**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Green Bus Ticket System**

|  |  |
| --- | --- |
| **Group 1** | |
| **Group members** | Đỗ Ngọc Hoàng – SE61246  Trần Quang Trường – SE61129  Đoàn Minh Đức – SE61486 |
| **Supervisor** | Kiều Trọng Khánh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | GBTS |

- Ho Chi Minh city, September 5th 2016 -

Table of Contents

[List of Tables 4](#_Toc468625835)

[List of Figures 4](#_Toc468625836)

[Definitions, Acronyms, and Abbreviations 4](#_Toc468625837)

[A. Introduction 5](#_Toc468625838)

[1. Project Information 5](#_Toc468625839)

[2. Introduction 5](#_Toc468625840)

[3. Current Situation 5](#_Toc468625841)

[4. Problem Definition 5](#_Toc468625842)

[5. Proposed Solution 6](#_Toc468625843)

[5.1 Feature functions 6](#_Toc468625844)

[5.2 Advantages and disadvantages 6](#_Toc468625845)

[6. Functional Requirements 7](#_Toc468625846)

[7. Role and Responsibility 8](#_Toc468625847)

[B. Software Project Management Plan 8](#_Toc468625848)

[1. Problem Definition 8](#_Toc468625849)

[1.1. Name of this Capstone Project 8](#_Toc468625850)

[1.2. Problem Abstract 8](#_Toc468625851)

[1.3. Project Overview 8](#_Toc468625852)

[2. Project organization 11](#_Toc468625853)

[2.1. Software Process Model 11](#_Toc468625856)

[2.2. Roles and responsibilities 12](#_Toc468625857)

[2.3. Tools and Techniques 13](#_Toc468625858)

[3. Project Management Plan 14](#_Toc468625859)

[3.1. Software development life cycle 14](#_Toc468625861)

[3.2. Phase Detail 16](#_Toc468625862)

[3.3. All Meeting Minutes 17](#_Toc468625863)

[C. Software Requirement Specification 18](#_Toc468625864)

[1. User Requirement Specification 18](#_Toc468625865)

[1. Unauthorized User Requirement 18](#_Toc468625866)

[2. Authorized User Requirement 18](#_Toc468625867)

[3. Passenger Requirement 18](#_Toc468625868)

[4. Staff Requirement 18](#_Toc468625869)

[5. Manager Requirement 18](#_Toc468625870)

[6. Admin Requirement 18](#_Toc468625871)

[7. Emulator Requirement 19](#_Toc468625872)

[8. Auto Handler Requirement 19](#_Toc468625873)

[2. System Requirement Specification 19](#_Toc468625874)

[3. Conceptual Diagram 20](#_Toc468625875)

[D. Software Design Description 21](#_Toc468625876)

[1. Design Overview 21](#_Toc468625877)

[2. System Architecture Design 22](#_Toc468625878)

[1.2 Web Application Architecture Description 23](#_Toc468625879)

[1.3 Android Application Architecture Description 23](#_Toc468625880)

[3. Component Diagram 24](#_Toc468625881)

[4. Detailed Description 25](#_Toc468625882)

[5. Entity Relationship Diagram 26](#_Toc468625883)

[6. Algorithms 26](#_Toc468625884)

[E. Appendix 31](#_Toc468625885)

## List of Tables

[Table 1 : Roles and Responsibilities 10](#_Toc468347681)

[Table 2 : Hardware Requirement for Server 10](#_Toc468347681)

[Table 3 : Hardware Requirement for Mobile 11](#_Toc468347682)

[Table 4 : Software requirements 11](#_Toc468347683)

[Table 5 : Roles and responsibilities 13](#_Toc468347684)

[Table 6 : Tools List 13](#_Toc468347685)

[Table 7 : Technique List 13](#_Toc468347686)

[Table 8 : Software Development Life Cycle Detail 15](#_Toc468347687)

[Table 9 : Phase 1: Infrastructure 16](#_Toc468347688)

[Table 10 : Phase 2: System & Web app 17](#_Toc468347689)

[Table 11 : Phase 3: Web service 17](#_Toc468347690)

[Table 12 : Phase 4: Mobile app 17](#_Toc468347691)

[Table 13 : Conceptual Diagram Data Dictionary 21](#_Toc468347692)

[Table 14 : Component Diagram Data Dictionary 21](#_Toc468347692)

## List of Figures

[Figure 1 : Scrum model 12](#_Toc468348286)

[Figure 2: System Overview Use case 19](#_Toc468348287)

[Figure 3: Conceptual Diagram 20](#_Toc468348288)

[Figure 4: Web Application Architecture 22](#_Toc468348289)

[Figure 5: Mobile Application Architecture 23](#_Toc468348290)

[Figure 6: Component Diagram 24](#_Toc468348291)

[Figure 7: Component Diagram 24](#_Toc468348292)

[Figure 8: Class Diagram 25](#_Toc468348293)

[Figure 9: Entity Relationship Diagram 26](#_Toc468348294)

[Figure 10: Entity Relationship Diagram 25](#_Toc468348294)

[Figure 11: Flow Daily Usage Prediction 28](#_Toc468348294)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| GBTS | Green Bus Ticket System |
| NFC | Near Field Communication |

# Introduction

## Project Information

* Project name: **Green Bus Ticket System**
* Abbreviation: **GBTS**
* Product Type: **Web app & Mobile app**
* Start Date: **September 5th 2016**
* End Date: **December 2016**

## Introduction

Nowadays, bus is the most popular public transportation, buses are very safe, time efficient and cheap. In Vietnam, many people daily travel in these local buses from their house to their offices, schools and other places. For traveling by bus, people have to buy a paper bus ticket on a bus. The bus sometimes is crowded, buying ticket is not comfortable and inefficient. Moreover, the ticket is thrown anywhere after had used. The cost is spent very large and the ticket is created to garbage that cause wasting of wood resource. Based on researches and analysis, we proposed a solution for bus managers, passengers in Vietnam.

We build a system, which help bus managers, passengers to solve their current problems. In the process of analysis, we believe that NFC technology is suitable to resolve the problem by using NFC card to save passenger’s account information, so these NFC cards will be electronic bus tickets. NFC cards are cheap, small, convenient to bring with and easy to perform checking or validating. NFC cards are very durable, no-battery required and easy to rewrite information which is suitable to reuse. Beside of that, we also provide a system to help passengers manage their account, NFC cards and find buses based on their starting point and destination. Moreover, this system can help bus company manage their buses and tickets for income report.

## Current Situation

* For using bus, the passengers have to buy a paper bus ticket on a bus and pay cash for the driver assistant.
* Some buses route has already provided the buy ticket machine but the driver must take a cash, then must pay cash in return.

## Problem Definition

Research the current buying bus ticket process, we found that the traditional process has many advantages and disadvantages below:

* Advantages:
  + Easy for anyone to use the bus, they need to bring cash along and pay for the paper ticket.
  + No technical skills needed.
  + No need to bring NFC card along.
* Disadvantages:
  + Buying a paper bus ticket on a bus which sometimes is crowded is not comfortable and inefficient.
  + The ticket is thrown anywhere after had used is wasting of money and wood resource.
  + The ticket sets (monthly or annually) are easily being lost or fray. Besides, the driver assistance has not good behavior with those passengers.
  + Some buses route has already provided the buy ticket machine but the driver must take cash, then pay cash in return. It is not comfortable and inefficient both passenger and driver.

## Proposed Solution

Our solution is a new system, which will cover the whole bus ticket buy and sell process combine with NFC technology for buying bus ticket easy and efficient. In addition, our solution is also help passengers find buses route.

GBTS includes a web app and a mobile app, with the following features.

### Feature functions

* + Using NFC technology for storing passenger’s account information. The passengers can buy NFC card from bus companies or via online obly one time for using many times. The passengers use this NFC card as an electronic ticket each time they use bus.
  + There is an emulator on each bus which directly communicates with NFC card to access passenger account information. The emulator then sends information to the system to process. Next, the system checks, minus credit on related account and notify ticket details to passenger’s mobile phone (if they have already installed mobile application).
  + With web application, passengers can manage their account and outcome report. In the other hand, bus managers can manage their buses, tickets, passengers and income report. They can also notify or suggest the promotion campaign to passengers.
  + With mobile application, passengers can receive ticket details or promotion notification, manage their account and outcome report. The mobile application can also help them to find buses route.
  + In the other hand, bus companies also issue anonymous NFC cards with fixed balance. Passengers who forget their card can buy these anonymous cards for temporary using.

### Advantages and disadvantages

* Advantages:
  + - NFC card is cheaper and very durable, no-battery required, small and easy to bring with.
    - The system can replace the traditional way which always need paper bus tickets. We try to reduce garbage and save wood resource.
    - The system provides a new way to accost passengers with promotion or promotion.
    - Standardize the process and make them available to more and more transportation services.
  + Disadvantages:
    - Each passenger must have at least one primary NFC card to use the system and up to three additional NFC cards.
    - Cost of buying each bus an emulator in order to communicate with NFC card.

## Functional Requirements

Passenger component:

* + Activate for a new account.
  + Get NFC cards.
  + Add credit to card.
  + Edit card name
  + Get outcome report.
  + Find bus.

Staff component:

* + Search for passenger.
  + Edit passengers.
  + Manage NFC cards.

Manager component:

* + Manage ticket type.
  + Manage credit plan.
  + Get income report.
  + Create promotions.

Admin component:

* + Manage all accounts.

Emulator component:

* + Read & write NFC card.
  + Verify card.

Auto Handler component:

* + Suggest promotions
  + Send notification
  + Parse bus route

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | [khanhkt@fpt.edu.vn](mailto:khanhkt@fpt.edu.vn) |
| 2 | Đỗ Ngọc Hoàng | Developer | Leader | [hoangdnse61246@fpt.edu.vn](mailto:hoangdnse61246@fpt.edu.vn) |
| 3 | Trần Quang Trường | Developer | Member | [truongtqse61129@fpt.edu.vn](mailto:truongtqse61129@fpt.edu.vn) |
| 4 | Đoàn Minh Đức | Developer | Member | [ducdmse61486@fpt.edu.vn](mailto:ducdmse61486@fpt.edu.vn) |

Table 1: Roles and Responsibilities

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* **Official name**: Green Bus Ticket System
* **Vietnamese name**: Hệ thống bán vé xe buýt tiện lợi
* **Abbreviation**: GBTS

### Problem Abstract

For the goal of improving current bus system, especially the buying ticket process. We provide the solution for both passengers in buying, using bus ticket and the bus managers in managing efficiently. But there are many kinds of passengers may use our system, some of them don’t have any information technology skills like the old people. So we have to find the best convenient way to make our system simplest and easy to use for anyone.

Our system use NFC technology, this technology is quite new, so we may need times to research and integrate NFC to our system. Currently, only Android and Window Phone are supporting NFC technology, so we need knowledge on these operating systems for implementation.

### Project Overview

#### Current Situation

Below are the problems encountered in this project:

* **NFC security**: working with NFC, there are some problems may happen, any device support NFC like smartphone can read and write to this, so it can be counterfeited, attacked during data transmission caused data loss, data, corruption.
* **Emulator’s problem:** using emulator on bus in order to validate and process NFC cards is sometime not work or damaged.
* **Passenger’s habitats**: passengers are used to buying paper ticket with cash, so deploy the system in real life may take long time.
* **Account information security**: the system allow passenger to buy ticket credits, this function may become target for hacking and cheating.

#### The Proposed System

* After doing many researches on technology for saving information, we choose NFC technology as this technology is very capable of resolve the current situations in selling bus tickets. The basic idea is to use a NFC tag that each NFC card contain a unique card ID as a ticket that can be reused instead of using paper.
* In task assignment, we assign to member using vertical model to make sure if any member in this problem cannot continue to work in our team there will be the least harmful to the project processes.
* Our system includes three subsystems:
  + An online web application for passengers, staffs, managers and administrator.
  + A mobile application for passengers.
  + A mobile application for emulator.

##### ***Web Application***

Web application consists of three main parts:

* For passengers:
  + Activate for a new account.
  + Get NFC cards.
  + Add credit to card.
  + Edit card name
  + Get outcome report.
  + Find bus.
* For staff:
  + Search for passenger.
  + Edit passengers.
  + Manage NFC cards.
  + Publish scratch cards.
* For manager:
  + Manage ticket type.
  + Manage credit plan.
  + Get income report.
  + Manage offer subscription
  + Create promotions.
* For administrator:
  + Manage accounts.

Besides, website application also provides an API interface for two mobile applications to retrieve, update data from mobile applications.

##### ***Mobile Application***

There will be 2 applications which will be used by passengers and emulator. The mobile applications included functions as below:

* For passengers:
  + Activate for a new account.
  + Get NFC cards.
  + Add credit to card.
  + Edit card name
  + Get outcome report.
  + Find bus.
  + Buy ticket by phone.
* For emulator:
  + Read & write NFC card.
  + Verify card.

#### Boundaries of the System

* The system is mostly built based on real processes of bus ticket in Ho Chi Minh City. Our main target is improving the current process and makes it more convenient and efficient in Ho Chi Minh City.
* Any bus system which deployed this system must set up devices to operate, includes:
  + Emulator can read a NFC Card, with internet connection.
  + NFC cards with account information.
* The completed product includes:
  + Website application
  + Android mobile application for passengers and for emulator.

#### Future Plans

With further research and development, the system can apply the following features:

* Bus companies can place many emulators at bus stop with an assistant for helping the passengers recharge their NFC card credit without using mobile app or website. It is suitable for any passenger who lack of knowledge in information technology.
* Allow passengers add credit to their account. The system will minus credit on their account automatically while buying ticket in case of their cards is out of credit.

#### Development Environment

##### ***Hardware requirements***

* **For web application server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Window Server 2008 R2 | Window Server 2012 R2 |
| Computer Processor | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core |
| Computer Memory | 2GB of RAM | 4GB of RAM or more |

Table 2 : Hardware Requirement for Server

* **For Mobile**

|  |  |  |
| --- | --- | --- |
| Android | Minimum | Recommended |
| Internet Connection | Wi-Fi or 3G (1 Mbps) | Wi-Fi or 3G (8 Mbps) |
| Operating System | Android 4.4.2 | Android 6.0.0 |
| Mobile Processor | Cortex-A7 Dual-Core 1.3GHz | Cortex-A7 Dual-Core 1.3GHz |
| Mobile Memory | 1GB of RAM | 2GB of RAM or more |

Table 3 : Hardware Requirement for Mobile

##### ***Software requirements***

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Window Server 2012 R2 | Operating system and platform for development |
| Environment | .NET Framework 4.5 | Specification for developing web application |
| IDE | Visual Studio 2015, Android Studio v2.1 | Used for implement website and Android Mobile App. |
| Design Model tool | StartUML v2.5.1 | Used for creating modal and diagrams. |
| DBMS | Microsoft SQL Server 2014 | Used to create & manage the database for system |
| Document storage | Github | Used for storing document |
| Store and manage source code | Github & SourceTree | Used to store all source code |

Table 4 : Software requirements

## Project organization



### Software Process Model

The project is developed under scrum model. Scrum model is capable with current situation in our team. We choose this model because the following reasons:

* The bus ticket problem not fully defined and the bus business in company cannot be fully understood. The users of our system are vary, so we may have many changes during development process to adapt the requirements.
* This project use NFC technology, which is a new technology that may need many times to research and implement.
* Scrum adopts an empirical approach, accepting that the problem is not fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly, to respond to emerging requirements and to adapt to evolving technologies and changes in market conditions.



Figure 1 : Scrum model

Reference: <http://skytechnovation.com/scrum-development-model/>

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Kiều Trọng Khánh | Supervisor, Project Manager | * Specify user requirements * Control the development process * Give out technique and business analysis support |
| 2 | Đỗ Ngọc Hoàng | Team leader, B.A, Developer, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 3 | Trần Quang Trường | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |
| 4 | Đoàn Minh Đức | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * Create test plan * Coding * Test |

Table 5 : Roles and responsibilities

### Tools and Techniques

|  |  |
| --- | --- |
| Tool | Name / version |
| Web server | IIS |
| Development tool | Visual Studio, Android Studio |
| DBMS | SQL Server 2014 |
| Source control | Github & SourceTree |
| Modeling tool | StarUML v5.0.1 |
| Document tool | Microsoft Word 2010 |

Table 6: Tools List

|  |  |
| --- | --- |
| Technique | Name / version |
| Frontend | HTML5, CSS, JavaScript, jQuery |
| Backend | ASP.Net, Android, NFC |

Table 7: Technique List

## Project Management Plan



### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Description | Deliverables | Resource needed | Dependencies and Constrains | Risks |
| Infrastructure | - Identify and clarify overall requirements.  - Determine the system architecture.  - Build infrastructure for the project. | - Database design.  - System main structure. | 20 man-days |  | - Unclear project scope.  - Lack of member share of understand. |
| System  &  Web app | - Identify software and hardware requirements.  - Implements all web app modules.  - Design the web UI  - Build the web app | - Complete web app for all roles of the system. | 60 man-days | - Depends on “Infrastructure” | - Unclear project scope.  - Lack of experience. |
| Web services | - Identify requirements for mobile app.  - Build required API for mobile app. | - API for mobile app. | 20 man-days | - Depends on “Web app & System” | - Lack of experience. |
| Mobile apps | - Design the mobile UI  - Build mobile apps for end users and emulator. | - Complete Android Apps | 20 man-days | - Depends on “Web services” | - Lack of experience.  - Lack of NFC knowledge |

Table 8: Software Development Life Cycle Detail

### Phase Detail

#### Phase 1: Infrastructure

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements.  - Create product backlog. | * HoangDN * DucDM * TruongTQ |
| 2. Selection | - Determine system architecture: ASP .NET MVC.  - Determine software design pattern: Repository & Service.  - Determined all core functions. | * HoangDN * DucDM * TruongTQ |
| 3. Development | - Create the main structure of project. | * HoangDN * DucDM * TruongTQ |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * HoangDN * DucDM * TruongTQ |

Table 9: Phase 1: Infrastructure

#### Phase 2: System & Web app

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements for System and Web app.  - Update product backlog. | * HoangDN * DucDM * TruongTQ |
| 2. Selection | - Determine all functions according to requirements of System and Web app. | * HoangDN * DucDM * TruongTQ |
| 3. Development | - Design and build prototype for web UI  - Create conceptual diagram  - Design class diagram  - Design database  - Implement the entire web UI: layouts, detail pages, etc.  - Implement all the functions in controllers.  - Build needed utility classes | * HoangDN * DucDM * TruongTQ |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * HoangDN * DucDM * TruongTQ |

Table 10: Phase 2: System & Web app

#### Phase 3: Web service

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements for Web service.  - Update product backlog. | * HoangDN * DucDM * TruongTQ |
| 2. Selection | - Determine all functions according to requirements of Web service. | * HoangDN * DucDM * TruongTQ |
| 3. Development | - Create API for mobile app based on functions on the web app. | * HoangDN * DucDM * TruongTQ |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * HoangDN * DucDM * TruongTQ |

Table 11: Phase 3: Web service

#### Phase 4: Mobile app

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements for System and Mobile app.  - Update product backlog. | * HoangDN * DucDM * TruongTQ |
| 2. Selection | - Determine all functions according to requirements of Mobile app. | * HoangDN * DucDM * TruongTQ |
| 3. Development | - Implement all the functions based on the designed UI and the provided API. | * HoangDN * DucDM * TruongTQ |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * HoangDN * DucDM * TruongTQ |

Table 12: Phase 4: Mobile app

### All Meeting Minutes

Meeting minutes are contained in folder “Meeting minutes” in the attached CD.

# Software Requirement Specification

## User Requirement Specification

### Unauthorized User Requirement

Unauthorized is user does not login to this system. Unauthorized only has two functions.

* Activate new account.
* Login.

### Authorized User Requirement

Authorized User is user who has logged into system. Authorized User only has two functions.

* Edit profile.
* Logout.

### Passenger Requirement

Passenger can use some following functions:

* + Get NFC cards
* Add credit to card
* Edit card name
* Get outcome report
* Find bus

### Staff Requirement

Staff is the user who interacts directly to passengers, they can use some following functions:

* Search for passenger
* Edit passengers
* Manage NFC cards
* Publish scratch cards

### Manager Requirement

Manager is the Staff supervisor, they can use some following functions:

* Manage ticket type
* Manage credit plan
* Get income report
* Manage offer subscription
* Create promotions

### Admin Requirement

Admin is the user who responsible for managing accounts for the whole system, they can use some following functions:

* Manage account

### Emulator Requirement

Emulator is the device which can interact with NFC card, it can use some following functions:

* Read & write NFC card
* Verify card

### Auto Handler Requirement

Auto Handler can use some following functions:

* + Suggest promotions
  + Auto extend subscription
  + Crawl bus routes
  + Auto generate server token

## System Requirement Specification



Figure 2 - System Overview Use case

## Conceptual Diagram



Figure 3 - Conceptual Diagram

**Data Dictionary:**

|  |  |
| --- | --- |
| Entity Name | Description |
| User | Contains all properties about user. |
| Passenger | Contains all properties about passenger info. |
| Admin | Contains all properties about admin info. |
| Staff | Contains all properties about staff info. |
| Manager | Contains all properties about manager info. |
| Card | Contains all properties about NFC card. |
| Ticket Type | Contains all properties about ticket type for based on bus route, each ticket type has different price. |
| Bus Route | Contains all properties about bus route in Ho Chi Minh City. |
| Ticket | Contains all properties about ticket to specify which card buy ticket belongs to which ticket type on which bus route. |
| Credit Plan | Contains all info about credit plan which will be chosen while adding credit to card. |
| Payment Transaction | Contains all info about transaction when adding credit to card via cash, credit plan or scratch card. |
| Promotion | Contains all info about promotion. |
| Scratch Card | Contains all info about scratch card. |
| Offer Subscription | Contains all info about offer subscription. |
| User Subscription | Contains all info about user subscription to specify who subscribe which offer subscription. |
| User Subscription | Contains all info about user subscription to specify who subscribe which offer subscription. |

Table 14 – Component Diagram Data Dictionary

# Software Design Description

## Design Overview

* 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: gives the interface design description, which includes component interface, web application interface, and mobile application design.
* Section 6: describe a fully attributed Entity Relationship Diagram.
* Section 7: describe the algorithms that apply in the system.

## System Architecture Design

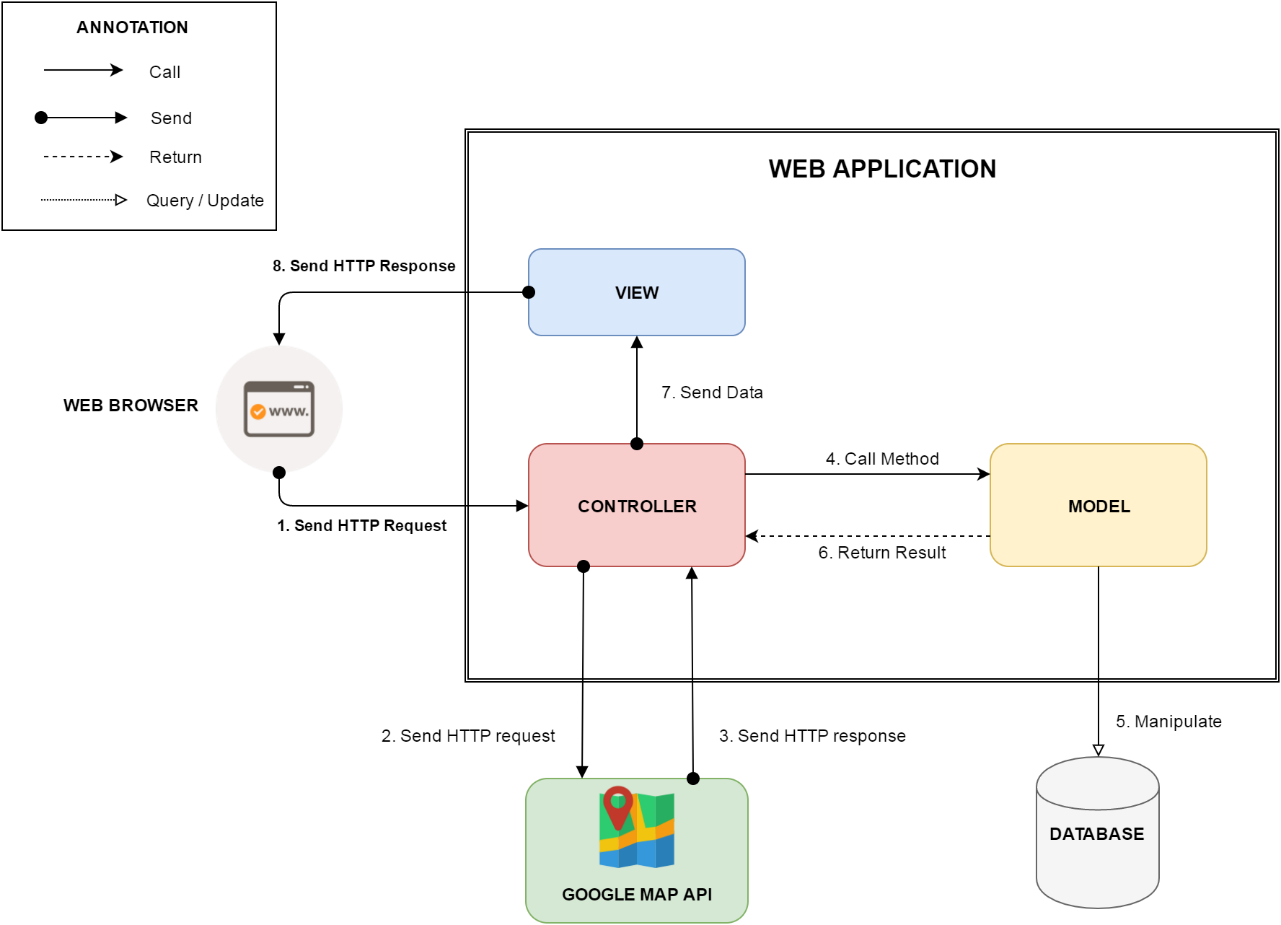


Figure 4: Web Application Architecture

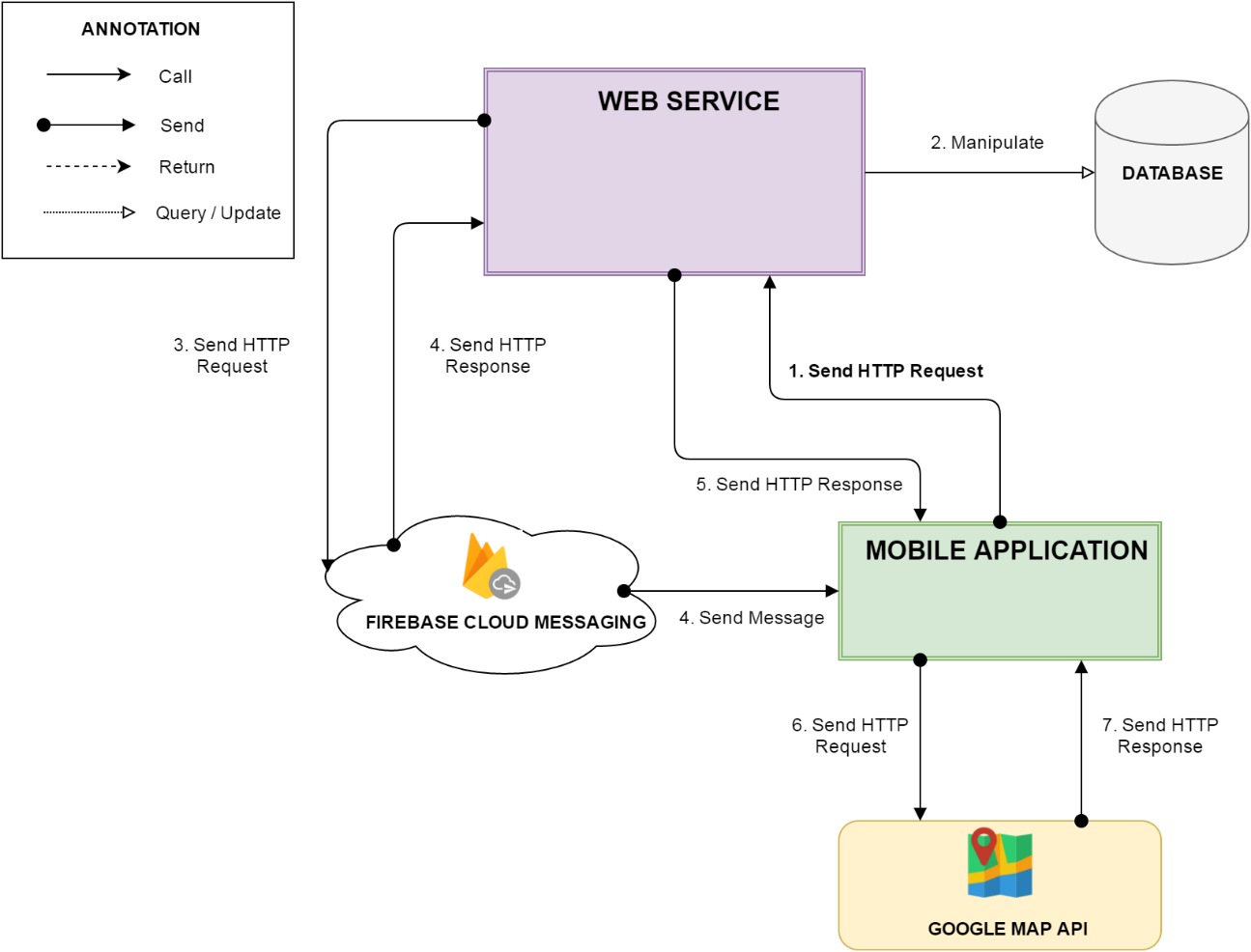


Figure 5: Mobile Application Architecture

### Web Application Architecture Description

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

* Web application contains Web services 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.
* We can organize the code better for maintainability, extensibility, reusability so we can expand the scope to other kind of illnesses such as flu, fever…
* In scope of 3-members team, MVC architecture makes it easier to split the big project into small modules and make it easier to assign each module for members in our team.

### Android Application Architecture Description

In our Android application, the application is developed under MVC architecture style. We choose this architecture for Android application because of following advantages:

* We can organize the code better for maintainability, extensibility and reusability.

## Component Diagram



Figure 6: Component Diagram

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| Mobile application | Mobile application package |
| Web service | Provide API for mobile application to interact with the system. |
| Service | Common component is used to handle system’s business operations. |
| Admin Component | Component to handler admin activities in the system |
| Passenger Component | Component to handler passenger activities in the system |
| Staff Component | Component to handler staff activities in the system |
| Manager Component | Component to handler manager activities in the system |
| Repository | Component is used to handle interaction between the system and database. |
| Auto Handler Component | Component is used to handle scheduler in the system |
| Model | Entity framework |
| Google Firebase Service | Handle push notification with Google Firebase Message |

Figure 7 – Component Diagram

## Detailed Description



Figure 8: Class Diagram

## Entity Relationship Diagram

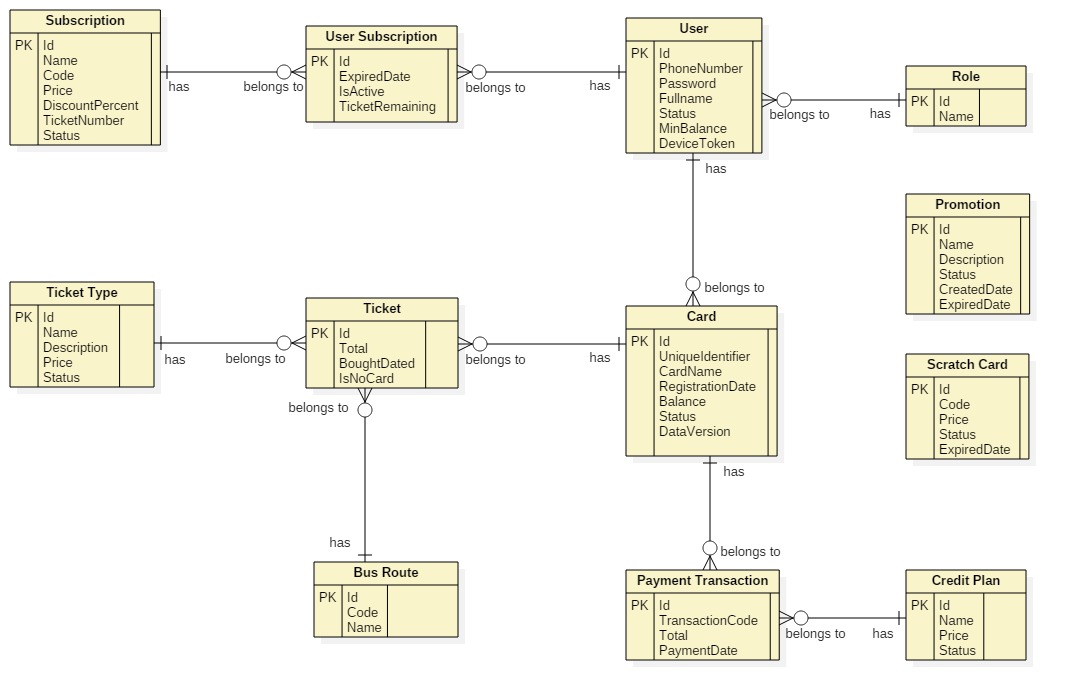


Figure 9: Entity Relationship Diagram

## Algorithms

1. **AES Encryption**
   1. Definition

AES (acronym of Advanced Encryption Standard) is a symmetric encryption algorithm. The algorithm was developed by two Belgian cryptographer Joan Daemen and Vincent Rijmen. AES was designed to be efficient in both hardware and software, and supports a block length of 128 bits and key lengths of 128, 192, and 256 bits.

* 1. Define Problem

For convenience purposes of allowing passenger to buy ticket in case of no internet connection from emulator to server, we save card’s balance and data version to NFC card. We need to encrypt these information to make sure no one can change the original information.

* 1. Solution

In order to secure these information, we use AES Encryption Algorithm to encrypt information before saving to NFC card.

Algorithm reference: <http://howtodoinjava.com/security/java-aes-encryption-example/>

1. **Token Authentication**
2. Definition

Token authentication is a technique to prove that the passengers is who they claim to be. Requests which have the right token will be processed.

1. Define problem

When the passengers use mobile application to buy ticket instead of their cards. Mobile application sends some information to emulator for verification, we have to make sure that these information only valid in a specific time for buying ticket, anyone who can capture these information will not be able to use it and buy ticket later.

1. Solution

Token we use in this solution is a GUID. GUIDs are stored as 128-bit values, and are displayed as 32 hexadecimal digits with groups separated by hyphens.

GUID Reference: <https://en.wikipedia.org/wiki/Globally_unique_identifier>

To solve this problem, we should follow these steps:

* Server creates a token if there is no token on the server.
* Server change the token frequently (every 30 minutes).
* When mobile application is used to buy ticket, it asks the token from server.
* Then mobile application sends a request along with this token to emulator.
* Emulator sends to server. Server verifies the received token, if it matches current server token, request will be processed.

1. Complexity: O(n)
2. Flowchart

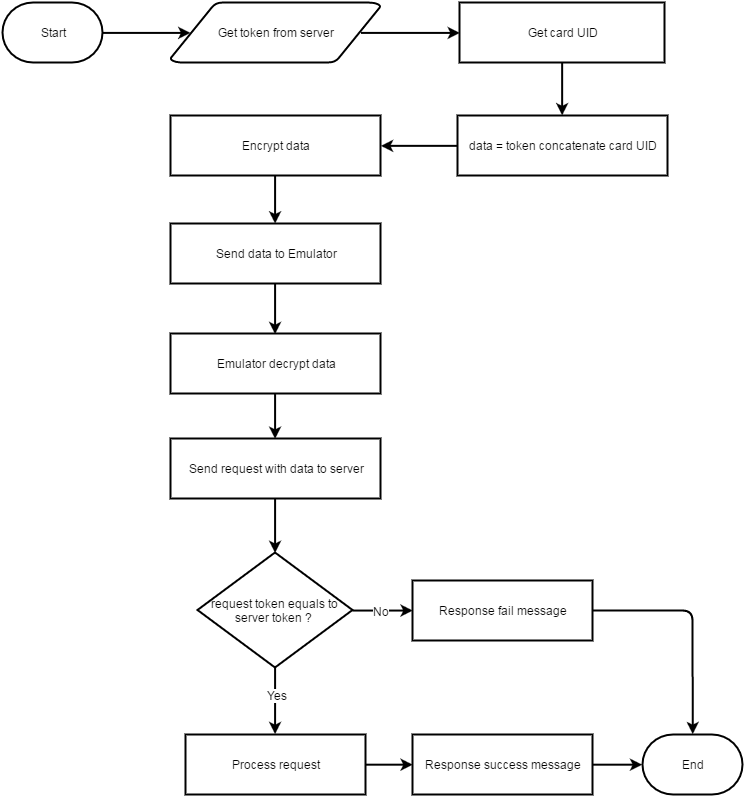


Figure 10: Flow Token Authentication

1. **Daily Usage Prediction**
2. Definition

Daily Usage Prediction is the way to predict how much a passenger need to spend for buying ticket in the next day base on the average bought tickets in the past.

1. Define Problem

Passenger sometimes forget to add credit to their cards. Each passenger has a diffirent travel demand. We should notify them in the morning if the remaining balance in their card is not enough for spending in the next day.

1. Solution

To solve this problem, we should follow these steps:

* Everyday, at a specific time, Auto Handler will filter the current passenger list to get passengers who have already installed mobile application, bought at least 30 tickets before and the last bought ticket must not exceed 7 days from the current date.
* Calculate the average spending for each customer and on each card of them:

**Average spending** =

**Total Spending**: Total money spent on a card in last 30 days.

**Number of dates**: Number of distinct dates which have at least one ticket was bought in last 30 days.

* If the current card’s balance less than the average spending above, a notification will be send to passenger’s mobile for reminding them.

1. Complexity: O(n2)
2. Flowchart

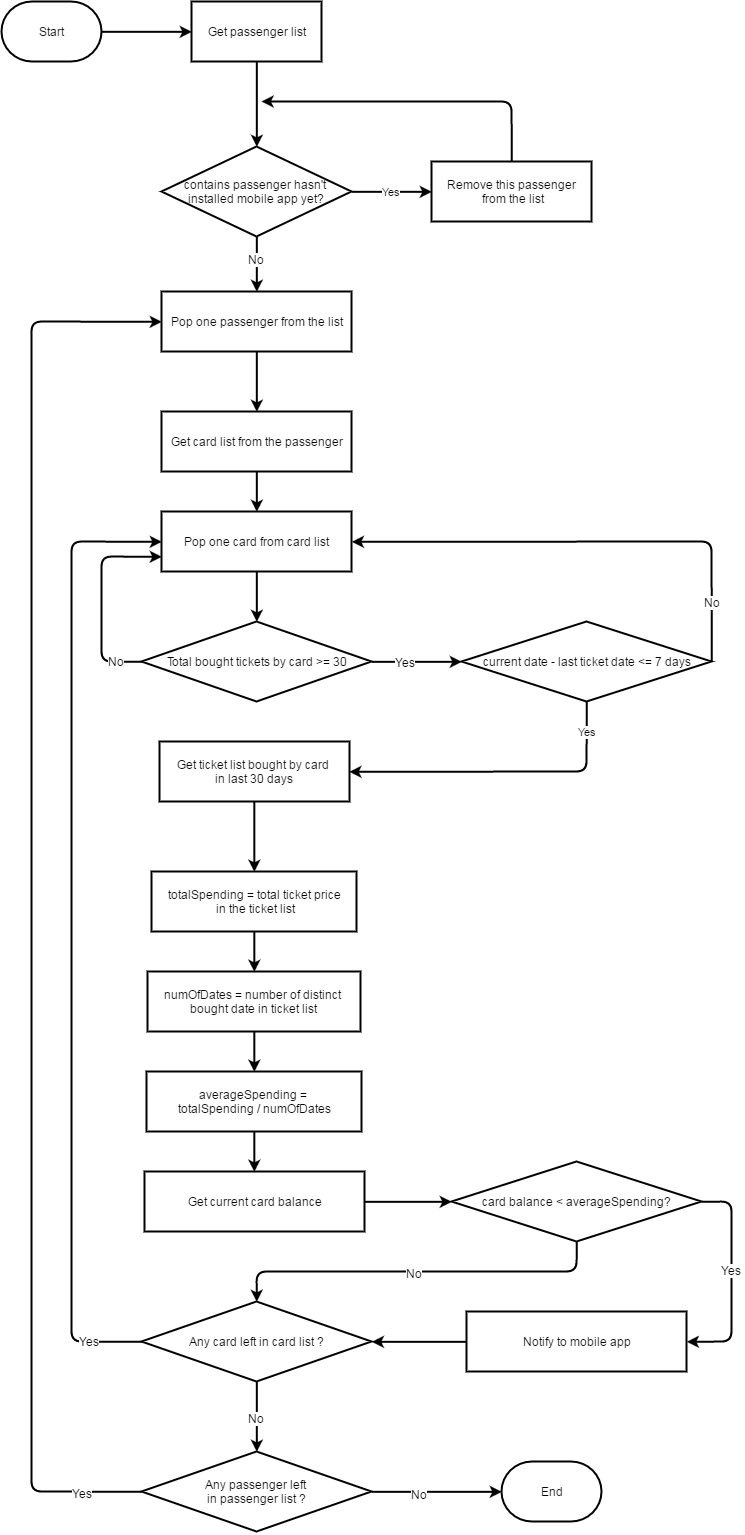


Figure 11: Flow Daily Usage Prediction

# Appendix

1. UML Documentation of IBM <http://www.ibm.com/developerworks/rational/library/769.html?ca=drs->
2. AES encryption

[http://](http://www.ibm.com/developerworks/rational/library/769.html?ca=drs-)www.java2s.com/Code/Java/Security/EncryptionanddecryptionwithAESECBPKCS7Padding.htm

1. Android Developer Guide – Application Fundamentals

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

1. FPT Vietnamese Accentizer

[http://doc.openfpt.vn/services/vnaccent/documentation.html#/README](http://doc.openfpt.vn/services/vnaccent/documentation.html%23/README)

1. Google Firebase Api

<https://firebase.google.com/docs/reference/>