# 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

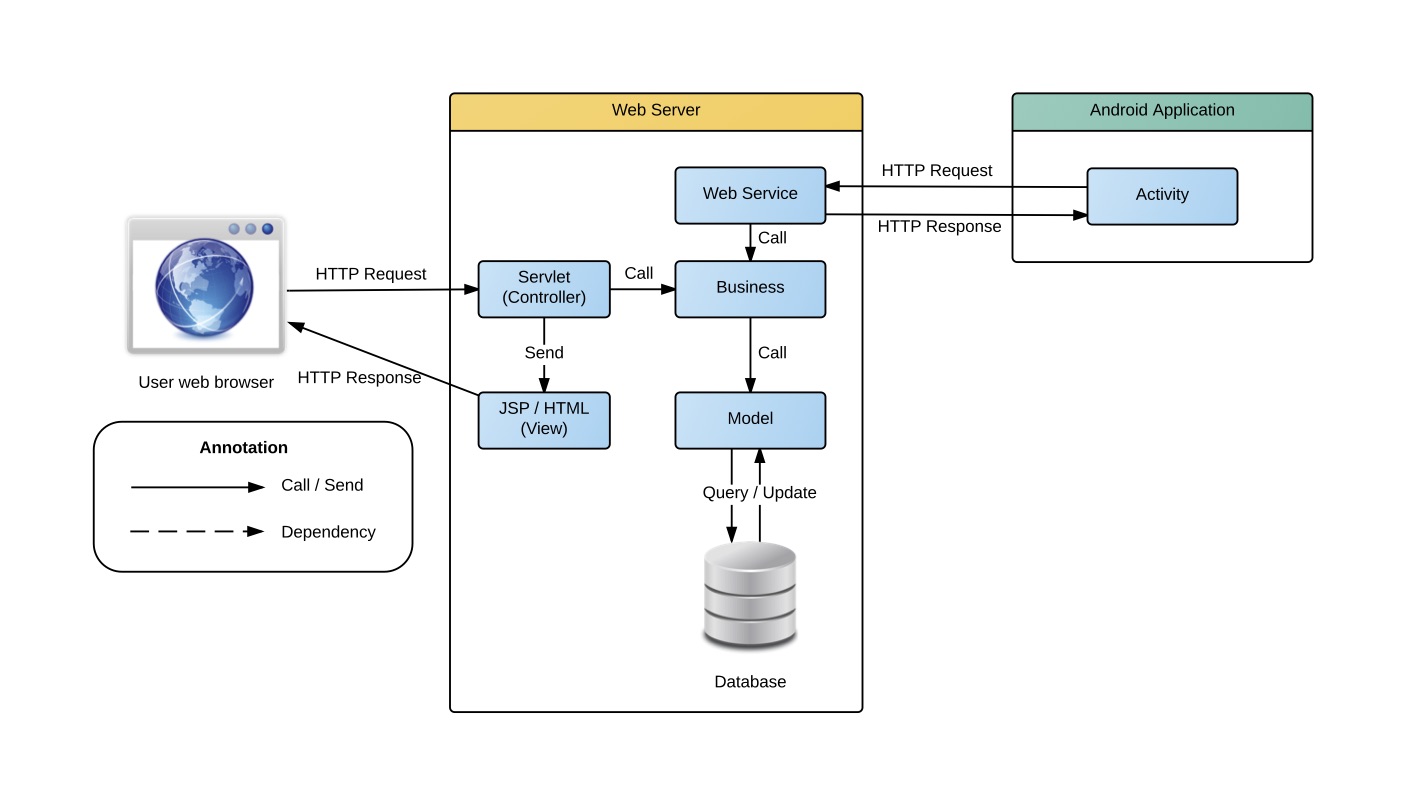


Figure 1 System architecture design

### Web Application architecture description

In this Web Application, the system is developed under J2EE MVC architecture style.

* **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.

* **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

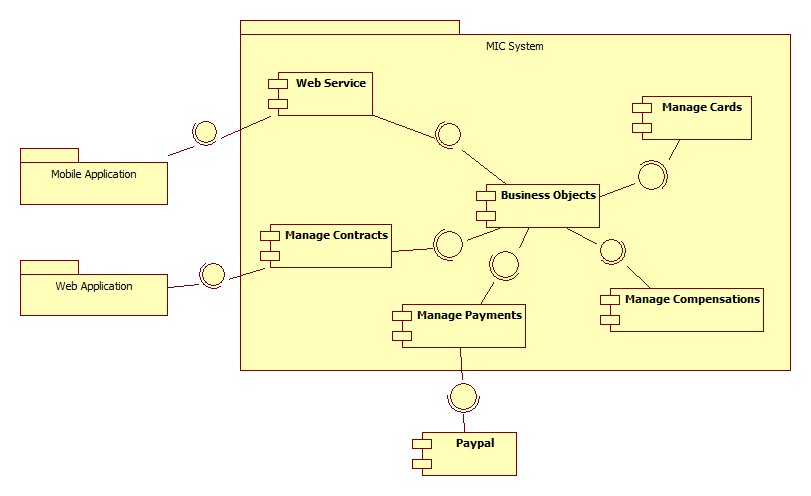


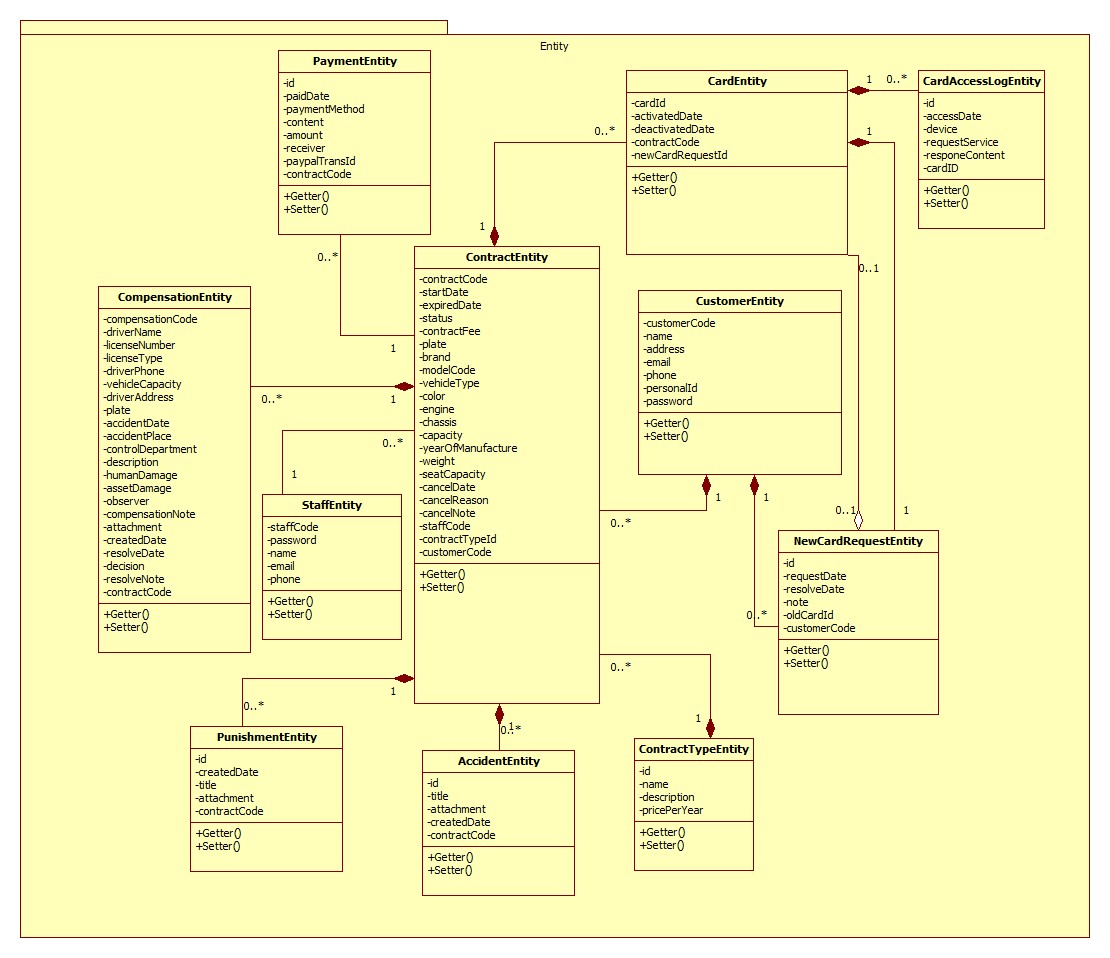
Figure 2 Component Diagram

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| Web Application | Web application package: View, Controller |
| Mobile Application | Mobile application package |
| Web Service | Includes all API controller of the system |
| Manage Contracts | Business logic to manage contracts |
| Manage Cards | Business logic to manage cards |
| Manage Compensations | Business logic to manage compensations |
| Manage Payments | Business logic to manage payments |
| PayPal | Handle payment process with PayPal API |

Table 1 Component Dictionary

## Detailed Description

### Class Diagram



|  |  |
| --- | --- |
| **Class dictionary: describe Class** | |
| **Class Name** | **Description** |
| PaymentEntity | Contain the payment informations. |
| CardEntity | Contain the card informations. |
| CustomerEntity | Contain the customer informations. |
| ContractEntity | Contain the contract informations. |
| StaffEntity | Contain the staff informations. |
| CompensationEntity | Contain the compensation informations. |
| PunishmentEntity | Contain the punishment informations. |
| AccidentEntity | Contain the accident informations. |
| ContractTypeEntity | Contain the contract type informations. |
| NewCardRequestEntity | Contain the new card request informations. |
| CardAccessLogEntity | Contain the card access log informations. |

### 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. |

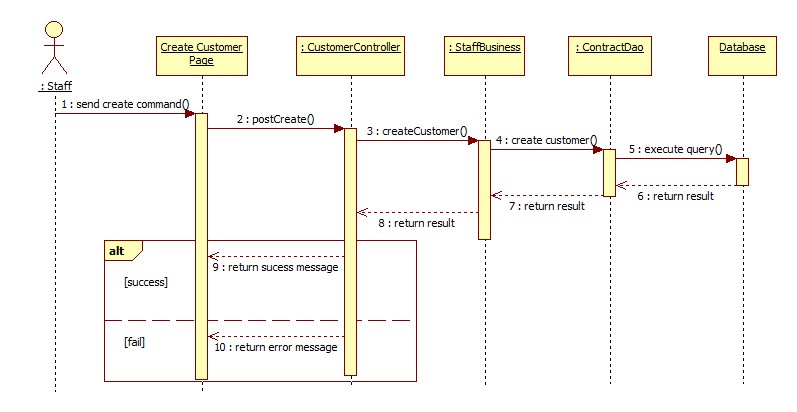
|  |  |  |  |
| --- | --- | --- | --- |
| Method | Return Type | Visibility | Description |
| Getter | Attribute type | Public | Get attribute value. |
| Setter | Void | Public | Set value of attribute. |

### Interactive Diagram

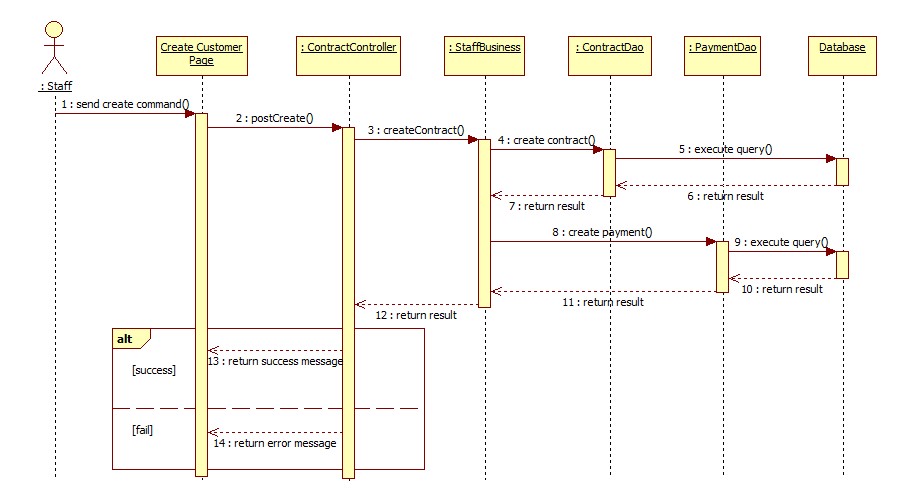
#### Web Application

##### Staff

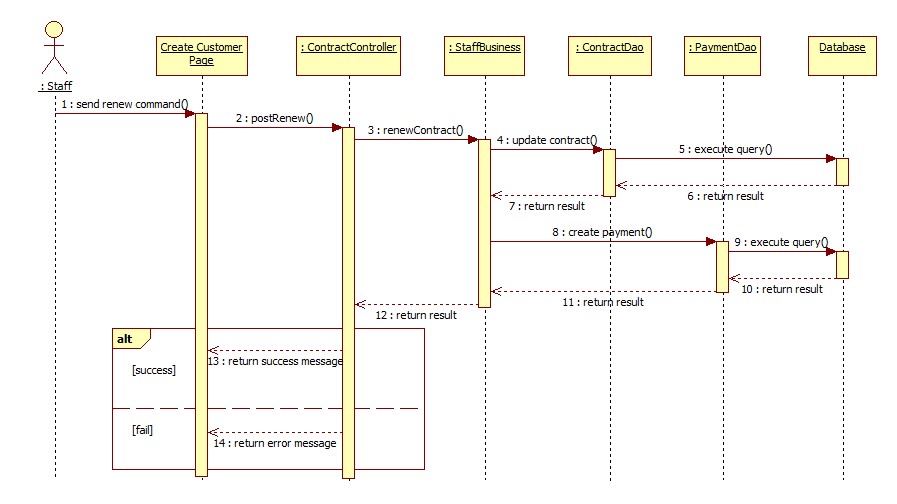
###### Create customer



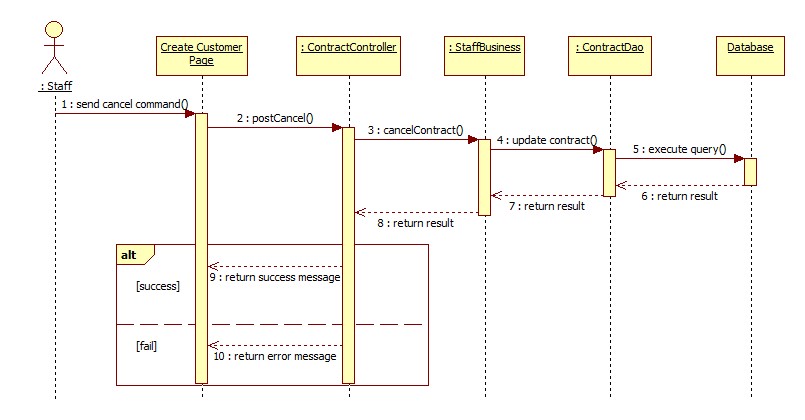
###### Create new contract



###### Renew contract

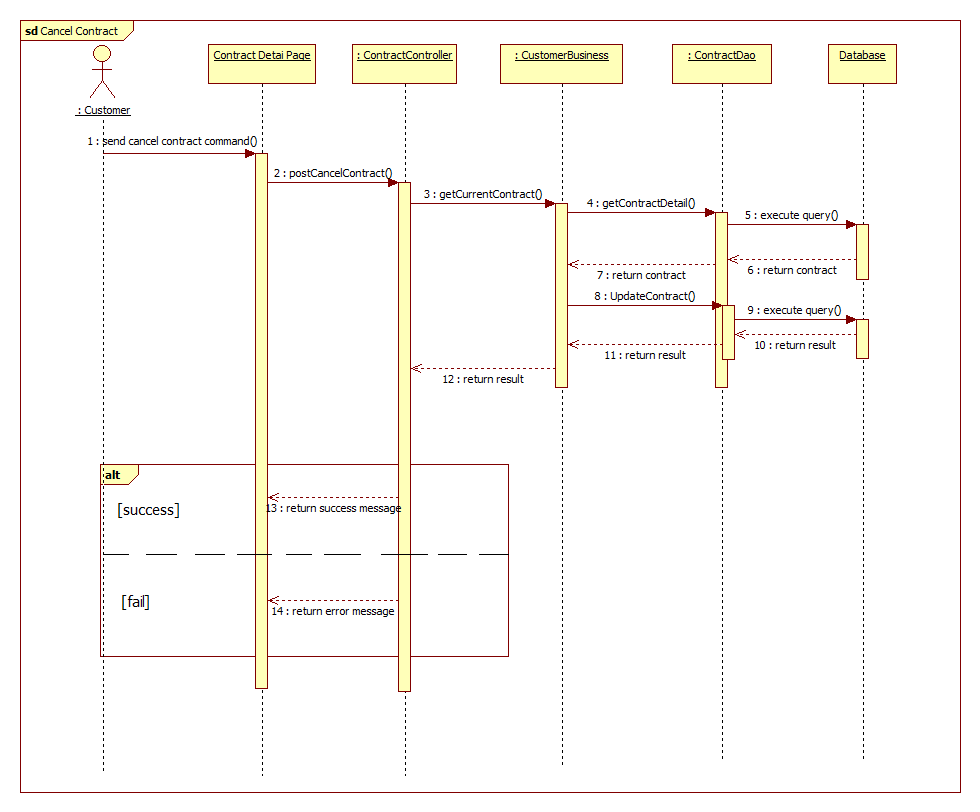


###### Cancel contract

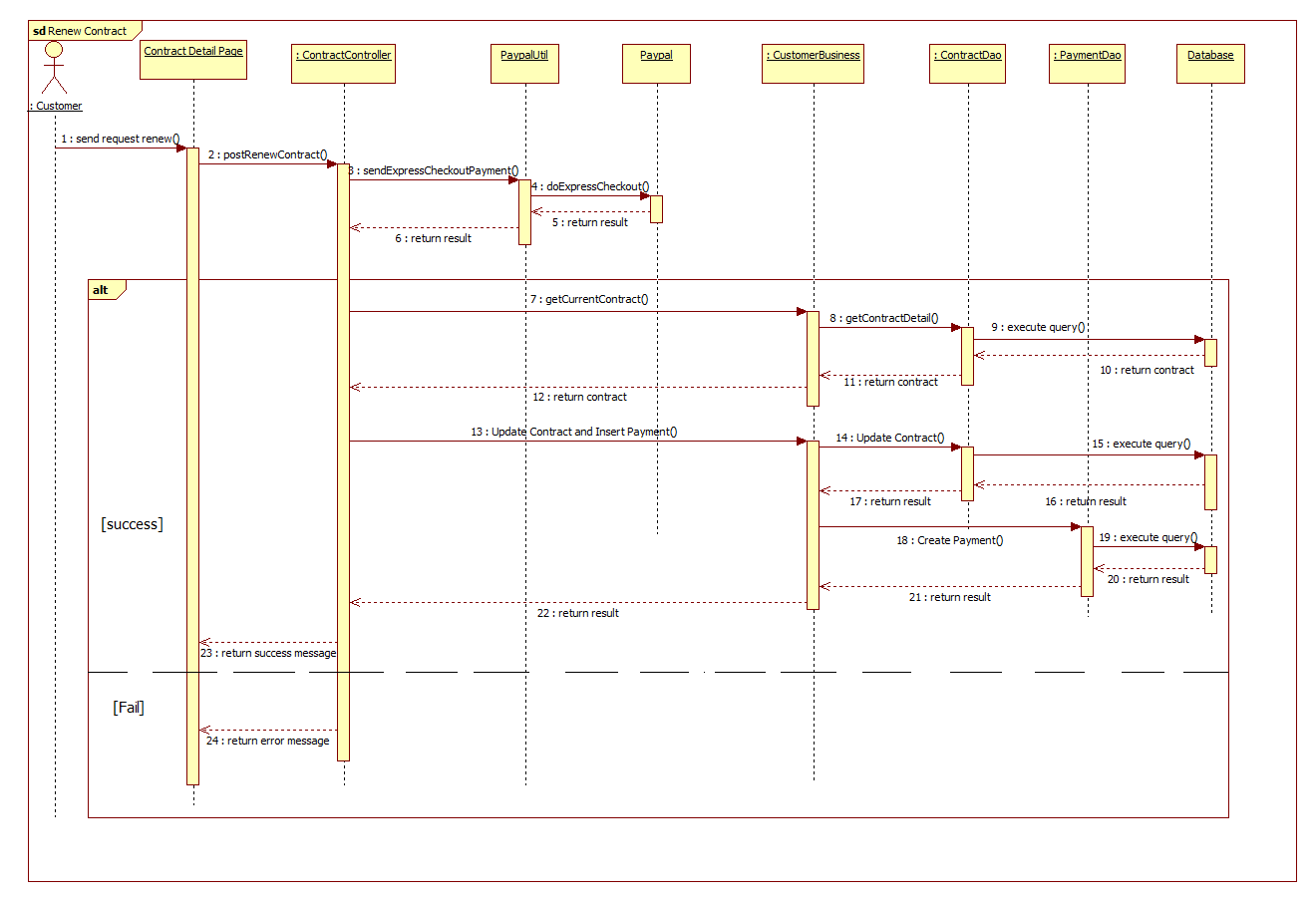


##### Customer

###### Cancel contract

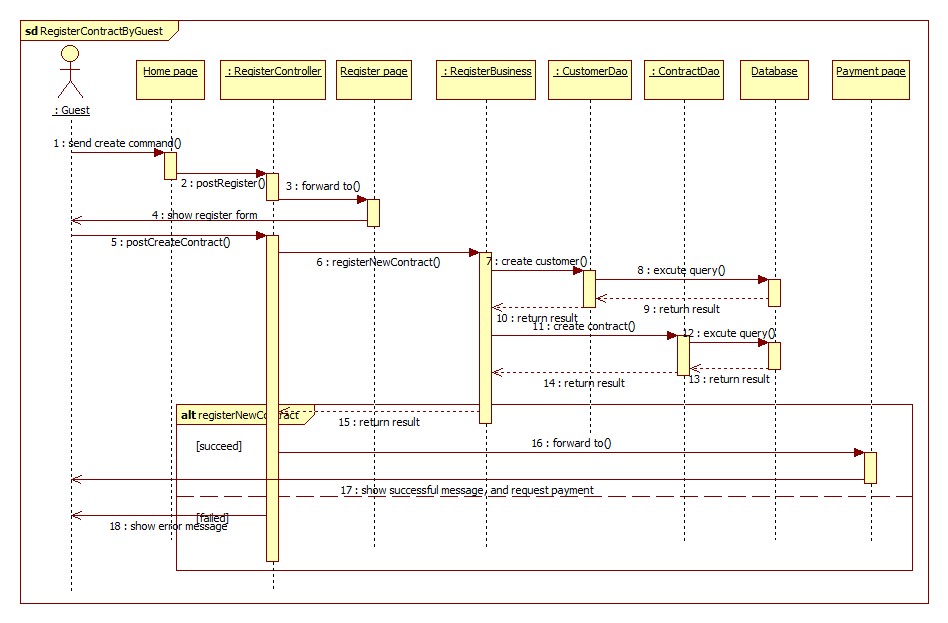


###### Renew contract

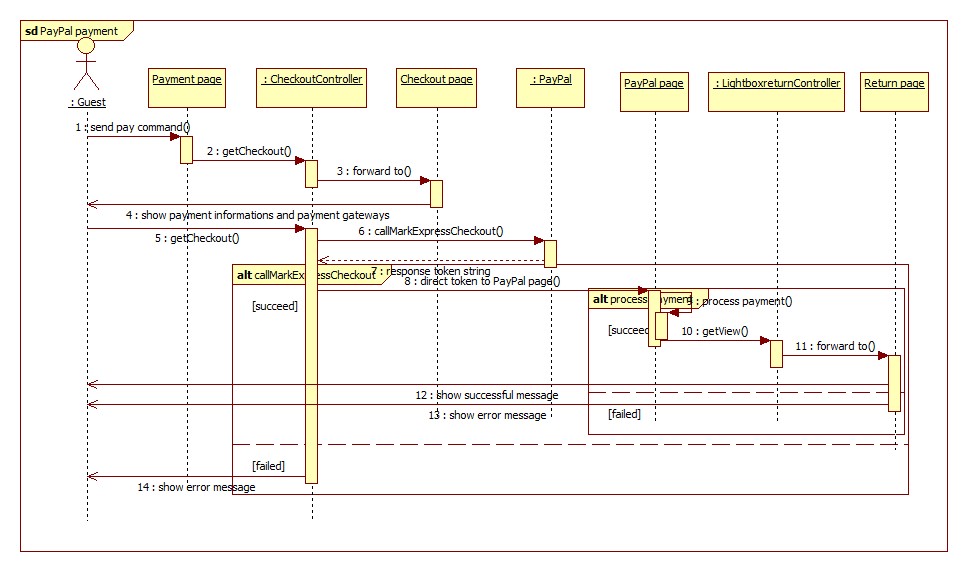


##### Guest

###### Register new contract



###### PayPal payment



#### Mobile Application

##### Checker Mobile Application

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

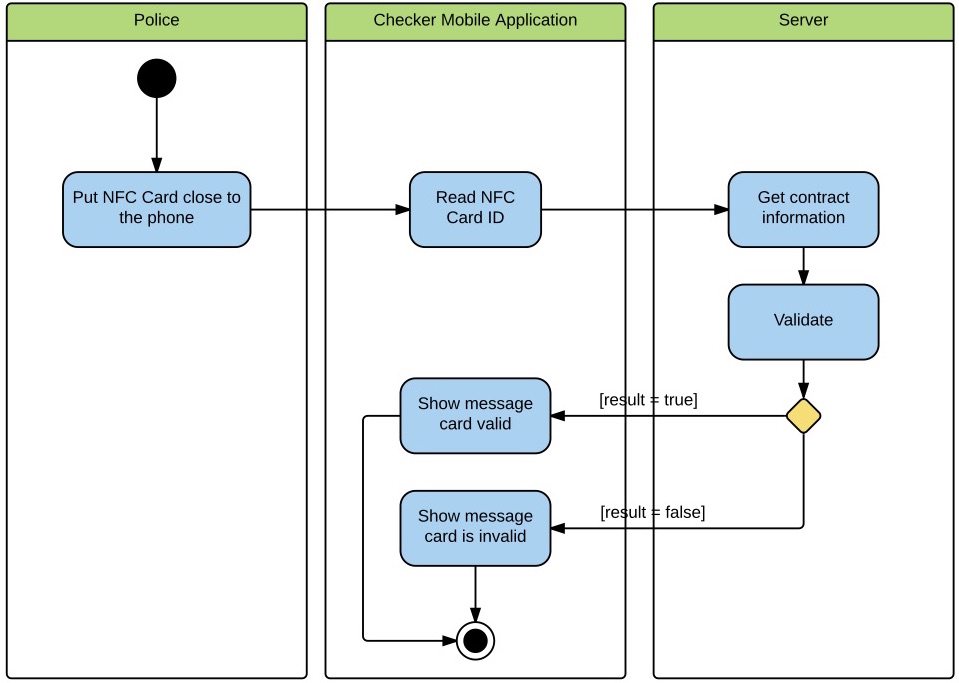


Figure 3 <Police> Verify card validation

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

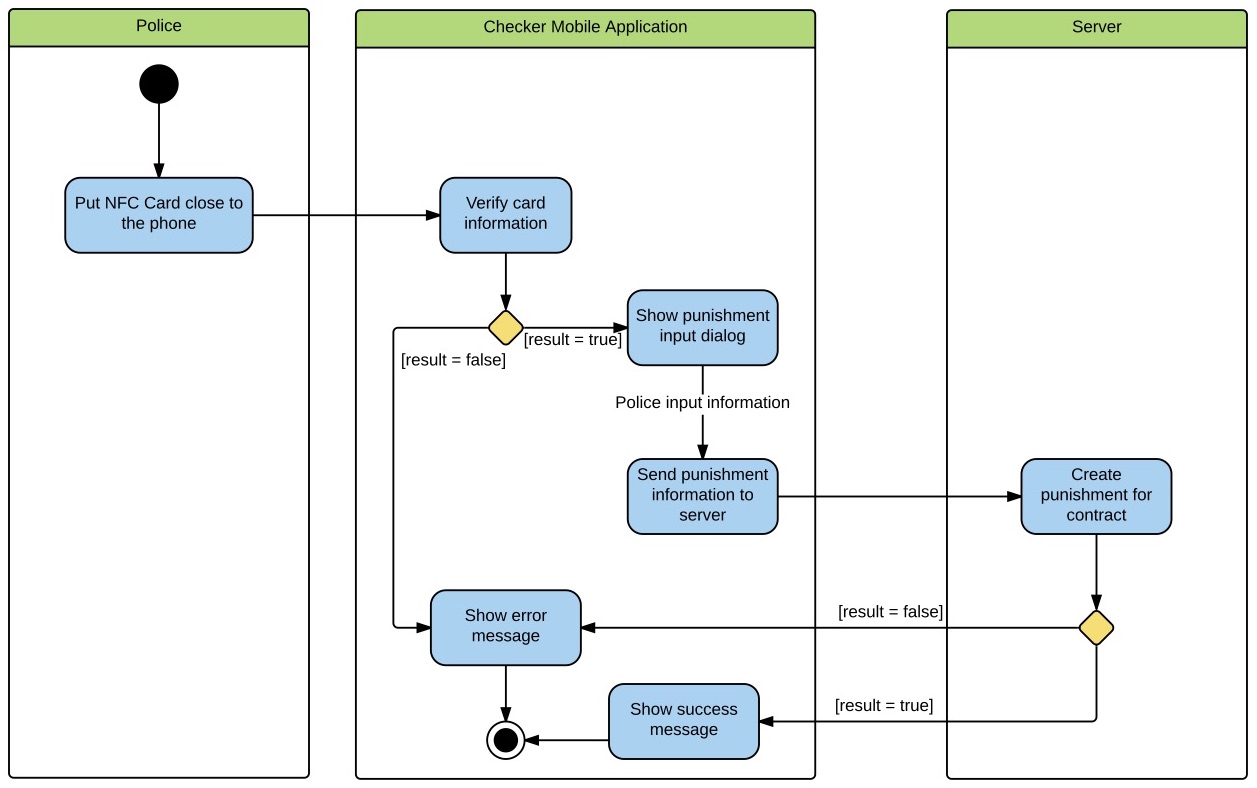


Figure 4 <Police> Add punishment information

##### Printer Mobile Application

###### <Staff> Search contract

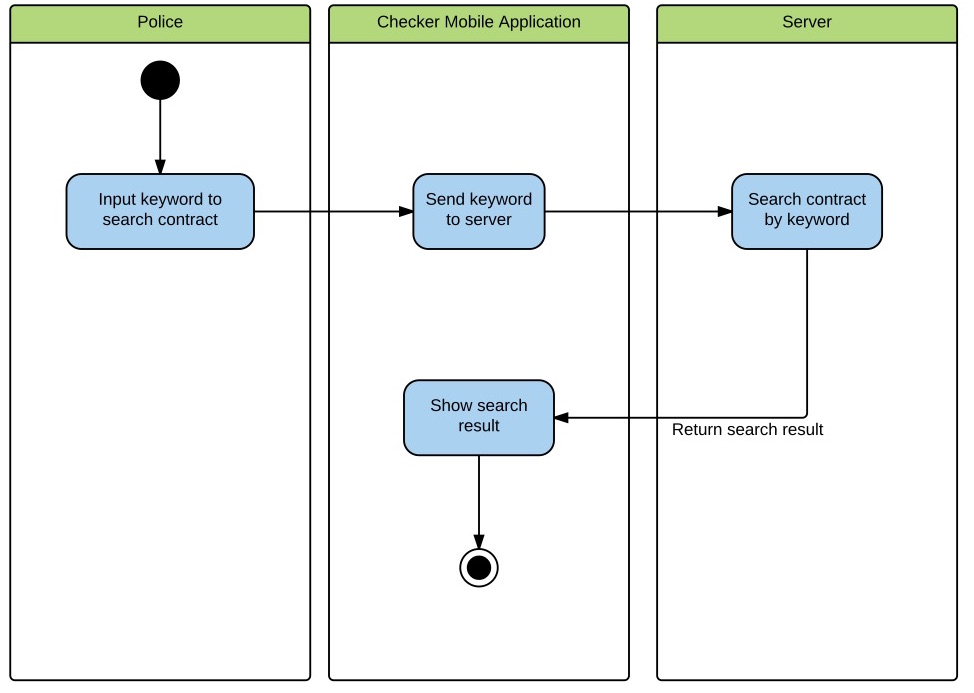


Figure 5 <Staff> Search contract

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

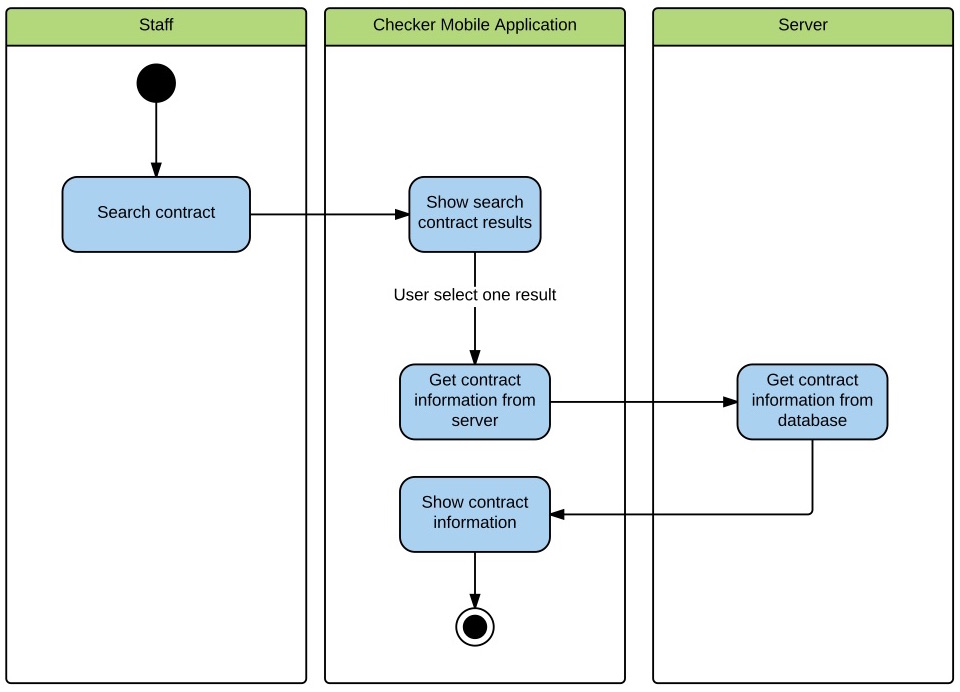


Figure 6 <Staff> View contract information

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

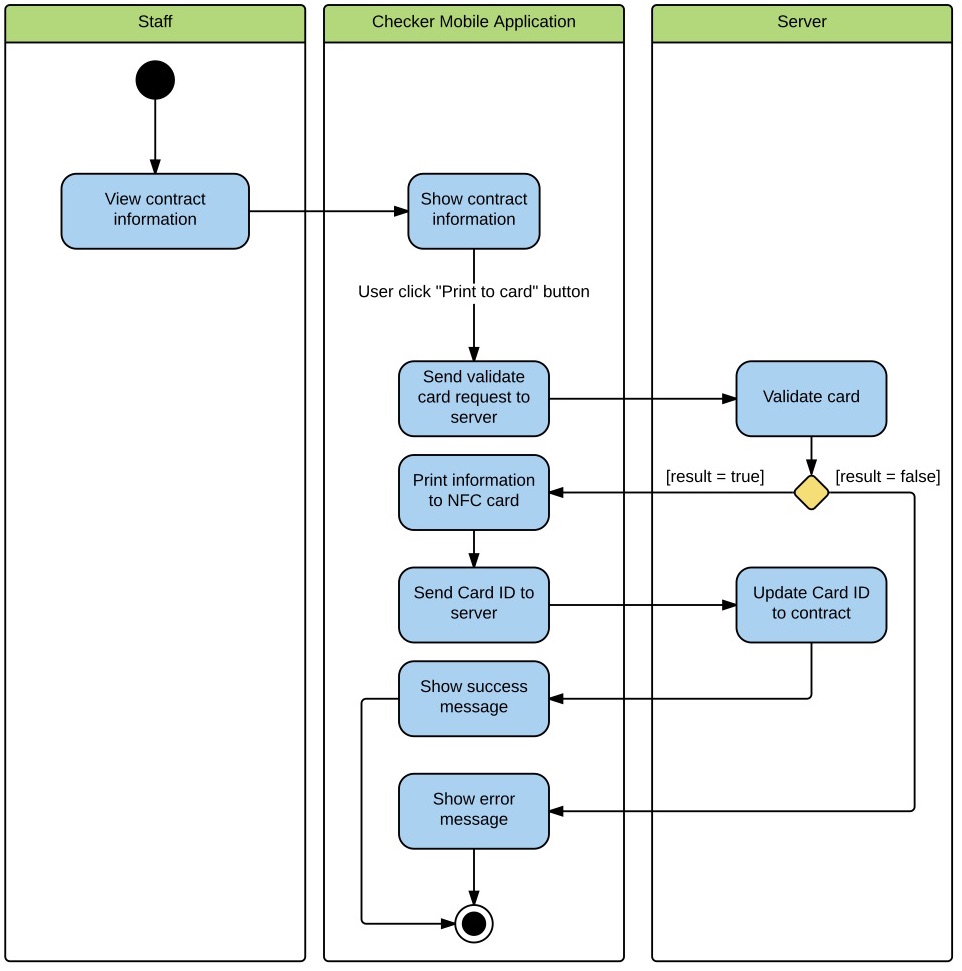


Figure 7 <Staff> Print information to NFC card

## 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 | View | List all contract | N/A | Transfer to list contract page |
| 4 | View | List all compensation | N/A | Transfer to list compensation page |
| 5 | View | List all new card request | N/A | Transfer to list new card request page |
| 6 | View | 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 | Search | 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 | Create | Create new customer | N/A | Transfer to create new customer page |
| 7 | Search | Search by input keyword | N/A | Transfer to search result page |
| 8 | View | 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 | Name | Fill customer name | No | Yes | Textbox | String | 3 – 80 |
| 4 | Address | Fill address | No | Yes | Textbox | String | 3 – 250 |
| 5 | Email | Fill email | No | Yes | Textbox | String | 3 – 250 |
| 6 | Phone | Fill phone | No | Yes | Textbox | Integer | 8 – 15 |
| 7 | Personal ID | Fill personal ID | No | No | Textbox | Integer | 8 – 15 |

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Function | Description | Validation | Outcome |
| 8 | Create | 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 | Search | 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 | Create | Create new contract | N/A | Transfer to create new contract page |
| 7 | Search | Search by input keyword | N/A | Transfer to search result page |
| 8 | View | View contract detail | N/A | Transfer to contract detail page |
| 9 | View | 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 | Customer code | Fill customer code | No | Yes | Textbox | String | 6 |
| 4 | Contract type | Select contract type | No | Yes | Select | N/A | N/A |
| 5 | Start date | Select start date | No | Yes | Date input form | N/A | N/A |
| 6 | Expired date | Select expired date | No | Yes | Date input form | N/A | N/A |
| 7 | Contract fee | Contract fee | Yes | Yes | Label | N/A | N/A |
| 8 | Plate | Fill vehicle plate | No | Yes | Textbox | String | 4 – 15 |
| 9 | Brand | Fill vehicle brand | No | Yes | Textbox | String | 2 – 20 |
| 10 | Engine | Fill vehicle engine | No | Yes | Textbox | String | 2 – 20 |
| 11 | Chassis | Fill vehicle chassis | No | Yes | Textbox | String | 2 – 20 |
| 12 | Capacity | Fill vehicle capacity | No | Yes | Textbox | String | 2 – 20 |
| 13 | Color | Fill vehicle color | No | No | Textbox | String | 2 – 20 |
| 14 | Type | Fill vehicle type | No | No | Textbox | String | 2 – 20 |
| 15 | Model code | Fill vehicle model code | No | No | Textbox | String | 2 – 20 |
| 16 | Year of manufacture | Fill vehicle year of manufacture | No | No | Textbox | String | 4 |
| 17 | Weight | Fill vehicle empty weight | No | No | Textbox | String | 1 – 4 |
| 18 | Seat capacity | Fill vehicle seat capacity | No | No | Textbox | String | 1 – 3 |
| 19 | Paid date | Select paid contract fee date | No | Yes | Date input form | N/A | N/A |

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Function | Description | Validation | Outcome |
| 20 | Select | Select customer | N/A | Transfer to select customer modal |
| 21 | Create | 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 | Expired date | Select expired date | No | Yes | Date input form | N/A | N/A |
| 4 | Renew contract fee | Renew contract fee | Yes | Yes | Label | N/A | N/A |
| 5 | Paid date | Select paid contract fee date | No | Yes | Date input form | N/A | N/A |

**Button/Hyperlinks**

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

##### Cancel contract



**Fields**

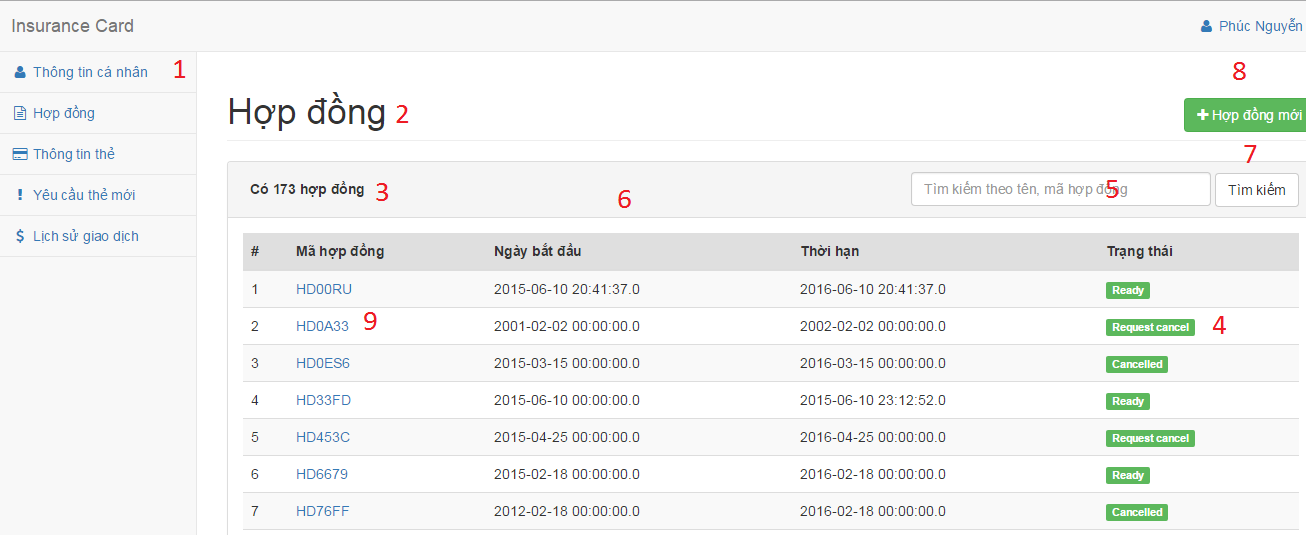
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Title | Title of the modal | Yes | Yes | Label | N/A | N/A |
| 2 | Cancel date | Select cancel contract date | No | Yes | Date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

**Button/Hyperlinks**

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

#### Customer Interface Design

##### Manage contract



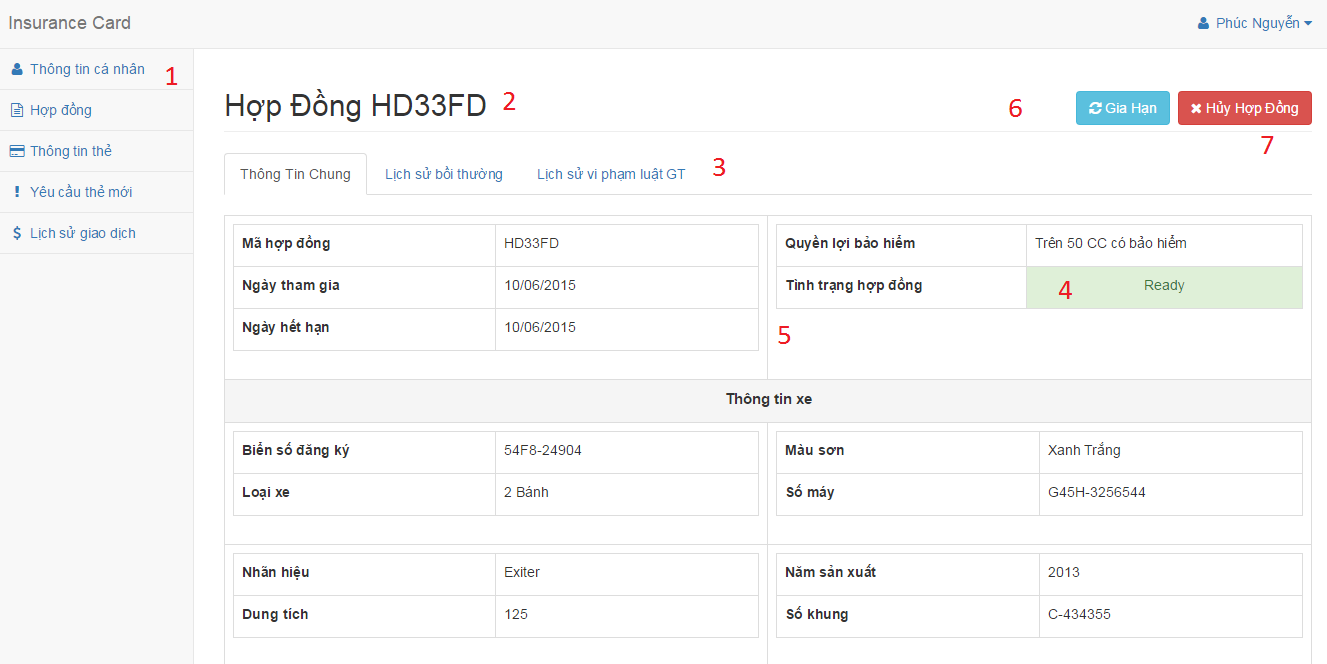
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | 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 | Text | Status of contract | Yes | No | Label | N/A | N/A |
| 5 | Search | Fill search keyword | No | Yes | Textbox | String | N/A |
| 6 | Table | Table of available contract | Yes | Yes | Table | N/A | N/A |

**Buttons / Hyperlinks**

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

##### Contract detail



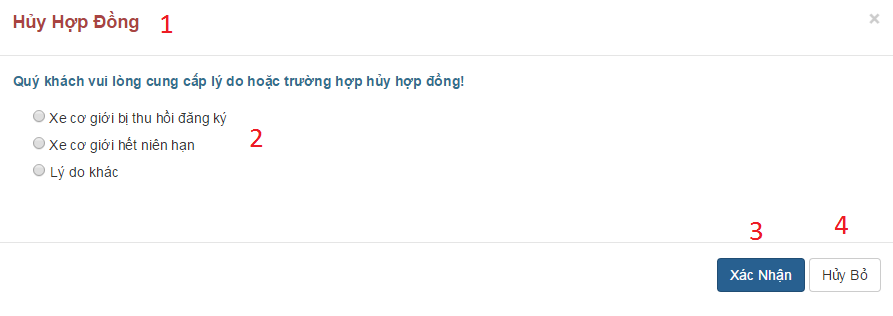
**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | Menu | Navigation bar | Yes | Yes | Menu bar | N/A | N/A |
| 2 | Title | Contract code | Yes | Yes | Label | N/A | N/A |
| 3 | Text | Information about contract | Yes | No | Tab | N/A | N/A |
| 4 | Text | Status of contract | Yes | No | Label | N/A | N/A |
| 5 | Text | Information about contract | No | Yes | Textbox | String | N/A |

**Buttons / Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Function | Description | Validation | Outcome |
| 6 | Renew | Renew contract | N/A | Show popup to choose the reason cancel |
| 7 | Cancel | Cancel contract | N/A | Show popup to provide new time for contract and the way to pay for renew contract |

##### Cancel contract



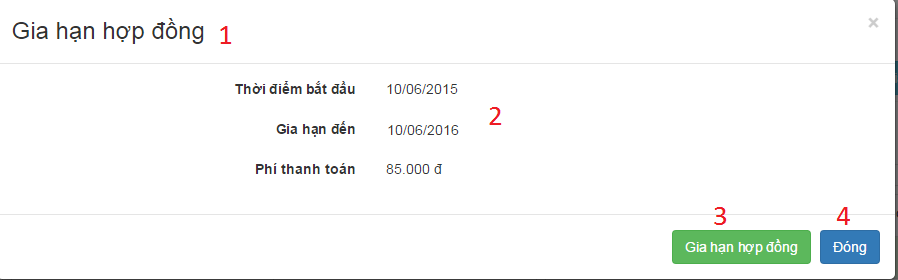
**Fields**

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

**Buttons / Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Function | Description | Validation | Outcome |
| 2 | Option cancel | Choose reason why customer cancel contract | N/A | N/A |
| 3 | Confirm | 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 | Leave | 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 | Title | Title | Yes | Yes | Label | N/A | N/A |
| 2 | Text | Information about renew contract | Yes | Yes | Label | N/A | N/A |

**Buttons / Hyperlinks**

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

**Select paypment methodSelect paypment method**

**Select paypment method**

### Checker Mobile Application Design

#### Scan NFC card screen

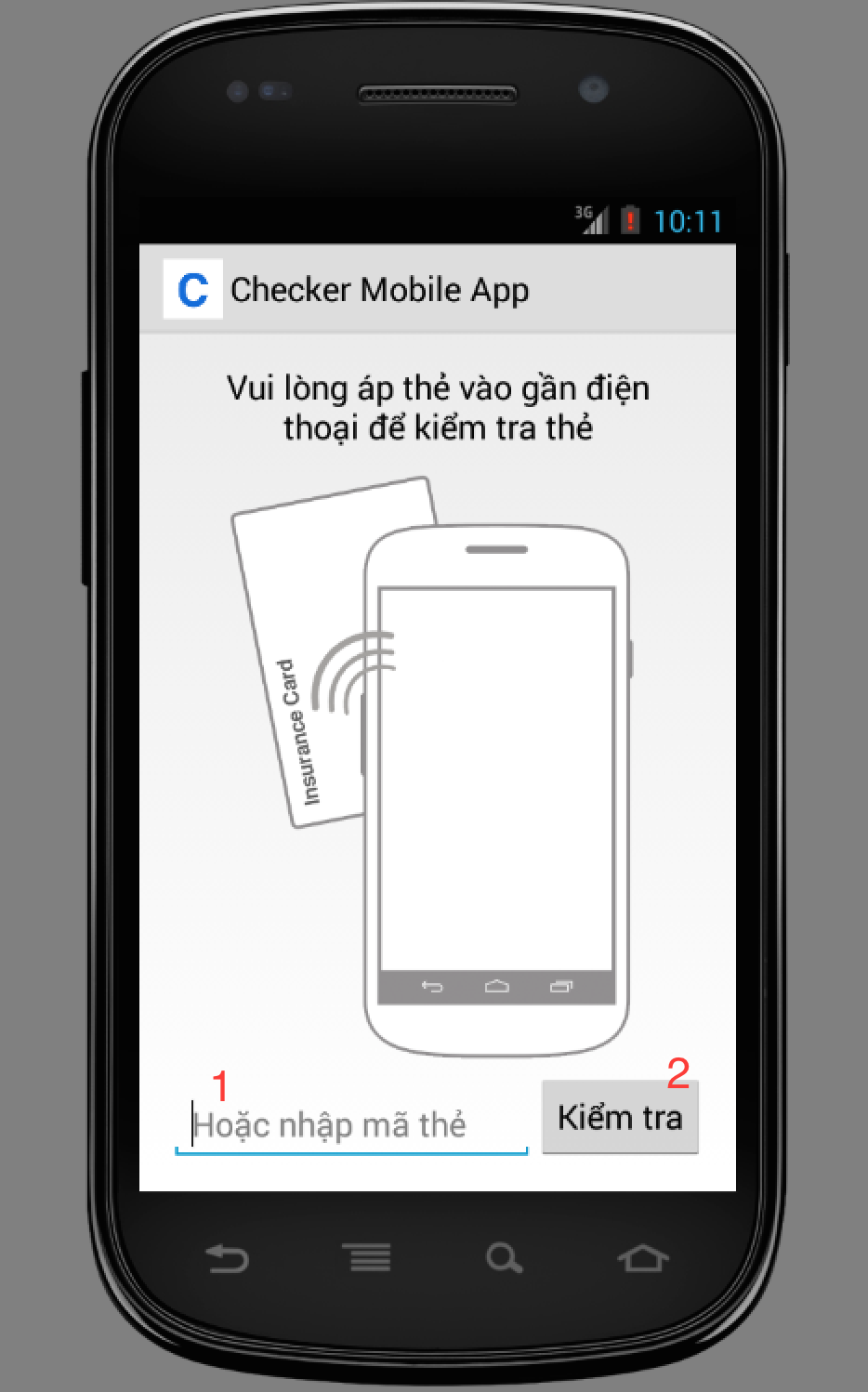


Figure 8 Scan NFC card screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | CardIDnpen 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 | Cnpen print card screen supporto view detail information of that contractheck Card | Verify card validation | N/A | Go to view card information screen |

Table 3 Scan NFC card screen - Buttons

#### View card information screen

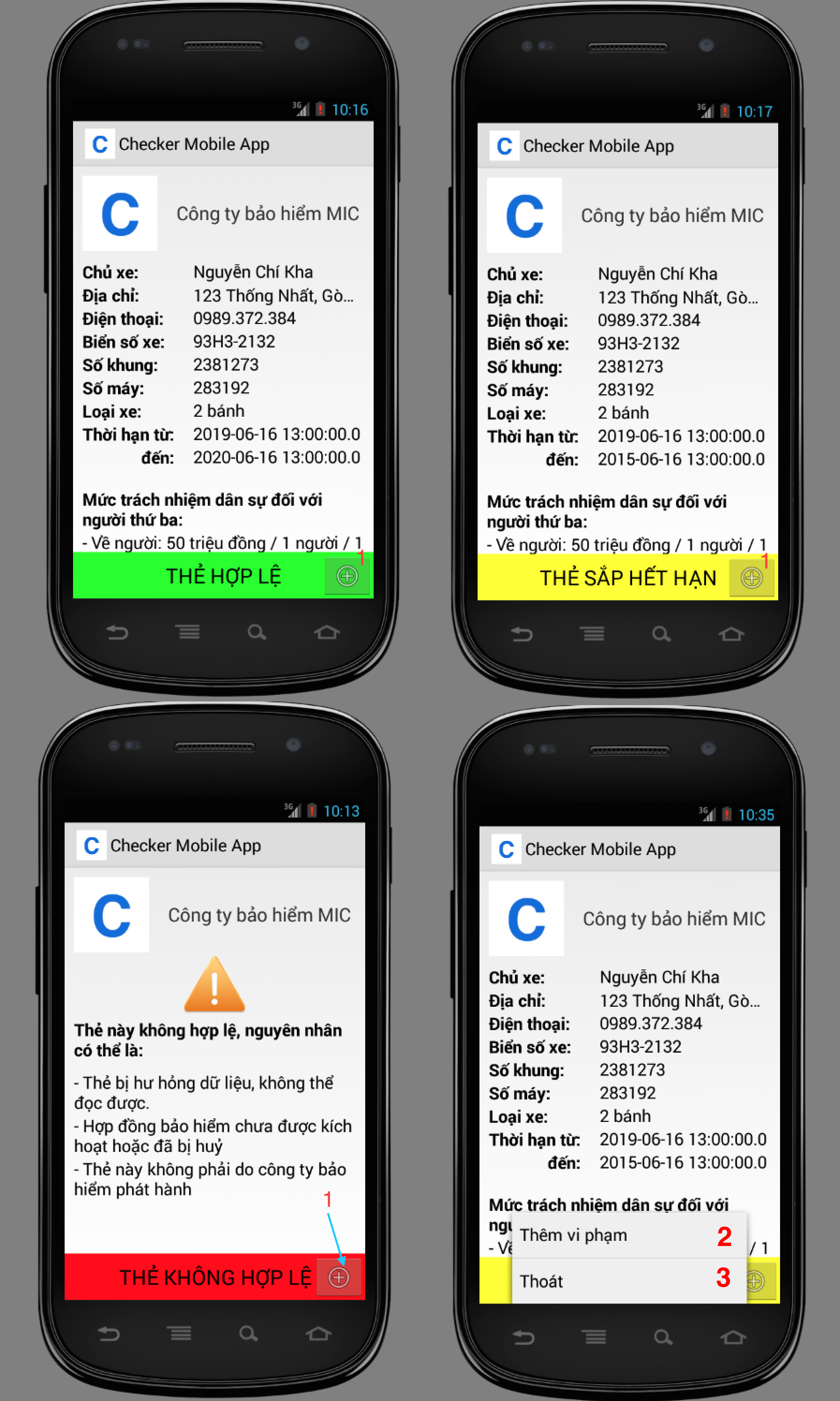


Figure 9 View card information screen

**Buttons**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Function | Description | Validation | Outcome |
| 1 | Show menu | Show menu which contains “Add punishment” button and “Exit” button | N/A | Open menu |
| 2 | Add punishment | Add punishment information | N/A | Open add punishment dialog |
| 3 | Exit | 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

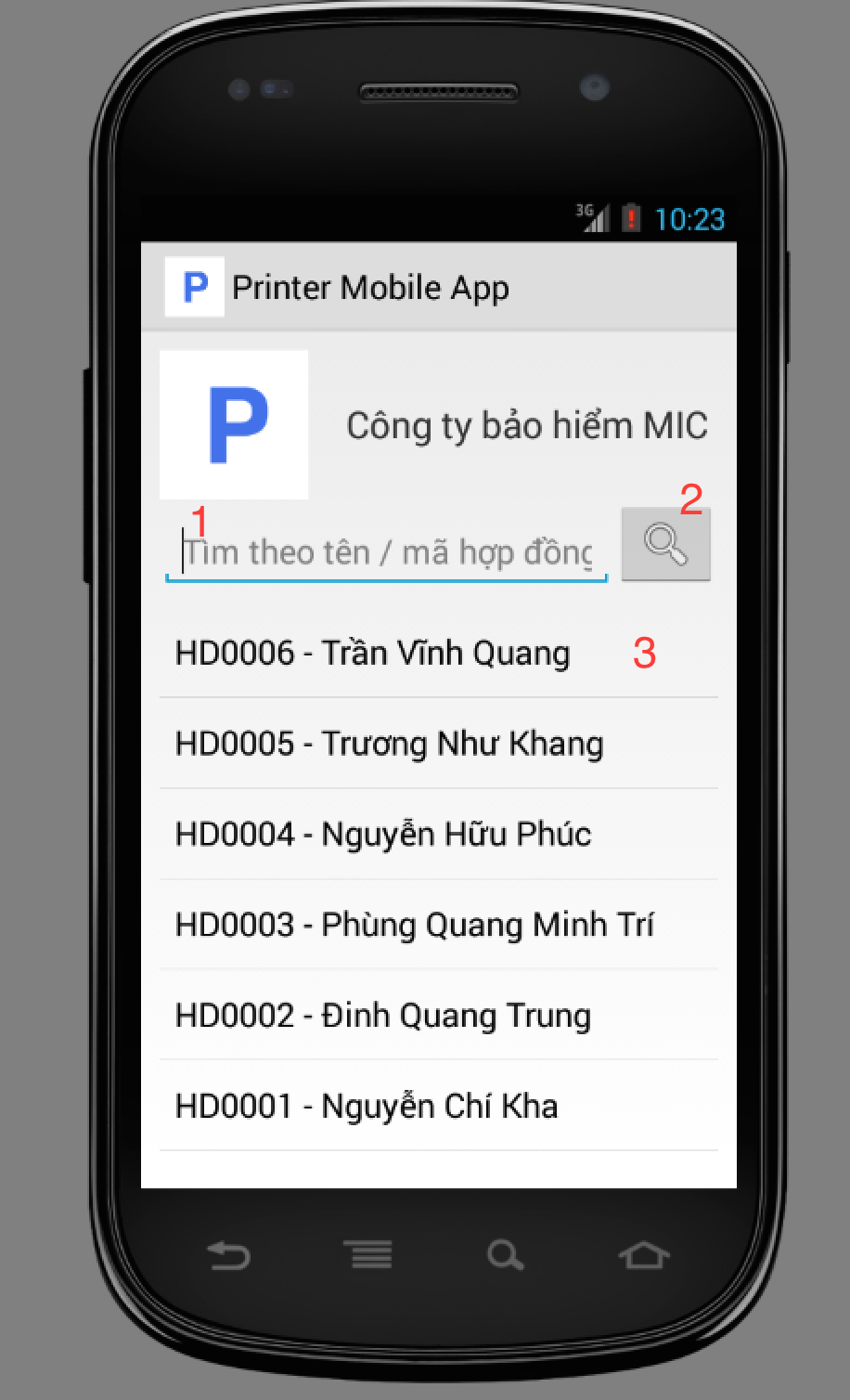


Figure 10 Search contract screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | SearchBox | 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 | Search | Search contract with keyword | N/A | The list will be updated with search results |
| 3 | View contract detail | 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 | Print to card | 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

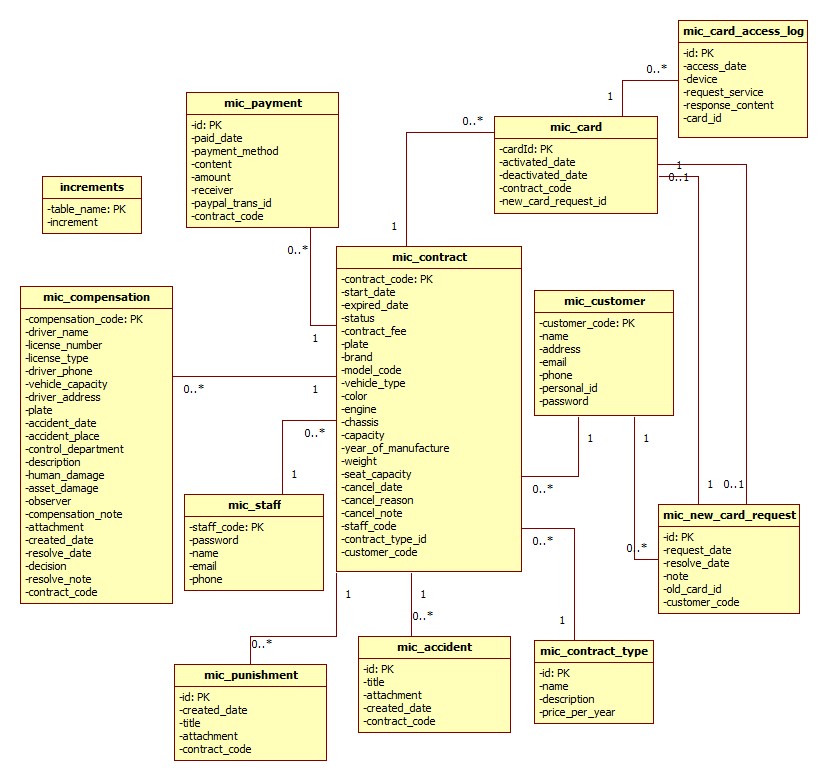
#### Print card screen



Figure 12 Print card screen

## Database Design

### Logical diagram



### Data Dictionary

|  |  |
| --- | --- |
| Data Dictionary: describes content of all tables | |
| Entity name | **Description** |
| mic\_customer | Contain the customer information. |
| mic\_contract | Contain the contract information. |
| mic\_card | Contain the card information. |
| mic\_payment | Contain the payment information. |
| mic\_staff | Contain the staff information. |
| mic\_compensation | Contain the compensation information. |
| mic\_punishment | Contain the punishment information. |
| mic\_accident | Contain the accident information. |
| mic\_contract\_type | Contain the contract type information. |
| mic\_new\_card\_request | Contain the new card request information. |
| mic\_card\_access\_log | Contain the new card access log. |
| increments | Contain the number identity auto increment of other table (compensation, contract, customer, staff). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Attributes | Description | Domain | Null |
| 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 number | VARCHAR(15) | NO |
| personal\_id | Customer personal id. | VARCHAR(15) | YES |
| password | Customer password. | VARCHAR(32) | NO |
| mic\_contract | contract\_code {PK} | Unique identifier of contract. | VARCHAR(10) | NO |
| start\_date | Start date of the contract | DATETIME | NO |
| expired\_date | Expired date of the contract | DATETIME | NO |
| status | Contract’s status | VARCHAR(50) | NO |
| contract\_fee | Contract’s fee. | FLOAT | NO |
| plate | Plate number of the motor. | VARCHAR(15) | NO |
| brand | Brand of the motor. | VARCHAR(20) | NO |
| model\_code | Model code of the motor. | VARCHAR(20) | YES |
| vehicle\_type | Vehicle type of the motor. | VARCHAR(20) | YES |
| color | Motor’s color. | VARCHAR(20) | YES |
| engine | Motor’s engine. | VARCHAR(20) | NO |
| chassis | Motor’s chassis | VARCHAR(20) | NO |
| capacity | Motor’s capacity. | VARCHAR(20) | NO |
| year\_of\_manufature | Year of manufacture. | INT UNSIGNED | YES |
| weight | Motor’s weight | INT UNSIGNED | YES |
| seat\_capacity | Motor’s seat capacity. | INT UNSIGNED | YES |
| cancel\_date | The contract cancel date. | DATETIME | YES |
| cancel\_reason | The cancel reason. | VARCHAR(250) | YES |
| cancel\_note | The cancel note. | VARCHAR(2000) | YES |
| staff\_code | The staff code. | VARCHAR(10) | YES |
| contract\_type\_id | The contract type id. | INT UNSIGNED | NO |
| customer\_code | The customer code. | VARCHAR(10) | NO |
| mic\_card | card\_id {PK} | The unique identifier of card. | VARCHAR(20) | NO |
| activated\_date | The activated date of card. | DATETIME | NO |
| deactivated\_date | The deactivated date of card. | DATETIME | YES |
| contract\_code | The contract code | VARCHAR(10) | NO |
| new\_card\_request\_id | The new card request id. | INT UNSIGNED | YES |
| mic\_payment | id {PK} | The unique identifier of payment. | INT UNSIGNED | NO |
| paid\_date | The date payment was processed. | DATETIME | NO |
| payment\_method | The payment method | VARCHAR(20) | NO |
| content | The payment content. | VARCHAR(250) | NO |
| amount | The total amount. | FLOAT | NO |
| receiver | The receiver. | VARCHAR(80) | YES |
| paypal\_trans\_id | The paypal transaction id. | VARCHAR(50) | YES |
| contract\_code | The contract code. | VARCHAR(10) | NO |
| mic\_staff | staff\_code | The unique identifier of staff. | VARCHAR(10) | NO |
| password | The staff password. | VARCHAR(32) | NO |
| name | The staff name | VARCHAR(80) | NO |
| email | The staff email. | VARCHAR(250) | NO |
| phone | The staff phone number. | VARCHAR(15) | NO |
| mic\_compensation | compensation\_code {PK} | The unique identifier of compensation | VARCHAR(10) | NO |
| driver\_name | The driver name. | VARCHAR(80) | NO |
| license\_number | The license number of the motor. | VARCHAR(15) | NO |
| license\_type | The license type. | VARCHAR(10) | NO |
| vehicle\_capacity | The vehicle capacity. | VARCHAR(20) | NO |
| driver\_address | The driver’s address. | VARCHAT(250) | NO |
| plate | The plate number | VARCHAR(15) | NO |
| accident\_date | The accident date. | DATETIME | NO |
| accident\_place | The accident place. | VARCHAR)(250) | NO |
| control\_department | The police control department. | VARCHAR(2000) | NO |
| description | The compensation description. | VARCHAR(2000) | NO |
| human\_damage | The human damage. | VARCHAR(2000) | NO |
| asset\_damage | The asset damage | VARCHAR(2000) | NO |
| observer | The observer name. | VARCHAR(80) | NO |
| compensation\_note | The compensation note. | VARCHAR(2000) | YES |
| attachment | The attachment | VARCHAR(255) | YES |
| created\_date | The created date of the compensation. | DATETIME | NO |
| resolve\_date | The resolve date of the compensation | DATETIME | YES |
| desicion | The decision of the compensation | VARCHAR(250) | YES |
| resolve\_note | The resolve note | VARCHAR(2000) | YES |
| contract\_code | The contract code. | VARCHAR(10) | NO |
| mic\_punishment | id {PK} | The unique identifier of punishment. | INT UNSIGNED | NO |
| created\_date | The created date of the punishment. | DATETIME | NO |
| title | The punishment title | VARCHAR(250) | NO |
| attachment | The attachment file link. | VARCHAR(255) | NO |
| contract\_code | The contract code. | VARCHAR(10) | NO |
| mic\_accident | id {PK} | The unique identifier of accident. | INT UNSIGNED | NO |
| title | The accident title | VARCHAR(250) | NO |
| attachment | The attachment link file. | VARCHAR(255) | NO |
| created\_date | The accident created date | DATETIME | NO |
| contract\_code | The contract code | VARCHAR(10) | NO |
| mic\_contract\_type | id | The unique identifier of the contract type. | INT UNSIGNED | NO |
| name | The contract type name | VARCHAR(250) | NO |
| description | The contract type description | VARCHAR(2000) | NO |
| price\_per\_year | The price per year. | FLOAT | NO |
| mic\_new\_card\_request | id | The unique identifier of new card request. | INT UNSIGNED | NO |
| request\_date | The request date. | DATETIME | NO |
| resolve\_date | The resolve date. | DATETIME | YES |
| note | The new card request note. | VARCHAR(2000) | NO |
| old\_card\_id | The card id of the old motor. | VARCHAR(20) | NO |
| customer\_code | The customer code | VARCHAR(10) | NO |
| mic\_card\_access\_log | id {PK} | The unique identifier of card access log. | INT UNSIGNED | NO |
| access\_date | The access date. | DATETIME | NO |
| device | The access device. | VARCHAR(50) | NO |
| request\_service | The request service. | VARCHAR(200) | NO |
| response\_content | The response content. | VARCHAR(200) | NO |
| card\_id | The card id. | VARCHAR(20) | NO |
| increments | table\_name | The unique identifier of the increments table. | VARCHAR(255) | NO |
|  | increment | The auto increment number of other table (staff, contract, customer, compensation) | INT | YES |

## 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.

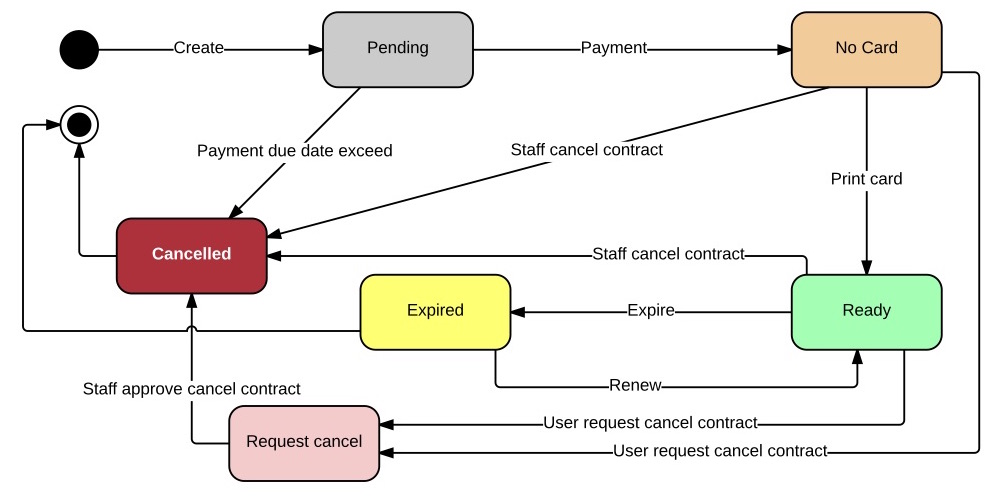


Figure 13 Contract State Chart

|  |  |
| --- | --- |
| State Dictionary: Describes States | |
| 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

* First customer create new contract, contract is in “Pending” state.
* Next, customer pay for the contract via PayPal or direct payment, contract state change from “Pending” to “No card”.
* Next, staff print the card for customer, the card ID is saved to the system, contract state change from “No card” to “Ready”.
* Next, when the contract due date is exceed, system will change the contract status from “Ready” to “Expired”
* Next, when customer renew the contract, the contract state change from “Expired” to “Ready”. Customer can renew the contract when the contract is in “Ready” or “Expired” state.
* When the contract is in “No card” or “Ready”, staff can cancel the contract to change the state to “Cancelled”.
* When the contract is in “No card” or “Ready”, customer can send contract cancel request, contract state change to “Request cancel”
* When the contract is in “Request cancel”, staff can approve the request to change contract state from “Request cancel” to “Cancelled”.

### 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.

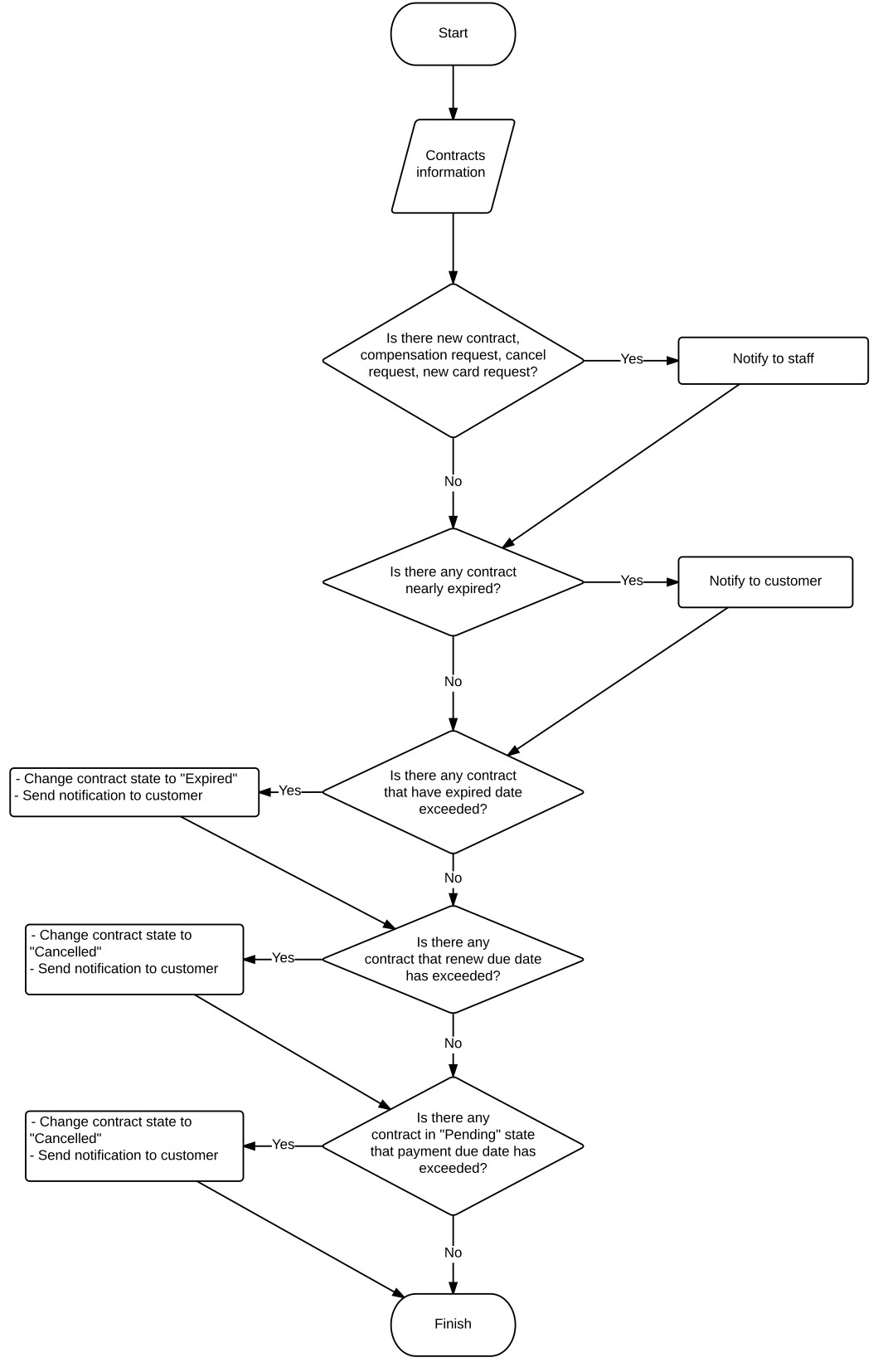


Figure 14 System Scheduler Process

### 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:

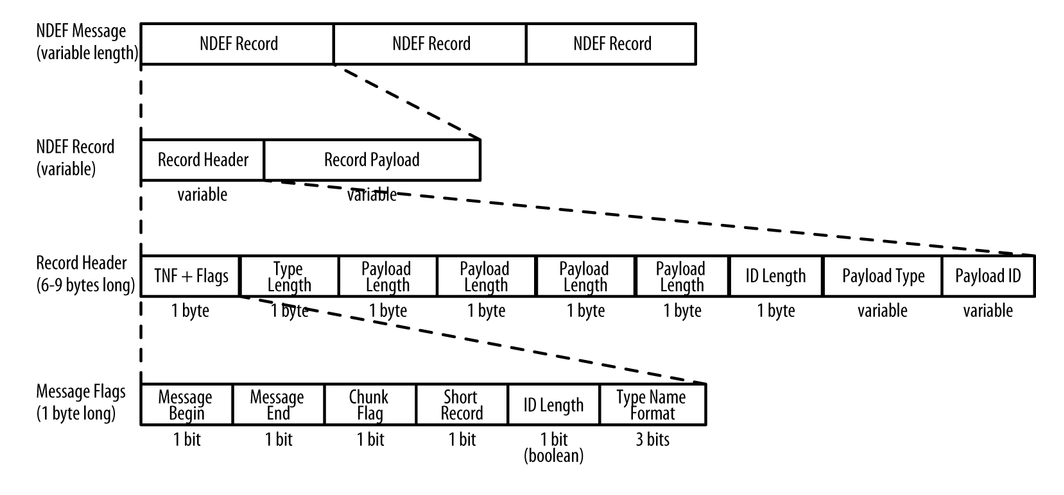


Figure 15 NDEF message structure

**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 9 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