|  |  |
| --- | --- |
| logo ngan.png | MINISTRY OF EDUCATION AND TRAINING |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Build a Web-based application  that manages the activities of  delivery service system by coach |
|  |
| |  |  | | --- | --- | | **Group 18** | | | **Group Members** | Đào Bảo Long – Team Leader – SE60690  Lê Phúc Lữ – Team Member – 60563  Nguyễn Thanh Tùng – Team Member – SE60609  Nguyễn Tấn Đức – Team Member – SE60660 | | **Supervisor** | Nguyễn Trọng Tài | | **Ext Supervisor** | N/A | | **Capstone Project Code** | i-Deliver | |
| Hồ Chí Minh City, April 2014 - |

*Acknowledgement*

To write these words for expressing our gratitude, we first say thanks to Mr. Nguyen Trong Tai about his help for all of us for more than three months duration. He always willing to answer our questions, guide to solve the problems not only at the two meeting days each week but also whenever we need him. We believe that each of time our team faced to difficulties gave caused of his concern. We also appreciate to his kindness and his trust to choose our team among two teams applied to the same his project at the beginning of the course.

After a long time of learning in FPT University, we have been taught by a lot of lectures. Each of them has a specific style but all give us the chances to gain the huge of precious knowledge for the future works and we will not forget their work. End of this duration, capstone project is really a milestone for our study journey. We all believe this is one of best subject we have learned and we also would like to say thanks to education departure of FU for set this subject with the real importance meanings and effects.   
When receive the decision of forming our team, each member of our team do not know the others and we have not worked together in any project before that. It is quite hard at some first weeks for sharing the issues within team members. But after that, with the same objectives, we found out the solidarity spirit and everything became better. No doubts that our team not very good and we need to try day by day to finish our tasks for build the website for the release day.   
In these days, we also receive the assistance from our friends and the encouragement from our family. Without these help, we may not get anything done and we have to say thanks to all of them as a duty.   
And we also say thanks to Mr. Kieu Trong Khanh about his accept to the invitation to take part in our trial defense capstone project and gave us a lot of comments for improving our project. Our project may be contains some bugs but we do not afraid to find and fix them because this should be the main of software engineering daily work.

Group 18  
*Nguyễn Tấn Đức,*

*Đào Bảo Long,*

*Lê Phúc Lữ,*

*Nguyễn Thanh Tùng*

*This page is intentionally left blank*

# Table of contents

[Table of contents 4](#_Toc385651133)

[Table of figures 6](#_Toc385651134)

[Terminology 6](#_Toc385651135)

[1. Introduction 7](#_Toc385651136)

[1.1. Introduction 7](#_Toc385651137)

[1.2. The initial idea of group 7](#_Toc385651138)

[1.3. Overview of existing methods 7](#_Toc385651139)

[1.1.1. Requests management 7](#_Toc385651140)

[1.1.2. Packages arrangement 7](#_Toc385651141)

[1.1.3. Packages management 7](#_Toc385651142)

[1.4. Limitations of existing system 7](#_Toc385651143)

[1.1.1. Requests management 7](#_Toc385651144)

[1.1.2. Packages management and arrangement 8](#_Toc385651145)

[1.1.3. Schedules and trips management 8](#_Toc385651146)

[1.5. Benefits of expected system 8](#_Toc385651147)

[1.6. Business outline 8](#_Toc385651148)

[1.7. Approaches 8](#_Toc385651149)

[1.8. Group of functions 9](#_Toc385651150)

[2. Software Project Management Plan (SPMP) 10](#_Toc385651151)

[2.1. Problem Definition 10](#_Toc385651153)

[2.1.1. Name of this Capstone Project 10](#_Toc385651156)

[2.1.2. Problem Abstract 10](#_Toc385651157)

[2.1.3. Project Overview 10](#_Toc385651158)

[2.2.3.1. The Current System 10](#_Toc385651165)

[2.2.3.2. The Proposed System 10](#_Toc385651166)

[2.2.3.3. Boundaries of the System 12](#_Toc385651167)

[2.2.3.4. Development Environment 12](#_Toc385651168)

[2.2. Project organization 13](#_Toc385651169)

[2.3.1. Software Process Model 13](#_Toc385651172)

[2.3.2. Roles and Responsibilities 13](#_Toc385651173)

[2.3.3. Tools and Techniques 14](#_Toc385651174)

[2.3. Project management plan 14](#_Toc385651175)

[2.4.1. Tasks 14](#_Toc385651177)

[2.2.1.1. Task 1: Initiating 14](#_Toc385651181)

[2.2.1.2. Task 2: Planning 15](#_Toc385651182)

[2.2.1.3. Task 3: Specifying requirements 15](#_Toc385651183)

[2.2.1.4. Task 4: Designing database 15](#_Toc385651184)

[2.2.1.5. Task 5: Creating Software Design Description 15](#_Toc385651185)

[2.2.1.6. Task 6: Implementing 16](#_Toc385651186)

[2.2.1.7. Task 7: Performing Testing 16](#_Toc385651187)

[2.2.1.8. Task 8: Writing User’s Manual 16](#_Toc385651188)

[2.2.1.9. Task 9: Deploying the Website 16](#_Toc385651189)

[2.2.1.10. Task 10: Finalizing and Closing 17](#_Toc385651190)

[2.4.2. Task sheet 17](#_Toc385651191)

[3. Software Requirement Specifications (SRS) 17](#_Toc385651192)

[3.1. User Requirement Specification 17](#_Toc385651194)

[3.1.1. Guest Requirements 17](#_Toc385651197)

[3.1.2. Logged User Requirements 18](#_Toc385651198)

[3.1.3. Customer Requirements 18](#_Toc385651199)

[3.1.4. System Admin Requirements 18](#_Toc385651200)

[3.1.5. Staff Requirements 18](#_Toc385651201)

[3.2. System Requirement Specification (Specific Requirements) 18](#_Toc385651202)

[3.2.1. Overall use case 18](#_Toc385651204)

[3.3. Entity Relationship Diagram 19](#_Toc385651205)

[4. Software Design Description (SDD) 21](#_Toc385651206)

[4.1. Design Overview 21](#_Toc385651208)

[4.2. System Architectural Design 21](#_Toc385651209)

[4.3. Component Diagram 22](#_Toc385651210)

[4.4. Database Design 23](#_Toc385651211)

[3.2.2. Logical database design 23](#_Toc385651212)

[3.2.3. Physical database design 23](#_Toc385651217)

# Table of figures

[Figure 1. An overview of the i-Deliver system 12](#_Toc385651120)

[Figure 2. The waterfall software lifecycle model 13](#_Toc385651121)

[Figure 3. Overall use case 19](#_Toc385651122)

[Figure 4. Model-View-Controller Architecture 21](#_Toc385651123)

[Diagram 1. Actor overview diagram 17](#_Toc385651116)

[Diagram 2. Entity relationship diagram 20](#_Toc385651117)

[Diagram 3. Component Diagram 22](#_Toc385651118)

[Diagram 24. Physical database design 24](#_Toc385651119)

# Terminology

|  |  |  |
| --- | --- | --- |
| No. | Terminology | Explanation |
| 1. | Customer | A person who needs to request for goods delivery from their province to another province. |
| 2. | Staff | A person in charge of managing delivery requests and related services of journeys. |
| 3. | System Administrator | A person take responsibilities of manage staff and price changes |
| 4. | Delivery Request | A request sent by a customer to the staff to ask for a delivery request with the i-Deliver system |

# Introduction

## Introduction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Title:** | | *Building a web-based application that manages the activities of delivery service system by coach* | | | |
| **Start Date:** | | Jan 6, 2014 | | **Finished Date:** |  |
| **No** | **Full name** | | **Role** | **Position** | **Contact** |
| 1. | Nguyễn Trọng Tài | | Supervisor | Lecturer | [TaiNT@fpt.edu.vn](mailto:PhuongLHK@fpt.edu.vn) |
| 2. | Đào Bảo Long | | Developer, Tester | Team Leader | [LongDBSE60690@fpt.edu.vn](mailto:LongDBSE60690@fpt.edu.vn) |
| 3. | Lê Phúc Lữ | | Developer, Tester | Member | [LuLP60563@fpt.edu.vn](mailto:DucBM60324@fpt.edu.vn) |
| 4. | Nguyễn Thanh Tùng | | Developer, Tester | Member | [TungNTSE60609@fpt.edu.vn](mailto:TungNTSE60609@fpt.edu.vn) |
| 5. | Nguyễn Tấn Đức | | Developer, Tester | Member | [DucNTSE60660@fpt.edu.vn](mailto:DucNTSE60660@fpt.edu.vn) |

## The initial idea of group

Nowadays, freight traffic between cities in a day is huge. We have the passenger transportation service providers like Mai Linh or the goods delivery service providers like Tin Thanh, Hop Nhat. And we also have the combination like Phuong Trang. Before working on this project, our team conducted a survey in a passenger transportation service provider. Here are the major findings during the survey:

**Phuong Trang Travel & Transportation Company:**

This company provides passenger transport service using coach. Beside this service, this company also provides goods delivery service. They receive a lot delivery requests per day but all of them still managed by staff manually using papers/books. It makes planning and scheduling for goods delivery become complicated beside passenger transportation arrangement, also makes hard to manage all goods to ensure integrity.

From the problems above, our team decided to develop a delivery service system for passenger transportation service providers which using coach. It operates difference from professional delivery service which using cars specially made for goods delivery.

## Overview of existing methods

### Requests management

The goods delivery service’s staffs have to write down all of delivery requests, monitoring – planning – scheduling manually using papers/books and their memory.

### Packages arrangement

They just simply fill-up empty cargo compartments of each coach as much as possible.

### Packages management

Tracking, searching … is temporary unavailable

## Limitations of existing system

### Requests management

By using papers/books and memory, staffs cannot ensure information accuracy; controllable requests, delivery status of requests.

### Packages management and arrangement

They don’t have any specific fee calculating formula, it’s just estimation.

They don’t have any plan to arrange package on each coach so it’s hard to balance the freight on each route.

### Schedules and trips management

They don’t have detail plans to schedule for each trip, all still managed by demands on real-time.

## Benefits of expected system

The project aims to develop a web-based application that

* Create an easy way for customers to make goods delivery request by allow them to post a new requests on website with registered account then they can tracking their package to ensure delivery.
* Mainly assists service providers’ staff to manage all goods delivery request by using optimized planning and scheduling algorithms, manage the fee of the delivery requests
* Allow customers to comment and rating for the service.

## Business outline

Following the project objectives above, the scope of the project is constrained to the following statements:

* The users of the web-application should interact with the web-application itself through a friendly and attractive user interface.
* *For* ***customers***, this web-application should provide fundamental functions such as register, login then post, edit and cancel goods delivery requests. Through integrated e-payment services, they can pay for their delivery requests. They also can search for posted requested, tracking their package. After using service, customers can comment and rating for the service.
* *For* ***system administrators***, the web-application should allow them to monitor fundamental information about customers, staffs, coaches, routes, and fee.
* *For* ***staffs****,* i-Deliver is a management system allows them to import request, fee calculating, invoice making, planning and scheduling for goods delivery, assign packages for specific coach, specific route.

## Approaches

* Adopt the perspective of a customer during the whole development process, in order to develop a web-based application that promotes usability and interactivity as much as possible.
* Negotiate with web service providers to use their services, and then agree upon how the application communicates with the web services, what information should be retained and what should be eliminated.
* Try to provide staffs the customized tools for manage requests easily.
* Conduct research on how to determine the most appropriate fee based on the average fee of the same service providers in the market.
* Conduct research on how to optimize planning and scheduling for goods delivery.

## Group of functions

|  |  |
| --- | --- |
| Functions for customers | * Allow customers register, login, post/edit/cancel requests, search for, and tracking requests; * Allow customers to rate by different criteria and write comments/reviews on service; |
| Functions for staffs | * Allow staffs to manage fundamental information about requests, fee calculating, invoice making; * Allow staffs to planning and scheduling for goods delivery; * Allow staffs to monitoring and assigning for coaches and routes; |
| Functions for system admins | * Allow system admins to manage all relevant information about the system, customers, staffs, coaches and routes; * Allow system admins to grant access rights to other users of the system; * Allow system admins to decide fee calculating formulas; * Allow system admins to collect and export data to statistic. |

# Software Project Management Plan (SPMP)



## Problem Definition



### Name of this Capstone Project

|  |  |
| --- | --- |
| **Official name** | Building a web-based application that manages the activities of delivery service system by coach |
| **Vietnamese name** | Xây dựng ứng dụng web quản lý các hoạt động cho dịch vụ vận chuyển hàng hóa thông qua hệ thống xe khách đường dài |
| **Abbreviation** | i-Deliver |

### Problem Abstract

The idea of the project is to develop a web-based application that assists delivery service staffs/administrators in manages goods delivery requests, package arrangement, and planning and scheduling, fee management. It also helps customers to make goods delivery request, searching and tracking their package, make a payment online.

### Project Overview



#### The Current System

The idea of this project is developing a delivery service system for passenger transportation service providers which using coach. Before working on this project, we conducted a survey about traditional goods delivery process.

**Traditional goods delivery process:**

Customers will pick their package to delivery service station and provide the name of receiver, destination, and type of goods, weight and size. Then staffs of delivery service will calculate transport fee and give the customer a package invoice which contain provided information and destination station address.

#### The Proposed System

By working on this project, we will develop a service system that assists delivery services providers in Ho Chi Minh City to be closer to their customers. The system has some significant features:

**Create user-oriented interfaces for administrators to simplify management:**

The i-Deliver system support administrators to manage delivery fee like define or edit fee factor. It also helps them to manage related activities of a journey, included trips, stages, stations and routes. Of course the system will provide mechanisms to manage users/staffs.

**Support staffs in delivery requests management, package arrangement, planning and scheduling:**

This system provide easy-to-use interface for requests management like make a new request, edit request information, fee calculating, invoice making, packages arrangement and planning and scheduling, assign packages for specific coach/route.

**Easy-to-use tool for customer to make goods delivery requests, searching and review/edit requests, tracking their package:**

The i-Deliver system is integrated with some technique to help making goods delivery requests online, searching then review or edit requests information, tracking customer’s packages by using request code.

**Optimize arrangement, planning and scheduling for package delivery process**

In traditional way, delivery service staffs have to planning, scheduling and arrange packages for each coaches manually. They just simply fill-up empty cargo compartments as much as possible. This system helps them do their works easily, efficiently by using tools, which optimized by arrangement, planning and scheduling algorithms.

**The i-Deliver system’s users:**

1. **Guests**: non-authorized members can register new account of the i-Deliver website to be granted full access permission or they just can search for routes which delivery service providers operating.
2. **Members:** guests had an authorized account can login to the i-Deliver website to

* Make goods delivery requests;
* Search for posted requests;
* Edit posted requests;
* Cancel posted requests (also have constraints);
* Tracking packages;
* Comment and rating for service;
* Statistics

1. **Administrators**: owners of the i-Deliver website who have highest permission can

* Create new staff account;
* Edit staff account information;
* Delete staff account;
* Define fee calculating formula and fixed value;
* Edit fee calculating formula and fixed value;
* Statistics

1. **Staffs**: users who has account which created by administrators have right to

* Manage requests: approve, reject, update status, assign and scheduling;
* Search for packages, routes or members;
* Manage journeys: include add new, edit or delete routes, stations, trips and coaches;
* Manage comments and rates: delete violated/spam comments

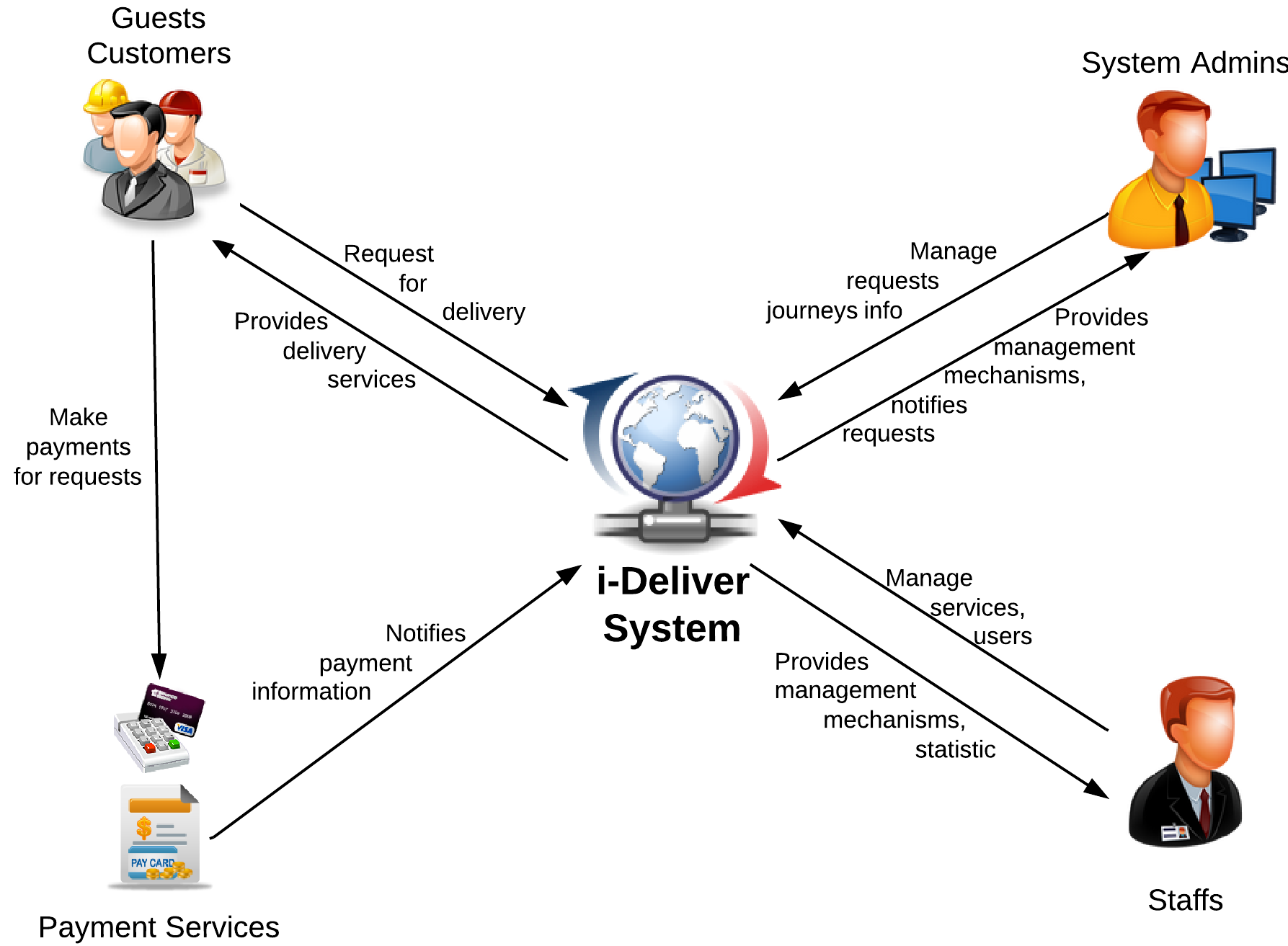


Figure 1. An overview of the i-Deliver system

#### Boundaries of the System

There is no previous version of this system. The product will be developed from scratch, independent of any current system.

As said previously, the system under development is not a delivery service management system. It does not provide mechanisms to manage all activities related to goods delivery. In our team’s scope, the i-Deliver is a system that manage related activities of transportation service providers which using coaches. It means that the service provider only working on 2-stations routes (likes Saigon-Nha Trang, Saigon-Vung Tau, Saigon-Hanoi); they don’t provide mechanism to deliver packages to stations between starting point and destination. That is our team’s future plan for this system.

It focuses mainly on providing easy-to-use interfaces and tools, which support both customers and delivery service staffs.

The final product of this Capstone Project includes

* A service portal which helps customers make goods delivery requests;
* A management module for staffs/administrators of delivery service;
* All the documents involved in the development process.

#### Development Environment

Below is the list of hardware and software requirements needed for the development environment of the project.

**Hardware requirements**

* Personal computers for developing with the minimum configuration: 2 Gb of RAM, 100GB of hard disk, Core 2 Duo 2.0 Ghz;
* A server computer for testing with the minimum configuration: 4 Gb of RAM, 100GB of hard disk, Core 2 Duo 2.0 Ghz;
* All computers must be connected to the Internet.

**Software requirements**

* Operating system: Windows 7 or above;
* Web Server: IIS Express 8;
* Framework: .NET Framework 4.5;
* IDE: Visual Studio 2012;
* DBMS: SQL Server 2008 R2;
* Source Control: Tortoise SVN 1.8.4.

## Project organization



### Software Process Model

The waterfall software lifecycle model will be used to guide the development of the system. The waterfall model includes five major phases as in the figure below, enforcing moving to the next phase only after completion of the previous phase.

Requirement Specifications

System and Software Designs

Implementation and Unit Testing

Integration and System Testing

Operation and Maintenance

Figure 2. The waterfall software lifecycle model

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in group | Responsibilities |
| 1 | Nguyễn Trọng Tài | Supervisor | * Give advice on business and technical problems; * Review and approve of project documents and product deliverables; * Assess the performance of team members. |
| 2 | Đào Bảo Long | Team Leader,  Developer,  Tester | * Create project management plan and distribute tasks to the other members; * Monitor the development process and review the deliverables; * Work on system architecture and detailed designs; * Implement; * Prepare documents; * Perform unit testing; * Deploy the final product. |
| 3 | Lê Phúc Lữ | Developer, Tester | * Research on; * Design user interfaces; * Work on detailed designs; * Implement; * Prepare documents; * Perform unit testing, system testing, and integration test. |
| 4 | Nguyễn Thanh Tùng | Developer, Tester | * Research on; * Design user interfaces; * Work on detailed designs; * Implement; * Perform unit testing, system testing, and integration test. |
| 5 | Nguyễn Tấn Đức | Developer, Tester | * Research on; * Design user interfaces; * Work on detailed designs; * Implement; * Perform unit testing, system testing, and integration test. |

### Tools and Techniques

The tools that will be used to develop the system include:

* *Developing tools:* Microsoft Visual Studio 2012; Tortoise SVN 1.8.4; Microsoft SQL Server 2008 RC;
* *Modeling tools:* StarUML 5.0.2.1570;
* *Document tools:* Microsoft Office 2010.

## Project management plan



### Tasks

Below are all the major tasks that need to be performed sequentially during the development of the system.



#### Task 1: Initiating

|  |  |
| --- | --- |
| ***Task name*** | Initiating |
| ***Descriptions*** | Perform research/survey on some delivery service providers and pricing model; decide upon the technology that will be used to develop the system. |
| ***Deliverables*** | Report 1 – Project Introduction |
| ***Resources needed*** | All team members; 6 days |
| ***Dependencies and constraints*** | N/A |
| ***Risks*** | Performing survey on the delivery service providers can be difficult because of their business; the chosen technology is new to some members. |

#### Task 2: Planning

|  |  |
| --- | --- |
| ***Task name*** | Planning |
| ***Descriptions*** | Create the project management plan; break the system into modules and assign tasks to each member. |
| ***Deliverables*** | Report 2 – Software Project Management Plan |
| ***Resources needed*** | All team members; 6 days |
| ***Dependencies and constraints*** | Task 1 has finished |
| ***Risks*** | Team leader has no experience in managing software projects; all members are still not acquainted with the new technology. |

#### Task 3: Specifying requirements

|  |  |
| --- | --- |
| ***Task name*** | Specifying requirements |
| ***Descriptions*** | Discuss and agree upon the software requirements, what is to be developed and what is not; generate detailed descriptions of all the functions to be developed. |
| ***Deliverables*** | Report 3 – Software Requirement Specification |
| ***Resources needed*** | All team members; 15 days |
| ***Dependencies and constraints*** | Task 2 has finished |
| ***Risks*** | Many aspects of the problem are still unclear to team members; has no experience of working in a delivery service management project. |

#### Task 4: Designing database

|  |  |
| --- | --- |
| ***Task name*** | Designing database |
| ***Descriptions*** | Design the database based on the requirements collected, through three major steps: Conceptual, Logical, and Physical Design |
| ***Deliverables*** | ERD and the physical database with sample data |
| ***Resources needed*** | All team members; 3 days |
| ***Dependencies and constraints*** | Task 3 has finished |
| ***Risks*** | Some of the requirements specified are not clear and cannot be translated into corresponding entities; little experience in organizing data. |

#### Task 5: Creating Software Design Description

|  |  |
| --- | --- |
| ***Task name*** | Creating Software Design Description |
| ***Descriptions*** | Agree upon the system architecture; work on the detailed design of each module; decide which techniques are appropriate to which modules; design the user interfaces for users to interact with. |
| ***Deliverables*** | Report 4 – Software Design Description |
| ***Resources needed*** | All team members; 12 days |
| ***Dependencies and constraints*** | Task 4 has finished |
| ***Risks*** | Some functions are difficult to find appropriate methods to implement; initial development environment setup is also difficult. |

#### Task 6: Implementing

|  |  |
| --- | --- |
| ***Task name*** | Implementing |
| ***Descriptions*** | Each team member implement all the functions that he or she was assigned and regularly check in the code to the Tortoise SVN; regularly validate that the implementation is consistent with the system and detailed designs. |
| ***Deliverables*** | The implemented website |
| ***Resources needed*** | All team members; 24 days |
| ***Dependencies and constraints*** | Task 5 has finished |
| ***Risks*** | Some design documents contain errors; implementation is not always consistent with the system and detailed designs. |

#### Task 7: Performing Testing

|  |  |
| --- | --- |
| ***Task name*** | Performing Unit Testing |
| ***Descriptions*** | Create and perform appropriate test cases for all main functions; record the test results for later reference; fix all the bugs found during the testing sessions. |
| ***Deliverables*** | Report 5 – Software Test Documentation |
| ***Resources needed*** | All team members; 12 days |
| ***Dependencies and constraints*** | Task 6 has finished |
| ***Risks*** | Lack of test cases for some non-critical functions; not enough time to intensively test all the functions. |

#### Task 8: Writing User’s Manual

|  |  |
| --- | --- |
| ***Task name*** | Writing User’s Manual |
| ***Descriptions*** | Writing a user’s manual to instruct the users, including guest, users, staffs, and system administrators, how to use the system. |
| ***Deliverables*** | Report 6 – User’s Manual |
| ***Resources needed*** | All team members; 5 days |
| ***Dependencies and constraints*** | Task 7 has finished |
| ***Risks*** | Some of the functions are not consistent with the user requirements, causing the user’s manual to be inconsistent with the user requirements. |

#### Task 9: Deploying the Website

|  |  |
| --- | --- |
| ***Task name*** | Deploying the Website |
| ***Descriptions*** | Perform acceptance testing and deploy the website to a host on the Internet |
| ***Deliverables*** | The complete website |
| ***Resources needed*** | All team members; all days left |
| ***Dependencies and constraints*** | Task 8 has finished |
| ***Risks*** | Little experience of deploying a website to a real host |

#### Task 10: Finalizing and Closing

|  |  |
| --- | --- |
| ***Task name*** | Finalizing and Closing |
| ***Descriptions*** | Finalize all related documents and prepare for presentation |
| ***Deliverables*** | The complete website and related documents, presentation |
| ***Resources needed*** | All team members; 4 days |
| ***Dependencies and constraints*** | Task 9 has finished |

### Task sheet

Refer to the Tasksheet Final.xlsx for the detailed task sheet of the project plan.

# Software Requirement Specifications (SRS)



## User Requirement Specification

The system should allow 4 types of actors, namely Guest, Customer, Staff, and System Administrator, and an abstract actor named Logged User, to interact with. Each of these types of user is granted a set of functions as specified below.

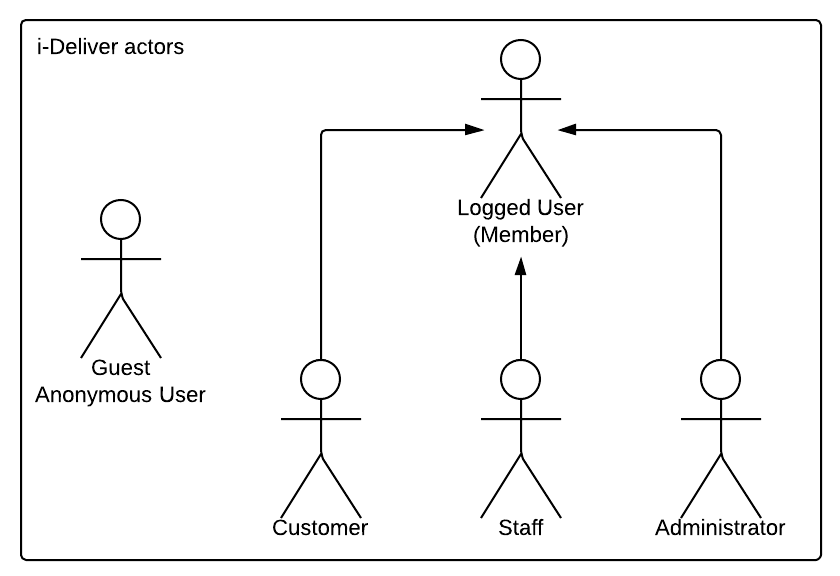


Diagram 1. Actor overview diagram



### Guest Requirements

A guest is an unauthenticated user of the website. He or she can:

* Log in with his authorized account;
* Register a new account;
* View, search for, and filter routes by a number of criteria that suit his or her needs;
* View details of an arbitrary routes.

### Logged User Requirements

A logged user is an authenticated user of the website (e.g., a user who logged on to the system with a valid username and password). They can:

* Log out;
* Change their own password;
* View and update their account details.

Note that that this actor is an abstract one and does not represent a real actor in practice.

### Customer Requirements

A Customer is a logged user (see 3.1.2. Logged User Requirement) and is given all the functions of a logged user. In addition, he or she is granted all the functions of a guest (see 3.1.1. Guest Requirement) except for the Log in and Register functions. A Customer also has his or her own set of functions:

* Create and submit the goods delivery requests;
* Cancel unwanted requests if they weren’t paid or approved;
* Edit submitted delivery information in a limit duration;
* Leave reviews and rate for the service of a station;
* Tracking for package (when the package left station, time left to be delivered, …);
* Make an online payment via third-party service.

### System Admin Requirements

A System Administrator is a logged user (see 3.1.2. Logged User Requirement) and is given all the functions of a logged user. In addition, a system Administrator also has his or her own set of functions:

* Manage Staff accounts;
* Add more System Administrator accounts;
* Manage fee value and calculating formula;
* Statistics: get data about revenue, performance.

### Staff Requirements

A Staff is a logged user (see 3.1.2. Logged User Requirement) and is given all the functions of a logged user. In addition, a system Administrator also has his or her own set of functions:

* Manage all requests: approve, reject online requests, create new, edit and cancel offline requests;
* Manage all related information about routes, stations, trips and coaches;
* Arrange and schedule for packages delivery;
* Searching for information about requests, routes, stations, trips and coaches;
* Manage comments and rating: delete violated comments and clear spam rating;

## System Requirement Specification (Specific Requirements)



### Overall use case

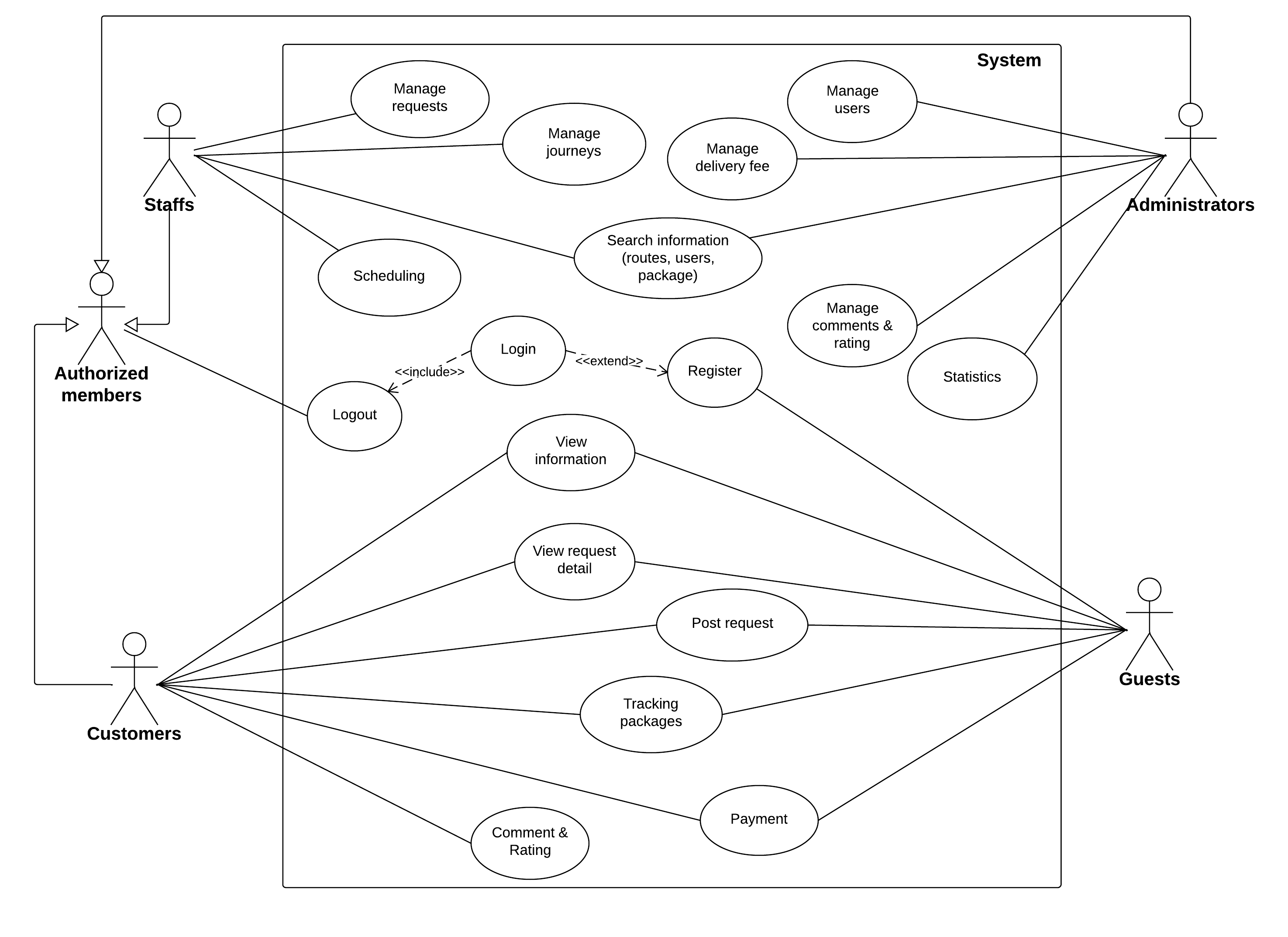


Figure 3. Overall use case

## Entity Relationship Diagram

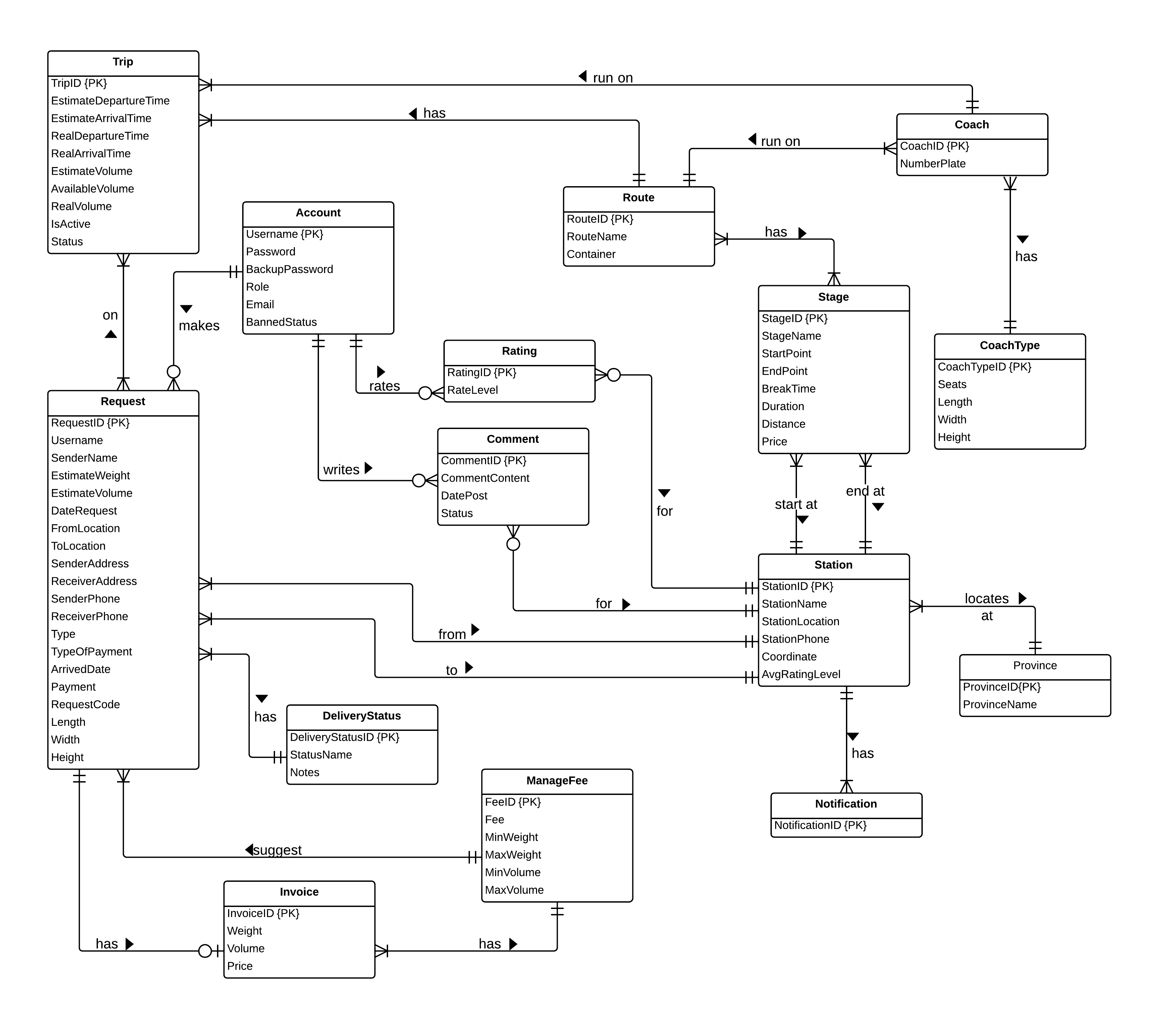


Diagram 2. Entity relationship diagram

# Software Design Description (SDD)



## Design Overview

The SDD describes the system architectural design and the detailed designs, including the user interfaces, of the system.

The components should communicate through interfaces. The detailed implementation of each component should be transparent to other components. The passive MVC III pattern is the preferable architecture for the website.

The detailed designs of the system should adopt the basic principles of software design, including “high cohesion, low coupling” and “open to extension, closed to modification” principles.

Each of the following sections is summarized below:

* Section 4.2: Gives a specification of the system architecture design, describing the overall architecture of the system and subsystems.
* Section 4.3: Specifies all the components that should be included in the system, and the communications between them.
* Section 4.4: Describes the class and package diagrams, as well as their descriptions in details, to provide a static view of the system.
* Section 4.4: Describes the sequence diagrams for the use cases specified in the SRS, to provide a dynamic view of the system.
* Section 4.5: Describes the user interfaces, to specify what will be seen by the users of the system.
* Section 4.5: Describes the Database Design, including the relations and the relationships between them.

## System Architectural Design

The MVC III (Model – View – Controller) pattern is used as the overall system architecture, because it specifies a clear distinction between the responsibilities of the components and is appropriate for developing web applications.

3. Exchange data

4. Results

**CONTROLLER**

* Intercepts user input;
* Coordinates the view and model;
* Handles communication between the model and data layer.

**VIEW**

* Binds to the model;
* Renders the UIs (HTML, CSS, JavaScript);
* Allows navigating between controllers.

**MODEL**

* Communicates with data source;
* Exposes functionalities, business logics, and data validation.

Database   
Server

Application Server

1. Request

8. Response

5. Select views

7. Changes made

2. Invoke methods

6. Query states

Method invocations

Change notifications

Figure 4. Model-View-Controller Architecture

## Component Diagram

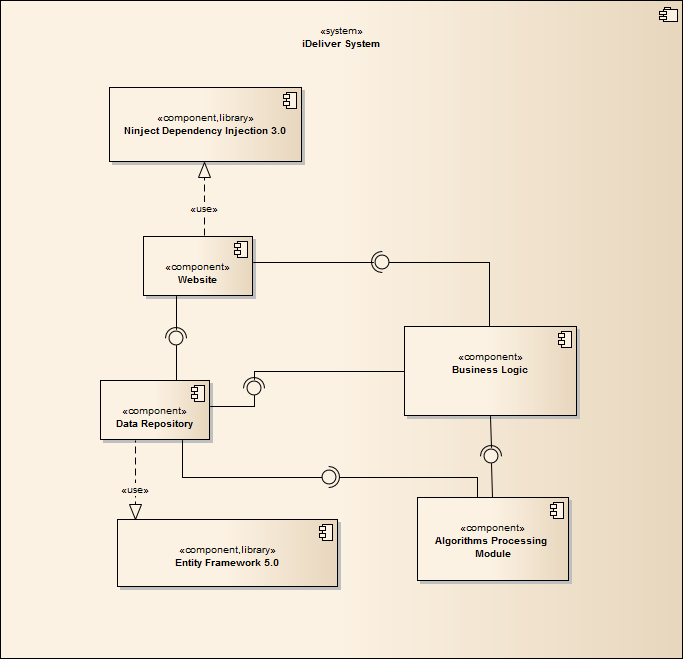


Diagram 3. Component Diagram

The i-Deliver system includes the following major components:

* **Entity Framework 5.0:** An object-relational mapper that enables working with relational data persistence using domain-specific objects;
* **Data Repository:** A data store that provides data access functionality, using the Entity Framework to communicates with the database;
* **Business Logic:** A module that uses interfaces exposed by the Data Repository to encapsulate important businesses of in the i-Deliver system;
* **Ninject Dependency Injection 3.0:** A lightweight dependency injection framework for .NET applications which helps split the application into a collection of loosely-coupled, highly-cohesive pieces, and then glue them back together in a flexible manner.
* **Website:** A web-application that helps exposes all the functionalities to end-users.
* **Algorithms Processing Module:**

From the point of view of the MVC-II pattern, the components Entity Framework 5.0, Data Repository, Business Logic, and Ninject Dependency Injection 3.0 all belong to the Model part. The Controller and the View parts are wrapped in the component Website.

## Database Design

### Logical database design

From the ERD specified in the SRS section, logical database design includes the following relations:

|  |  |  |
| --- | --- | --- |
| **Index** | **Table Name** | **Description** |
| 1 | Account | List of user that registered to the system with valid username and password. |
| 2 | UserInfo | Detail information of each user. |
| 3 | Coach | List of all coach the center has. |
| 4 | CoachType | List of type of coach base on the number of seat that the coach has. |
| 5 | Route | List of route that center can deliver to, including the route name. |
| 6 | Trip | Information of trip of each coach the center has every day. |
| 7 | Station | The list of all stations the center has. |
| 8 | Schedule | The schedule of coach for run on the specific route during a day. |
| 9 | Request | Information of each request that user posted to the system. |
| 10 | DeliveryStatus | Status of the request of base on the action of user and staff. |
| 11 | Invoice | Detail of invoice corresponding to the approve request. |
| 12 | ManageFee | The fee of service base on the range of volume and range of weight. |
| 13 | Comment | Content of comment of user posted for each station. |
| 14 | Rating | Rating level that user rate for each station. |
| 15 | Assigning | Table of mapping between coach and request. |
| 16 | Stage | The partial of the way of route, each stage contains the start and end station and addition information of this stage. |
| 17 | RouteStage | Mapping between route and stage |
| 18 | Notification | Check whether user has seen the notification or not yet. |
| 19 | Province | List of provinces that the system has some station in there. |



The details of each relation are specified below

### Physical database design



Diagram 4. Physical database design