, Ho Chi Minh City, Ho Chi Minh, Vietnam
----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The screenshot shows a web application window titled "Giao diện quản lý - Insurance". The URL is "localhost:8080/customer/compensation?action=Create&contractCode=HD0001". The form contains the following data:

Ngày xảy ra tai nạn *	08/10/2015
Nơi xảy ra tai nạn *	An Sương, District 12, Ho Chi Minh, Vietnam
CQCA giải quyết *	CSGT bình triệu
Diễn biến tai nạn *	xe chạy quá tốc độ mất thăng
Thiệt hại về người *	không có
Thiệt hại về tài sản *	hai xe máy hư hỏng nặng
Người làm chứng *	Đặng Thu Trinh
Địa chỉ liên hệ *	An Phú, Ho Chi Minh City, Ho Chi Minh, Vietnam
Yêu cầu bồi thường	(empty text area)

Gởi yêu cầu bồi thường

At the bottom of the page, there is a footer with the MIC logo, company name, address, and other links.

Step	Description
4	Click on “Gởi yêu cầu bồi thường” button

Yêu cầu bồi thường cho hợp đồng HD0001 sẽ được giải quyết từ ngày 10/08/2015 đến ngày 10/09/2015

Thông tin hợp đồng

Mã hợp đồng	HD0001
Tên khách hàng	Đinh Quang Trung

Thông tin yêu cầu bồi thường

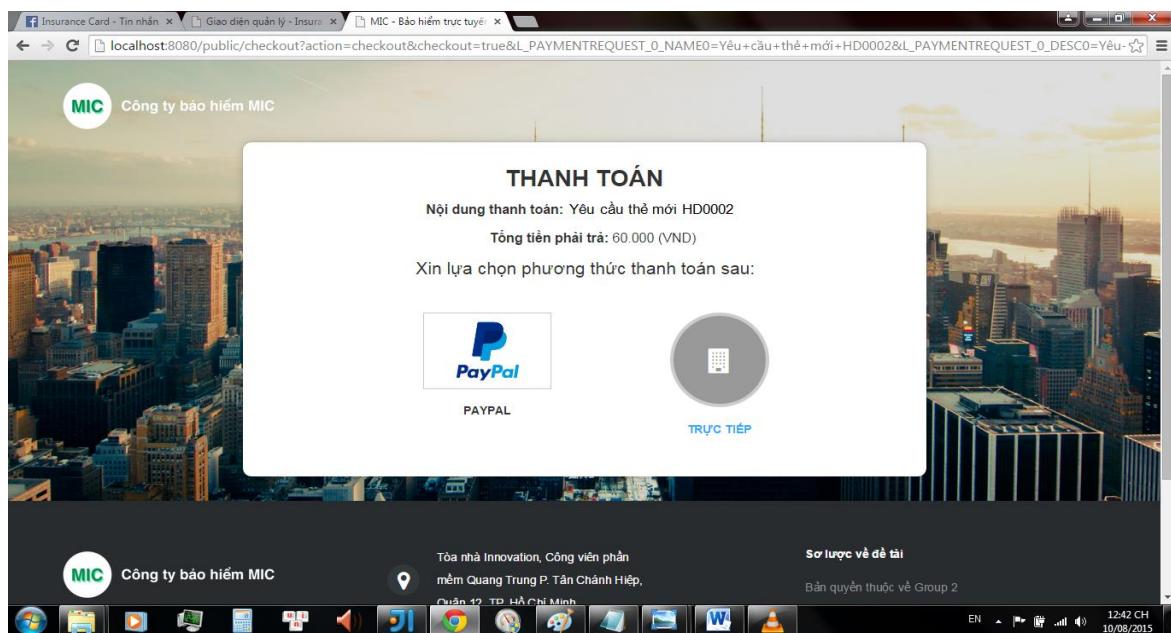
Ngày gửi yêu cầu	10/08/2015	Trạng thái	Chưa giải quyết
Họ tên lái xe	Đinh Quang Trung		
Địa chỉ liên hệ	Phường Tân Chánh Hiệp, Q 12, TP HCM		
Số điện thoại	01228847857		
Giấy phép lái xe số	598325893739242	Hạng	A2
Biển số xe gây tai nạn	736Y-23222	Trọng tài/số chỗ ngồi	2
Ngày xảy ra tai nạn	10/08/2015		

Ngày gửi yêu cầu	10/08/2015	Trạng thái	Chưa giải quyết
Họ tên lái xe	Đinh Quang Trung		
Địa chỉ liên hệ	Phường Tân Chánh Hiệp, Q 12, TP HCM		
Số điện thoại	01228847857		
Giấy phép lái xe số	598325893739242	Hạng	A2
Biển số xe gây tai nạn	736Y-23222	Trọng tài/số chỗ ngồi	2
Ngày xảy ra tai nạn	10/08/2015		
Nơi xảy ra tai nạn	An Sương, District 12, Ho Chi Minh, Vietnam		
Cơ quan CA giải quyết	CSGT bình triệu		
Điển biến và nguyên nhân	xe chạy quá tốc độ mất thắng		
Thiệt hại về người	không có		
Thiệt hại về tài sản	hai xe máy hư hỏng nặng		
Người làm chứng	Đặng Thu Trinh		
Địa chỉ của người làm chứng	An Phú, Ho Chi Minh City, Ho Chi Minh, Vietnam		
Yêu cầu bồi thường	Không có		

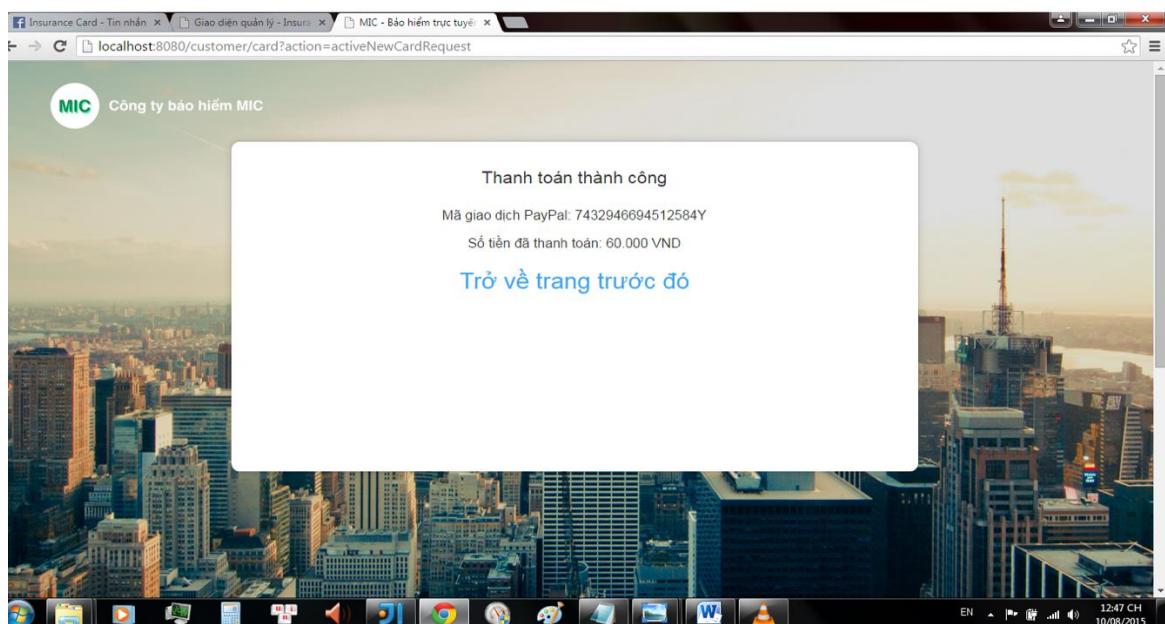
2.1.2.5.Customer request new card

Step	Description
1	Click on “Yêu cầu thẻ mới” button

Step	Description
2	<ul style="list-style-type: none"> - Fill “Xác nhận mật khẩu” : 123123 (for example) - “Ghi chú” : “cấp thẻ mới” (for example) - Check “Thanh toán qua PayPal” - Check “Hủy bỏ thẻ cũ” - Check “” Giao thẻ tận nơi
3	Click on “Xác nhận” button



Step	Description
4	<ul style="list-style-type: none"> - Click on “Trực tiếp” image link button if customer wants to pay direct at company - Click on “PayPal” image link button



Step	Description
5	<ul style="list-style-type: none"> - Click on “Trở về trang trước đó” link

Hợp Đồng HD0002

Thông tin chung | Lịch sử bồi thường | Lịch sử vi phạm luật GT | Lịch sử tai nạn | **Dang yêu cầu thẻ mới** | Hủy Hợp Đồng

Thông tin hợp đồng		
Mã hợp đồng	HD0002	Quyền lợi bảo hiểm
Ngày tham gia lúc	27/04/2015	Tình trạng hợp đồng
Ngày hết hạn	27/04/2016	Trạng thái

Thông tin xe		
Biển số đăng ký	56E4-5953	Nhà sản xuất
Số khung	251263	Số máy
Dung tích	150	Màu sơn

EN 12:49 CH 10/08/2015

Step	Description
6	- Click on “Yêu cầu thẻ mới” link menu

Yêu cầu thẻ mới

Có 1 yêu cầu thẻ mới chưa xử lý

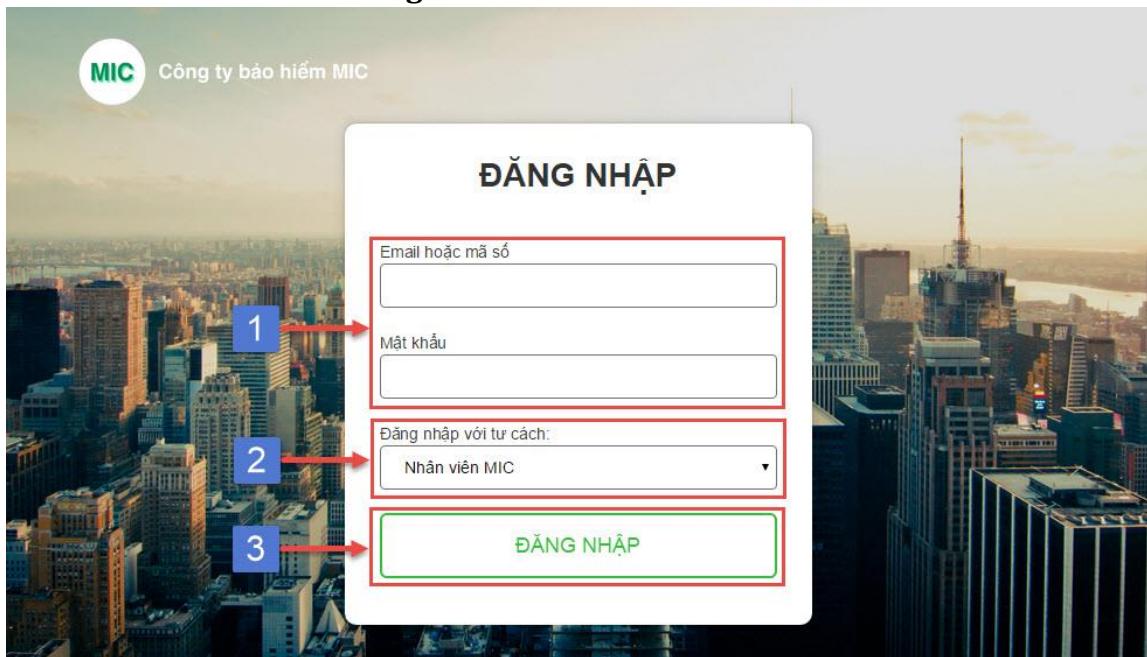
#	Thời gian	Ghi chú	Mã thẻ cũ	Hợp đồng	Ngày cấp mới	Thẻ mới cấp	Thực hiện
1	10/08/2015		9021748528325	HD0002	Chưa cấp		Đã thanh toán

localhost:8080/customer/card?action=viewNewCardRequests

EN 12:50 CH 10/08/2015

2.1.3. Staff

2.1.3.1. Staff login



Step	Description
1	Input email or code and password to login
2	Select role is "Nhân viên MIC"
3	Click "ĐĂNG NHẬP" button

2.1.3.2. Staff manage contract

Step	Description
1	Login with “Staff” role
2	Select menu “Hợp đồng” from navigation bar

2.1.3.3.Staff create new contract

#	Mã hợp đồng	Tên khách hàng	Ngày bắt đầu	Ngày kết thúc	Trạng thái
1	HD0018	Đương Phạm Huy Hùng	26/07/2015	26/07/2016	Sẵn sàng
2	HD0017	Vũ Thị Hậu	26/07/2015	26/07/2016	Sẵn sàng
3	HD0016	Đương Phạm Huy Hùng	26/07/2015	26/07/2016	Đã huỷ
4	HD0015	Nguyễn Tuấn Hùng	26/07/2015	26/07/2016	Sẵn sàng
5	HD0014	Trần Văn Thông	26/08/2015	26/04/2016	Chưa kích hoạt
6	HD0013	Đương Phạm Huy Hùng	16/09/2014	16/09/2015	Sẵn sàng
7	HD0012	Vũ Thị Hậu	26/09/2015	26/07/2016	Chưa kích hoạt
8	HD0011	Nguyễn Hoàng Trường	16/08/2014	16/08/2015	Sẵn sàng
9	HD0010	Đỗ Tiến Sỹ	26/07/2015	26/07/2016	Sẵn sàng
10	HD0007	Phạm Tiến Dũng	26/07/2015	26/05/2016	Sẵn sàng

Step	Description
1	Click on button “Hợp đồng mới”

Công ty bảo hiểm MIC

Nhân viên
Nguyễn Chí Kha

Thông tin khách hàng

Thông tin về dịch vụ bảo hiểm

Thông tin về xe cơ giới

Thông tin thanh toán

Các ô có dấu * là bắt buộc

Step Description

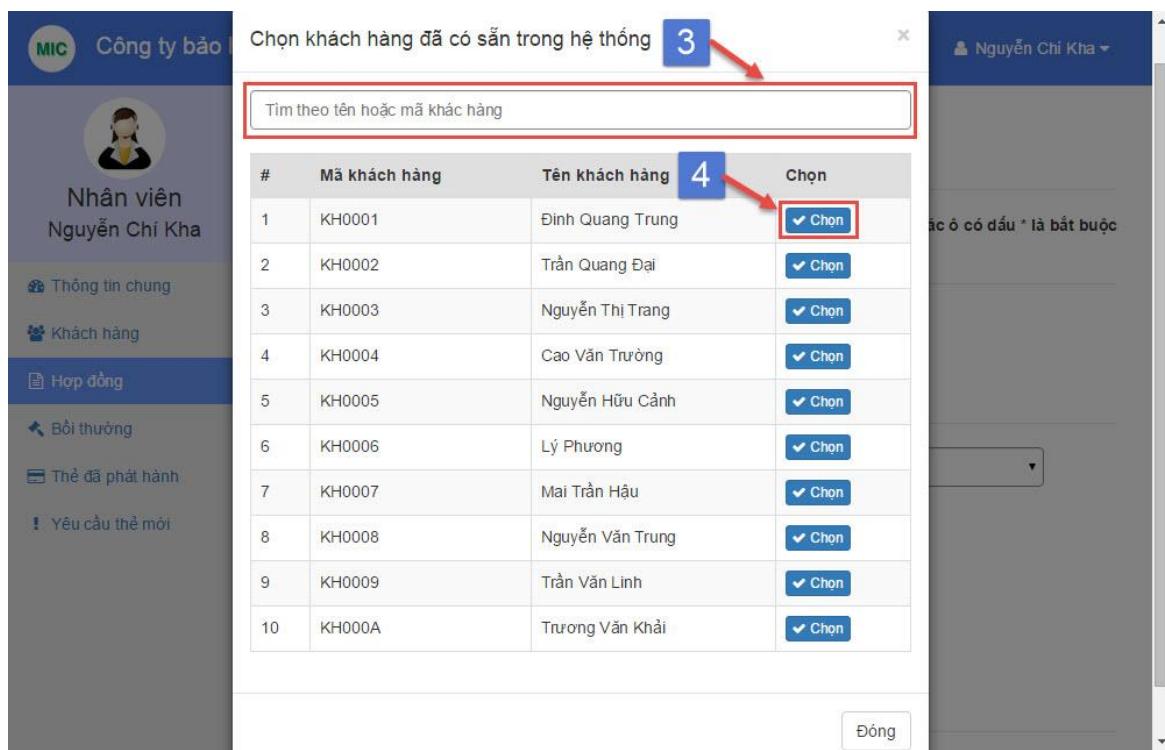
2 Click “Chọn” button

5 Select insurance type

6 Enter vehicle details

7 Enter payment date

8 Click “Kiểm tra thông tin và hoàn tất” button



Step	Description
3	Search customer by name or code
4	Click "Chọn" button

Thêm hợp đồng mới

Các ô có dấu * là bắt buộc

Thông tin khách hàng

2 → Mã khách hàng *

Thông tin về dịch vụ bảo hiểm

5 → Loại hình bảo hiểm *
 Thời điểm có hiệu lực *
 Thời điểm hết hiệu lực *

Phí bảo hiểm *

Thông tin về xe cơ giới

6 → Biển số đăng ký * Nhãn hiệu *
 Số khung * Số máy *
 Dung tích * Màu sơn
 Số loại Loại xe
 Năm sản xuất Tự trọng kg
 Số người được chở người

Thông tin thanh toán

7 → Ngày nộp phí *
 8 →

Step	Description
5	Select contract type in “Loại hình bảo hiểm” dropdown list (required) Choose “Thời điểm có hiệu lực” the start date of this contract (required) Choose “Thời điểm hết hiệu lực” the end date of this contract (required)
6	Fill vechile plate, brand, engine, chassis, capacity (required) Fill color, model code, vehicle type, year of manufacture, weight (optional)
7	Choose “Ngày nộp phí” the paid date of this contract (required)
8	Click “Kiểm tra thông tin và hoàn tất” button

Công ty bảo hiểm MIC

Nhân viên
Nguyễn Chí Kha

Thông tin chung

Khách hàng

Hợp đồng (Selected)

Bồi thường

Thẻ đã phát hành

Yêu cầu thẻ mới

Thêm hợp đồng mới

Vui lòng kiểm tra lại các thông tin trong hợp đồng chính xác.
Nhấn nút **Hoàn tất hợp đồng** ở cuối trang để hoàn tất hợp đồng

Thông tin khách hàng

Tên Khách hàng	Đinh Quang Trung		
Địa chỉ	Phường Tân Chánh Hiệp, Q 12, TP HCM		
Email	trungdq88@gmail.com		
Số điện thoại	123123123	Số CMND / Hộ chiếu	123123123

Thông tin về dịch vụ bảo hiểm

Loại hợp đồng	Xe trên 50cc có BH cho người trên xe
Bắt đầu có hiệu lực từ	ngày 10 tháng 08 năm 2015
Thời điểm hết hiệu lực	ngày 10 tháng 08 năm 2016

Thông tin về xe cơ giới

Biển số đăng ký	55S1-43563	Nhãn hiệu	Honda
Số khung	U600QU5FFUUF	Số máy	BK72I661WG0
Dung tích	108 cc	Màu sơn	Đen-Trắng
Số loại	Air Blade	Loại xe	Hai bánh
Năm sản xuất	2012	Tự trọng	120 kg
Số người được chở	2 người		

Thông tin thanh toán

Số tiền phí đã trả	86.000 VNĐ	Ngày nộp phí	10/08/2015
--------------------	------------	--------------	------------

9 → **Hoàn tất hợp đồng**

← Quay lại chỉnh sửa

Step	Description
9	Click “Hoàn tất hợp đồng” button

2.1.3.4.Staff view contract detail

#	Mã hợp đồng	Tên khách hàng	Ngày bắt đầu	Ngày kết thúc	Trạng thái
1	HD0018	Dương Phạm Huy Hùng	26/07/2015	26/07/2016	Sẵn sàng
2	HD0017	Vũ Thị Hậu	26/07/2015	26/07/2016	Sẵn sàng
3	HD0016	Dương Phạm Huy Hùng	26/07/2015	26/07/2016	Bã huỷ
4	HD0015	Nguyễn Tuấn Hùng	26/07/2015	26/07/2016	Sẵn sàng
5	HD0014	Trần Văn Thông	26/08/2015	26/04/2016	Chưa kích hoạt
6	HD0013	Dương Phạm Huy Hùng	16/09/2014	16/09/2015	Sẵn sàng
7	HD0012	Vũ Thị Hậu	26/09/2015	26/07/2016	Chưa kích hoạt
8	HD0011	Nguyễn Hoàng Trường	16/08/2014	16/08/2015	Sẵn sàng
9	HD0010	Đỗ Tiên Sỹ	26/07/2015	26/07/2016	Sẵn sàng
10	HD00007	Phan Tiên Dung	26/07/2015	26/05/2016	Sẵn sàng

Step	Description
1	Click on view contract detail link

2.1.3.5.Staff complete payment

Mã hợp đồng	Trạng thái
HD0019	Chưa kích hoạt

Thông tin về dịch vụ bảo hiểm

Loại hợp đồng	Mô tả
Xe từ 50cc trở xuống có BH cho người trên xe	Mã hợp đồng: HD0019 Bắt đầu có hiệu lực từ: ngày 11 tháng 08 năm 2015 Thời điểm hết hiệu lực: ngày 11 tháng 08 năm 2015 Phí bảo hiểm: 81.000 VND

Step	Description
1	Click "Hoàn tất thanh toán" button

Thông tin về dịch vụ bảo hiểm

Mã hợp đồng: HD0019 | Trạng thái: Chưa kích hoạt

Loại hợp đồng: Xe từ 50cc trở xuống có BH cho người trên xe

Bắt đầu có hiệu lực từ: ngày 11 tháng 08 năm 2015

Thời điểm hết hiệu lực: ngày 11 tháng 08 năm 2015

Thông tin thanh toán

Ngày nộp phí*: 11/08/2015

Hoàn tất thanh toán

Phi bảo hiểm: 81.000 VNĐ

Step	Description
2	Choose “Ngày nộp phí” the payment paid date (required)
3	Click “Hoàn tất thanh toán” button

2.1.3.6.Staff renew contract

Gia hạn hợp đồng

Thông tin về dịch vụ bảo hiểm

Mã hợp đồng: HD000R | Trạng thái: Hết hạn

Loại hợp đồng: Xe trên 50cc có BH cho người trên xe

Bắt đầu có hiệu lực từ: ngày 24 tháng 09 năm 2014

Thời điểm hết hiệu lực: ngày 24 tháng 07 năm 2015

Thông tin gia hạn hợp đồng

Gia hạn đến*: 10/08/2016

Phi gia hạn: 86.000 VNĐ

Cấp thẻ mới:

Ngày nộp phí*: 10/08/2015

Gia hạn hợp đồng

Step	Description
1	Click “Gia hạn” button
2	Choose “Gia hạn đến” the renewal date (required)
3	Choose “Ngày nộp phí” the payment paid date (required)
4	Click “Gia hạn hợp đồng” button

Step	Description
1	Click “Gia hạn” button

2	Choose “Gia hạn đến” the contract new expired date (required)
3	Choose “Ngày nộp phí” the contract paid date (required)
4	Click “Gia hạn hợp đồng” button

2.1.3.7.Staff resolve cancel request

Công ty bảo hiểm MIC

Nhân viên
Nguyễn Chí Kha

Thông tin chung
Khách hàng
Hợp đồng
Bồi thường
Thẻ đã phát hành
Yêu cầu thẻ mới

HD0002

HỢP ĐỒNG NÀY ĐANG CÓ YÊU CẦU HỦY TỪ KHÁCH HÀNG

Thời điểm gửi yêu cầu: 11/08/2015 lúc 00:00:00
Lý do hủy hợp đồng: Xe cơ giới hỏng không sử dụng được hoặc bị phá huỷ do tai nạn giao thông được cơ quan công an xác nhận

1 → **✓ Giải quyết**

Thông tin về dịch vụ bảo hiểm

Mã hợp đồng: HD0002 | Trạng thái: **Yêu cầu hủy**
Loại hợp đồng: Xe từ 50cc trở xuống có BH cho người trên xe
Bắt đầu có hiệu lực từ: ngày 27 tháng 04 năm 2015
Thời điểm hết hiệu lực: ngày 27 tháng 04 năm 2016

Step	Description
1	Click “Giải quyết” button

Công ty bảo hiểm MIC

Nhân viên
Nguyễn Chí Kha

Thông tin chung
Khách hàng
Hợp đồng
Bồi thường
Thẻ đã phát hành
Yêu cầu thẻ mới

Giải quyết yêu cầu hủy hợp đồng

Thời điểm gửi yêu cầu: 11/08/2015 lúc 00:00:00
Lý do hủy hợp đồng: Xe cơ giới hỏng không sử dụng được hoặc bị phá huỷ do tai nạn giao thông được cơ quan công an xác nhận

Quyết định của công ty *: **2** Hủy hợp đồng này

3 Ghi chú

4 → **✓ Xác nhận**

Thông tin về dịch vụ bảo hiểm

Mã hợp đồng: HD0002 | Trạng thái: **Yêu cầu hủy**
Loại hợp đồng: Xe từ 50cc trở xuống có BH cho người trên xe
Bắt đầu có hiệu lực từ: ngày 27 tháng 04 năm 2015
Thời điểm hết hiệu lực: ngày 27 tháng 04 năm 2016

Step	Description
2	Choose “Quyết định của công ty” (required)
3	Input “Ghi chú” if company decision is cancel this contract (optional)
4	Click “Xác nhận” button

2.1.3.8.Staff cancel contract

The screenshot shows the MIC insurance system interface. A modal window titled "Hủy hợp đồng" (Cancel Contract) is open. Step 1 is indicated by a red box around the "Hủy hợp đồng" button. Step 2 is indicated by a red box around the "Ngày hủy hợp đồng" input field containing "10/08/2015". Step 3 is indicated by a red box around the "Lý do hủy hợp đồng" and "Ghi chú" input fields. Step 4 is indicated by a red box around the "Đồng ý hủy" (Agree to cancel) button, which has a checked checkbox.

Step	Description
1	Click “Hủy hợp đồng” button
2	Choose “Ngày hủy hợp đồng” the contract cancel date (required)
3	Fill “Lý do hủy hợp đồng” the contract cancel reason (required) Fill “Ghi chú” the contract cancel note (optional)
4	Click “Đồng ý hủy” button

2.1.3.9.Staff manage compensation

The screenshot shows the MIC insurance management system interface. At the top, there is a header bar with the company logo and name "Công ty bảo hiểm MIC". On the right side of the header, there is a user profile for "Nguyễn Chí Kha" and a notification badge indicating 1 unread message. Below the header, the main content area has a sidebar on the left containing navigation links: Thông tin chung, Khách hàng, Hợp đồng, Bồi thường (highlighted with a red box and number 2), Thẻ đã phát hành, and Yêu cầu thẻ mới. The main content area displays four cards under the heading "Thông tin chung":

- A blue card for "Hợp đồng" showing 37 items, with a "Xem danh sách" button.
- A green card for "Yêu cầu bồi thường" showing 1 item, with a "Xem danh sách" button.
- An orange card for "Yêu cầu cấp thẻ mới" showing 0 items, with a "Xem danh sách" button.
- A red card for "Yêu cầu hủy hợp đồng" showing 1 item, with a "Xem danh sách" button.

Step	Description
1	Login with “Staff” role
2	Select menu “Bồi thường” from navigation bar

2.1.3.10. Staff create new compensation

The screenshot shows the MIC insurance management system interface for creating new compensation. At the top, there is a header bar with the company logo and name "Công ty bảo hiểm MIC". On the right side of the header, there is a user profile for "Nguyễn Chí Kha" and a notification badge indicating 1 unread message. Below the header, the main content area has a sidebar on the left containing navigation links: Thông tin chung, Khách hàng, Hợp đồng, Bồi thường (highlighted with a red box and number 1), Thẻ đã phát hành, and Yêu cầu thẻ mới. The main content area displays a list titled "Yêu cầu bồi thường" with a "Tìm kiếm" search bar and a "Tìm kiếm" button. A red box highlights the "Yêu cầu bồi thường mới" button. Below the search bar, there is a table showing two existing compensation requests:

#	Mã yêu cầu	Ngày yêu cầu	Mã hợp đồng	Tên khách hàng	Trạng thái
1	BT0002	25/07/2015	HD000B	Trương Văn Khải	Chưa giải quyết
2	BT0001	26/07/2015	HD000B	Trương Văn Khải	Đã giải quyết

Step	Description
1	Click on button “Yêu cầu bồi thường mới”

Thông báo tai nạn và yêu cầu bồi thường

Lưu ý quan trọng: Người kê khai phải kê khai đầy đủ và trung thực các nội dung dưới đây.
Doanh nghiệp bảo hiểm có thể từ chối một phần số tiền bồi thường nếu nhận được nội dung kê khai thiếu trung thực

Các ô có dấu * là bắt buộc

2 Mã hợp đồng *

Ngày gửi yêu cầu *

Chọn hình thức nhập * Nhập liệu bằng tay Nhập từ ảnh chụp biên bản

Step	Description
2	Click “Chọn” button

Chọn hợp đồng đã có sẵn trong hệ thống

Tìm theo mã hợp đồng hoặc tên khách hàng

#	Mã hợp đồng	Tên khách hàng	Chọn
1	HD0001	Đinh Quang Trung	<input checked="" type="button" value="Chọn"/>
2	HD0002	Đinh Quang Trung	<input checked="" type="button" value="Chọn"/>
3	HD0019	Đinh Quang Trung	<input checked="" type="button" value="Chọn"/>
4	HD0003	Trần Quang Đại	<input checked="" type="button" value="Chọn"/>
5	HD0004	Nguyễn Thị Trang	<input checked="" type="button" value="Chọn"/>
6	HD0005	Cao Văn Trường	<input checked="" type="button" value="Chọn"/>
7	HD000C	Cao Văn Trường	<input checked="" type="button" value="Chọn"/>
8	HD0006	Nguyễn Hữu Cảnh	<input checked="" type="button" value="Chọn"/>
9	HD0007	Lý Phương	<input checked="" type="button" value="Chọn"/>
10	HD0008	Mai Trần Hậu	<input checked="" type="button" value="Chọn"/>

Step	Description
------	-------------

3	Search contract by contract code or customer name
4	Click “Chọn” button

Thông báo tai nạn và yêu cầu bồi thường

Các ô có dấu * là bắt buộc

Lưu ý quan trọng: Người kê khai phải kê khai đầy đủ và trung thực các nội dung dưới đây.
Doanh nghiệp bảo hiểm có thể từ chối một phần số tiền bồi thường nếu nhận được nội dung kê khai thiếu trung thực

5

Mã hợp đồng * HD0001

Ngày gởi yêu cầu *

6

Chọn hình thức nhập * Nhập liệu bằng tay Nhập từ ảnh chụp biên bản

Step	Description
5	Choose “Ngày gởi yêu cầu” the compensation created date (required)
6	Select “Chọn hình thức nhập” from two option (required)

The screenshot shows the MIC Insurance Card application interface. At the top, there is a header with the logo 'Công ty bảo hiểm MIC' and a user profile for 'Nguyễn Chí Kha'. On the left sidebar, under the 'Bồi thường' (Compensation) section, there is a step number '7' with a red arrow pointing to the main form area. The main form is titled 'Thông báo tai nạn và yêu cầu bồi thường' (Report of accident and compensation request). It includes a note about mandatory fields and a summary of the accident details. Step 7 covers filling in driver information and accident details, while step 8 covers clicking the 'Gởi yêu cầu bồi thường' button.

Lưu ý quan trọng: Người kê khai phải kê khai đầy đủ và trung thực các nội dung dưới đây. Doanh nghiệp bảo hiểm có thể từ chối một phần số tiền bồi thường nếu nhận được nội dung kê khai thiếu trung thực

Các ô có dấu * là bắt buộc

Mã hợp đồng * HD0001 Q Chon Ngày gởi yêu cầu * 11/08/2015

Chọn hình thức nhập *

- Nhập liệu bằng tay
- Nhập từ ảnh chụp biển bản

Họ tên lái xe *	Đinh Quang Trung	Điện thoại *	123123123
Địa chỉ *	Phường Tân Chánh Hiệp, Q 12, TP HCM		
Giấy phép lái xe số *	Ví dụ: 740118000357	Hạng *	Ví dụ: A1
Biển số xe gây tai nạn *	Ví dụ: 78Y9-15383	Trọng tải/số chỗ ngồi *	Ví dụ: 7 (tần/chỗ)
Ngày xảy ra tai nạn *	11/08/2015		
Nơi xảy ra tai nạn *	Ví dụ: 405, Hoàng Văn Thụ, Quận Tân Bình, TP.HCM		
CQCA giải quyết *	Ví dụ: Đội CSGT Bình Triệu		
Diễn biến tai nạn *			
Thiệt hại về người *			
Thiệt hại về tài sản *			
Người làm chứng *	Ví dụ: Đặng Thu Thảo		
Địa chỉ liên hệ *	Ví dụ: 73B, Nguyễn Trãi, Quận 5, TP.HCM		
Yêu cầu bồi thường			

8 → Gởi yêu cầu bồi thường

Step	Description
7	Fill driver name, phone, address, license number, license type, vehicle plate, seat capacity, accident date, accident place, control department, description, human damage, asset damage, observer name, observer address (required) Fill compensation request note (optional)
8	Click "Gởi yêu cầu bồi thường" button

2.1.3.11. Staff view compensation detail

#	Mã yêu cầu	Ngày yêu cầu	Mã hợp đồng	Tên Khách hàng	Trạng thái
1	BT0002	25/07/2015	HD000B	Truong Văn Khải	Chưa giải quyết
2	BT0001	26/07/2015	HD000B	Truong Văn Khải	Đã giải quyết

Step	Description
1	Click on view compensation detail link

2.1.3.12. Staff compensation cancel request

Step	Description
1	Click “Giải quyết yêu cầu” button

Giải quyết yêu cầu

Ngày giải quyết * 11/08/2015 **2**

Ghi chú **3**

4 Từ chối bồi thường Đồng ý bồi thường

Mã hợp đồng HD000B
Tên khách hàng Trương Văn Khải

Thông tin yêu cầu bồi thường

Ngày gửi yêu cầu ngày 25 tháng 07 năm 2015
Họ tên lái xe Trương Văn Khải Điện thoại 0984283338
Địa chỉ liên hệ Phường 1, Q. 9, TP HCM

Chưa giải quyết

Step	Description
2	Choose “Ngày giải quyết” the compensation resolved date (required)
3	Input “Ghi chú” the compensation resolve note (optional)
4	Select one from two option by click “Từ chối bồi thường” or “Đồng ý bồi thường” button

2.1.4. Admin

2.1.4.1. Add staff by admin

The screenshot shows a web application window titled "Giao diện quản lý" (Management Interface) with the URL "localhost:8080/admin/staff?action=viewCreateStaff". The main content area is titled "Thêm nhân viên mới" (Add new staff). It contains four input fields: "Email *", "Mật khẩu *", "Tên nhân viên *", and "Số điện thoại *". A note at the top right says "Các ô có dấu * là bắt buộc" (Fields with an asterisk are required). Below the fields is a green button labeled "+ Thêm nhân viên". The sidebar on the left has links for "Nhân viên", "Thiết lập", and "Loại hợp đồng". The footer contains the MIC logo, company address, contact information (+84 37337009, hydrangea8604@gmail.com), and a copyright notice (MIC Group 2 © 2015).

Step	Description
1	Fill in fields: “Email”: Staff’s email “Mật khẩu”: Password “Tên nhân viên”: Staff’s name “Số điện thoại”: Phone number
2	Click “Thêm nhân viên” button.

2.1.4.2.Add contract type:

The screenshot shows a web-based application for managing insurance contracts. The title bar indicates it's running on localhost:8080/admin/contractType. The main header is 'Quản lý loại hợp đồng'. On the left sidebar, there are three menu items: 'Nhân viên', 'Thiết lập', and 'Loại hợp đồng', with 'Loại hợp đồng' being the active tab. The main content area has a sub-header 'Thêm loại hợp đồng mới'. It contains three input fields: 'Tên loại hợp đồng *' (Name of contract type), 'Miêu tả *' (Description), and 'Phí hằng năm (VND) *' (Annual fee). Below these fields is a green 'Thêm' (Add) button. Underneath the button is a section titled 'Lịch sử' (History) which displays a table of existing contract types. The table columns are '#', 'Tên' (Name), 'Miêu tả' (Description), 'Phí hằng năm' (Annual fee), and 'Tình trạng' (Status). The data in the table is as follows:

#	Tên	Miêu tả	Phí hằng năm	Tình trạng
1	Xe trên 50cc có BH cho người trên xe		86.000 VNĐ	Đang hoạt động Ngừng hoạt động
2	Xe trên 50cc không có BH cho người trên xe		66.000 VNĐ	Đang hoạt động Ngừng hoạt động
3	Xe từ 50cc trở xuống có BH cho người trên xe		80.500 VNĐ	Đang hoạt động Ngừng hoạt động
4	Xe từ 50cc trở xuống có BH cho người trên xe		60.500 VNĐ	Đang hoạt động Ngừng hoạt động
5	Xe mô tô ba bánh, xe gắn máy và các loại xe tương tự		319.000 VNĐ	Đang hoạt động Ngừng hoạt động

Step	Description
1	Fill in fields: “Tên loại hợp đồng”: Contract type's name “Miêu tả”: Description “Phí hằng năm”: Price per year
2	Click “Thêm” button.

2.1.4.3.Edit business rules

The screenshot shows a web application interface for managing business rules. The left sidebar has links for 'Nhân viên', 'Thiết lập' (selected), and 'Loại hợp đồng'. The main content area is titled 'Thiết lập cấu hình'.

Các giới hạn ngày theo chính sách của công ty

Thời hạn thanh toán hợp đồng mới:	7	ngày
Thời gian thông báo hợp đồng sắp hết hạn lần 1: Trước	15	ngày
Thời gian thông báo hợp đồng sắp hết hạn lần 2: Trước	7	ngày
Thời gian thông báo hợp đồng sắp hết hạn lần 3: Trước	3	ngày
Thời hạn mặc định của hợp đồng:	12	tháng
Thời hạn tối thiểu của hợp đồng:	1	tháng
Thời hạn tối đa còn lại để gia hạn hợp đồng:	60	ngày

Giới hạn chỉnh sửa ngày cho nhân viên

Thời gian thêm thông tin sau khi hợp đồng bị hủy/hết hạn:	30	ngày
Ngày bắt đầu hợp đồng trước ngày hiện tại:	7	ngày
Ngày bắt đầu hợp đồng sau ngày hiện tại:	1825	ngày
Ngày thanh toán hợp đồng trước ngày hiện tại:	7	ngày
Ngày thanh toán hợp đồng sau ngày hiện tại:	30	ngày
Ngày huỷ hợp đồng trước ngày hiện tại:	7	ngày
Ngày huỷ hợp đồng sau ngày hiện tại:	30	ngày

Các loại chi phí

Phí làm thẻ mới:	50000	VND
Phí vận chuyển thẻ:	10000	VND

Cập nhật

Step	Description
1	Change the business rule values.
2	Click “Cập nhật” button.

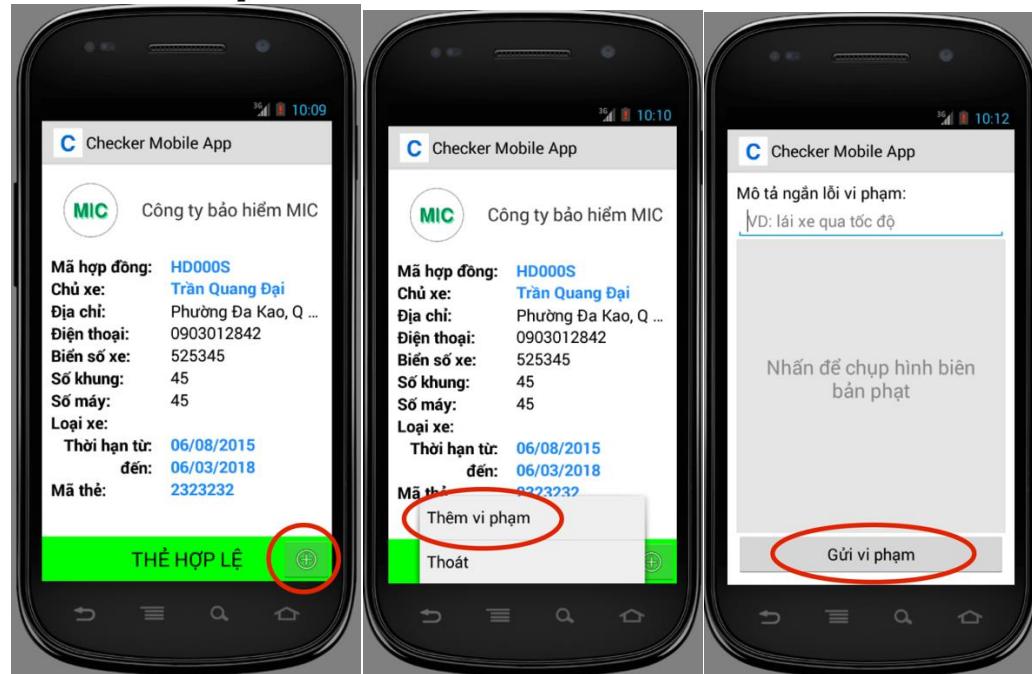
2.2. Checker Application

2.2.1. Validate card



Step	Description
1	Put Card close to the phone
2	Check information from the output screen

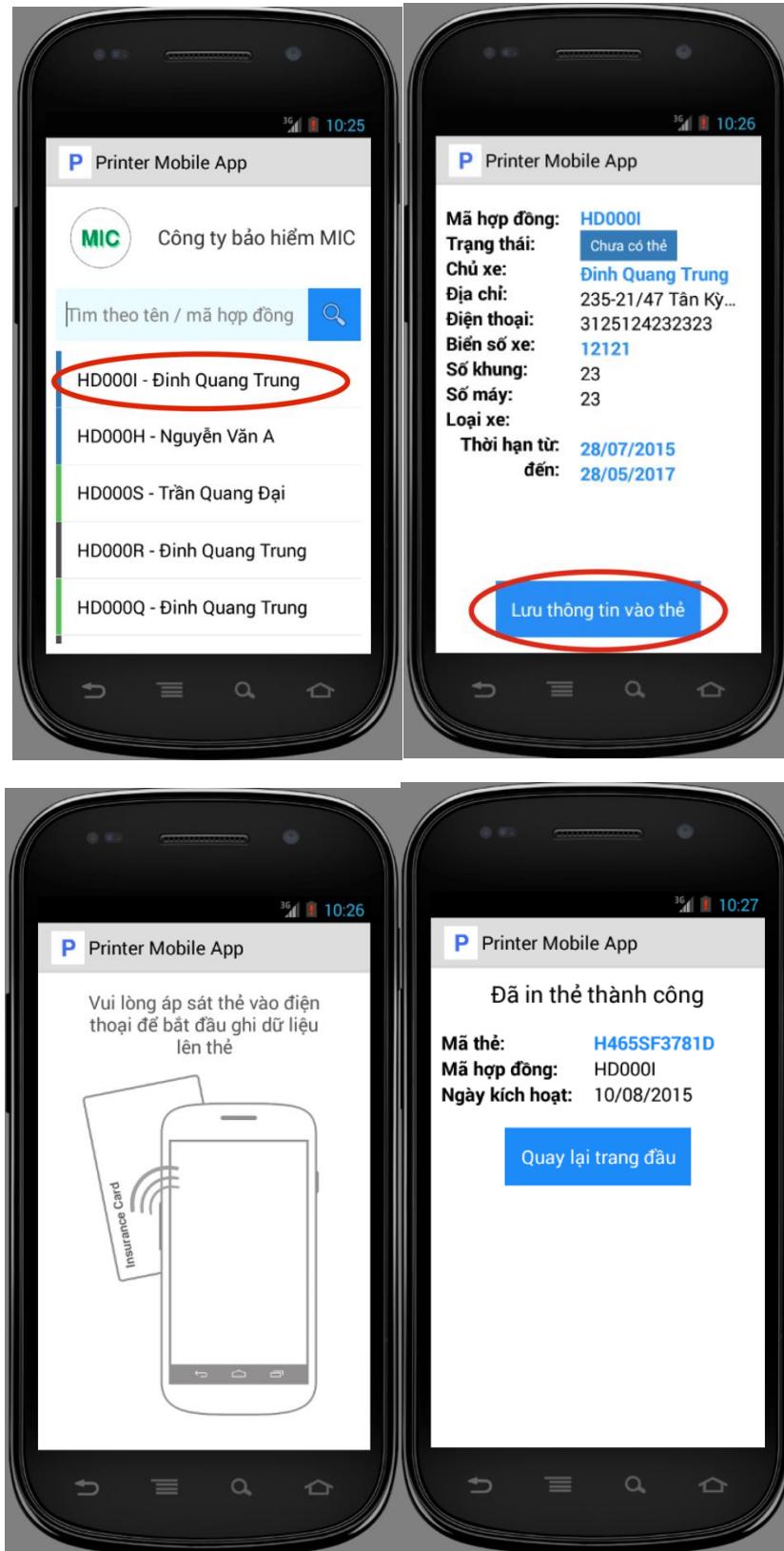
2.2.2. Add punishment information



Step	Description
1	At output screen, tap the plus button at bottom right corner.
2	Tap “Thêm vi phạm” button, application open new screen to enter punishment information
3	Press “Gửi vi phạm” to send punishment information

2.3. Printer Application

2.3.1. Print card



Step	Description
1	At home screen, select a contract from list.
2	Tap “Lưu thông tin vào thẻ” button
3	Put NFC card close to phone, application show print result

G. Appendix

1. SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.
2. Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999.
3. Android Developer Guide - Application Fundamentals
<http://developer.android.com/guide/components/fundamentals.html>
4. NFC ForumTM Type 3 Tag Operation Specification 1.1, 2011-06-28.
5. Circular No. 126/2008/TT-BTC: On rules, terms, table of premium rates and levels of liability of compulsory insurance for civil liability of motor vehicle owners - The Ministry of Finance - Hanoi, December 22, 2008.
<http://thuvienphapluat.vn/van-ban/Bao-hiem/Thong-tu-126-2008-TT-BTC-quy-tac-dieu-khoan-bieu-phi-muc-trach-nhiem-bao-hiem-bat-buoc-dan-su-chu-xe-co-gioi-83284.aspx>