# 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

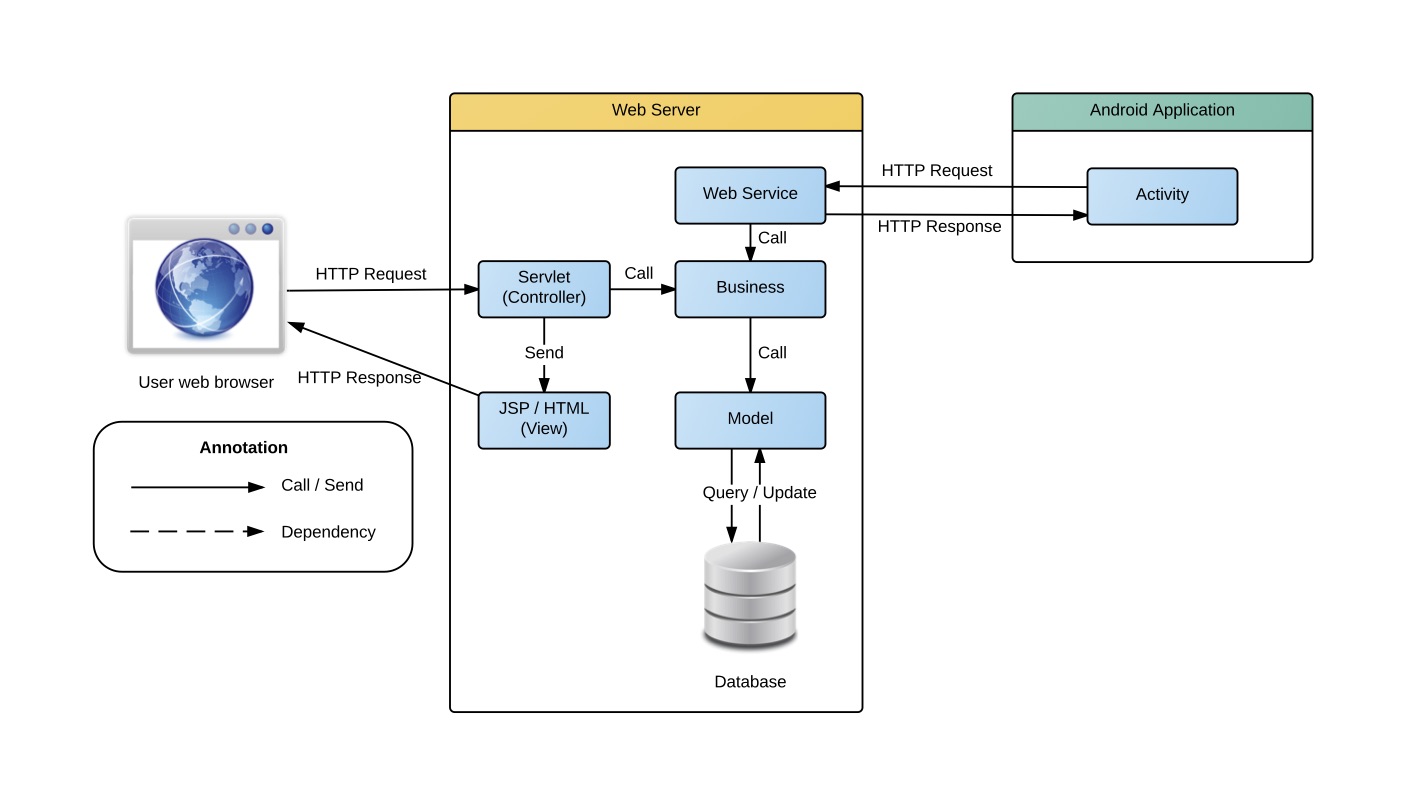


Figure 1 System architecture design

### 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

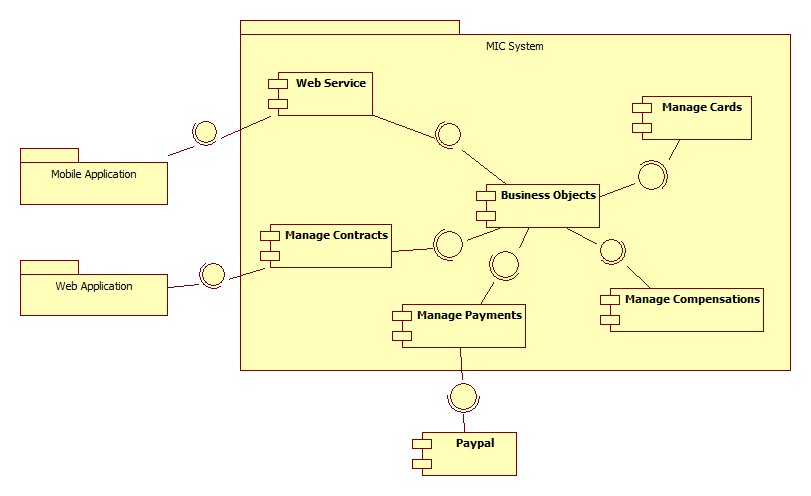


Figure Component Diagram

|  |  |
| --- | --- |
| 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 Component Dictionary

## Detailed Description

### Class Diagram

### Class Diagram Explanation

### Interactive Diagram

#### Web Application

#### Mobile Application

##### Checker Mobile Application

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

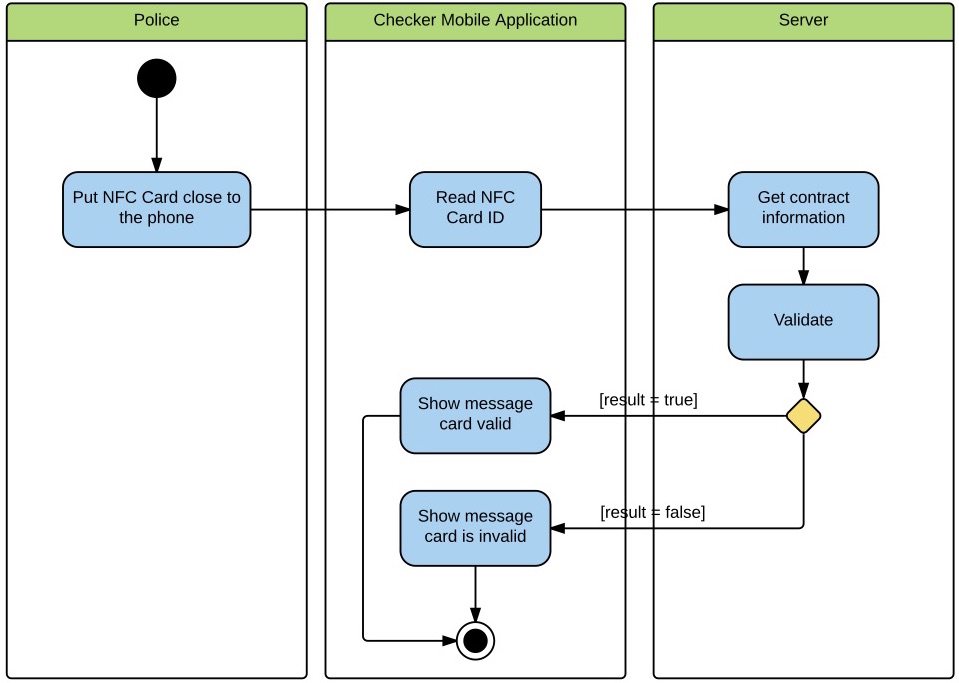


Figure <Police> Verify card validation

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

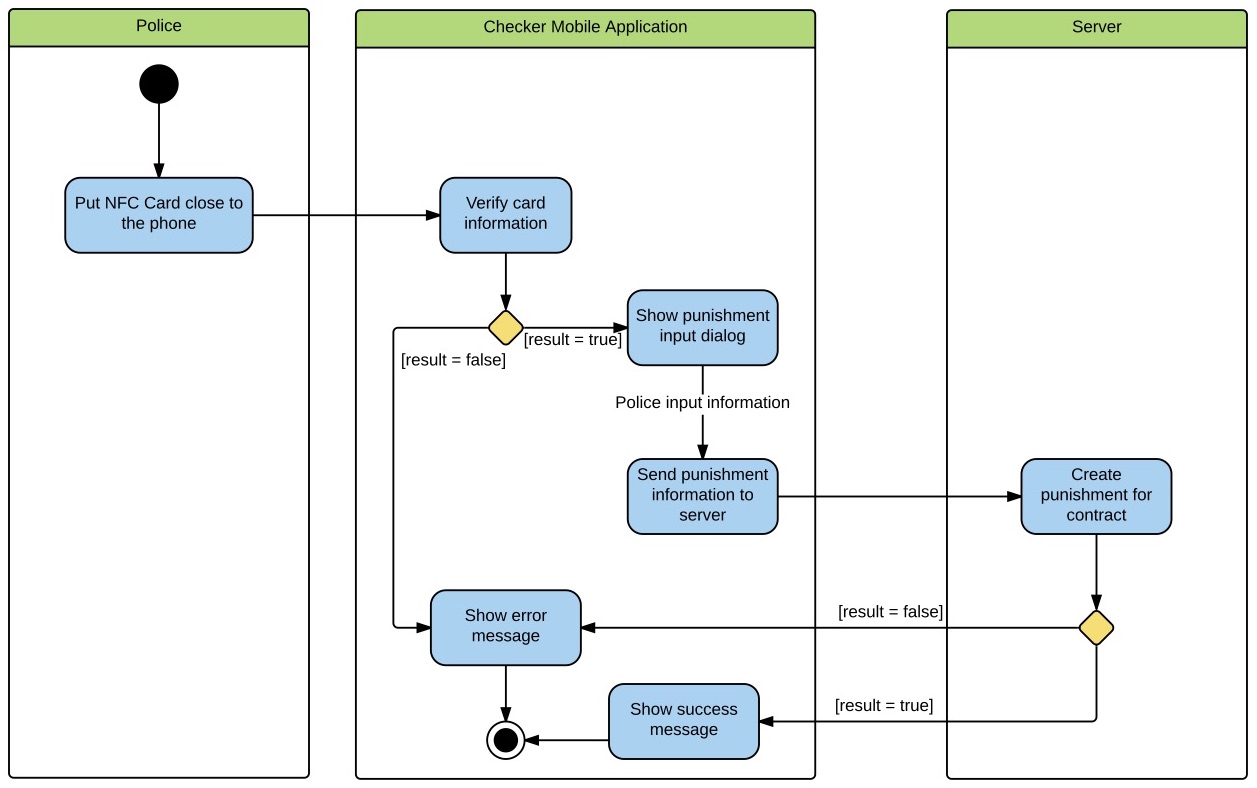


Figure <Police> Add punishment information

##### Printer Mobile Application

###### <Staff> Search contract

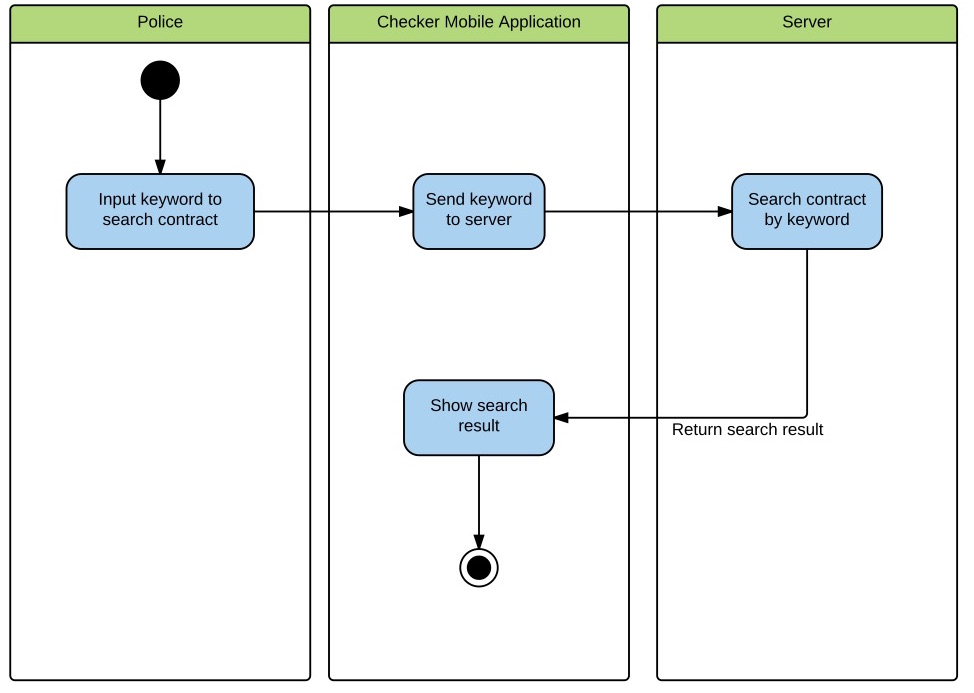


Figure <Staff> Search contract

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

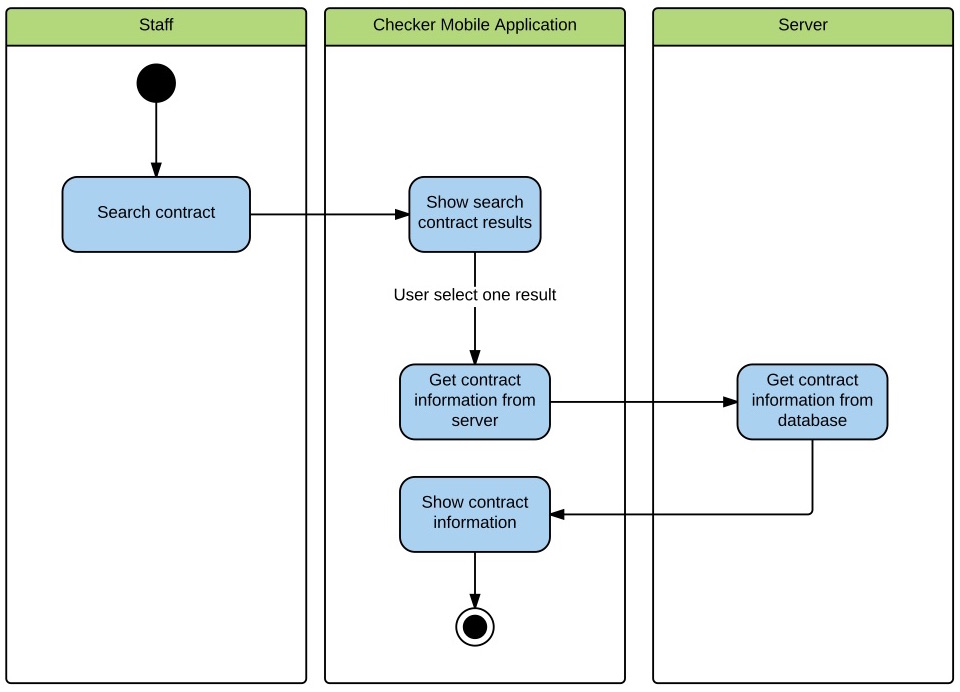


Figure <Staff> View contract information

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

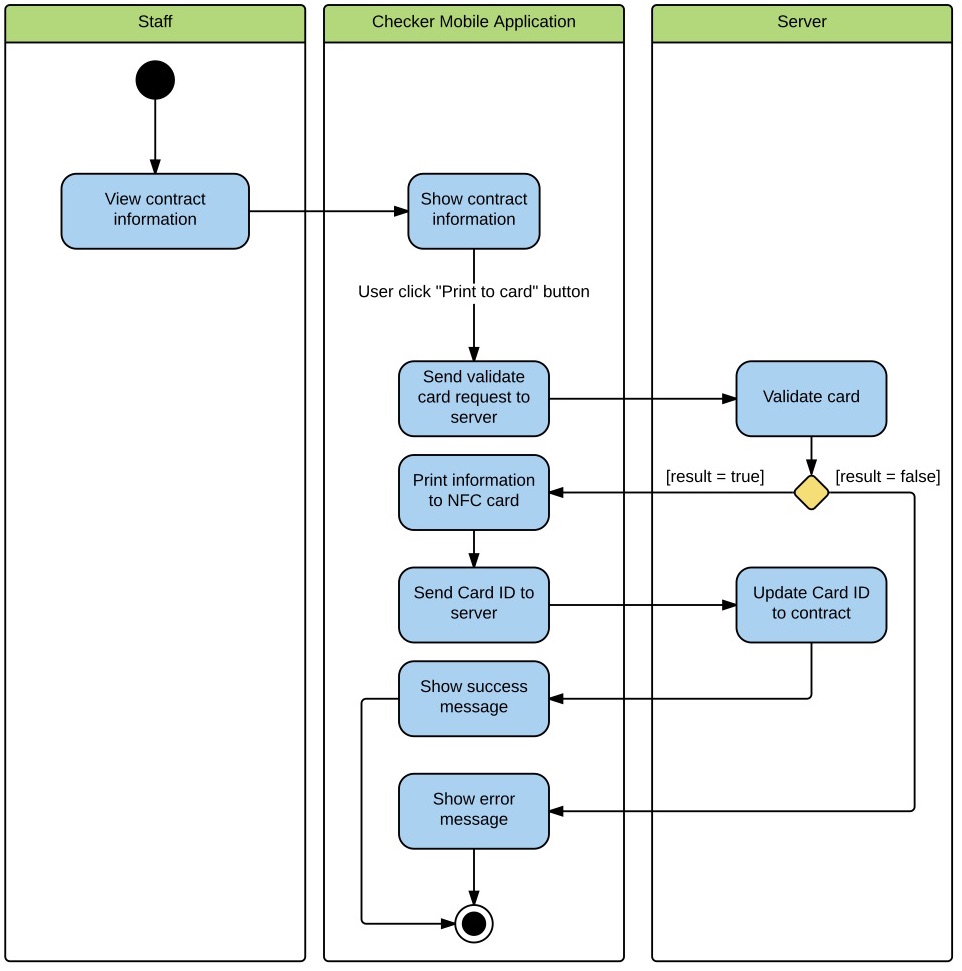


Figure <Staff> Print information to NFC card

## User Interface Design

### Web application Design

### Checker Mobile Application Design

### Printer Mobile Application Design

## Database Design

### Logical Diagram

### Data Dictionary

## Algorithms

### 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.

#### Complexity

With ***N*** is the number of contract in system, the complexity of the system scheduler process is ***N***.

#### Flow chart

The following flow chart diagram describes all the process of the System scheduler.

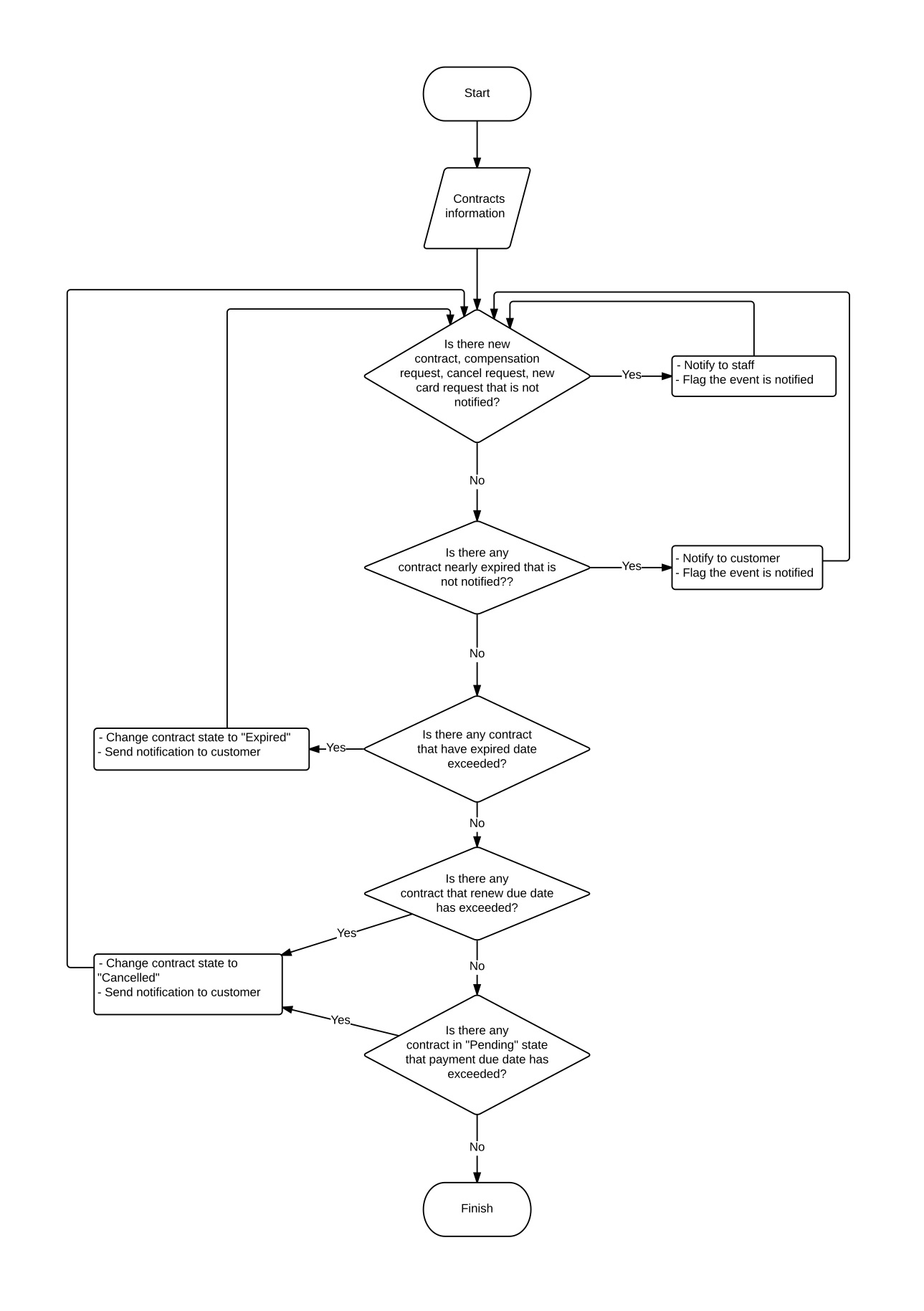


Figure System Scheduler Process

### 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.

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.

To make sure the card can only be read by our Checker application, we write a special record named Android Application Record (AAR).

**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

#### Complexity

#### Flow chart

### 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.

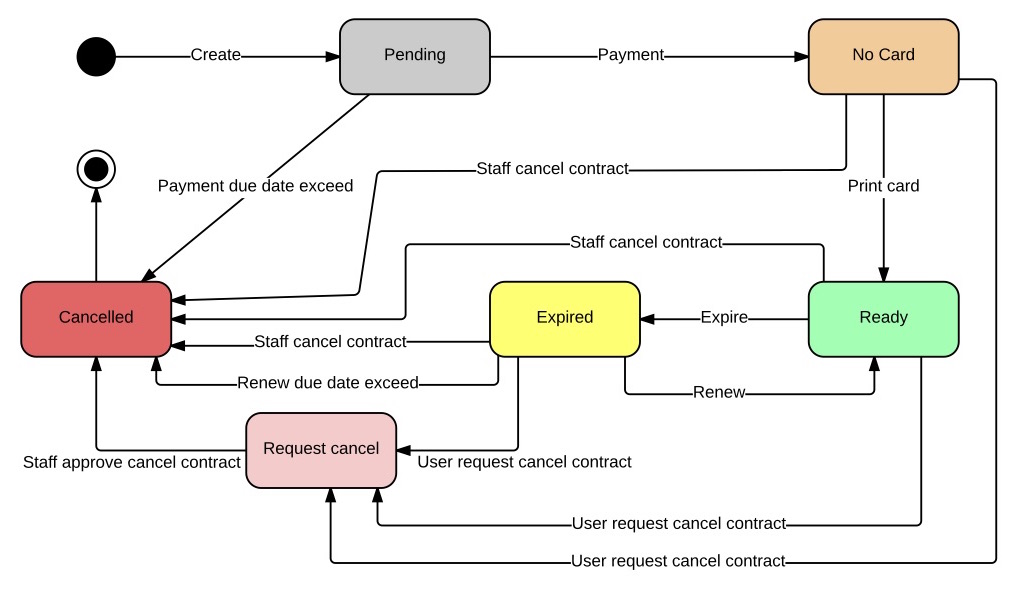


Figure 9 Contract State Chart

|  |  |
| --- | --- |
| 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 2 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.
* If customer does not renew the contract, after 1 month the contract state will change from “Expired” to “Cancelled”.
* When the contract is in “No card” or “Ready” or “Expired”, staff can cancel the contract to change the state to “Cancelled”.
* When the contract is in “No card” or “Ready” or “Expired”, 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”.