**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Freight Truck Services**

|  |  |
| --- | --- |
| **GROUP 1** | |
| **Group member** | Bui Duc Huy – Leader – HuyBDSE61006  Nguyen Quang Thien – Member – ThienNQSE61000  Dang Phu Thinh – Member – ThinhDPSE60879  Nguyen Duy Khuong – Member – KhuongND60493 |
| **Supervisor** | Kieu Trong Khanh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | FTS |

-Ho Chi Minh City, ***January 6, 2015***

# Table of Contents

[Table of Contents 3](#_Toc408433082)

[List of Tables 6](#_Toc408433083)

[List of Figures 7](#_Toc408433084)

[Definitions, Acronyms, and Abbreviations 7](#_Toc408433085)

[A. Introduction 8](#_Toc408433086)

[1. Project Information 8](#_Toc408433087)

[2. Introduction 8](#_Toc408433088)

[3. Current Situation 8](#_Toc408433089)

[4. Problem Definition 8](#_Toc408433090)

[5. Proposed Solution 8](#_Toc408433091)

[5.1 Feature functions 8](#_Toc408433092)

[5.2 Advantages and disadvantages 9](#_Toc408433093)

[6. Functional Requirement 9](#_Toc408433094)

[7. Role and Responsibility 10](#_Toc408433095)

[B. Software Project Management Plan 11](#_Toc408433096)

[1. Problem Definition 11](#_Toc408433097)

[1.1 Name of this Capstone Project 11](#_Toc408433098)

[1.2 Problem Abstract 11](#_Toc408433099)

[1.3 Project Overview 11](#_Toc408433100)

[2. Project organization 11](#_Toc408433101)

[2.1 Software Process Model 11](#_Toc408433102)

[2.2 Roles and responsibilities 11](#_Toc408433103)

[2.3 Tools and Techniques 12](#_Toc408433104)

[3. Project Management Plan 12](#_Toc408433105)

[3.1 Software Development Life Cycle 12](#_Toc408433106)

[3.2 All Meeting Minutes 12](#_Toc408433107)

[4. Coding Convention 12](#_Toc408433108)

[C. Software Requirement Specification 12](#_Toc408433109)

[1. User Requirement Specification 12](#_Toc408433110)

[1.1 Guest Requirement 12](#_Toc408433111)

[1.2 Member Requirement 12](#_Toc408433112)

[1.3 Partner Requirement 12](#_Toc408433113)

[1.4 Staff Requirement 12](#_Toc408433114)

[1.5 Admin Requirement 12](#_Toc408433115)

[2. System Requirement Specification 12](#_Toc408433116)

[2.1 External Interface Requirement 12](#_Toc408433117)

[2.2 System Overview Use Case 12](#_Toc408433118)

[2.3 List of Use Case 12](#_Toc408433119)

[3. Software System Attribute 13](#_Toc408433120)

[3.1 Usability 13](#_Toc408433121)

[3.2 Reliability 13](#_Toc408433122)

[3.3 Availability 14](#_Toc408433123)

[3.4 Security 14](#_Toc408433124)

[3.5 Maintainability 14](#_Toc408433125)

[3.6 Portability 14](#_Toc408433126)

[3.7 Performance 14](#_Toc408433127)

[4. Conceptual Diagram 14](#_Toc408433128)

[Data Dictionary 14](#_Toc408433129)

[D. Software Design Description 14](#_Toc408433130)

[1. Design Overview 14](#_Toc408433131)

[2. System Architectural Design 14](#_Toc408433132)

[2.1 Web Application architecture description 14](#_Toc408433133)

[2.2 Mobile Application architecture description 14](#_Toc408433134)

[3. Component Diagram 14](#_Toc408433135)

[4. Detailed Description 15](#_Toc408433136)

[4.1 Class Diagram 15](#_Toc408433137)

[4.2 Class Diagram Explanation 15](#_Toc408433138)

[4.3 Interaction Diagram 15](#_Toc408433139)

[5. User Interface Design 15](#_Toc408433140)

[5.1 Guest Interface Design 15](#_Toc408433141)

[6. Database Design 15](#_Toc408433142)

[6.1 Logical Diagram 15](#_Toc408433143)

[6.2 Data Dictionary 15](#_Toc408433144)

[7. Algorithms 15](#_Toc408433145)

[7.1 Document Breakdown 15](#_Toc408433146)

[7.2 String Comparison 16](#_Toc408433147)

[E. System Implementation & Test 16](#_Toc408433148)

[1. Introduction 16](#_Toc408433149)

[1.1 Overview 16](#_Toc408433150)

[1.2 Test Approach 16](#_Toc408433151)

[2. Database Relationship Diagram 16](#_Toc408433152)

[2.1 Physical Diagram 16](#_Toc408433153)

[2.2 Data Dictionary 16](#_Toc408433154)

[3. Performance Measures 17](#_Toc408433155)

[3.1 Clustering Performance 17](#_Toc408433156)

[4. Test Plan 17](#_Toc408433157)

[4.1 Features to be tested 17](#_Toc408433158)

[4.2 Features not to be tested 17](#_Toc408433159)

[5. System Testing Test Case 18](#_Toc408433160)

[5.1 Guest Test Case 18](#_Toc408433161)

[F. Software User’s Manual 18](#_Toc408433162)

[1. Installation Guide 18](#_Toc408433163)

[1.1 Setting up environment at server side 18](#_Toc408433164)

[1.2 Deployment at server side 18](#_Toc408433165)

[1.3 Setting up the environment at client side 19](#_Toc408433166)

[2. User Guide 19](#_Toc408433167)

[G. Appendix 19](#_Toc408433168)

# List of Tables

# List of Figures

# Definitions, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| **No.** | **Abbreviation & Acronym** | **Definition** |
|  |  |  |

# Introduction

## Project Information

* Project name: Freight Truck Services
* Project Code: FTS
* Product Type: Website & Android Application
* Start Date: Jan 6th, 2015
* End Date: May 2nd, 2015

## Introduction

Nowadays, trucks are used to carry goods across the country. However, after they deliver the goods, the trucks usually come back empty. Our project looks to solve this problem by proposing a solution: when drivers come back after a delivery, they will also pick up and deliver additional goods from other owners on their way. We provide the mobile application for drivers to search for goods when they return and for good owner to send their request for delivery.

## Current Situation

Truck driver only deliver goods for one owner at a time. Therefore, when they finish deliver goods to the owner, they usually return home without anything. Sometimes good owners and truck drivers make verbal agreement without delivery contract. Therefore, there is no guarantee that items will be delivered safely.

* **Backload4u.co.nz, Freightfinder.com:** is a website based on this idea and work very well in New Zealand and U.S. However there are some limitations: Do not support mobile devices; information of truck drivers and owner operators are public on website for unregisted users, which does not warrant the deal between truck drivers and owner operators. These websites don't allow real-time dealing, too.

## Problem Definition

The problem arises when truck has delivered goods, it usually return with empty container on the way while owners always want to optimize shipping costs and need to ensure the safety of your items, but they can only contact each other through phone calls based on the relationship had known before.

## Proposed Solution

Our system supports two types: the mobile application for truck drivers, goods owners; the website application for admin, staff, and goods owners. There are following functions in the system:

### **Feature functions**

Admins can

* Manage accounts.
* Manage configuration.

Staff can

* Tracking orders and transactions.

System can

* Support the suggestion to driver in making decision choosing the goods owners.
* Give the optimal road to drivers.

Truck drivers can

* Post their road, container size, contact info…
* Make arrangement with goods owners and delivery items on their way back.

Goods owners can

* Search for drivers.
* Request for delivery and make a deal with truck drivers.
* Track their order’s status.

### Advantages and disadvantages

Advantages:

- Guarantee the goods are delivered with care.

- Increase income for truck drivers.

- Goods owners can find appropriate drivers quickly.

Disadvantages:

- Truck driver must have smartphone with GPS.

- Internet must be available so users can utilize all features.

## Functional Requirement

### Account management:

* Admins can manage users’s account.
* Truck drivers and good owners can create and manage the properties of their account.

### User information

* Truck drivers can input their phone number, container size, bank account, e-mail address.
* Goods owners can phone number, address, bank account, e-mail address.

### Search

* Truck drivers can search for goods based on their pre-defined road.
* Goods owners can search for truck by criteria.

### Suggestion

* System can support truck drivers in choosing the goods owners and suggest truck drivers for goods owners after they post their goods.
* System can suggest the optimal road for drivers.

### Mobile device

* Truck drivers can publish the destination with specified date, container size, contract information…
* Define the proposed road and/or adjust depending on the goods owners.
* Goods owners can make deal with truck drivers and make the order to deliver their goods.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Kiều Trọng Khánh | Project Manager | Instructor | KhanhKT@fpt.edu.vn |
| 2 | Bùi Đức Huy | Developer | Leader | HuyBDSE61006@fpt.edu.vn |
| 3 | Nguyễn Quang Thiện | Developer | Member | ThienNQSE61000@fpt.edu.vn |
| 4 | Đặng Phú Thịnh | Developer | Member | ThinhDPSE60879@fpt.edu.vn |
| 5 | Nguyễn Duy Khương | Developer | Member | KhuongND60493@fpt.edu.vn |

Table 1: Roles and Responsibility

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

Freight Truck Services

### Problem Abstract

### Project Overview

#### Current Situation and Disadvantages

#### The Proposed System

#### Boundaries of the System

#### Development Environment

##### Hardware requirements

**For Server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wifi (4 Mbps) | Cable, Wifi (8 Mbps) |
| Operating System | XP, Vista, 7, 8 | XP, Vista, 7, 8 |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 1GB RAM | 3GB or more |

Table 2: Hardware Requirement for Server

**For Mobile Application**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wifi (4 Mbps) | Cable, Wifi (8 Mbps) |
| Operating System | XP, Vista, 7, 8 | XP, Vista, 7, 8 |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 1GB RAM | 3GB or more |

**Table 3: Hardware Requirement for Mobile**

##### Software requirements

* Microsoft Windows 8 Professional: operating system and platform for development.
* SQL Server 2008 R2 Express: used to create and manage the database for system.
* Eclipse Luna 4.4.1, JDK 7 update 51, Apache Tomcat 7; Android SDK 14, ADT 23.0.2: used to implement website, web services and mobile application.
* GitHub & TortoiseSVN: used for source control.
* Visio: used to create models and diagrams.
* Skype: used for communication and meeting.

## Project organization

### Software Process Model

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in group** | **Responsibilities** |
| 1 | Kieu Trong Khanh | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| 2 | Bui Duc Huy | Team leader, BA, DEV, Tester | * Managing process * Support team members with technical problems. * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 3 | Nguyen Quang Thien | Team member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |
| 4 | Dang Phu Thinh | Team member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |
| 5 | Nguyen Duy Khuong | Team member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |

Table 4: Roles and Responsibility Details

### Tools and Techniques

* Front – end technologies: HTML5, CSS3, JavaScript, jQuery, AJAX.
* Back – end:
  + Website: Model – View – Controller Pattern – Java.
  + Web Service: RESTful.
  + Mobile App: Android – Java.
* Web Server: Apache Tomcat 7.0.
* Database Management System: MS SQL Server 2008 R2 Express.

## Project Management Plan

### Software Development Life Cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constrains** | **Risks** |
| Requirements Definition | * Collect requirements from customer * Identify and clarify requirements for the system in general | * Introduction of proposed system. * Main functions. | 20 man-days | N/A | Misunderstand the requirements.  Unclear scope of project |
| System and Software Design | * Choose the software development model * Research technology needed for project * Design main structure of system | * Project management plan. * Demo technology * Use case * ERD * SRS | 20 man-days | Depend on “Requirements Definition” | Lack of experience.  Technology is difficult to use  Not cover all use case of project |
| Implementation and Unit Testing | * Coding core functions first * Coding other functions * Update user interface * Unit test | * System design description * Main user’s functions on web and mobile * Suggestion of system * Test case | 50 man-days | Depend on “System and Software Design” | Lack of experience.  Not found suitable suggestion algorithm  Web interface not friendly |
| Integration and System Testing | * Integration test * System test | * Test case | 20 man-days | Depend on “Implementation and Unit Testing” | Lack of experience.  Test case not cover all situation. |
| Operation and Maintenance | * Deploy on server and mobile | * Installation Guide * User’s Guide | 10 man-day | Depend on all previous phase | Lack of experience. |

Table 5: Software Development Life Cycle Detail

### Phase Detail

#### Phase 1: Requirement definition

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| 1. Collect requirements | Find which systems currently provide similar service, their strengths and weakness. | HuyBD, ThienDQ, ThinhDP, KhuongND |
| 2. Identify and clarify main functions. | Define which main functions system should provide. | HuyBD, ThienDQ, ThinhDP, KhuongND |
| 3. Create System  Introduction. | Complete Introduction Report. | HuyBD |
| 4. Software Project Management Plan. | Prepare Project Management Plan. | HuyBD |
| 5. Website Prototype. | Build a prototype of proposed system (Website/Mobile). | ThienDQ, ThinhDP, KhuongND |
| 6. SRS | Create SRS document. | HuyBD, ThienDQ, ThinhDP, KhuongND |

Table 6: Requirement Definition

#### Phase 2: System and software design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Architecture Design** | Implement function import and breakdown data from docx files. | HuyBD, ThienDQ, ThinhDP, KhuongND |
| **2. Detailed Design** | Compare new document with existed documents of system. | HuyBD, ThienDQ, ThinhDP, KhuongND |
| **3. Database Design** | Get jobs from other server to recommendation. | HuyBD, ThienDQ, ThinhDP, KhuongND |
| **4. Technology research** | Create search engine for basic search and advance search. | HuyBD, ThienDQ, ThinhDP, KhuongND |
| **5. Design Document** | Create software design document | HuyBD, ThinhDP |

Table 7: System and Software Design

#### Phase 3: Implementation and unit testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Front-end web functions** | Implement front-end functions on web | KhuongND, HuyBD |
| **2. Back-end web functions** | Implement back-end functions on web | HuyBD, KhuongND |
| **3. Mobile functions** | Implement mobile application | HuyBD, ThienDQ, ThinhDP |
| **4. Suggestion algorithms** | Research and implement suggestion algorithms | HuyBD |
| **5. Unit testing** | Write test case and testing for web functions | HuyBD, KhuongND |
| Write test case and testing for mobile functions | ThienDQ, ThinhDP |

Table 8: Implementation and Unit testing

#### Phase 4: Integration and system testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Integration testing** | Write test case and testing system | HuyBD, ThienDQ, ThinhDP, KhuongND |
| **2. Alpha testing** | Do alpha test with customer | HuyBD, ThienDQ, ThinhDP, KhuongND |

Table 9: Integration and system testing

#### Phase 5: Operation and maintenance

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Installation guide** | Write installation guide | HuyBD |
| **2. User Manual** | Write user manual | HuyBD, ThienDQ, ThinhDP, KhuongND |

Table 10: Opration and maintenance

### All Meeting Minutes

Refer to **Meeting Minutes** folder.

## Coding Convention

* Naming Convention.
* Use camel case for both variable and function name.
* Use Pascal case for class name.
* Indentation.
* Four spaces should be used as the unit of indentation. The exact construction of the indentation (spaces vs. tabs) is unspecified. Tabs must be set exactly every 8 spaces (not 4).
* Avoid lines longer than 80 characters.
* Declaration.
* One declaration per line is recommended since it encourages commenting.
* In absolutely no case should variables and functions be declared on the same line.
* Code Examples

Follow “Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999”.

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

# Software Requirement Specification

## User Requirement Specification

### Guest Requirement

### Member Requirement

### Partner Requirement

### Staff Requirement

### Admin Requirement

## System Requirement Specification

### External Interface Requirement

#### User Interface

#### Hardware Interface

#### Software Interface

#### Communication Protocol

### System Overview Use Case

Figure 2: System Overview Use Case

### List of Use Case

#### <Guest>Overview Use Case

Figure 3: <Guest> Overview Use Case

##### <Guest> Register

Use Case Diagram

Figure 4: <Guest> Register

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – CPS001** | | | |
| **Use Case No.** |  | **Use Case Version** |  |
| **Use Case Name** |  | | |
| **Author** |  | | |
| **Date** |  | **Priority** |  |
| **Actor:**  **Summary:**  **Goal:**  **Triggers:**  **Preconditions:**  **Post Conditions:**   * **Success:** * **Fail:**   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Respone | |  |  |  | |  |  |  | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | ctor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  |   **Relationships:** N/A  **Business Rules:** | | | |

Table 10: <Guest> Register

## Software System Attribute

### Usability

#### Graphic User Interface

#### Usability

#### Installation

### Reliability

### Availability

### Security

### Maintainability

### Portability

### Performance

## Conceptual Diagram

### Data Dictionary

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

# Software Design Description

## Design Overview

## System Architectural Design

### Web Application architecture description

### Mobile Application architecture description

## Component Diagram

## Detailed Description

### Class Diagram

### Class Diagram Explanation

#### Role

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

### Interaction Diagram

#### <Member> View Friend List

## User Interface Design

### Guest Interface Design

#### Login

Figure 12: Login

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
|  |  |  |  |  |  |  |  |

**Buttons/ Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
|  |  |  |  |  |

## Database Design

### Logical Diagram

### Data Dictionary

## Algorithms

### Document Breakdown

#### Definition

#### Define Problem

#### Solution

#### Complexity

#### Flow Chart

### String Comparison

#### Define Problem

#### Requirement

#### Solution

#### Example

# System Implementation & Test

## Introduction

### Overview

### Test Approach

## Database Relationship Diagram

### Physical Diagram

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| Entity Name | Description |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 14: Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
|  |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 15: Attribute Data Dictionary

## Performance Measures

### Clustering Performance

## Test Plan

### Features to be tested

### Features not to be tested

* qweqwe
* qweqw

## System Testing Test Case

### Guest Test Case

#### Search Event

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Inter-test Case Dependence** | **Result** | **Test Date** | **Note** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

# Software User’s Manual

## Installation Guide

### Setting up environment at server side

#### Hardware requirements

#### Software requirements

### Deployment at server side

#### Prepare deployment package

#### Configure Server before deploy

#### Deploy web application on server

### Setting up the environment at client side

#### Setting up for computer

## User Guide

# Appendix