# Software Design Description

## 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:
  + Section 2: gives an overall description of the system architecture design.
  + Section 3: gives component diagrams that describe the connection and integration of the system.
  + Section 4: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
  + Section 5: describe a fully attributed Entity Relationship Diagram.

## System Architecture Design

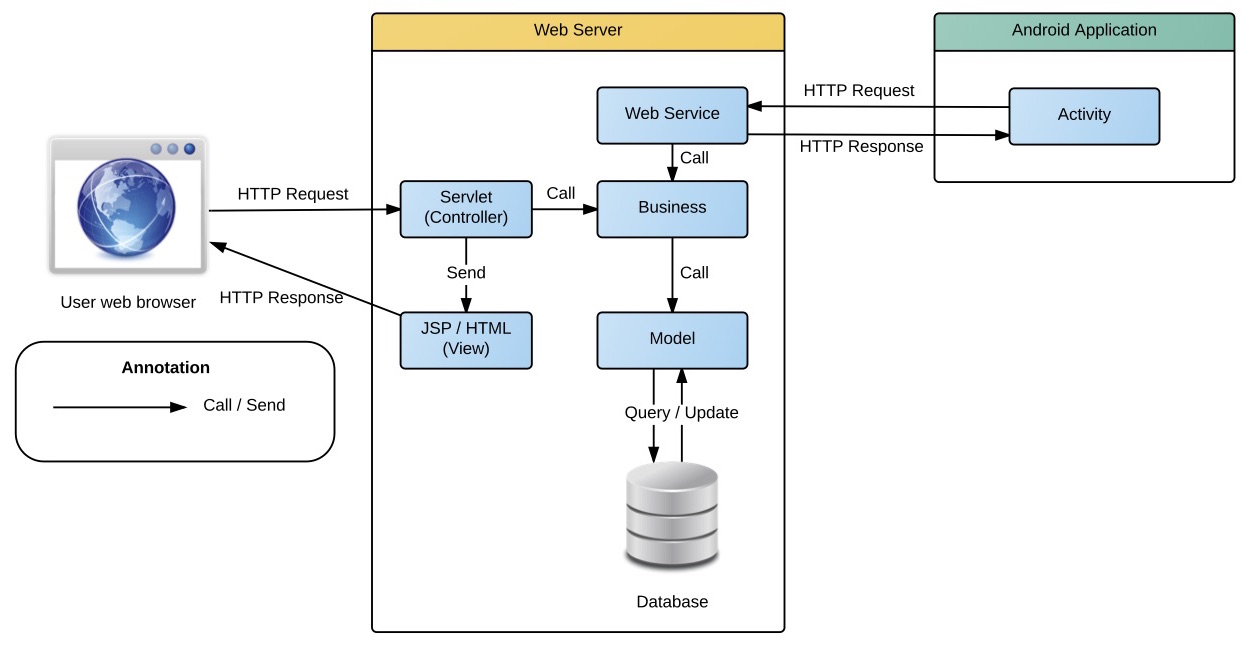


Figure 1 System architecture design

### 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 Heath insurance or Asset insurance…
* In scope of 4-member team this, 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 follow 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.

### 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 follow Android application architecture with following components:

* **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 ...

## Component Diagram

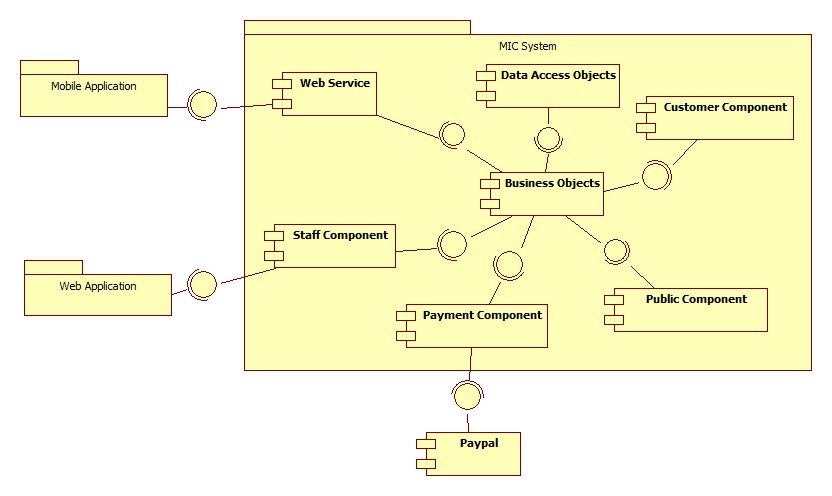


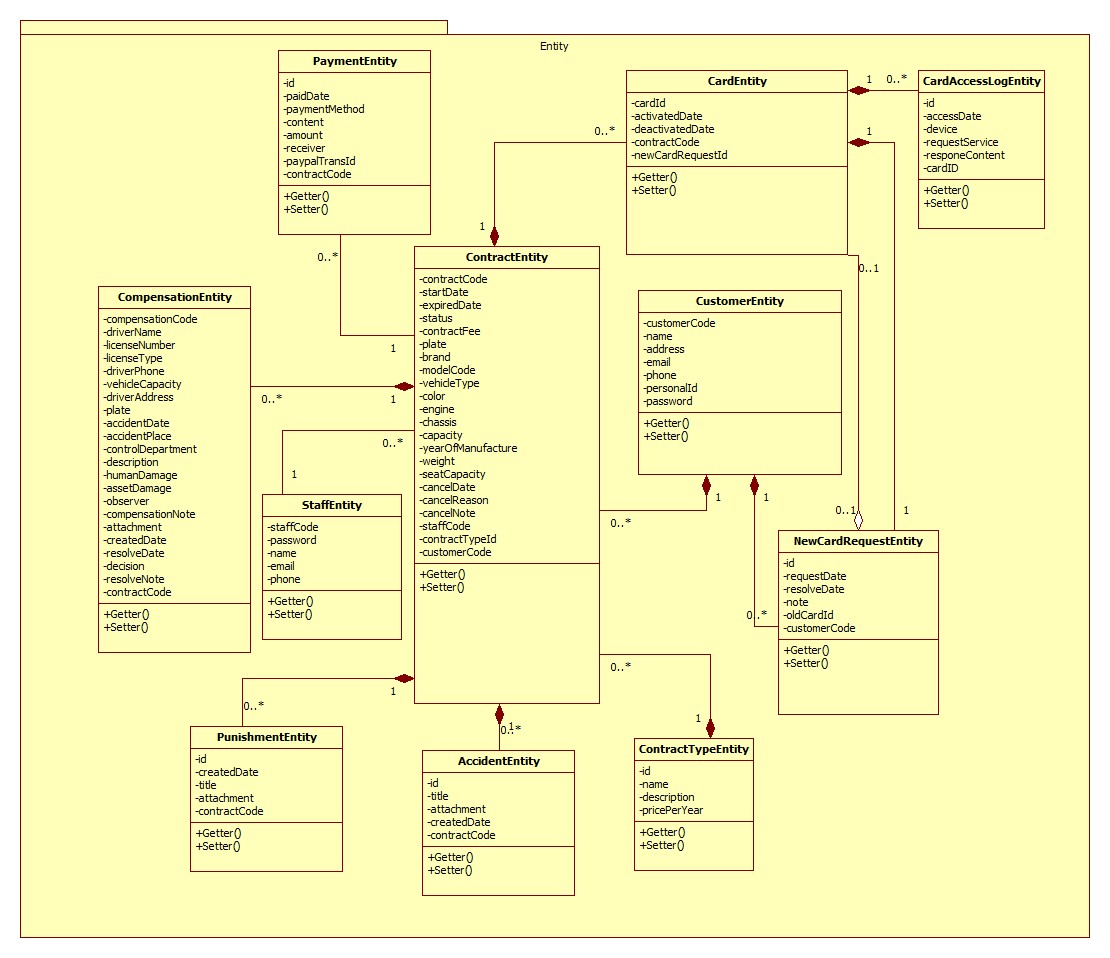
Figure 2 Component Diagram

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| Web Application | Web application package: View, Controller |
| Mobile Application | Mobile application package |
| PayPal | Handle payment process with PayPal API |
| Web Service | operations o handle interaction between the system and database.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 |
| 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 1 Component Dictionary

## Detailed Description

### 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. |
| 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. |

### Class Diagram Explanation

#### 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. |

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

#### CardEntity

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| cardId | String | Private | Unique identifier of an card |
| 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. |

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

#### 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. |

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

#### 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. |

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

#### 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. |

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

#### 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. |

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

#### 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. |

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

#### 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. |

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

#### 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. |

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

#### 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. |

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

#### 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. |

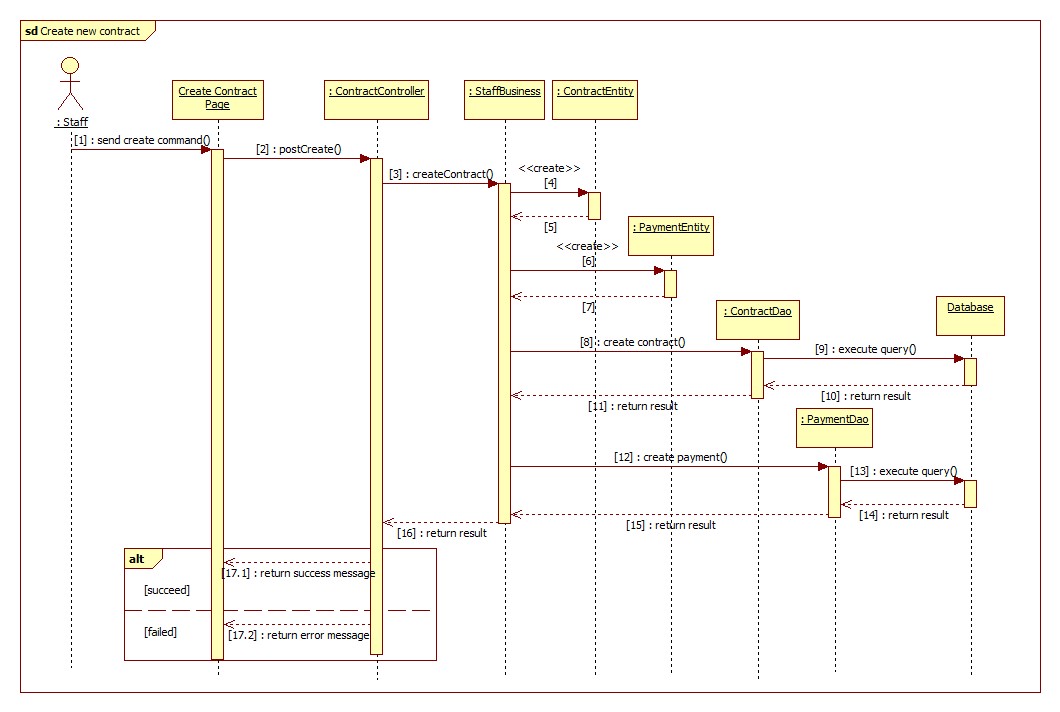
|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value. |
| Setter | Void | Public | Set value of attribute. |

### Interactive Diagram

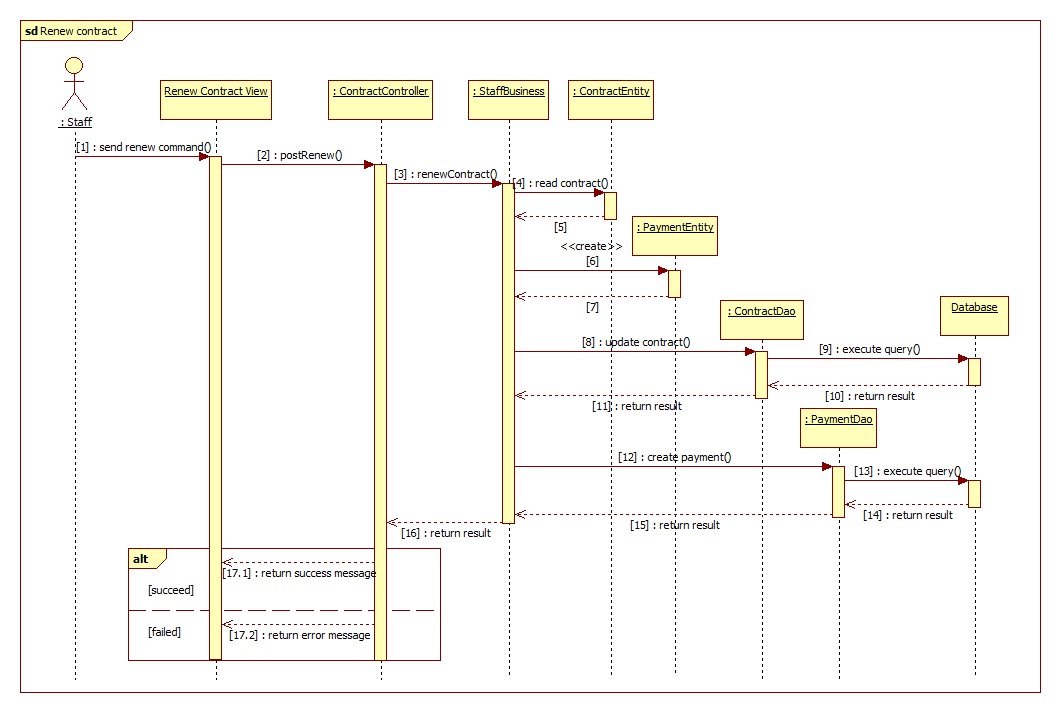
#### Web Application

##### Staff

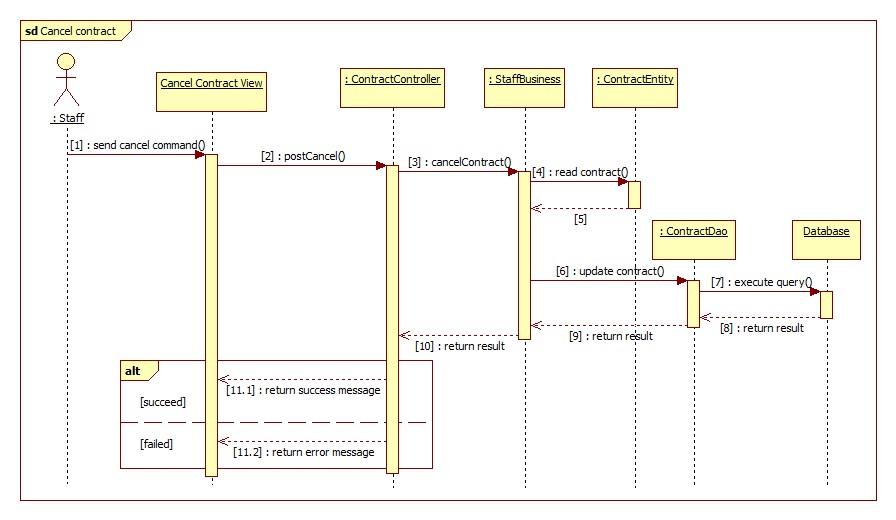
###### Create new contract



###### Renew contract

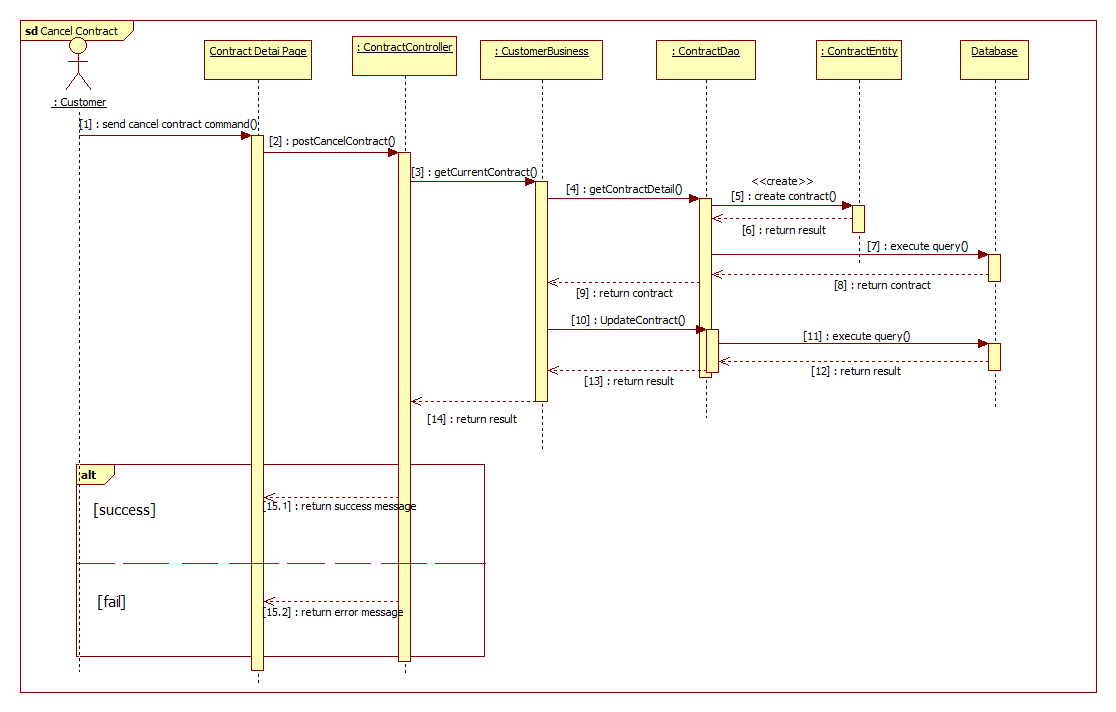


###### Cancel contract

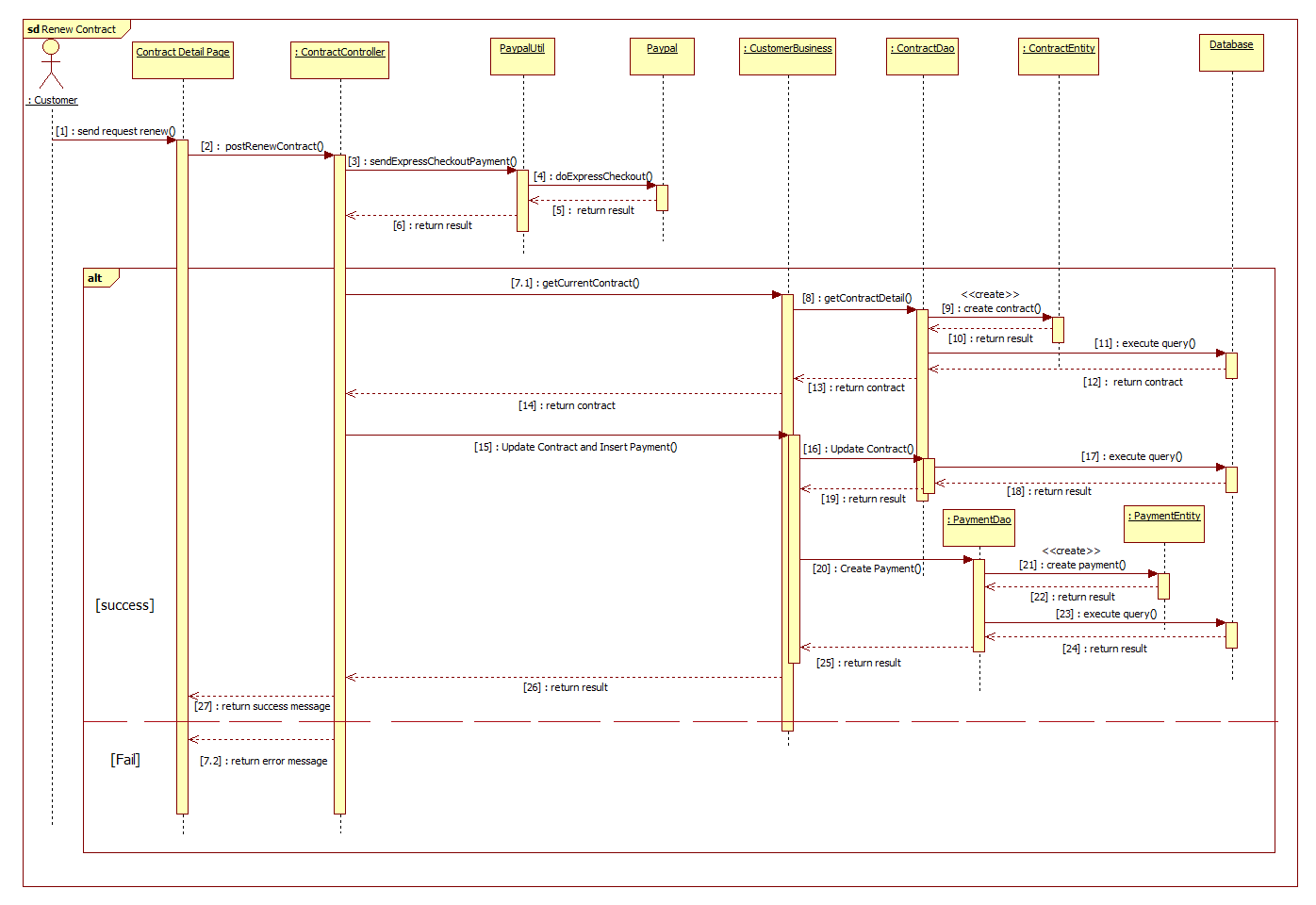


##### Customer

###### Cancel contract

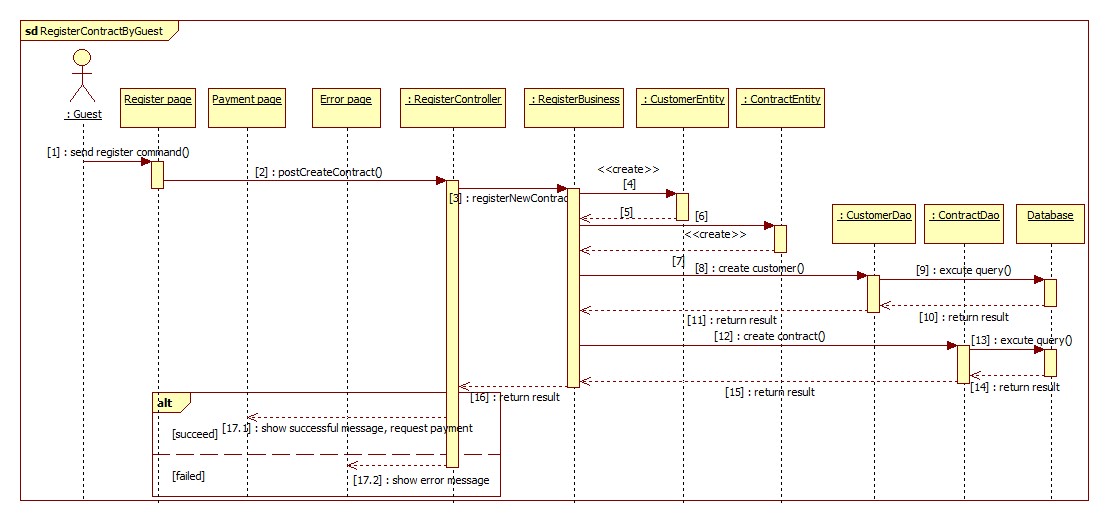


###### Renew contract

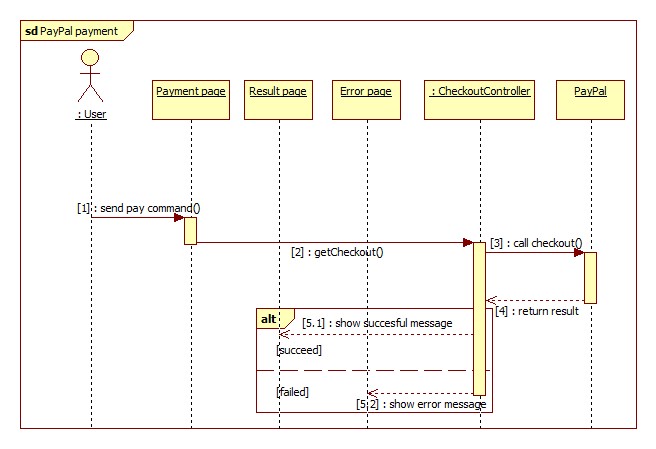


##### Guest

###### Register new contract



###### PayPal payment



#### Mobile Application

##### Checker Mobile Application

###### <Police> Verify card validation

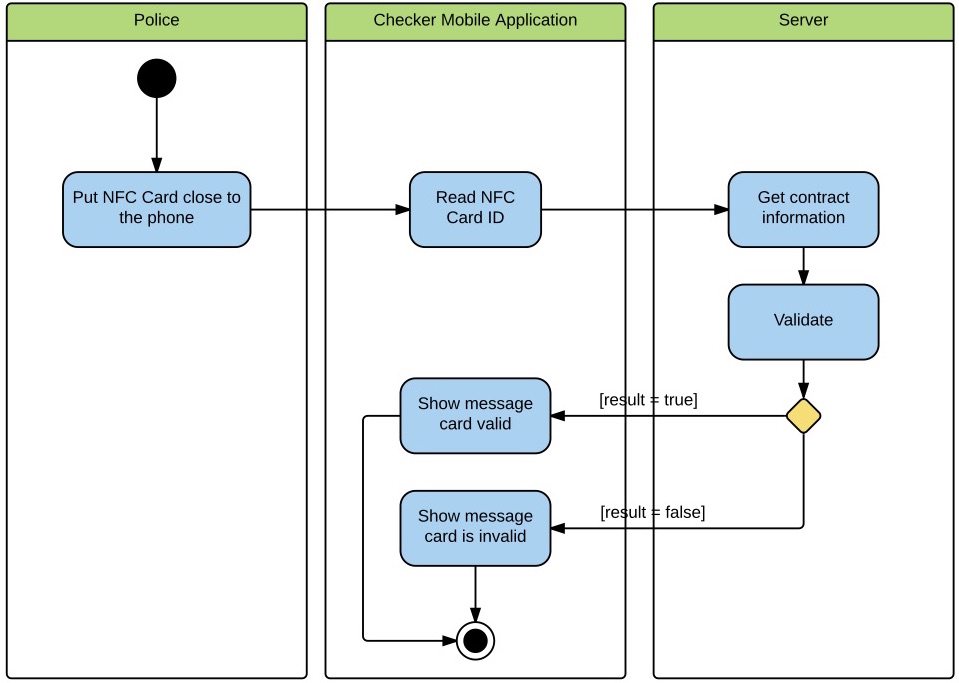


Figure 3 <Police> Verify card validation

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

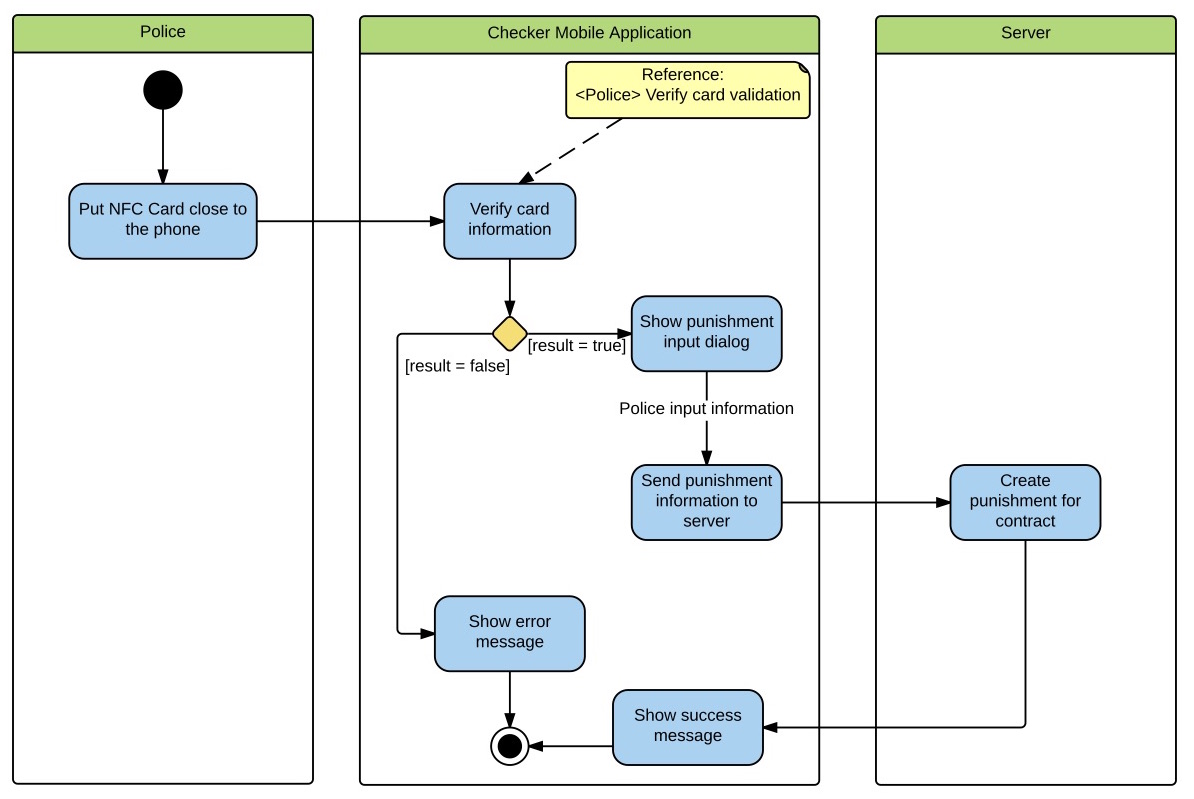


Figure 4 <Police> Add punishment information

##### Printer Mobile Application

###### <Staff> Search contract

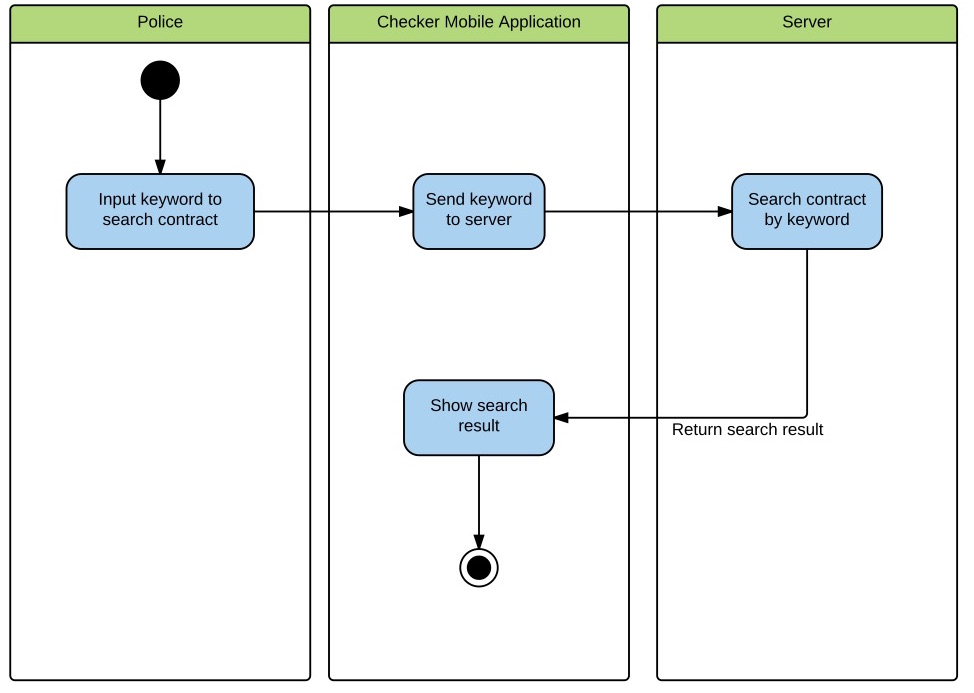


Figure 5 <Staff> Search contract

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

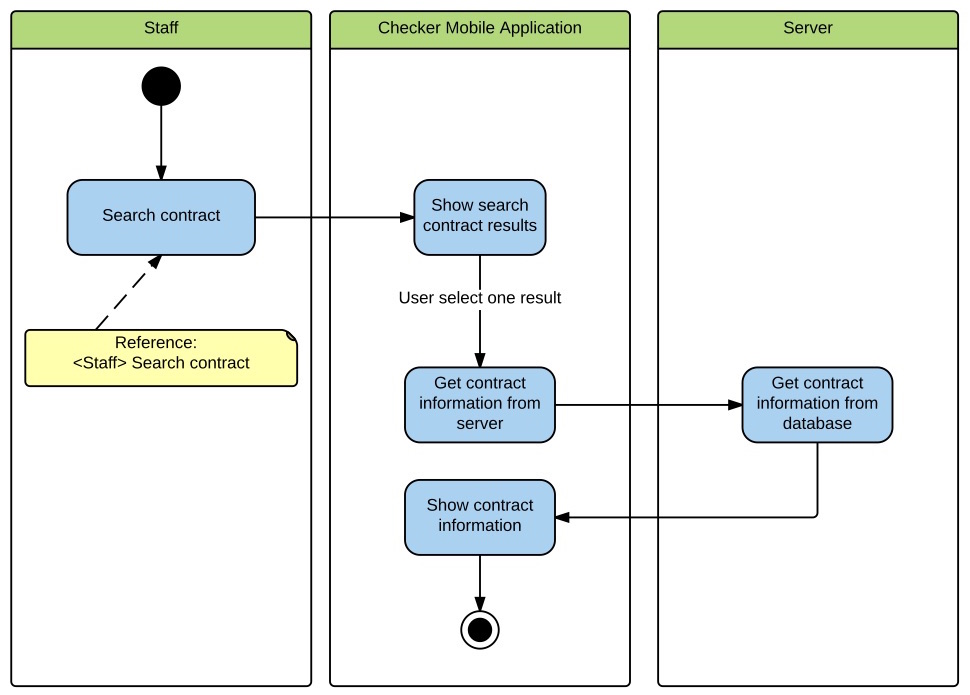


Figure 6 <Staff> View contract information

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

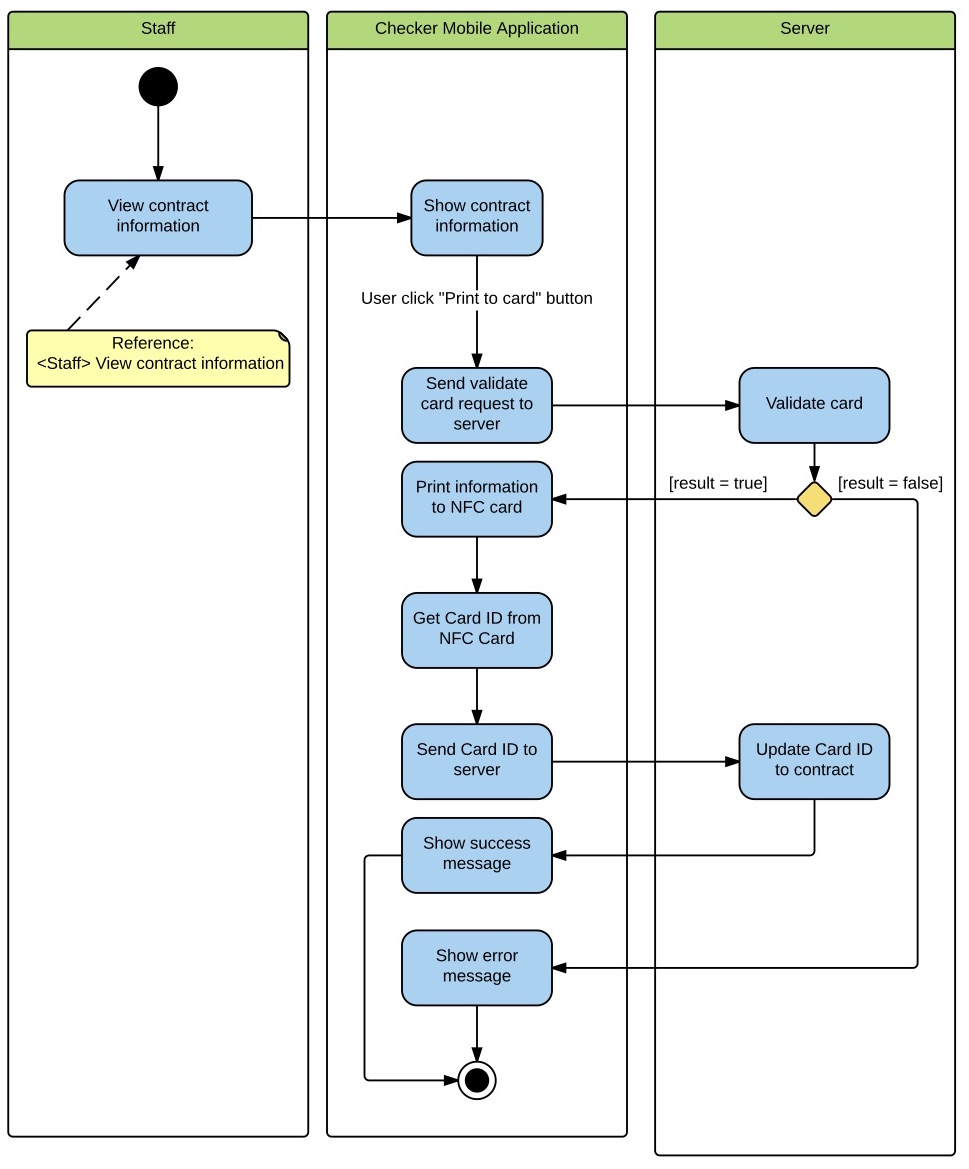


Figure 7 <Staff> Print information to NFC card

## User Interface Design

### Web application Design

#### Staff Interface Design

##### 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 |

**Button/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 |

##### 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 |

**Button/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 |

##### Create 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 | 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 |

**Button/Hyperlinks**

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

##### 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 |

**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 |

##### Create 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 | 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 | txtModel code | 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 |

**Button/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 |

##### Renew 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 | Start date | Contract’s start date | Yes | Yes | Label | N/A | N/A |
| 3 | txtExpiredDate | Select expired date | No | Yes | Date input form | N/A | N/A |
| 4 | txtAmount | Renew contract fee | Yes | Yes | Label | N/A | N/A |
| 5 | txtPaidDate | Select paid contract fee date | No | Yes | Date input form | N/A | N/A |

**Button/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 |

##### Cancel contract



**Fields**

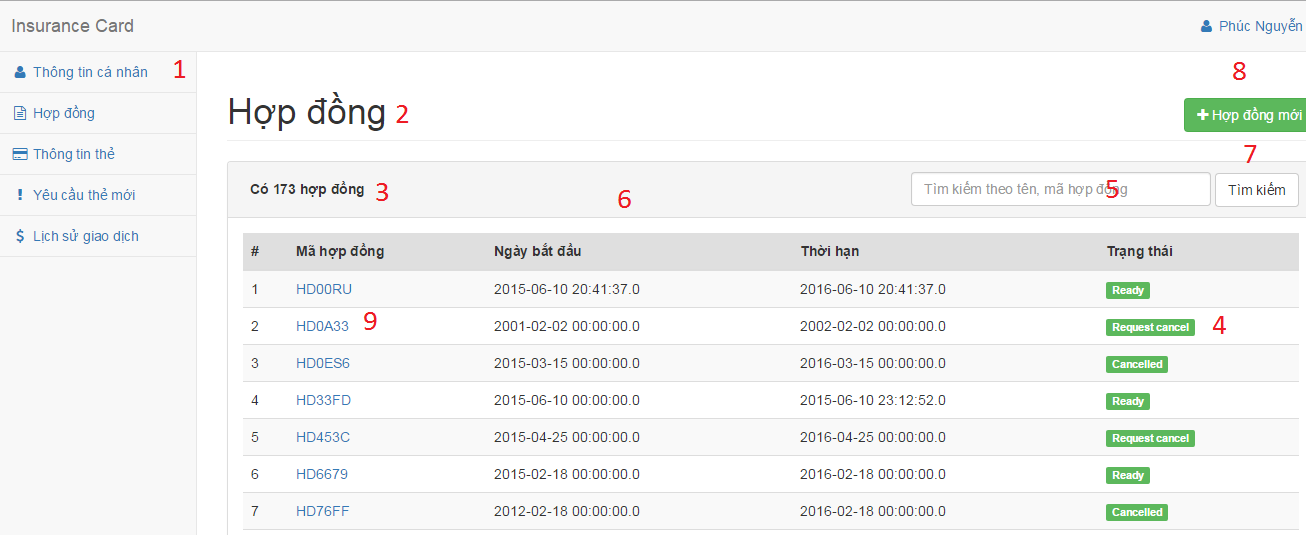
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Title | Title of the modal | Yes | Yes | Label | N/A | N/A |
| 2 | txtCancelDate | Select cancel contract date | No | Yes | Date 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 |

**Button/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 |

#### Customer Interface Design

##### Manage contract



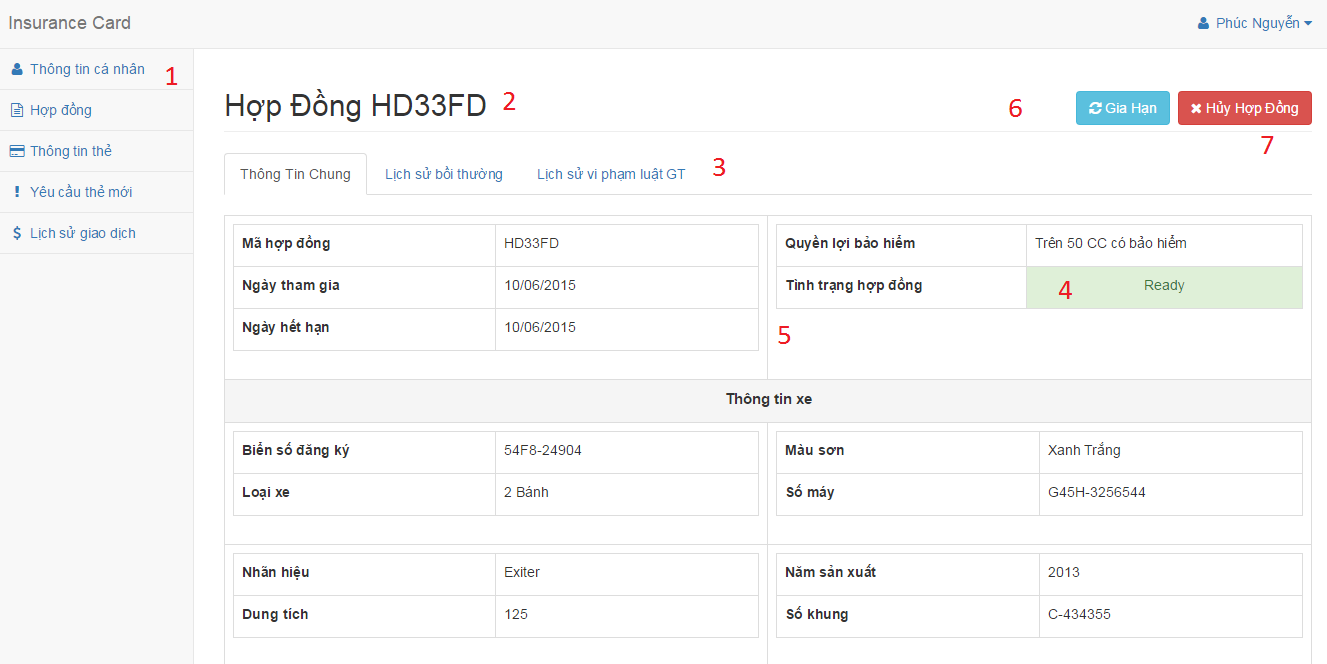
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 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 |

**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 |

##### Contract detail



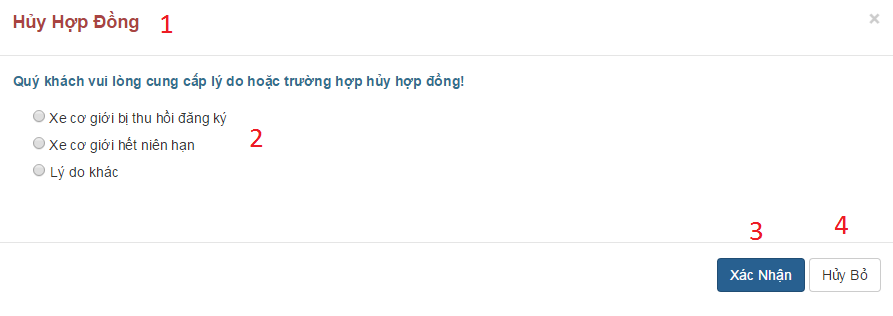
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

**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 |

##### Cancel contract



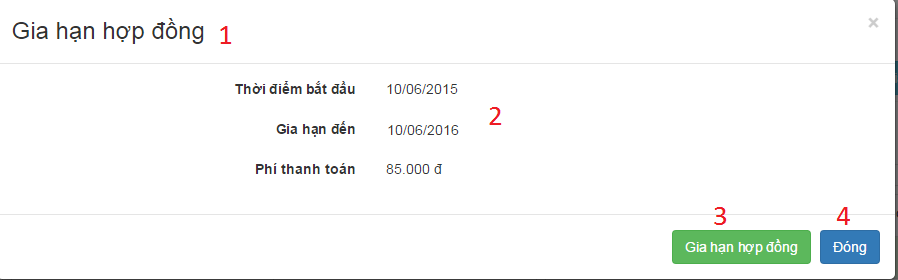
**Fields**

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

**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 |

##### Renew contract



**Select paypment method**

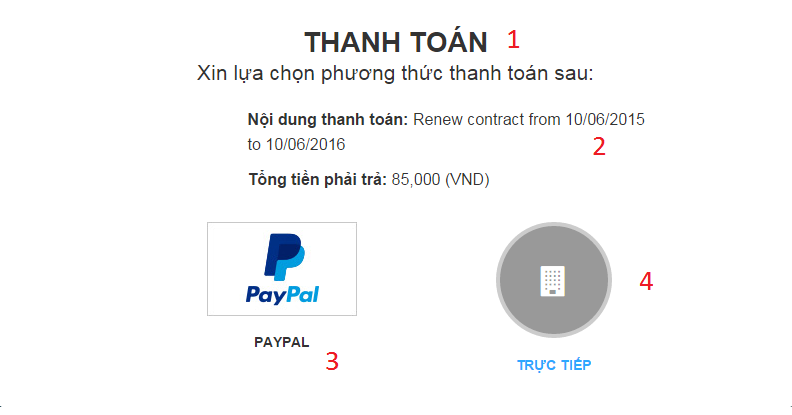
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

**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 |

##### 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 |

**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 |

**Select paypment method**

### Checker Mobile Application Design

#### Scan NFC card screen

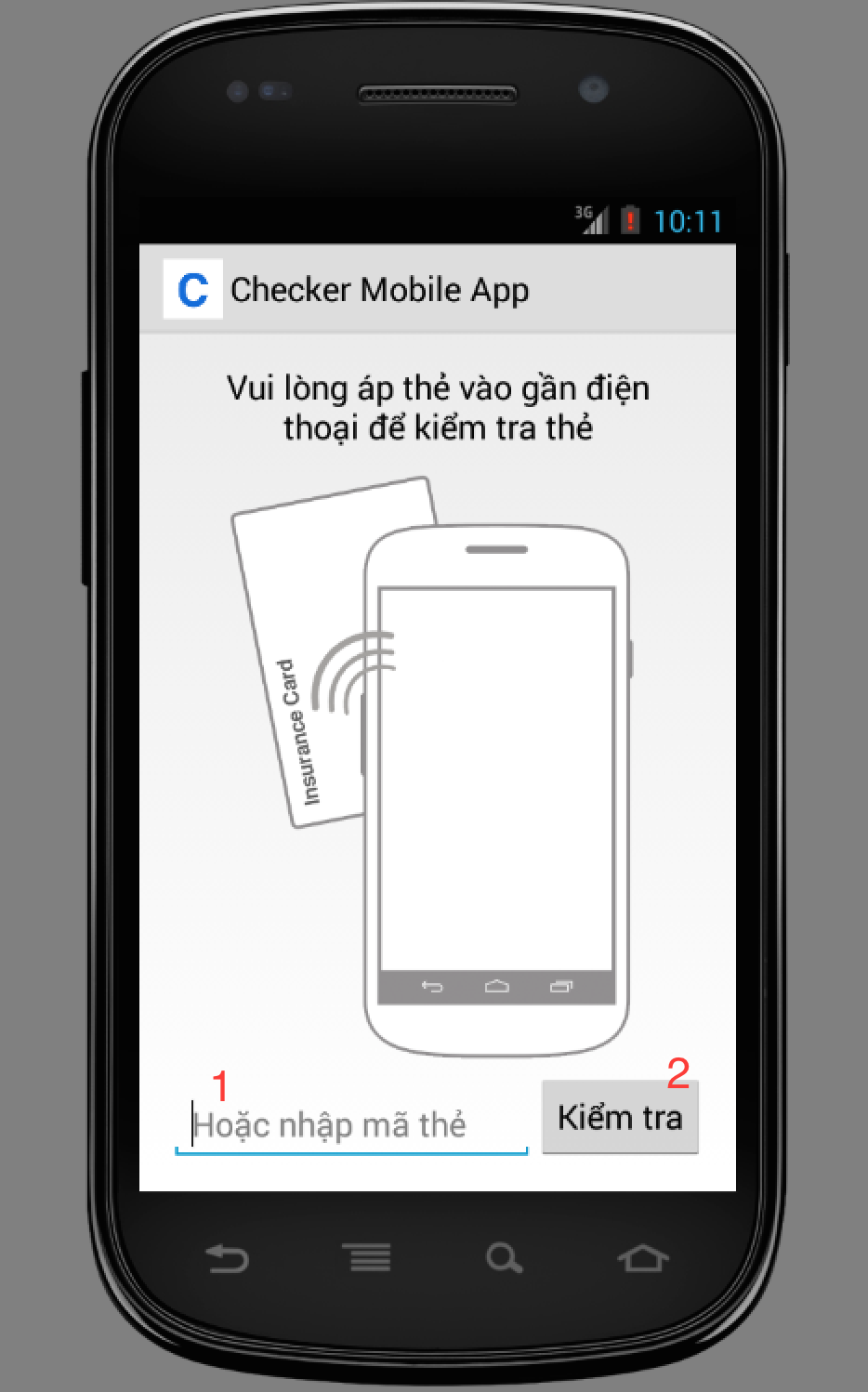


Figure 8 Scan NFC card screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | txtCardIDnpen print card screen supporto view detail information of that contractCard | ID of the card | No | No | Textbox | String | 20 |

Table 2 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 3 Scan NFC card screen - Buttons

#### View card information screen

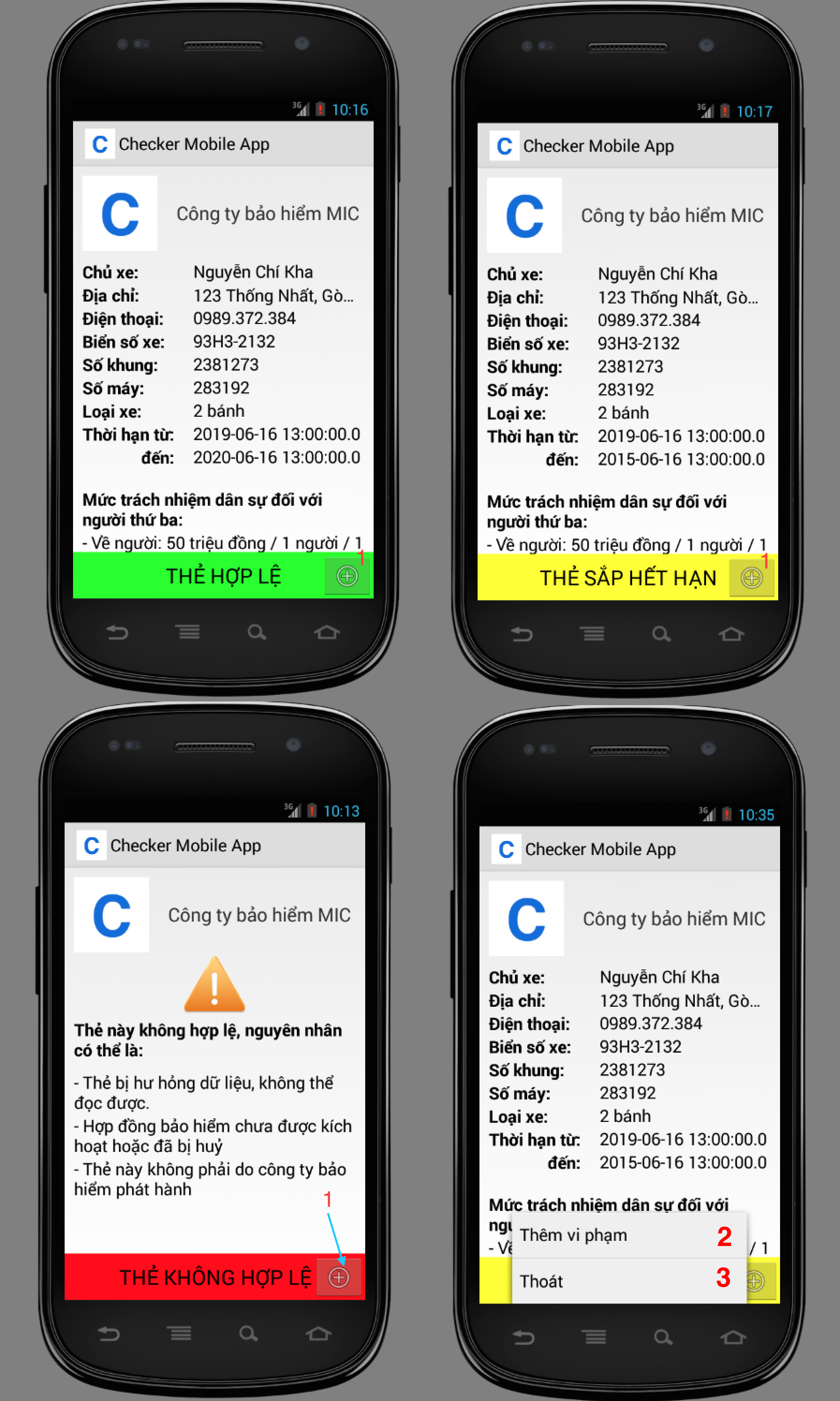


Figure 9 View card information screen

**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 4 View card information screen - Buttons

#### Add punishment screen

### Printer Mobile Application Design

#### Search contract screen

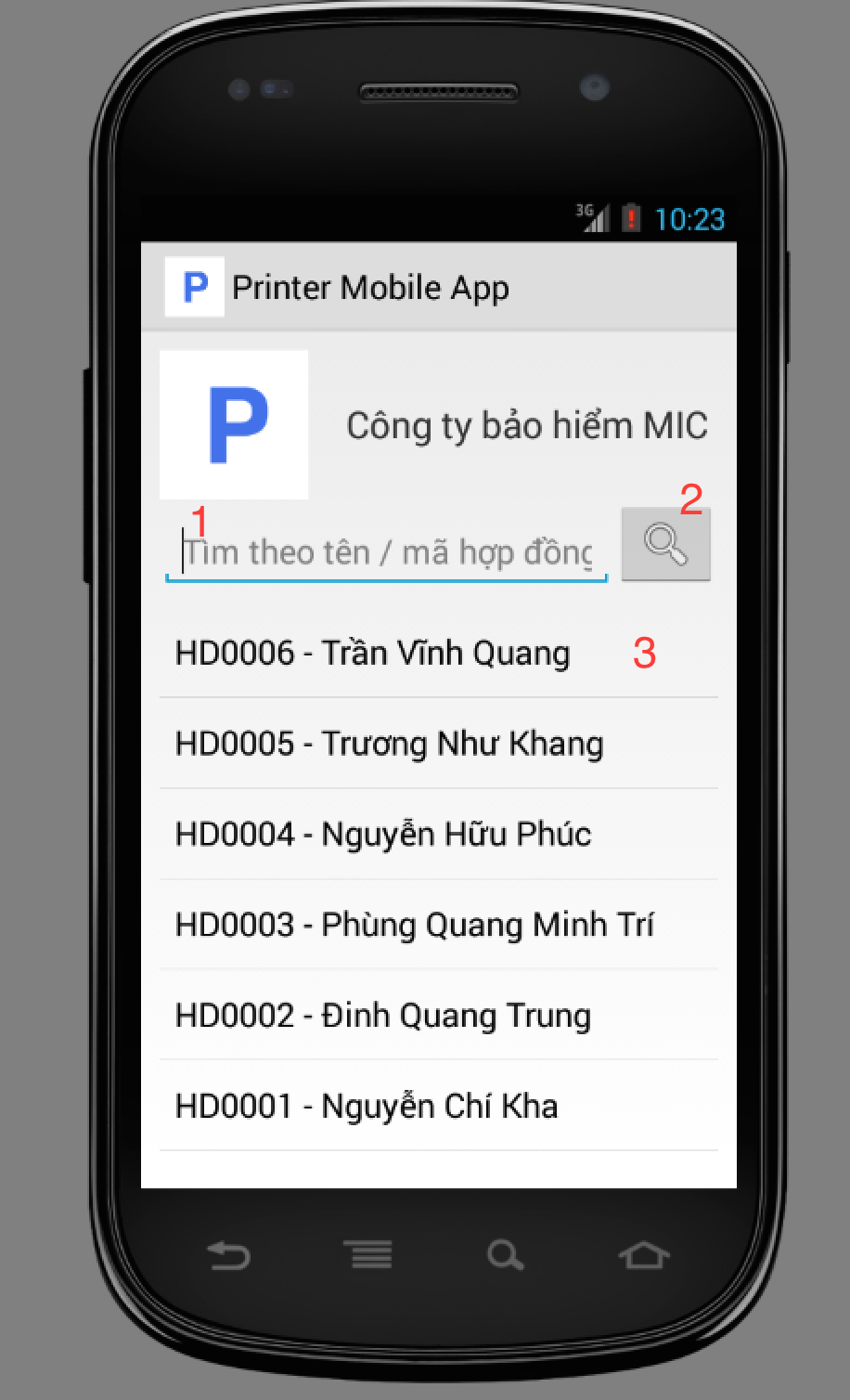


Figure 10 Search contract screen

**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 5 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 6 Search contract screen - Buttons

#### View contract screen



Figure 11 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 7 View npen print card screen supporto view detail information of that contractcontract screen - Buttons

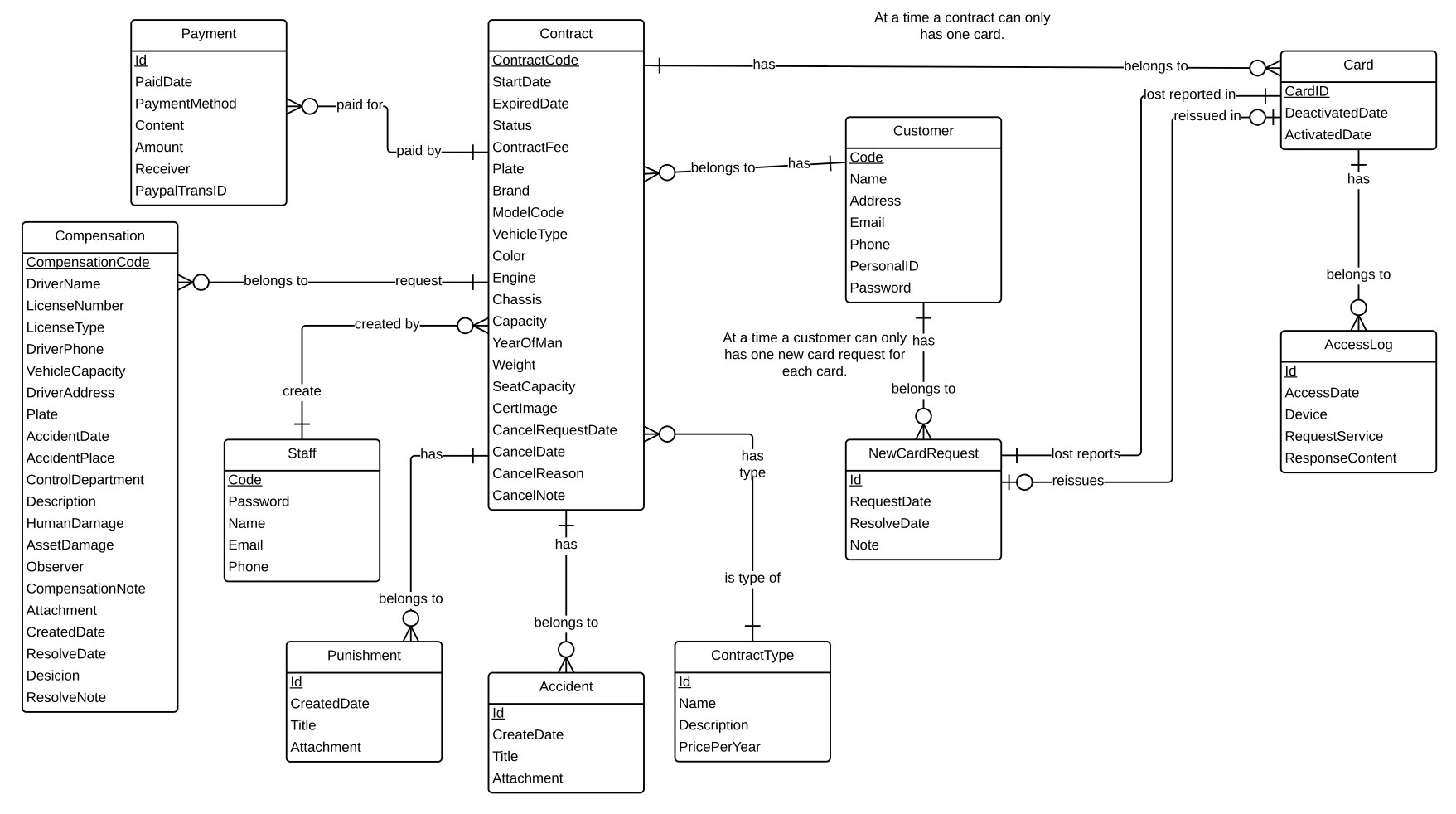
#### Print card screen



Figure 12 Print card screen

## Database Design

### Entity relationship diagram



### Entity Dictionary

|  |  |
| --- | --- |
| Entity Data Dictionary: describe content of all entities | |
| Entity name | **Description** |
| Customer | Contain the customer information. |
| Contract | Contain the contract information. |
| Card | Contain the card information. |
| 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. |

## Algorithms

### 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.

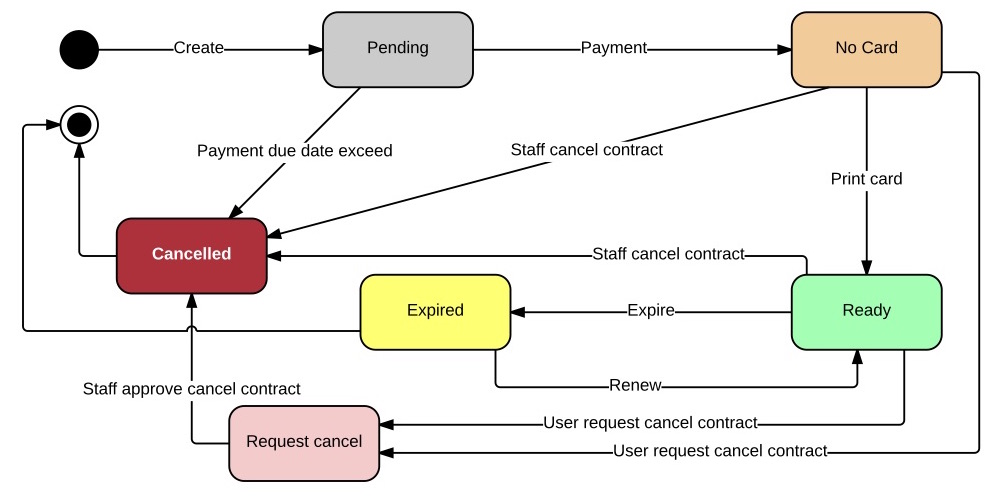


Figure 13 Contract State Chart

|  |  |
| --- | --- |
| SStimulusion﷽﷽﷽﷽﷽> ERDquence numberecution bar em and database.tate | Description |
| Pending | The contract is created and do not have payment |
| No card | The contract had have payment 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 8 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 | 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 | Rancelledancelcancel request from customer via web application .eady | 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 cancelledancelcancel request from customer via web application .ancel contract | Staff approve cancel request from customer via web application | Request cancel | Cancelled |

Table 9 Contract State Flow

### System Scheduler Process

#### 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.

#### 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 dates is exceed… etc.

#### Solution

We create a Cronjob in Unix operating system that run on the same server of the Web application, Cronjob will automatically run at specific time that system administrator define when deploy the system. The Cronjob 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:
   1. Contract status
   2. Contract expired date
   3. Contract renew due date
   4. Contract payment due date
   5. Compensation requests
   6. New card requests
   7. Cancel request
3. Update the contract state follow the business rules.

#### Flow chart

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

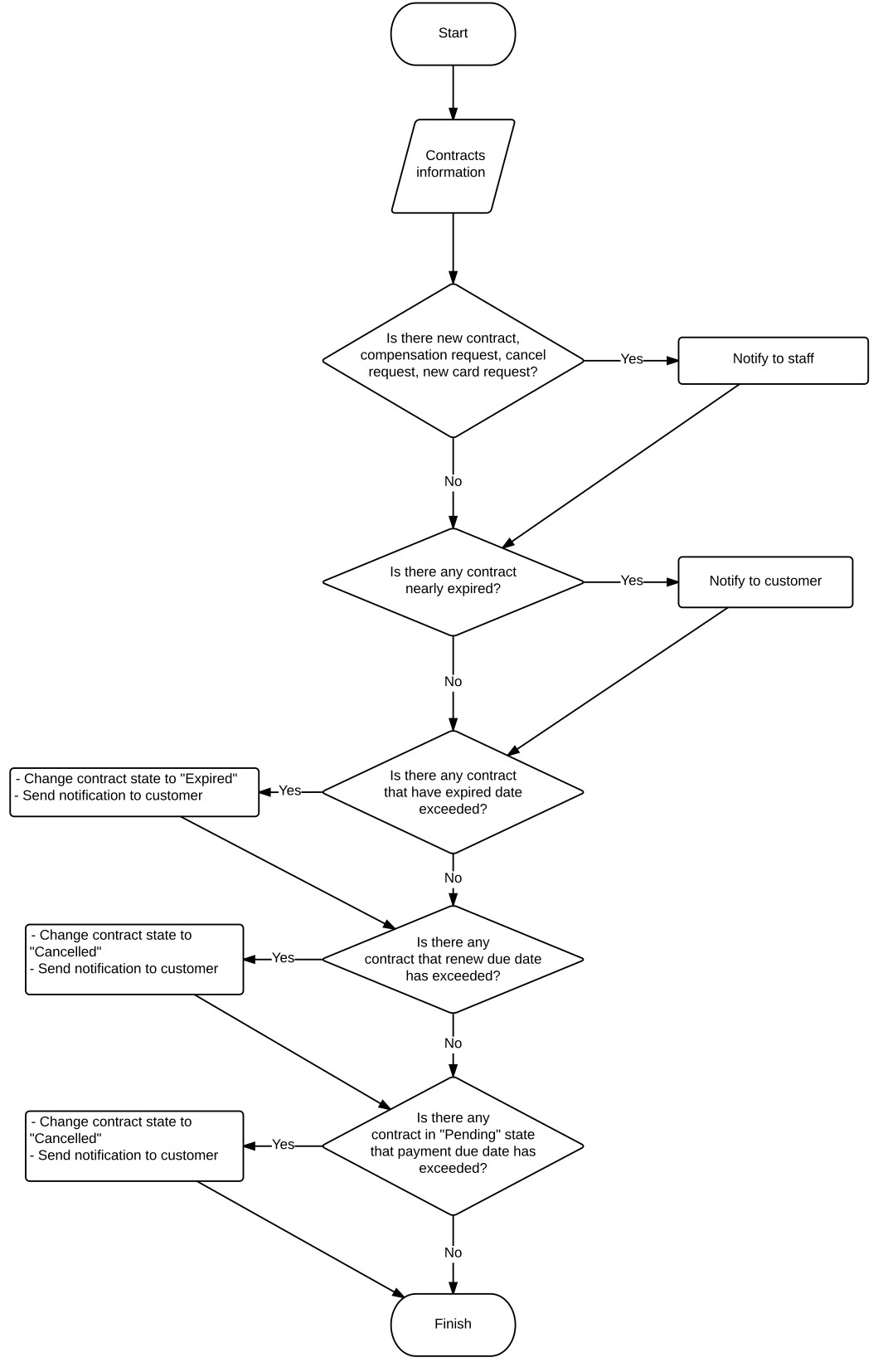


Figure 14 System Scheduler Process

### NFC Card Data Format

#### Definition

According to [Android Developer Forum](https://developer.android.com/guide/topics/connectivity/nfc/index.html) 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.

#### 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 ForumTM NDEF 1.0 Specification (2006-07-24)*, NDEF Messages is stored in tag in this structure:

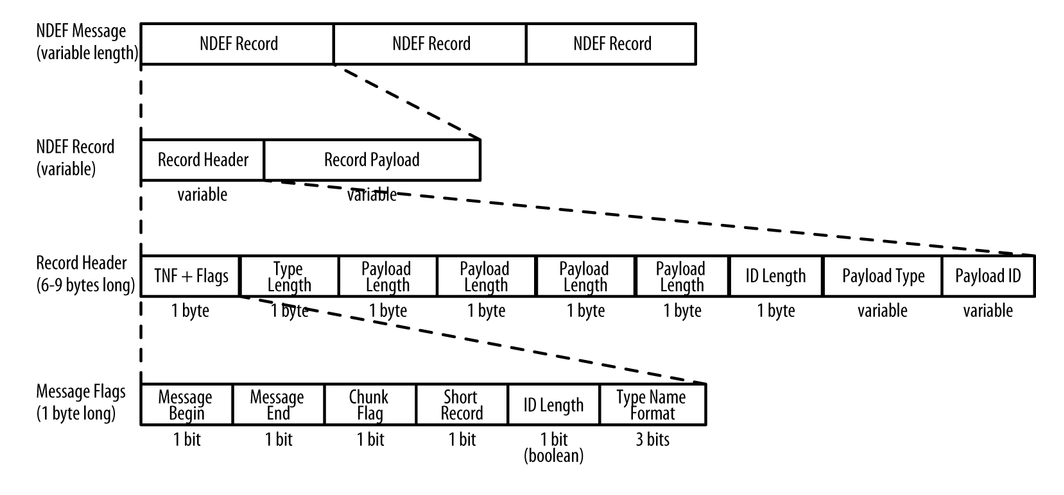


Figure 15 NDEF message structure

**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 10 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.

#### 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 ForumTM 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.

**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 exists 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