**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 5](#_Toc464338621)

[List of Figures 6](#_Toc464338622)

[Definitions, Acronyms, and Abbreviations 7](#_Toc464338623)

[A. Report No. 1 Introduction 8](#_Toc464338624)

[1. Project Information 8](#_Toc464338625)

[2. Introduction 8](#_Toc464338626)

[3. Current Situation 8](#_Toc464338627)

[4. Problem Definition 8](#_Toc464338628)

[5. Proposed Solution 9](#_Toc464338629)

[5.1 Feature functions 9](#_Toc464338630)

[5.2 Advantages and disadvantages 9](#_Toc464338631)

[6. Functional Requirements 10](#_Toc464338632)

[7. Role and Responsibility 10](#_Toc464338633)

[B. Report No.2 Software Project Management Plan 12](#_Toc464338634)

[1. Problem Definition 12](#_Toc464338635)

[1.1. Name of this Capstone Project 12](#_Toc464338636)

[1.2. Problem Abstract 12](#_Toc464338637)

[1.3. Project Overview 12](#_Toc464338638)

[2. Project organization 15](#_Toc464338639)

[2.1. Software Process Model 15](#_Toc464338642)

[2.2. Roles and responsibilities 16](#_Toc464338643)

[2.3. Tools and Techniques 16](#_Toc464338644)

[3. Project Management Plan 18](#_Toc464338645)

[3.1. Software development life cycle 18](#_Toc464338647)

[3.2. Phase Detail 20](#_Toc464338648)

[3.3. All Meeting Minutes 22](#_Toc464338649)

[4. Coding Convention 22](#_Toc464338650)

[C. Report No. 3 Software Requirement Specification 23](#_Toc464338653)

[1. User Requirement Specification 23](#_Toc464338654)

[1. Unauthorized User Requirement 23](#_Toc464338655)

[2. Authorized User Requirement 23](#_Toc464338656)

[3. Passenger Requirement 23](#_Toc464338657)

[4. Staff Requirement 23](#_Toc464338658)

[5. Manager Requirement 23](#_Toc464338659)

[6. Admin Requirement 23](#_Toc464338660)

[7. Emulator Requirement 23](#_Toc464338661)

[8. Auto Handler Requirement 24](#_Toc464338662)

[2. System Requirement Specification 24](#_Toc464338663)

[1. External Interface Requirement 24](#_Toc464338665)

[2. System Overview Use Case 25](#_Toc464338666)

[3. List of Use Case 26](#_Toc464338667)

[3. Software System Attribute 84](#_Toc464338668)

[1. Usability 84](#_Toc464338670)

[2. Reliability 85](#_Toc464338671)

[3. Availability 85](#_Toc464338672)

[4. Security 85](#_Toc464338673)

[5. Maintenanability 85](#_Toc464338674)

[6. Portability 85](#_Toc464338675)

[7. Performance 85](#_Toc464338676)

[4. Conceptual Diagram 86](#_Toc464338677)

[D. Report No. 3 Software Requirement Specification 87](#_Toc464338679)

[1. Design Overview 87](#_Toc464338680)

[2. System Architecture Design 87](#_Toc464338681)

[4.2 Web Application Architecture Description 88](#_Toc464338682)

[4.3 Android Application Architecture Description 88](#_Toc464338683)

[3. Component Diagram 88](#_Toc464338684)

[4. Detailed Description 90](#_Toc464338685)

[4.1 Class Diagram 90](#_Toc464338688)

[4.2 Class diagram explanation 90](#_Toc464338689)

[4.3 Interactive Diagram 93](#_Toc464338690)

[4.4 User Interface 99](#_Toc464338691)

## List of Tables

[Table 1 : Roles and Responsibilities 11](#_Toc464338694)

[Table 2 : Hardware Requirement for Server 14](#_Toc464338695)

[Table 3 : Hardware Requirement for Mobile 14](#_Toc464338696)

[Table 4 : Software requirements 15](#_Toc464338697)

[Table 5 : Roles and responsibilities 16](#_Toc464338698)

[Table 6: Tools List 17](#_Toc464338699)

[Table 7: Technique List 17](#_Toc464338700)

[Table 8: Software Development Life Cycle Detail 19](#_Toc464338701)

[Table 9: Phase 1: Infrastructure 20](#_Toc464338702)

[Table 10: Phase 2: System & Web app 21](#_Toc464338703)

[Table 11: Phase 3: Web service 21](#_Toc464338704)

[Table 12: Phase 4: Mobile app 22](#_Toc464338705)

[Table 13: <Unauthorized User> Login 28](#_Toc464338706)

[Table 14: <Unauthorized User> Activate account 29](#_Toc464338707)

[Table 15: < Authorized User> Overview use case 31](#_Toc464338708)

[Table 16: < Authorized User> Log out 32](#_Toc464338709)

[Table 17: < Passenger> Get NFC card 35](#_Toc464338710)

[Table 18: < Passenger> Add credit 37](#_Toc464338711)

[Table 19: < Passenger> Edit card name 38](#_Toc464338712)

[Table 20: < Passenger> Get outcome report 41](#_Toc464338713)

[Table 21: < Passenger> Find bus 42](#_Toc464338714)

[Table 22: < Passenger> Get route 44](#_Toc464338715)

[Table 23: < Staff > Add card 46](#_Toc464338716)

[Table 24: < Staff > Activate/Deactivate card 48](#_Toc464338717)

[Table 25: < Staff > Search passenger 49](#_Toc464338718)

[Table 26: < Staff > Edit passenger 52](#_Toc464338719)

[Table 27: < Manager > Add ticket type 54](#_Toc464338720)

[Table 28: < Manager > Edit ticket type 57](#_Toc464338721)

[Table 29: < Manager > Delete ticket type 58](#_Toc464338722)

[Table 30: < Manager > Add credit plan 60](#_Toc464338723)

[Table 31: < Manager > Edit credit plan 62](#_Toc464338724)

[Table 32: < Manager > Delete credit plan 64](#_Toc464338725)

[Table 33: < Manager > Get income report 66](#_Toc464338726)

[Table 34: < Manager > Create promotion 68](#_Toc464338727)

[Table 35: < Admin> Search account 69](#_Toc464338728)

[Table 36: < Admin> Add account 72](#_Toc464338729)

[Table 37: < Admin> Edit account 73](#_Toc464338730)

[Table 38: < Admin> Delete account 75](#_Toc464338731)

[Table 39: < Auto Handler > Parse bus route 77](#_Toc464338732)

[Table 40: < Auto Handler > Suggest promotion 79](#_Toc464338733)

[Table 41: < Auto Handler > Send notification 81](#_Toc464338734)

[Table 42: < Emulator > Write card 83](#_Toc464338735)

[Table 43: < Emulator > Verify ticket 84](#_Toc464338736)

[Table 44: Conceptual Diagram Data Dictionary 87](#_Toc464338737)

[Table 45: Component Dictionary 89](#_Toc464338738)

## List of Figures

[Figure 1 : Scrum model 17](#_Toc464338739)

[Figure 2 : <Use case> Web application overview 26](#_Toc464338740)

[Figure 3 : <Use case> Mobile application overview 27](#_Toc464338741)

[Figure 4: <Unauthorized User> Overview use case 27](#_Toc464338742)

[Figure 5: <Unauthorized User> Login 27](#_Toc464338743)

[Figure 6: <Unauthorized User> Activate account 29](#_Toc464338744)

[Figure 7: < Authorized User> Overview use case 30](#_Toc464338745)

[Figure 8: < Authorized User> Overview use case 31](#_Toc464338746)

[Figure 9: < Authorized User> Log out 32](#_Toc464338747)

[Figure 10: < Passenger> Overview use case 34](#_Toc464338748)

[Figure 11: < Passenger> Get NFC card 34](#_Toc464338749)

[Figure 12: < Passenger> Add credit 36](#_Toc464338750)

[Figure 13: < Passenger> Edit card name 38](#_Toc464338751)

[Figure 14: < Passenger> Get outcome report 40](#_Toc464338752)

[Figure 15: < Passenger> Find bus 42](#_Toc464338753)

[Figure 16: < Passenger> Get route 44](#_Toc464338754)

[Figure 17: < Staff > Overview use case 45](#_Toc464338755)

[Figure 18: < Staff > Add card 46](#_Toc464338756)

[Figure 19: < Staff > Activate/Deactivate card 48](#_Toc464338757)

[Figure 20: < Staff > Search passenger 49](#_Toc464338758)

[Figure 21: < Staff > Edit passenger 51](#_Toc464338759)

[Figure 22: < Manager > Overview use case 53](#_Toc464338760)

[Figure 23: < Manager > Add ticket type 53](#_Toc464338761)

[Figure 24: < Manager > Edit ticket type 56](#_Toc464338762)

[Figure 25: < Manager > Delete ticket type 58](#_Toc464338763)

[Figure 26: < Manager > Add credit plan 59](#_Toc464338764)

[Figure 27: < Manager > Edit credit plan 61](#_Toc464338765)

[Figure 28: < Manager > Delete credit plan 63](#_Toc464338766)

[Figure 29: < Manager > Get income report 65](#_Toc464338767)

[Figure 30: < Manager > Create promotion 67](#_Toc464338768)

[Figure 31: < Admin> Overview use case 69](#_Toc464338769)

[Figure 32: < Admin> Search account 69](#_Toc464338770)

[Figure 33: < Admin> Add account 71](#_Toc464338771)

[Figure 34: < Admin> Edit account 73](#_Toc464338772)

[Figure 35: < Admin> Delete account 75](#_Toc464338773)

[Figure 36: < Auto Handler > Overview use case 77](#_Toc464338774)

[Figure 37: < Auto Handler > Parse bus route 77](#_Toc464338775)

[Figure 38: < Auto Handler > Suggest promotion 79](#_Toc464338776)

[Figure 39: < Auto Handler > Send notification 81](#_Toc464338777)­

[Figure 40: < Emulator > Overview use case 82](#_Toc464338778)

[Figure 41: < Emulator > Write card 83](#_Toc464338779)

[Figure 42: < Emulator > Verify ticket 84](#_Toc464338780)

[Figure 43: Conceptual Diagram 87](#_Toc464338781)

[Figure 44: System Architecture 88](#_Toc464338782)

[Figure 45: Component Diagram 89](#_Toc464338783)

[Figure 46: Class Diagram 91](#_Toc464338784)

[Figure 47: <Sequence Diagram> Add Credit 95](#_Toc464338785)

[Figure 48: <Sequence Diagram> Get Cards 95](#_Toc464338786)

[Figure 49: <Sequence Diagram> Get Income Report 96](#_Toc464338787)

[Figure 50: <Sequence Diagram> Edit Passenger 96](#_Toc464338788)

[Figure 51: <Sequence Diagram> Activate/Deactivate Card 97](#_Toc464338789)

[Figure 52: <Sequence Diagram>Add Card 97](#_Toc464338790)

[Figure 53: <Sequence Diagram>Get Outcome Report 98](#_Toc464338791)

[Figure 54: <Sequence Diagram>Create Promotion 98](#_Toc464338792)

[Figure 55 : <Activity Diagram> Add Credit 99](#_Toc464338793)

[Figure 56: <Activity Diagram>Find Bus 100](#_Toc464338794)

[Figure 57: <Activity Diagram> Get Outcome Report 100](#_Toc464338795)

# Definitions, Acronyms, and Abbreviations

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

# 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 knowledges 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.
* **Passengers’s habitat**: passengers are used to buying paper ticket with cash, so deploy the system in real life may take long time.
* **Account information secutiry**: the system allow passenger to buy ticket credits, this function may becom tatget 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 tikets. 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.  - Determind 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 | - Determind 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 | - Determind 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 | - Determind 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 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 diffirent 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. |

# 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

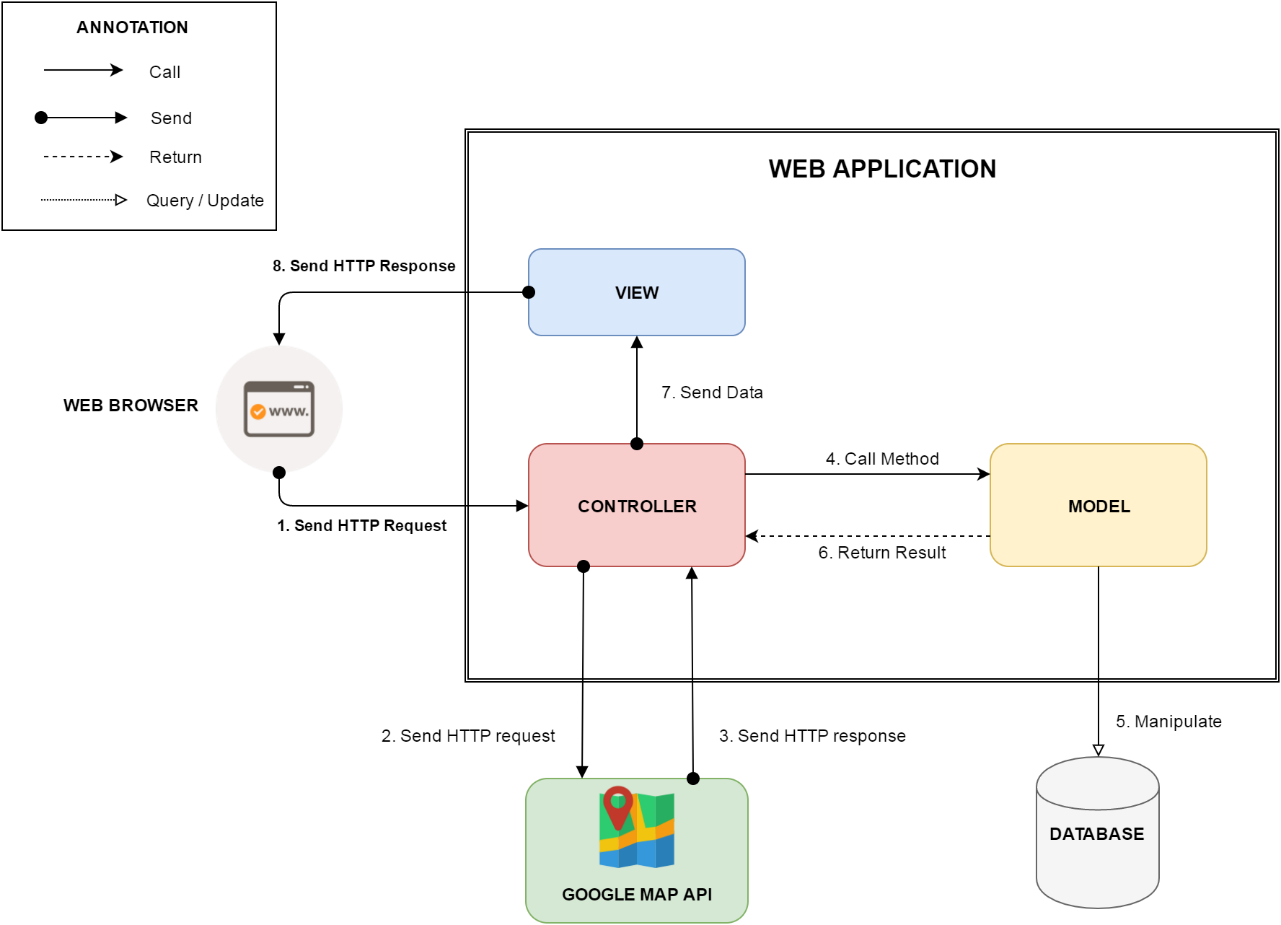


Figure 44: Web Application Architecture

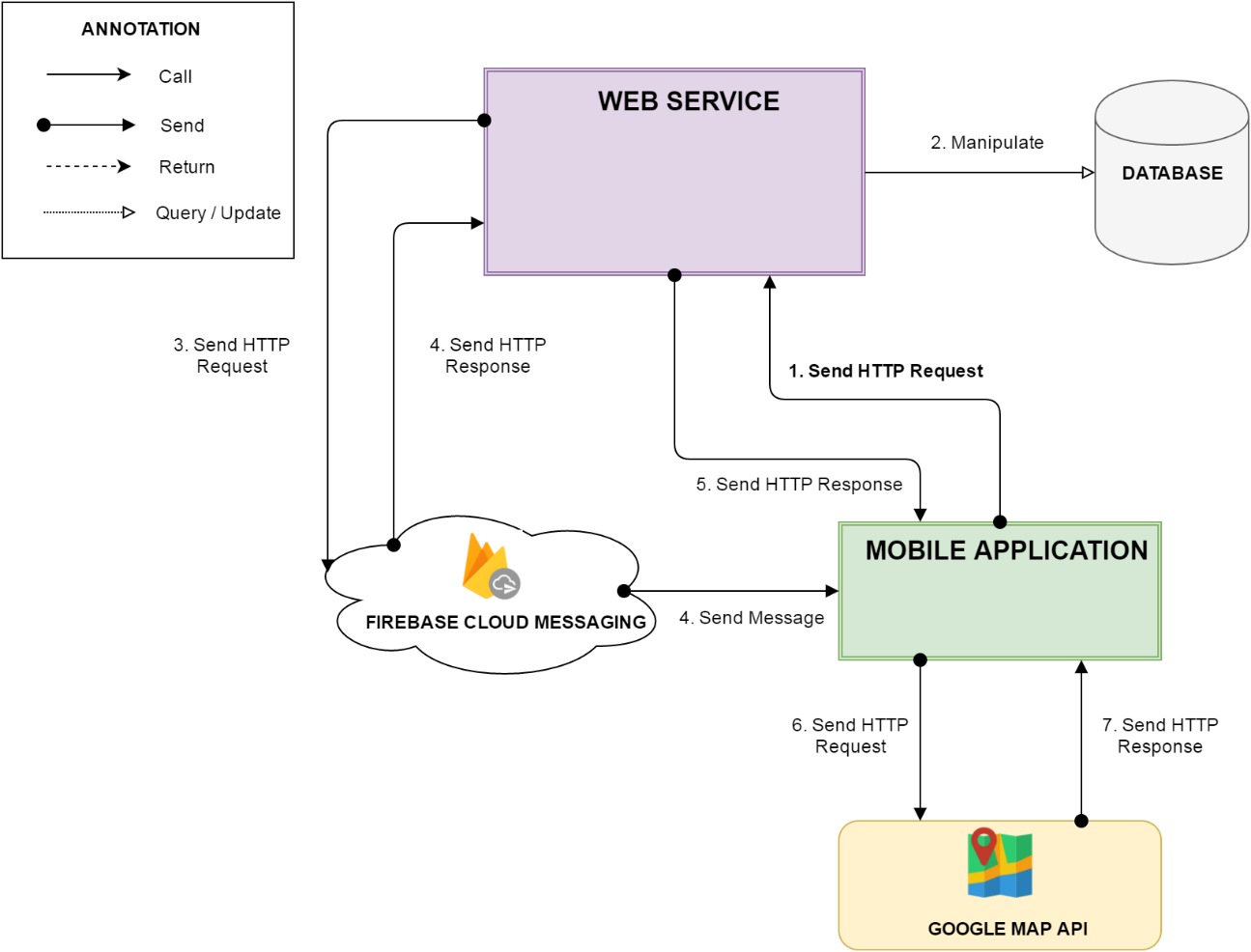


Figure 44: Mobile Application Architecture

### 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 45: 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 |

## Detailed Description



Figure 46: Class Diagram

## Entity Relationship Diagram

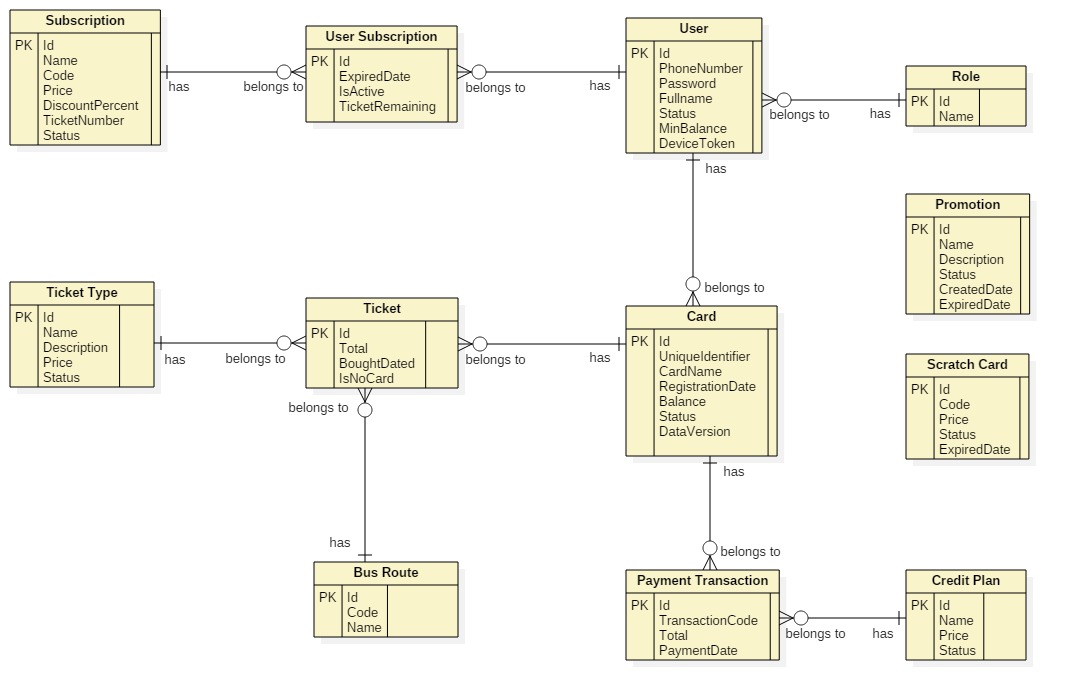


Figure 18 – Entity Relationship Diagram

## 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 convevient 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 informations, we use AES Encryption Algorithm to encrypt informations 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 informations to emulator for verification, we have to make sure that these informations only valid in a specific time for buying ticket, anyone who can capture these informations will not be able to use it and buy ticket later.

1. Solution

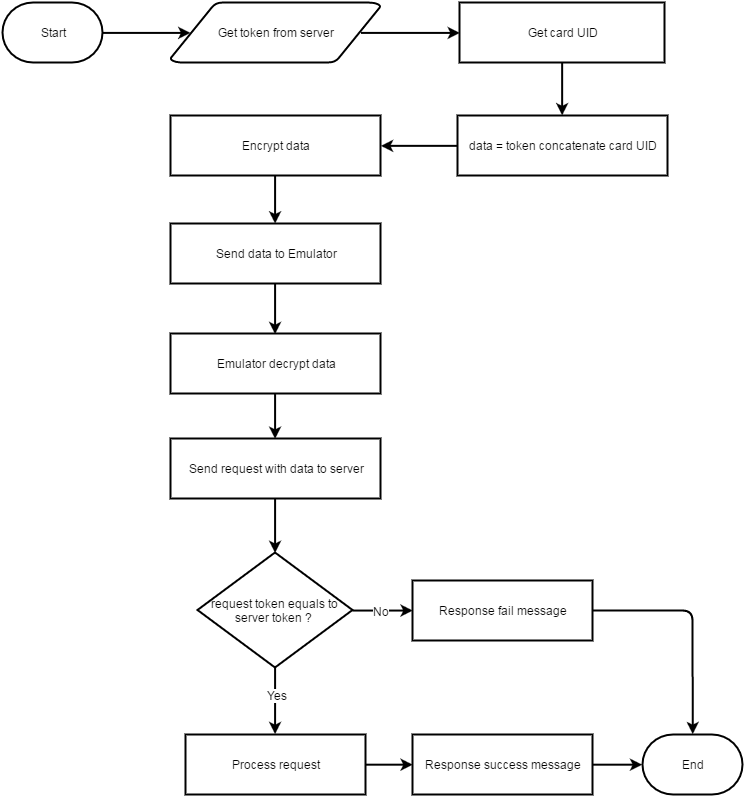
Token we use in this solution is a GUID. GUIDs are stored as 128-bit values, and are displayed as 32 hexadecimal digits with groups separated by hyphens.

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 matchs current server token, request will be processed.

1. Complexity: O(n)
2. Flowchart



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:

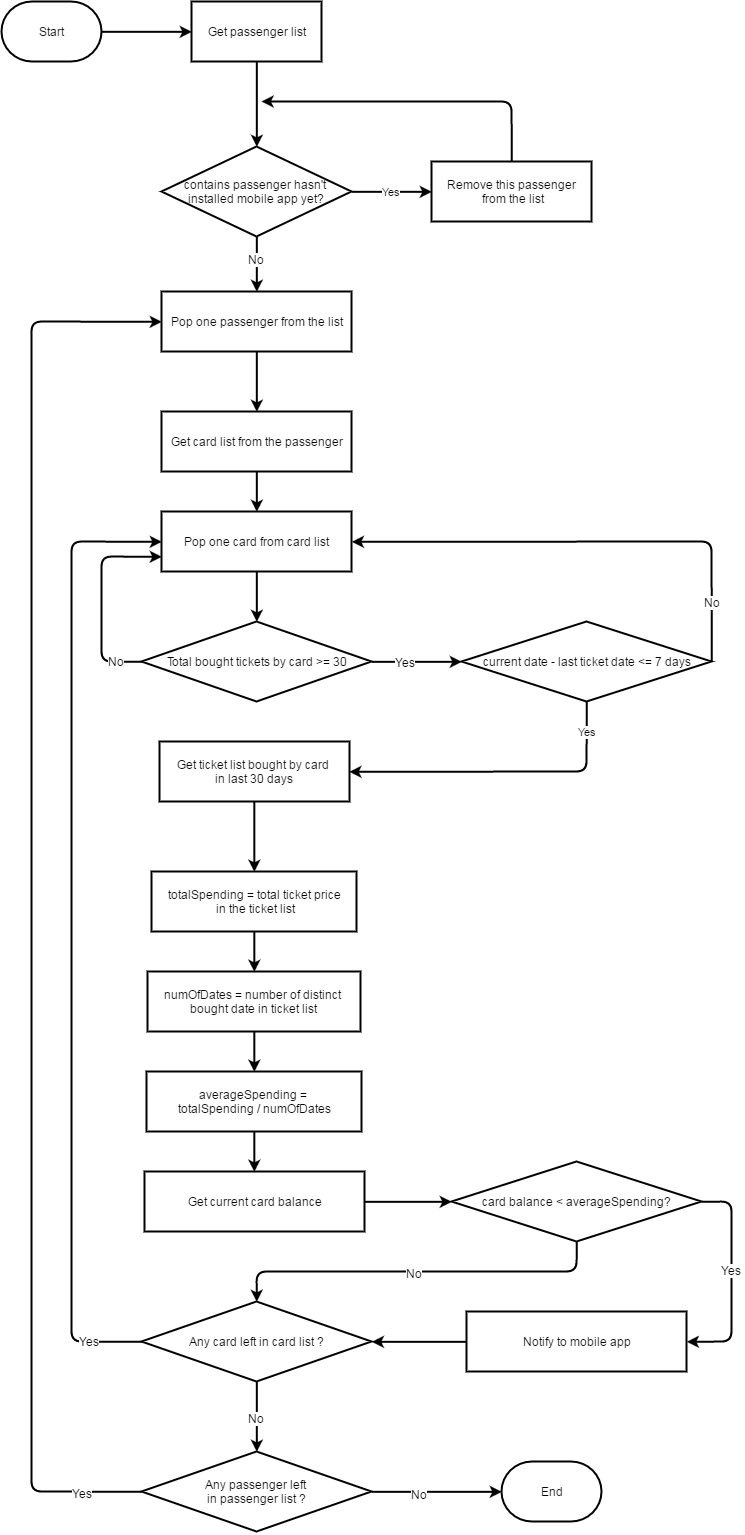
**Average spending** =

**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



# D. 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/>