



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 Tân Đức – Team Member – SE60660
Supervisor	Nguyễn Trọng Tài
Ext Supervisor	N/A
Capstone Project Code	i-Deliver

- Hồ Chí Minh City, May 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
Đào Bảo Long,

Lê Phúc Lữ,

Nguyễn Thành Tùng,

Nguyễn Tân Đức

Table of contents

Table of contents.....	3
Table of figures	10
Terminology.....	12
1. Introduction.....	13
1.1. Introduction	13
1.2. The initial idea of group	13
1.3. Overview of existing methods.....	13
1.1.1. Requests management.....	13
1.1.2. Packages arrangement.....	13
1.1.3. Packages management	13
1.4. Limitations of existing system	14
1.1.1. Requests management.....	14
1.1.2. Packages management and arrangement	14
1.1.3. Schedules and trips management	14
1.5. Benefits of expected system.....	14
1.6. Business outline.....	14
1.7. Approaches.....	14
1.8. Group of functions	15
2. Software Project Management Plan (SPMP).....	16
2.1. Problem Definition.....	16
2.1.1. Name of this Capstone Project.....	16
2.1.2. Problem Abstract	16
2.1.3. Project Overview	16
2.2.3.1. The Current System	16
2.2.3.2. The Proposed System	16
2.2.3.3. Boundaries of the System.....	18
2.2.3.4. Development Environment.....	18
2.2. Project organization.....	19
2.3.1. Software Process Model	19
2.3.2. Roles and Responsibilities	19
2.3.3. Tools and Techniques	20
2.3. Project management plan	20
2.4.1. Tasks	20

2.2.1.1.	Task 1: Initiating.....	21
2.2.1.2.	Task 2: Planning	21
2.2.1.3.	Task 3: Specifying requirements	21
2.2.1.4.	Task 4: Designing database	21
2.2.1.5.	Task 5: Creating Software Design Description	22
2.2.1.6.	Task 6: Implementing	22
2.2.1.7.	Task 7: Performing Testing	22
2.2.1.8.	Task 8: Writing User's Manual	23
2.2.1.9.	Task 9: Deploying the Website	23
2.2.1.10.	Task 10: Finalizing and Closing.....	23
2.4.2.	Task sheet.....	23
2.4.	Coding Convention	29
3.	Software Requirement Specifications (SRS)	30
3.1.	User Requirement Specification.....	30
3.1.1.	Guest Requirements	30
3.1.2.	Logged User Requirements.....	30
3.1.3.	Customer Requirements	30
3.1.4.	System Admin Requirements	31
3.1.5.	Staff Requirements.....	31
3.2.	System Requirement Specification (Specific Requirements)	31
3.2.1.	External Interface Requirements.....	31
3.2.1.1.	User Interfaces	31
3.2.1.2.	Hardware Interfaces.....	31
3.2.1.3.	Software Interfaces	32
3.2.2.	Functional Requirements	32
3.2.2.1.	Overall use case	32
3.2.2.2.	« Guest » Register.....	33
3.2.2.3.	« Customer, Guest » Post request.....	34
3.2.2.4.	« Customer » Cancel request	36
3.2.2.5.	« Customer, Guest » View request detail	37
3.2.2.6.	« Customer, Guest » Tracking package.....	39
3.2.2.7.	« Customer » Payment.....	40
3.2.2.8.	« Customer » Rating	42
3.2.2.9.	« Customer » Comment.....	44
3.2.2.10.	« System administrator » Define delivery fee value.....	46

3.2.2.11.	« System administrator » Edit delivery fee value/ calculating formula	48
3.2.2.12.	« System administrator » Create new staff account	49
3.2.2.13.	« System administrator » Edit account information	50
3.2.2.14.	« System administrator » Delete account	52
3.2.2.15.	« System administrator » Statistic	53
3.2.2.16.	« System administrator » Delete comment.....	55
3.2.2.17.	« System administrator » Clear rating	56
3.2.2.18.	« Customer/Guest » Edit request information	58
3.2.2.19.	« Staff » Input invoice	60
3.2.2.20.	« Staff » Update status of new trip	62
3.2.2.21.	« Staff » Update status of arrival trip	64
3.2.2.22.	« Staff » Update status of departed trip	66
3.2.2.23.	« Staff » Manage expired request	68
3.2.2.24.	« Staff » Manage late request	70
3.2.2.25.	« Staff » View request on way and request delivered	72
3.2.2.26.	« Staff » Approve request.....	74
3.2.2.27.	« Staff » Reject request.....	75
3.2.2.28.	« Staff » Assign package	76
3.2.2.29.	« Staff » Update package(s) status	78
3.2.2.30.	« Staff » Create a schedule	80
3.2.2.31.	« Staff » Create a trip.....	81
3.2.2.32.	« Staff » Delete a trip.....	83
3.2.2.33.	« Staff » Add a station	85
3.2.2.34.	« Staff » Edit station information	87
3.2.2.35.	« Staff » Add a route.....	88
3.2.2.36.	« Staff » Delete a route	90
3.2.2.37.	« Staff » Add new coach.....	92
3.2.2.38.	« Staff » Delete a coach	93
3.2.3.	Non-Functional Requirements	95
3.2.2.1.	Reliability	95
3.2.2.2.	Availability	95
3.2.2.3.	Security	95
3.2.2.4.	Maintainability.....	95
3.2.2.5.	Portability	95
3.2.2.6.	Performance.....	95

3.3.	Entity Relationship Diagram	95
4.	Software Design Description (SDD)	97
4.1.	Design Overview.....	97
4.2.	System Architectural Design.....	97
4.3.	Component Diagram	99
4.4.	Detailed Description of Components	102
4.4.1.	Entities Classes	102
4.4.2.	Algorithm Processing Classes	103
4.4.2.1.	Class diagram	103
4.4.2.2.	Algorithms description	103
4.4.2.3.	IRouteChooser	107
4.4.2.4.	IAssigning.....	108
4.4.3.	Data Repository Classes	108
4.4.3.1.	Class diagram	108
4.4.3.2.	IRequestRepository Interface	109
4.4.3.3.	IUserRepository Interface.....	110
4.4.3.4.	IJourneyRepository Interface	111
4.4.3.5.	IAssigningRepository Interface	112
4.4.3.6.	IFeedbackRepository Interface	113
4.4.3.7.	IPriceRepository Interface	114
4.4.4.	Business Logic Classes	115
4.4.4.1.	Class diagram	115
4.4.4.2.	IAccount Interface	115
4.4.4.3.	IAssigning Interface	115
4.4.4.4.	IStage Interface.....	117
4.4.4.5.	ITrip Interface.....	118
4.4.4.6.	ICustomer Interface	118
4.4.4.7.	IFee Interface	119
4.4.4.8.	IFeedback Interface	119
4.4.4.9.	IHome Interface	120
4.4.4.10.	IPaypal Interface.....	121
4.4.4.11.	IRequest Interface	122
4.4.4.12.	ISMS Interface.....	123
4.4.4.13.	IStaff Interface	123
4.4.4.14.	IStation Interface	124

4.5.	Behavioral Diagrams	124
4.5.1.	State Machine Diagram: Request Status Transition	124
4.5.2.	State Machine Diagram: Package Status Transition.....	125
4.5.3.	Sequence Diagram: Send Delivery Requests.....	126
4.5.4.	Sequence Diagram: Tracking.....	126
4.5.5.	Sequence Diagram: Payment	126
4.5.6.	Sequence Diagram: Feedback.....	127
4.5.7.	Sequence Diagram: Approve Requests.....	127
4.5.8.	Sequence Diagram: Assign Package.....	128
4.5.9.	Sequence Diagram: Package in-transit	128
4.5.10.	Sequence Diagram: Package arrived.....	129
4.5.11.	Sequence Diagram: Package delivered	129
4.5.12.	Sequence Diagram: Package return.....	130
4.5.13.	Sequence Diagram: Manage late payment requests	130
4.5.14.	Sequence Diagram: Create schedule	131
4.5.15.	Sequence Diagram: Create trip	132
4.5.16.	Sequence Diagram: Create staff account	133
4.6.	User Interface Design.....	134
4.6.1.	Home Page	134
4.6.2.	Send Delivery Request Page	134
4.6.3.	Online Payment Page.....	136
4.6.4.	Approve Request Page	136
4.6.5.	Assign Package Page	137
4.6.6.	Tracking Page	137
4.6.7.	Update Package Status Page	138
4.6.8.	Create Trip Page	140
4.6.9.	Feedback and Review Page	141
4.6.10.	Manage Fee Page	142
4.7.	Database Design	143
4.7.1.	Logical database design	143
4.7.1.1.	User.....	147
4.7.1.2.	UserInfo.....	147
4.7.1.3.	Coach	148
4.7.1.4.	CoachType.....	148
4.7.1.5.	Route.....	148

4.7.1.6.	Trip	148
4.7.1.7.	Station.....	149
4.7.1.8.	Schedule.....	150
4.7.1.9.	Request	150
4.7.1.10.	DeliveryStatus	151
4.7.1.11.	Invoice	151
4.7.1.12.	ManageFee.....	151
4.7.1.13.	Comment	152
4.7.1.14.	Rating.....	152
4.7.1.15.	Assigning	153
4.7.1.16.	Stage	153
4.7.1.17.	RouteStage.....	153
4.7.1.18.	Notification.....	154
4.7.1.19.	Province	154
4.7.2.	Physical database design.....	154
5.	Software Test Document.....	156
5.1.	Introduction	156
5.1.1.	System Overview	156
5.1.2.	Test Approach.....	159
5.2.	Test Plan.....	159
5.2.1.	Features to be tested.....	159
5.2.2.	Features not to be tested.....	160
5.3.	System Test Cases.....	161
5.3.1.	Send new delivery request	161
5.3.2.	Online payment.....	162
5.3.3.	Tracking package	163
5.3.4.	Feedback and review.....	164
5.3.5.	Approve delivery requests	165
5.3.6.	Assign packages.....	166
5.3.7.	Prepare for packages return	166
5.3.8.	Create the trip.....	167
5.3.9.	Create schedules.....	169
5.3.10.	Monitor price/fee.....	170
5.3.11.	Monitor price/fee.....	171
5.3.12.	Create staff account.....	171

5.4. Checklists	173
5.4.1. Checklist of Validation	173
5.4.2. Submission Checklist.....	173
6. Software User's Manual	175
6.1. Installation Guide	175
6.1.1. Prerequisites.....	175
6.1.2. Installation procedures	175
6.2. User's Guide.....	178
6.2.1. Guides for Customer	178
6.2.1.1. Log in to the system.....	178
6.2.1.2. Log out of the system	179
6.2.1.3. Register an account with Customer role.....	180
6.2.1.4. Send a delivery request.....	181
6.2.1.5. View station details and review	183
6.2.1.1. Online payment.....	185
6.2.1.2. Tracking package(s)	186
6.2.2. Guides for Staffs and System Admins	187
6.2.2.1. Approve pending requests	187
6.2.2.2. Create a trip	189
6.2.2.3. Assign package(s).....	191
6.2.2.4. Confirm package(s) departed from start-station	192
6.2.2.5. Confirm package(s) arrived in end-station	195
6.2.2.6. Confirm package(s) delivered	196
6.2.2.7. Prepare package(s) for return	197
6.2.2.8. Extend due-date for online payment.....	198
6.2.2.9. Statistics.....	199
6.2.2.10. Add a fee.....	201
6.2.2.11. Monitor fees.....	201
Appendix.....	202
Reference	202

Table of figures

Figure 1. An overview of the i-Deliver system.....	18
Figure 2. The waterfall software lifecycle model	19
Figure 3. Overall use case.....	32
Figure 4. Model-View-Controller Architecture	99
Figure 5. Home page.....	134
Figure 6. Send delivery request page	135
Figure 7. Select new request to make a payment.....	136
Figure 8. Make an online payment via Paypal.....	136
Figure 9. Approve request page	137
Figure 10. Assign packages page.....	137
Figure 11. Tracking package page	138
Figure 12. Confirm package departed.....	139
Figure 13. Confirm package arrived	139
Figure 14. Confirm package delivered.....	140
Figure 15. Prepare package to return	140
Figure 16. Extend due-date for late payment requests.....	140
Figure 17. Create trip page.....	141
Figure 18. Feedback and review page.....	142
Figure 19. Manage fee page.....	143
Figure 20. Logical database design.....	146
Figure 21. Physical database design	155
Figure 22. i-Deliver web application release folder.....	175
Figure 23. Create i-Deliver database using SQL Query	176
Figure 24. Add new web application	176
Figure 25. Configure database connection string	177
Figure 26. Run i-Deliver website.....	178
Screen 1. Login screen.....	179
Screen 2. Logout screen	180
Screen 3. Register screen	181
Screen 4. Customer screen.....	182
Screen 5. Send delivery request screen.....	182
Screen 6. Station information screen	183
Screen 7. Customer main screen	184
Screen 8. Station list for review screen.....	184
Screen 9. Review screen	185
Screen 10. Request view before make a payment screen	186
Screen 11. Invoice review before checkout screen	186
Screen 12. Tracking package screen.....	187
Screen 13. Requests review screen	188
Screen 14. Input invoice screen	188
Screen 15. Trip list screen.....	190
Screen 16. Create trip screen	190
Screen 17. Assign package(s) screen	192

Screen 18. Confirm departed package(s) screen.....	193
Screen 19. List trip from the other station screen	195
Screen 20. List request that delivered to customer screen	197
Screen 21. List late deliver request screen.....	198
Screen 22. List late payment requests screen.....	199
Screen 23. Statistic screen.....	200
Screen 24. Add a fee screen.....	201
Screen 25. Monitor fees screen.....	201
 Diagram 1. Actor overview diagram	30
Diagram 2. Entity relationship diagram.....	96
Diagram 3. Component Diagram.....	100
Diagram 4. Entities Class Diagram.....	102
Diagram 5. Algorithm Processing Class Diagram.....	103
Diagram 6. Data Repository Class Diagram.....	109
Diagram 7. Business Logic Class Diagram	115
Diagram 8. State machine diagram: Request status transition.....	125
Diagram 9. State machine diagram: Package status transition	125
Diagram 10. Sequence diagram: Send delivery request	126
Diagram 11. Sequence diagram: Tracking package	126
Diagram 12. Sequence diagram: Payment	127
Diagram 13. Sequence diagram: Feedback.....	127
Diagram 14. Sequence diagram: Approve Requests.....	128
Diagram 15. Sequence diagram: Assign packages	128
Diagram 16. Sequence diagram: Packages in-transit.....	129
Diagram 17. Sequence diagram: Package arrived	129
Diagram 18. Sequence diagram: Package delivered	130
Diagram 19. Sequence diagram: Package return	130
Diagram 20. Sequence diagram: Manage late payment requests	131
Diagram 21. Sequence diagram: Create schedule.....	132
Diagram 22. Sequence diagram: Create trip	133
Diagram 23. Sequence diagram: Create staff account	133

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

1. Introduction

1.1. Introduction

Project Title:		<i>Building a web-based application that manages the activities of delivery service system by coach</i>		
Start Date:		Jan 6, 2014	Finished Date:	
No	Full name	Role	Position	Contact
1.	Nguyễn Trọng Tài	Supervisor	Lecturer	TaiNT@fpt.edu.vn
2.	Đào Bảo Long	Developer, Tester	Team Leader	LongDBSE60690@fpt.edu.vn
3.	Lê Phúc Lữ	Developer, Tester	Member	LULP60563@fpt.edu.vn
4.	Nguyễn Thanh Tùng	Developer, Tester	Member	TungNTSE60609@fpt.edu.vn
5.	Nguyễn Tân Đức	Developer, Tester	Member	DucNTSE60660@fpt.edu.vn

1.2. 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.

1.3. Overview of existing methods

1.1.1. 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.

1.1.2. Packages arrangement

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

1.1.3. Packages management

Tracking, searching ... is temporary unavailable

1.4. Limitations of existing system

1.4.1. Requests management

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

1.4.2. 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.

1.4.3. Schedules and trips management

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

1.5. 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.

1.6. 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.

1.7. 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.

1.8. **Group of functions**

Functions for customers	<ul style="list-style-type: none"> ▫ 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	<ul style="list-style-type: none"> ▫ 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	<ul style="list-style-type: none"> ▫ 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.

2. Software Project Management Plan (SPMP)

2.1. Problem Definition

2.1.1. 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

2.1.2. Problem Abstract

The idea of the project is to develop a web-based application that assists delivery service staffs/administrators in managing 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.

2.1.3. Project Overview

2.2.3.1. The Current System

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

Traditional goods delivery process:

Customers will pick their package at 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 contains provided information and destination station address.

2.2.3.2. 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 supports administrators to manage delivery fee like defining or editing fee factor. It also helps them to manage related activities of a journey, including 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 provides an easy-to-use interface for requests management like making a new request, editing request information, fee calculating, invoice making, packages arrangement and planning and scheduling, assigning 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
3. **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
4. **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

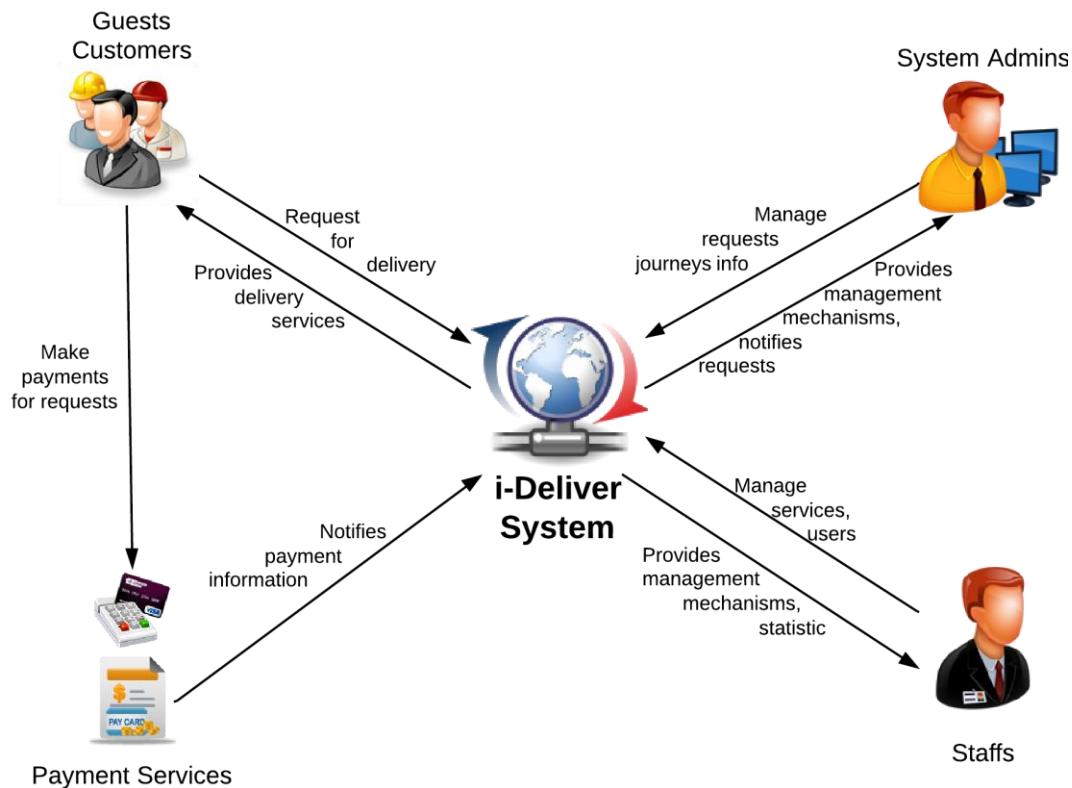


Figure 1. An overview of the i-Deliver system

2.2.3.3. 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.

2.2.3.4. 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.

2.2. Project organization

2.3.1. 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.

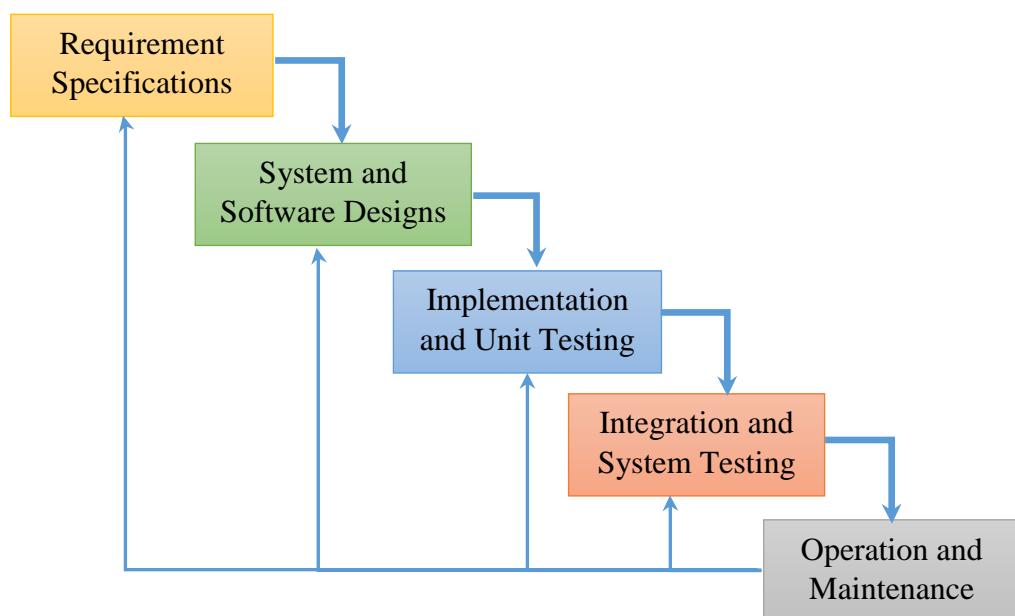


Figure 2. The waterfall software lifecycle model

2.3.2. Roles and Responsibilities

No	Full name	Role in group	Responsibilities
1	Nguyễn Trọng Tài	Supervisor	<ul style="list-style-type: none"> ▫ 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	<ul style="list-style-type: none"> ▫ 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	<ul style="list-style-type: none"> ▫ 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	<ul style="list-style-type: none"> ▫ Research on; ▫ Design user interfaces; ▫ Work on detailed designs; ▫ Implement; ▫ Perform unit testing, system testing, and integration test.
5	Nguyễn Tấn Đức	Developer, Tester	<ul style="list-style-type: none"> ▫ Research on; ▫ Design user interfaces; ▫ Work on detailed designs; ▫ Implement; ▫ Perform unit testing, system testing, and integration test.

2.3.3. 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.

2.3. Project management plan

2.4.1. Tasks

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

2.2.1.1. 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.

2.2.1.2. 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.

2.2.1.3. 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.

2.2.1.4. 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.

2.2.1.5. 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.

2.2.1.6. 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.

2.2.1.7. 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.

2.2.1.8. 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.

2.2.1.9. 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

2.2.1.10. 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

2.4.2. Task sheet

Refer to the next page for the detailed task sheet of the project plan.

Task Name	Length	Start	Finish	Predecessor	Resources
1. Initiating	6 days	Mon 06/01/14	Sat 11/01/14		
1.1. Identify key stakeholders	1 day	Mon 06/01/14	Mon 06/01/14		DucNT,LongDB,LuLP, TungNT
1.2. Research & discuss about business and technology	2 days	Tue 07/01/14	Wed 08/01/14	1.1	DucNT,LongDB,LuLP, TungNT
1.3. Research on similar existing system	2 days	Thu 09/01/14	Fri 10/01/14		DucNT,LongDB,LuLP, TungNT
1.4. Hold project kick-off meeting	0.5 days	Sat 11/01/14	Sat 11/01/14	1.2, 1.3	DucNT,LongDB,LuLP, TungNT
1.5. Report 1 – Project Introduction	0.5 days	Sat 11/01/14	Sat 11/01/14	1.4	DucNT,LongDB,LuLP, TungNT
2. Planning	6 days	Mon 13/01/14	Mon 20/01/14		
2.1. Hold team planning meeting	0.5 days	Mon 13/01/14	Mon 13/01/14	1.5	DucNT,LongDB,LuLP, TungNT
2.2. Prepare problem abstract, proposed solution, coding convention	2 days	Mon 13/01/14	Wed 15/01/14	2.1	TungNT
2.3. Prepare major tasks for the whole team	1.5 days	Wed 15/01/14	Thu 16/01/14	2.1	LuLP
2.4. Prepare management plan and determine task resources, durations, and dependencies	2 days	Wed 15/01/14	Thu 16/01/14	2.1	LongDB
2.5. Configure Tortoise SVN	0.5 days	Mon 20/01/14	Mon 20/01/14	2.4	LongDB
2.6. Report 2 – Project Management Plan	1.5 days	Fri 17/01/14	Mon 20/01/14	2.1-2.5	DucNT,LongDB,LuLP, TungNT
3. Specifying requirements	15 days	Mon 20/01/14	Fri 07/02/14		
3.1. Identify users and users' requirements	3 days	Mon 20/01/14	Thu 23/01/14	2.6	LuLP,TungNT,DucNT
3.2. Define system requirements	1 day	Thu 23/01/14	Fri 24/01/14	3.1	LongDB

3.3. Define non-functional requirements	1 day	Thu 23/01/14	Fri 24/01/14	3.1	DucNT
3.4. Determine main flows	3 days	Thu 23/01/14	Tue 28/01/14	3.1	LuLP,TungNT
3.5. Specify functional requirements for users/customers (requests making, searching and reviews posted requests)	3 days	Tue 28/01/14	Fri 31/01/14		TungNT
3.6. Specify functional requirements for staffs (requests management, package arrangement and scheduling)	3 days	Tue 28/01/14	Fri 31/01/14		LuLP
3.7. Specify functional requirements for administrators	3 days	Tue 28/01/14	Fri 31/01/14		DucNT
3.8. Specify functional requirements for coaches, trips, stages, stations, routes management, pricing model, fee calculating	3 days	Tue 28/01/14	Fri 31/01/14		LongDB
3.9. Specify functional requirements for statistics	2 days	Fri 31/01/14	Tue 04/02/14		TungNT
3.10. Report 3 – Software Requirement Specification	2 days	Wed 05/02/14	Fri 07/02/14	3.1-3.9	LongDB
4. Designing database	3 days	Fri 07/02/14	Tue 11/02/14		
4.1. Discuss on conceptual data model	0.5 days	Fri 07/02/14	Fri 07/02/14	3.11	DucNT,LuLP
4.2. Creating ERD	0.5 days	Fri 07/02/14	Fri 07/02/14	4.1	LongDB,TungNT
4.3. Validate ERD against software requirements	1 day	Mon 10/02/14	Mon 10/02/14	4.2	DucNT,LongDB,LuLP, TungNT
4.4. Create logical model and physical database	1 day	Tue 11/02/14	Tue 11/02/14	4.3	DucNT,LongDB,LuLP, TungNT
5. Creating Software Design Description	12 days	Wed 12/02/14	Thu 27/02/14		
5.1. Discuss on system architecture	0.5 days	Wed 12/02/14	Wed 12/02/14	3.11	DucNT,LuLP,TungNT
5.2. Configure development environment	0.5 days	Wed 12/02/14	Wed 12/02/14		LongDB

5.3. Design the master page	1.5 days	Wed 12/02/14	Thu 13/02/14	3.11	LongDB
5.4. Design the master customer layout	2.5 days	Wed 12/02/14	Fri 14/02/14	3.11	TungNT
5.5. Design the master staff layout	2.5 days	Wed 12/02/14	Fri 14/02/14		LuLP
5.6. Design the master admin layout	1.5 days	Wed 12/02/14	Thu 13/02/14		DucNT
5.7. Design pages for log in, log out, register of customer	1 day	Fri 14/02/14	Fri 14/02/14		DucNT
5.8. Design pages for posting new request, edit request, view request	2 days	Mon 17/02/14	Tue 18/02/14		LuLP
5.9. Design pages for customer to view and edit profile	1.5 days	Mon 17/02/14	Tue 18/02/14		DucNT
5.10. Design page for admin manage fee, stage, route, staff.	2.5 days	Mon 17/02/14	Wed 19/02/14		TungNT,LongDB
5.11. Design page for staff manage trip, request and invoice.	2.5 days	Wed 19/02/14	Fri 21/02/14		LongDB,TungNT
5.12. Design page for staff statistic request and trip information.	1.5 days	Tue 18/02/14	Wed 19/02/14		DucNT
5.13. Design page for staff schedule the package delivery time.	1 day	Wed 19/02/14	Wed 19/02/14		LuLP
5.14. Design layout for user rate and post comment of routes	1 day	Mon 24/02/14	Mon 24/02/14		TungNT
5.15. Design page for tracking the package	1 day	Mon 24/02/14	Mon 24/02/14		LongDB
5.16. Design page for admin manage comment, rating	1 day	Thu 20/02/14	Thu 20/02/14		LuLP
5.17. Design page for user statistic their request information	0.5 days	Thu 20/02/14	Thu 20/02/14		DucNT
5.18. Create main sequence diagrams of view, search, edit and delete request	1.5 days	Thu 20/02/14	Fri 21/02/14		DucNT
5.19. Create main sequence diagrams of manage comment, rating, fee, stage, route,	1.5 days	Mon 24/02/14	Tue 25/02/14		DucNT,LuLP

and staff						
5.20. Create main sequence diagrams of posting and rating a route		0.5 days	Tue 25/02/14	Tue 25/02/14		DucNT
5.21. Create main sequence diagrams of assigning package, edit status of request and manage time (departure/arrival) of coach.		1 day	Tue 25/02/14	Tue 25/02/14		TungNT
5.22. Draw entity class diagram		0.5 days	Tue 25/02/14	Tue 25/02/14		LongDB
5.23. Draw model class diagram		0.5 days	Tue 25/02/14	Tue 25/02/14		LongDB
5.24. Compose physical database description		1 day	Wed 26/02/14	Wed 26/02/14		DucNT,LongDB,LuLP, TungNT
5.25. Report 4 – Software Design Description		1 day	Thu 27/02/14	Thu 27/02/14		DucNT,LongDB,LuLP, TungNT
6. Implementing		24 days	Fri 28/02/14	Wed 02/04/14		
6.1. Log in, log out, register		1 day	Fri 28/02/14	Fri 28/02/14	5.5	DucNT
6.2. Search and view routes		1 day	Fri 28/02/14	Fri 28/02/14	5.6	LongDB
6.3. Post and view request		1.5 days	Fri 28/02/14	Mon 03/03/14	5.6	TungNT
6.4. Search, edit, delete request		1.5 days	Mon 03/03/14	Tue 04/03/14	5.7	TungNT
6.5. Rating for route		1 day	Fri 28/02/14	Fri 28/02/14	5.7	LuLP
6.6. Comment for route		1 day	Mon 03/03/14	Mon 03/03/14	5.3	LuLP
6.7. Tracking package		3 days	Mon 03/03/14	Wed 05/03/14	5.3	LongDB
6.8. Manage all coach and relative information		4 days	Mon 03/03/14	Thu 06/03/14	5.8	DucNT
6.9. Manage all user and relative information		4 days	Wed 05/03/14	Mon 10/03/14	5.9	TungNT
6.10. Manage all rating of user		2 days	Tue 04/03/14	Wed	5.11	LuLP

			05/03/14		
6.11. Manage all comment user	2 days	Thu 06/03/14	Fri 07/03/14	5.10	LuLP
6.12. Manage fee of service	2 days	Mon 10/03/14	Tue 11/03/14	5.10	DucNT
6.13. Statistics of users	2 days	Thu 06/03/14	Fri 07/03/14		LongDB
6.14. Statistics of staffs	2 days	Mon 10/03/14	Tue 11/03/14		LuLP
6.15. Statistics of administrators	2 days	Mon 10/03/14	Tue 11/03/14		LongDB
6.16. Manage all trips	3 days	Tue 11/03/14	Thu 13/03/14		TungNT
6.17. Manage package	2 days	Wed 12/03/14	Thu 13/03/14		DucNT
6.18. Scheduling for package	3 days	Wed 12/03/14	Fri 14/03/14		LuLP
6.19. Integrate all implemented functions	13 days	Mon 17/03/14	Wed 02/04/14		DucNT,LongDB,LuLP, TungNT
7. Performing Testing	12 days	Thu 03/04/14	Fri 18/04/14		
7.1. Write appropriate unit test cases for the functions each member implemented	3 days	Thu 03/04/14	Mon 07/04/14	6.14	DucNT,LongDB,LuLP, TungNT
7.2. Perform unit testing and record the results	2 days	Tue 08/04/14	Wed 09/04/14	7.1	DucNT,LongDB,LuLP, TungNT
7.3. Fix the bugs discovered during unit testing	3 days	Thu 10/04/14	Mon 14/04/14	7.2	DucNT,LongDB,LuLP, TungNT
7.4. Complete code review checklists	1 day	Tue 15/04/14	Tue 15/04/14	7.3	DucNT,LongDB,LuLP, TungNT
7.5. Report 5 – Software Test Documentation	1 day	Wed 16/04/14	Wed 16/04/14	7.4	DucNT,LongDB,LuLP, TungNT
7.6. Perform integration test and fix the bugs discovered	2 days	Thu 17/04/14	Fri 18/04/14	7.5	DucNT,LongDB,LuLP, TungNT
8. Writing Users' Manual	5 days	Mon 21/04/14	Fri 25/04/14		
8.1. Write users' manual for all the functions that	3.5 days	Mon 21/04/14	Thu 24/04/14	7.5	DucNT,LongDB,LuLP,

each member implemented					TungNT
8.2. Check if the users' manual and software requirements are consistent	1.5 days	Thu 24/04/14	Fri 25/04/14	8.1	DucNT,LongDB,LuLP, TungNT
9. Deploying the Website	6 days	Sat 26/04/14	Fri 02/05/14		
9.1. Add more data to the database	2 days	Sat 26/04/14	Mon 28/04/14		DucNT,LuLP,TungNT
9.2. Deploy the website the a real host	0.5 days	Tue 29/04/14	Tue 29/04/14	7.6	LongDB
9.3. Test the website on the real host	3.5 days	Tue 29/04/14	Fri 02/05/14		DucNT,LongDB,LuLP, TungNT
10. Finalizing and closing	4 days	Mon 05/05/14	Thu 08/05/14		
10.1. Prepare final project report	3 days	Mon 05/05/14	Wed 07/05/14		TungNT,LongDB
10.2. Prepare final project presentation	2 days	Mon 05/05/14	Tue 06/05/14		DucNT,LuLP
10.3. Final team meeting	1 day	Thu 08/05/14	Thu 08/05/14		DucNT,LongDB,LuLP, TungNT
10.4. Close the project	0 days	Thu 08/05/14	Thu 08/05/14		DucNT,LongDB,LuLP, TungNT

2.4. Coding Convention

The implantation of the system must strictly follow all the standard coding and naming convention specified by Microsoft, which can be found at <http://msdn.microsoft.com/en-us/library/ff926074.aspx>.

3. Software Requirement Specifications (SRS)

3.1. 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.

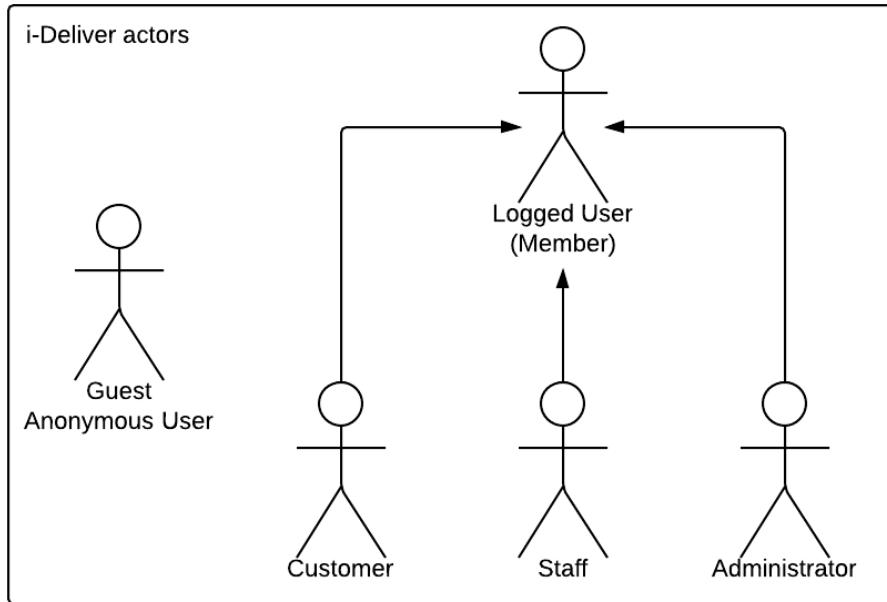


Diagram 1. Actor overview diagram

3.1.1. 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.

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

3.1.3. 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.

3.1.4. 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.

3.1.5. 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;

3.2. System Requirement Specification (Specific Requirements)

3.2.1. External Interface Requirements

3.2.1.1. User Interfaces

The i-Deliver website should adopt an attractive and user-friendly interface so that the users of the system can get a good experience browsing the site.

The pages should be well linked together to promote seamless navigation between them. The instructions should be specific and suggestive, so that the users will not be confused about what to do next.

Client postbacks should be limited wherever possible, in order to boost performance and ease of use.

Vietnamese should be the official language of the website as its target customers are mainly Vietnamese people. Vietnam Dong should be the official currency. Moreover, all other localizable information, such as date and time, should be Vietnamese-styled.

3.2.1.2. Hardware Interfaces

The i-Deliver website can be reached by personal computers that support Internet connection and web browsers. In order to get the best experience, the following conditions should be satisfied:

- 50 Kbps Internet connection or faster;

- 1 gigahertz (GHz) processor or faster;
- 512 megabytes (MB) RAM or more;

3.2.1.3. Software Interfaces

The software listed below is needed for the system to operate normally:

Software Name	Version	Cost	Provider
Web Browser <ul style="list-style-type: none"> ▫ Internet Explorer ▫ Google Chrome ▫ Mozilla Firefox ▫ Safari ▫ Opera 	7.0 or higher	Free Free Free Free Free	Microsoft Google Mozilla Apple Opera
SQL Server Express	10.0	Free	Microsoft
.NET Framework	4.5	Free	Microsoft

3.2.2. Functional Requirements

3.2.2.1. Overall use case

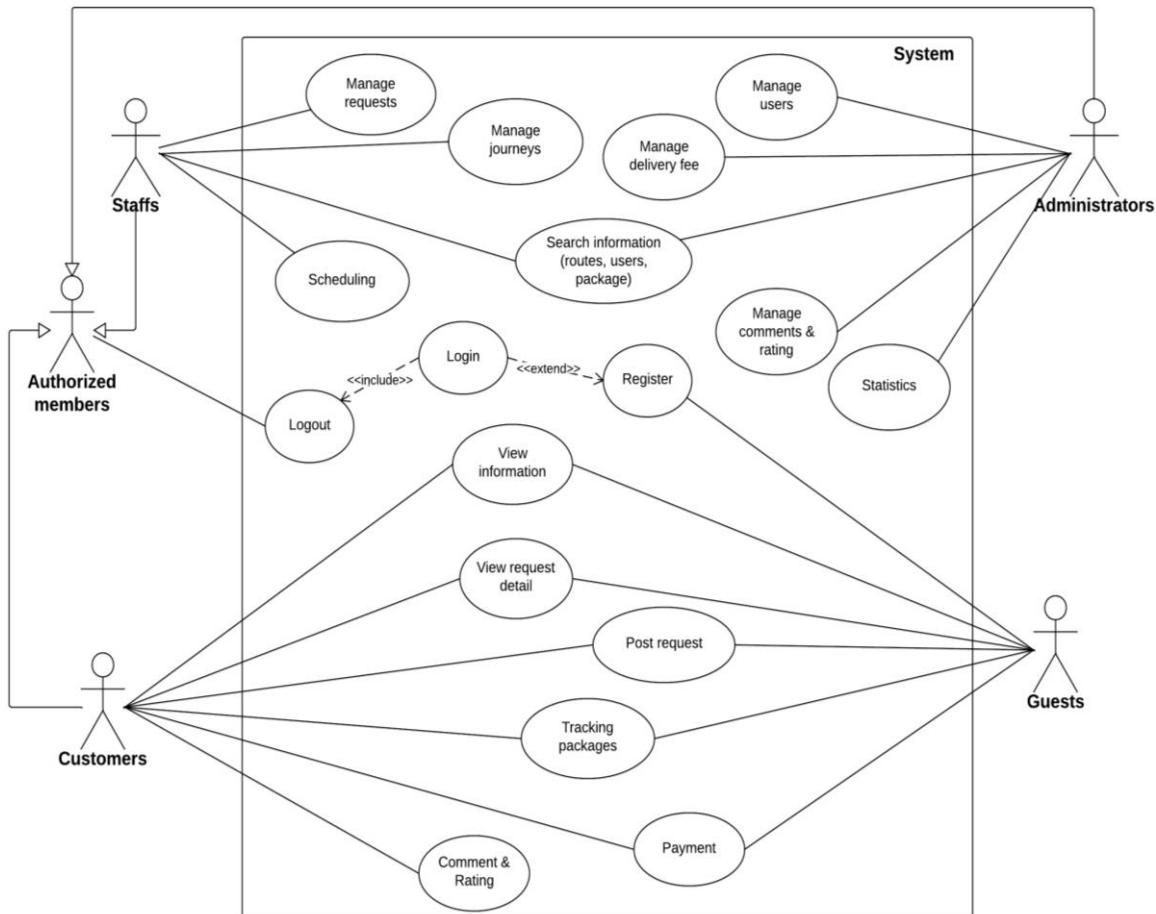
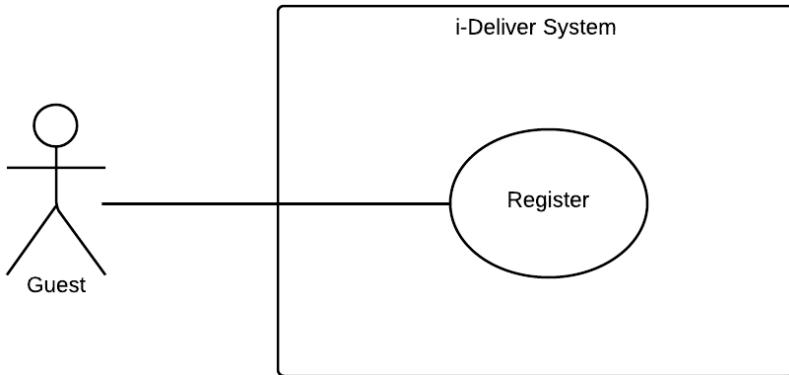


Figure 3. Overall use case

3.2.2.2. « Guest » Register



- SPECIFICATION

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Register		
Author	Nguyễn Tân Đức		
Date	07/02/2014	Priority	Normal

Actor: Guest

Summary:

This use case help guest to create an account in the system.

Goal:

Help Guest can have an account in the system. Guest can use this account to log in system and post their request.

Triggers:

- In the home page guest clicks on “Đăng Kí”.
- Input new information account.
- Click “Hoàn tất” to finish.

Preconditions:

Guest needs an account to login in to the system in other to have more authorization in the system.

Post Conditions:

On success: A new account is created in the system. This account is user account type.

On failure: Show error message describe invalid fields, ask guest enter again or ask guest register again.

Main Success Scenario:

Step	User Action	System Response
1.	Guest clicks on “Đăng Kí”	
2.		Display “Register” screen with text box for Guest to input all necessary info for creating new account.
3.	Guest inputs all required fields.	
4.		Validate all input data. Send invalid message (if have).
5.	Correct all invalid fields (if have).	
6.	Guest clicks on “Đồng Ý” button.	

7.		System creates new account for guest.
8.		Redirect to Login page.

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Hủy bỏ”.	
	2.		Redirect home page.

Exceptions:

No	User Action	System Response
1.	Not enough enter required information	Display message error: "Điền vào những ô yêu cầu"
2.		

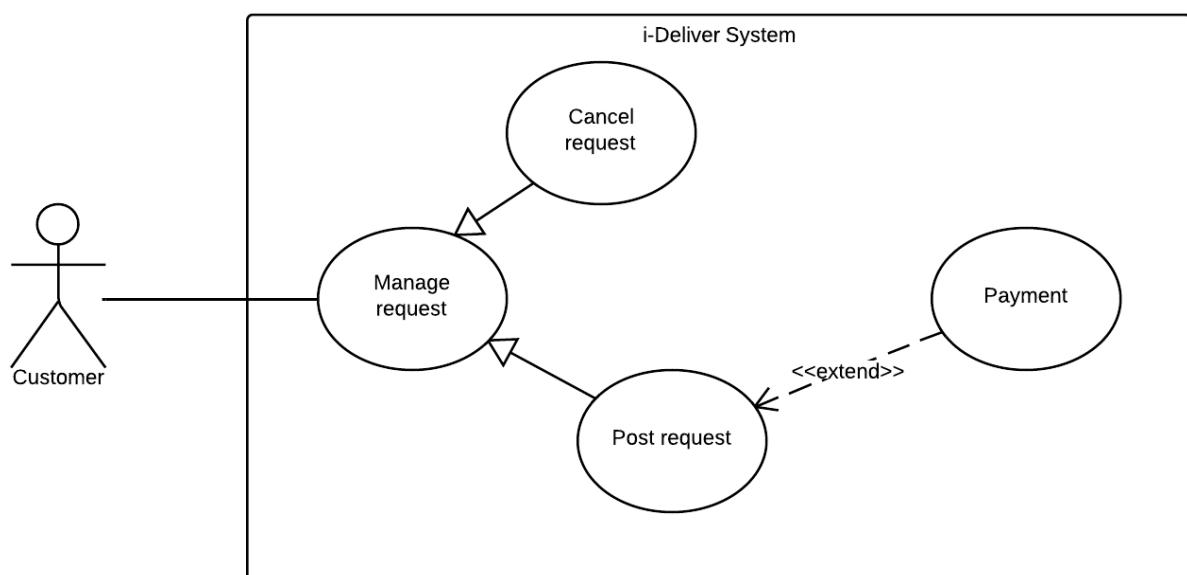
Relationships:

“Login” use case.

Business Rules:

No	Business Rule Description
1.	Guest must enter all required fields before submitting.
2.	Each registering, guest can create one account in the system.
3.	Each account in the system is distinguished by email address which assumes as username.
4.	When email fields of the new account are as same as the existing ones, the system will show invalid message.

3.2.2.3. « Customer, Guest » Post request



– SPECIFICATION

Use-case No.	UC002	Use-case	1.0
--------------	-------	----------	-----

		Version	
Use-case Name	Post request		
Author	Nguyễn Tân Đức		
Date	07/02/2014	Priority	Normal

Actor: Customer, Guest

Summary:

This use case is about allowing customer can post request to delivery their goods in the system.

Goal:

Customer can describe about their goods (size, weight, etc.), choose the place where to send the goods and get suggest fee from the system.

Triggers:

- In the home page, customer clicks on “Gửi yêu cầu” link.
- Input information to post request delivery.
- Click “Hoàn tất” to finish.

Preconditions:

- Customer logon with customer role.
- Customer post request for delivery goods.

Post Conditions:

On success: Request for transporting goods will be created.

On failure: Show message error. Ask customer request again.

Main Success Scenario:

Step	User Action	System Response
1.	Customer logon to the system with customer role.	
2.	Customer clicks “Gửi yêu cầu” on menu.	
3.		Display “Post request” screen with all textbox for customer to enter all necessary info for delivering goods.
4.	Customer enters all required fields.	
5.		Validate all input data. Send invalid message (if have).
6.	Correct all invalid fields (if have).	
7.	Customer click on “Hoàn tất” button.	
8.		System create new request for customer.

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Hủy bỏ”.	
	2.		Redirect home page.

Exceptions:

No	User Action	System Response
1.	Not enough enter required information	Display message error: "Điền vào những ô yêu cầu"
2.	Enter invalid size of package.	Display message error: "Chiều dài/ rộng/ cao vượt quá giới hạn"

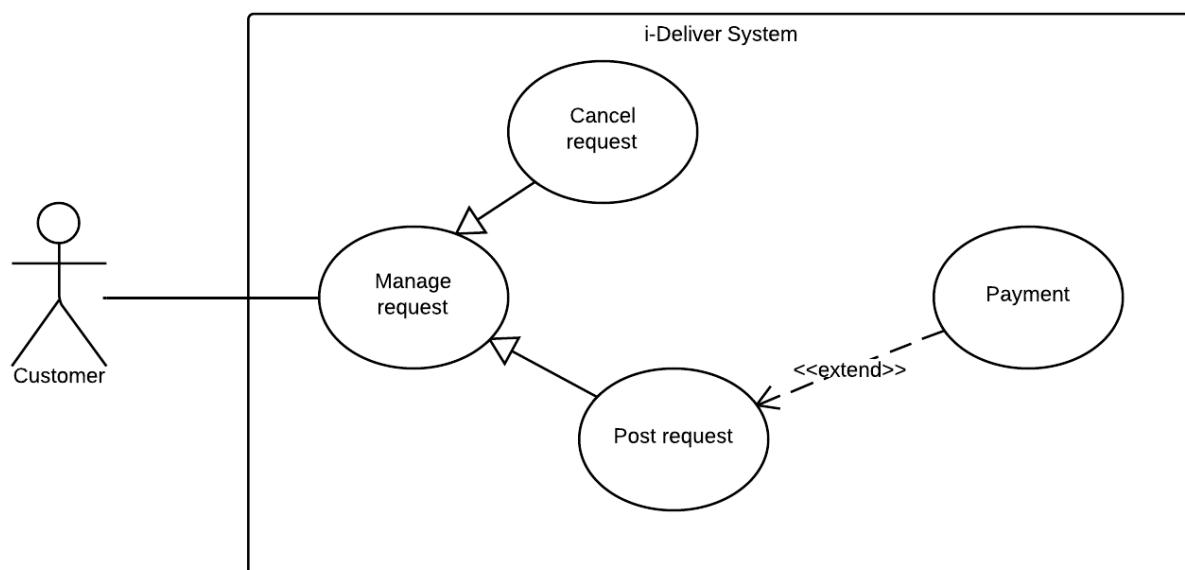
Relationships:

“Login” use case.

Business Rules:

No	Business Rule Description
1.	Customer must login to use the system function.
2.	Customer must enter all required fields before submitting.

3.2.2.4. « Customer » Cancel request



– SPECIFICATION

Use-case No.	UC003	Use-case Version	1.0
Use-case Name	Cancel request		
Author	Nguyễn Thanh Tùng		
Date	07/02/2014	Priority	Normal

Actor: Customer

Summary:

This use case is about allowing customer to cancel request for delivery goods.

Goal:

Cancel request for delivery goods.

Triggers:

N/A

Preconditions:

Customer login to the information with their role.

Customer must have previously submitted request.

Post Conditions:

On success: Request for delivery goods is removed.

On failure: N/A

Main Success Scenario:

Step	User Action	System Response
1.	Click “Quản lý yêu cầu” link in menu bar	
2.		Show list of request status on screen (Status of request: “Chờ xác nhận” and “Đã chuyển đi”)
3.	Click status “Chờ xác nhận”	
4.		Show list of request “Chờ xác nhận” with info each of request
5.	Choose the request want to cancel and click “Xoá” button.	
6.		Display screen to confirm cancel request.
7.	Click “Xác nhận” button	
8.		Redirect screen list request “Chờ xác nhận”

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Hủy bỏ”.	
	2.		Redirect screen list request “Chờ xác nhận”

Exceptions:

N/A

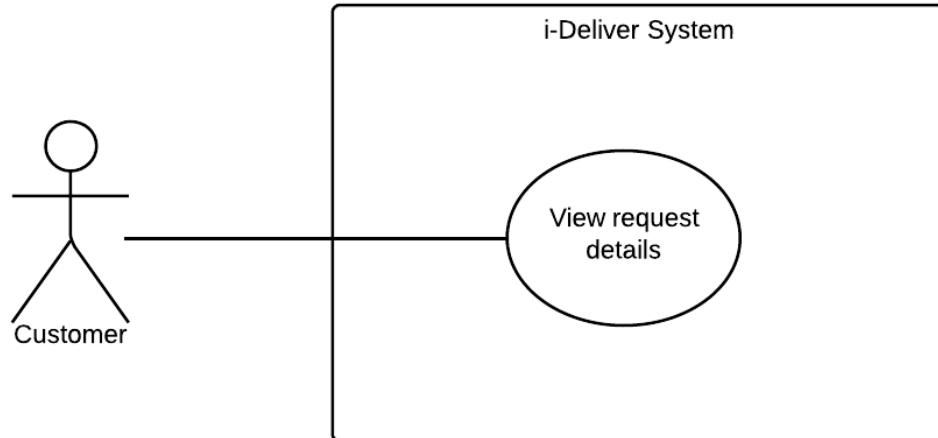
Relationships:

N/A

Business Rules:

No	Business Rule Description
1.	Customer must have previously submitted request.

3.2.2.5. « Customer, Guest » View request detail



– SPECIFICATION																		
Use-case No.	UC006	Use-case Version	1.0															
Use-case Name	View request detail																	
Author	Nguyễn Tân Đức																	
Date	11/02/2014	Priority	Normal															
Actor:	Customer, Guest																	
Summary:	This use case allows customer view all information of their request.																	
Goal:	View information of request.																	
Triggers:	Customer click on a request from list of requests.																	
Preconditions:	Customer must have previously submitted request. Customer login to the system. List of request page must be displayed.																	
Post Conditions:	<i>On success:</i> Show all information of request. <i>On failure:</i> Show message error.																	
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>User Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1.</td><td>Click “Quản lý yêu cầu” link in menu bar</td><td></td></tr> <tr> <td>2.</td><td></td><td>Show list of request status on screen(Status of request: “Chờ xác nhận” and “Đã chuyển đi”)</td></tr> <tr> <td>3.</td><td>Click status “Chờ xác nhận”</td><td></td></tr> <tr> <td>4.</td><td></td><td>Show list of request “Chờ xác nhận” with info each of request</td></tr> </tbody> </table>			Step	User Action	System Response	1.	Click “Quản lý yêu cầu” link in menu bar		2.		Show list of request status on screen(Status of request: “Chờ xác nhận” and “Đã chuyển đi”)	3.	Click status “Chờ xác nhận”		4.		Show list of request “Chờ xác nhận” with info each of request
Step	User Action	System Response																
1.	Click “Quản lý yêu cầu” link in menu bar																	
2.		Show list of request status on screen(Status of request: “Chờ xác nhận” and “Đã chuyển đi”)																
3.	Click status “Chờ xác nhận”																	
4.		Show list of request “Chờ xác nhận” with info each of request																
Alternative Scenario:																		

No	Step	User Action	System Response
1.	1.	Click status “Chờ xác nhận” button	
	2.		Show message “Không tìm thấy kết quả”.

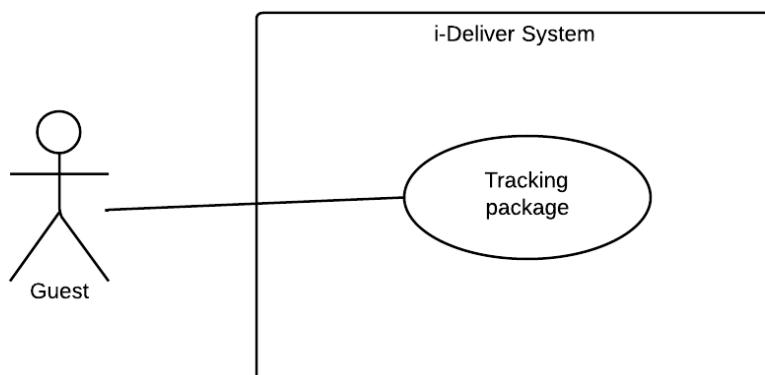
Exceptions:
N/A

Relationships:
N/A

Business Rules:

No	Business Rule Description
1.	Customer must have previously submitted request.

3.2.2.6. « Customer, Guest » Tracking package



– SPECIFICATION			
Use-case No.	UC007	Use-case Version	1.0
Use-case Name	Tracking package		
Author	Nguyễn Thanh Tùng		
Date	11/02/2014	Priority	Normal

Actor: Customer, Guest

Summary:
This use case allows customer can check amount of time that their package was sent.

Goal:
Help customer can track their package.

Triggers:
N/A

Preconditions:
Customer login to the system.

Customer must have previously submitted request and this request is pending.

Post Conditions:

On success:

On failure:

Main Success Scenario:

Step	User Action	System Response
1.	Click “Quản lý yêu cầu” link in menu bar	
2.		Show list of request status on screen (Status of request: “Chờ xác nhận” and “Đã chuyển đi”)
3.	Click status “Đã chuyển đi” button	
4.		Show list of request “Đã chuyển đi”
5.	Choose request need to tracking	
6.		Show result in Google map

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Click status “Đã chuyển đi” button	
	2.		Show message “Không tìm thấy kết quả”.

Exceptions:

N/A

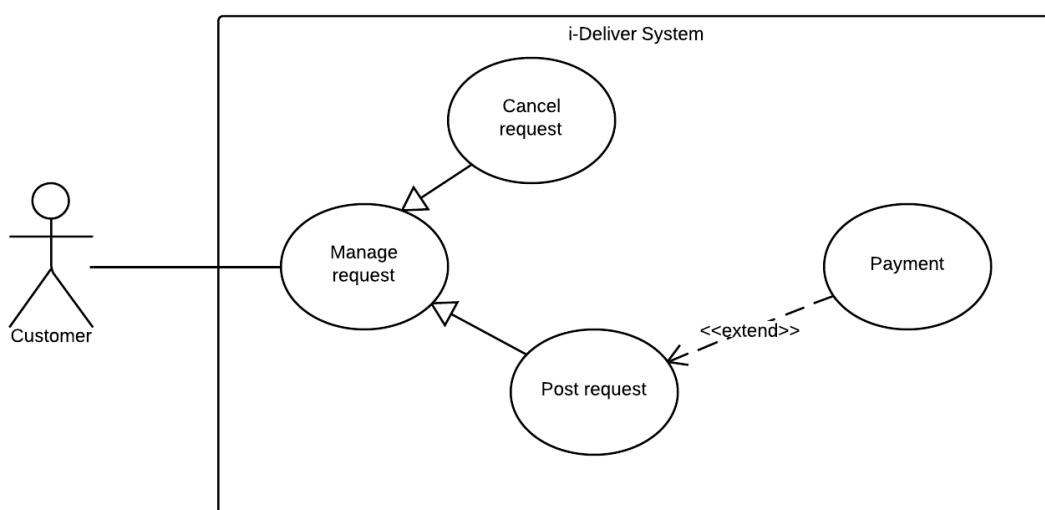
Relationships:

N/A

Business Rules:

N/A

3.2.2.7. « Customer » Payment



- SPECIFICATION

Use-case No.	UC008	Use-case Version	<1.0>
Use-case Name	Payment		
Author	Đào Bảo Long		
Date	11/02/2014	Priority	Normal

Actor: Customer

Summary:

This use case allows customer can choose a way to payment (direct or online payment).

Goal:

Pay to use the service. Customer can pay directly with staff or paying online through PayPal.

Triggers:

Customers/Guests can make a payment via PayPal online payment service.

Preconditions:

Customer must login to the system.

Guest must have request code.

Post Conditions:

On success: Customers/Guests can pay for delivery fee online.

On failure: System will alert error.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Thanh toán” link in left menu bar	
2.		Show list of requests required payment.
3.	Choose request to payment	
4.		Show information of request before payment.
5.	Click “Check out with PayPal” button	
6.		Redirect PayPal page
7.	Customer enters some required fields and click “Pay” button.	
8.		Redirect success payment page.

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Click “Đăng nhập” link in the left menu bar	
	2.		Browser popup a dialog with “Tra cứu thông tin gói hàng” link
	3.	Clicks on “Tra cứu thông tin gói hàng” link	
	4.		A textbox for request code input appear below link
	5.	Input request code from	

		SMS	
	6.		Display request information
	7.	Click “Check out with PayPal” button	
	8.		Redirect PayPal page
	9.	Customer enters some required fields and click “Pay” button.	
	10.		Redirect success payment page.

Exceptions:

No	User Action	System Response
1.	Guest input invalid request code.	System alert invalid code.

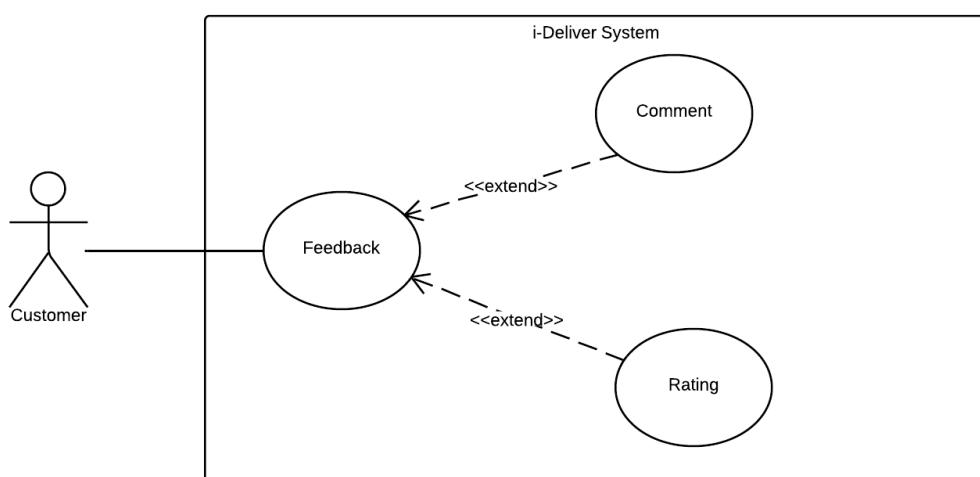
Relationships:

“Post request” use case

Business Rules:

- Customers/Guests posted delivery request

3.2.2.8. « Customer » Rating



RATING – SPECIFICATION			
Use-case No.	UC009	Use-case Version	1.0
Use-case Name	Rating		
Author	Nguyễn Tân Đức		
Date	09/02/2014	Priority	Normal

Actor: Customer.

Summary:

This use case allows customer can make a small feedback to center by rating the service of a specific route. When rating, he/she clicks to the star and the number of marked stars is the level of rating for this route.

Goal:

To rate the quality of the service of route of the center.

Triggers:

- Customer click on “Đóng góp ý kiến” link.
- Customer click “Góp ý” button in the station.
- Customer clicks to the stars to rate after he/she used the service of this route.

Preconditions:

The customer enters to the page contains the list of route and for each route; there is a rating level corresponding.

Post Conditions:

On success: The star from the leftmost to the star that user clicked to are all marked.

Then, there is a sentence notices the level of customer rated lies beside the list of stars, i.e., “Bạn đã đánh giá 4/5 điểm cho dịch vụ này”.

On failure: There is nothing changed and the star is not marked.

Main Success Scenario:

Step	User Action	System Response
1.	User click to one of five stars corresponding route to select the value that they think it should be.	
2.		<p>Display the information that user rated for the route: “Bạn đã đánh giá x/5 điểm cho tuyến đường này” with x is the rating level.</p> <p>Saves the rating point for the route and updates the overall rating average point and rating average point for the currently rated route.</p>

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	Customer clicks on the star but system encounters an exception when saving the rating point for the route.	Shows the popup to notice about the error: “Không thể lưu đánh giá này do hệ thống có vấn đề”.

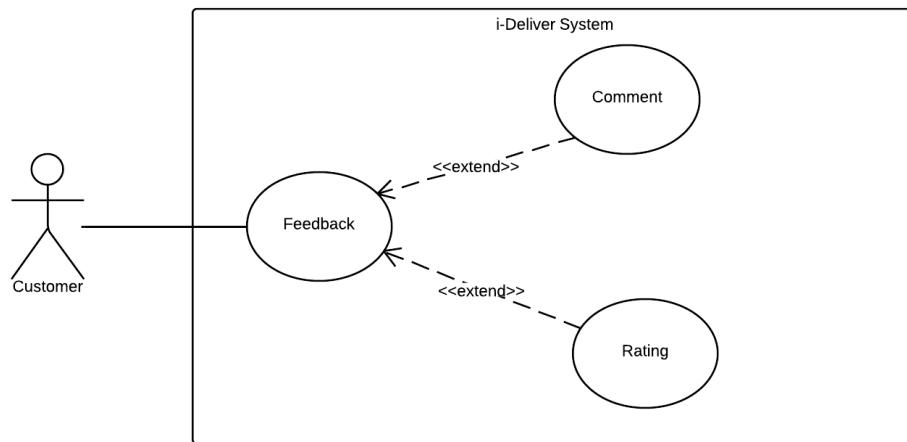
Relationships:

“Clear rating” use case.

Business Rules:

No	Business Rule Description
1.	This use case only for customer that logged in to the system so when guest rating, the i-Deliver check authentication and don't perform the rating level.
2.	When user re-rates for the route, the old rate level will be updated by new rating.
3	The rating level is the integer in interval [0; 5] inclusively but the average point maybe is a floating point number.

3.2.2.9. « Customer » Comment



COMMENT – SPECIFICATION			
Use-case No.	UC010	Use-case Version	1.0
Use-case Name	Comment		
Author	Nguyễn Tân Đức		
Date	09/02/2014	Priority	Normal
Actor:	Customer.		
Summary:	<p>This use case allows customer can make a small feedback to center by comment to the service of a specific route. When comment, he/she type the content of comment to the text box in the comment section of each route. After that, he/she click button “Gửi” or click “Enter” to submit the comment to system and this content will be display on the page of route. As long as submit the comment, he/she can edit or delete their own comment when they want.</p>		
Goal:	Comment the quality of the service of route of the center.		
Triggers:	<ul style="list-style-type: none"> - Customer click on “Đóng góp ý kiến” link. 		

- Customer click “Góp ý” button in the station.
- Customer can comment to route after he/she used the service of this route.

Preconditions:

The customer enter to the page contains the list of route and for each route, there is text box to input the content of comment corresponding.

Post Conditions:

On success: The content of comment display on the page of route that customer concerned.

On failure: User can input the comment to text box but cannot submit this content to system

Main Success Scenario:

Step	User Action	System Response
1.	Customer inputs the content of comment to the text box then clicks to button “Gửi”.	
2.		Displays the content of comment that user input. Name of button “Gửi” will be changed to “Chỉnh sửa” so when user want to modify their comment, he/she can click to his button, modify and resubmit.

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Customer doesn't input anything to the text box and clicks to button “Gửi”	
	2.		Displays the popup notice that they want to input the text at least 3 characters, i.e., “Nhận xét của bạn phải có ít nhất 3 kí tự, mời bạn nhập lại”.
2.	1.	Customer inputs the content of comment to the text box and he/she doesn't click to button “Gửi”, then they click to Back on browse or close window.	
	2.		Display the popup notice that they still not submit the comment to system and can be lost it by exit the window.

Exceptions:

No	User Action	System Response
----	-------------	-----------------

1.	Customer inputs the content of comment to the text box then clicks to button “Gửi”.	Shows the popup to notice about the error “Không thể hiện nhận xét này do hệ thống có vấn đề”.
2.	After customer submits the comment, he/she want to modify the content, they clicks to button “Chỉnh sửa”.	Shows the popup to notice about the error “Không thể chỉnh sửa nhận xét này do hệ thống có vấn đề”.

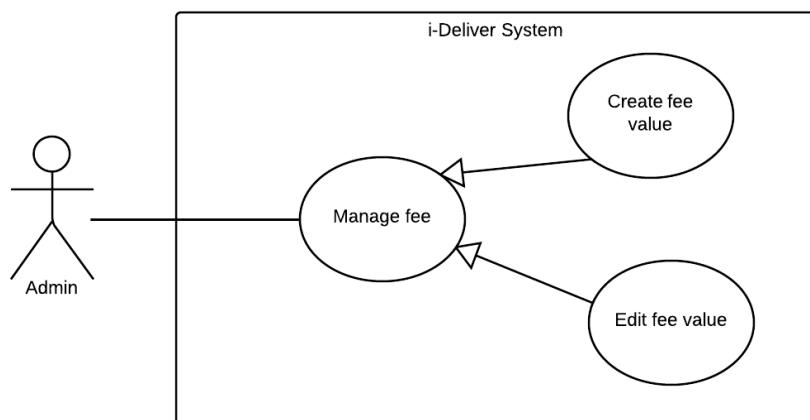
Relationships:

“Delete comment” use case.

Business Rules:

No	Business Rule Description
1.	When customer comments to the route, they can modify the content but cannot delete it. The comment is point to quality of the service not for reply the other comment so can be deleting if illegal.
2.	The content of comment is at least 3 characters differ to “space” character.

3.2.2.10. « System administrator » Define delivery fee value



DEFINE DELIVERY FEE VALUE – SPECIFICATION			
Use-case No.	UC011	Use-case Version	1.0
Use-case Name	Define delivery fee value/calculating formula		
Author	Lê Phúc Lữ		
Date	10/02/2014	Priority	Normal

Actor: System administrator (admin).

Summary:

This use case allows admin can create the fee of delivery service base on the weight and volume of package. In detail, he/she can define the range of weight including max and min value of package's weight; then, for each range, define the range of volume that also including max and min value of package's volume. For a specific range, create the fee for deliver this corresponding package.

Goal:

To create the fee for deliver package base on its weight and volume.

Triggers:

N/A.

Preconditions:

Admin logged in to the system and enter to the fee setting page and he/she doesn't setting anything about the fee before.

Post Conditions:

On success: The information of fee value will be inserted successfully.

On failure: The information of fee value not inserted or insert incorrectly.

Main Success Scenario:

Step	User Action	System Response
1.	Admin click “Tạo mức phí mới” link button in menu bar	
2.		Display form “Tạo mới phí mới ” on screen
3.	Admin enter all fields required. Click “Tạo mới” button.	
4.		If some fields wrong, system ask admin enter again.
5.	Admin enter all fields required. Click “Tạo mới” button.	
6.		New fee is created. Redirect list of fee page.

Alternative Scenario:

N/A

Exceptions:

N/A.

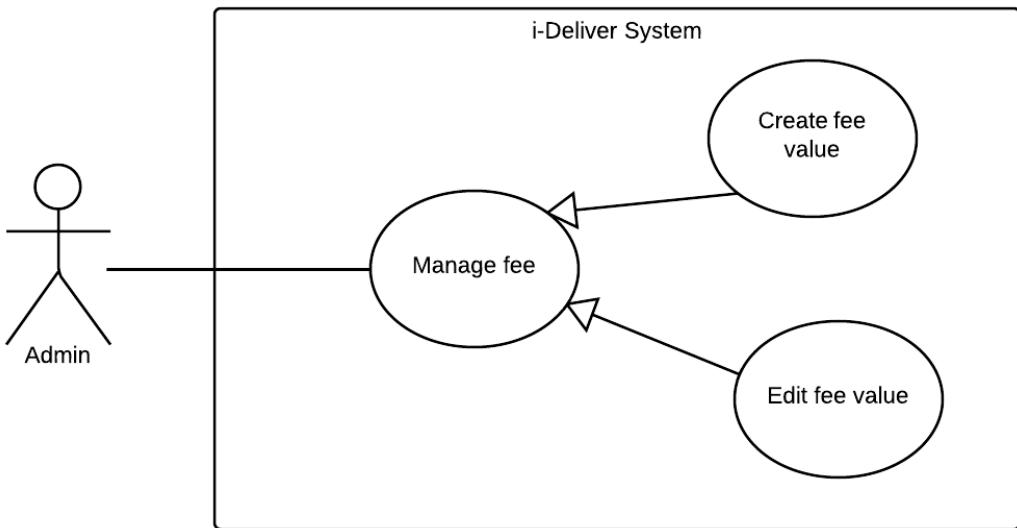
Relationships:

“Edit fee value” use case.

Business Rules:

No	Business Rule Description
1.	Login into the system with admin role.

3.2.2.11. « System administrator » Edit delivery fee value/ calculating formula



EDIT DELIVERY FEE VALUE – SPECIFICATION

Use-case No.	UC012	Use-case Version	1.0
Use-case Name	Edit delivery fee value / calculating formula		
Author	Lê Phúc Lữ		
Date	10/02/2014	Priority	Normal

Actor: System administrator (admin).

Summary:

This use case allows admin can modify the fee of delivery service base on the weight and volume of package. After he/she define information before, he/she can update the range of weight including max and min value of package's weight; the range of volume that also including max and min value of package's volume and also the fee for each range of weight and volume.

Goal:

To edit the fee for deliver package base on its weight and volume.

Triggers:

Admin define the fee value before.

Preconditions:

Admin logged in to the system and enter to the fee setting page.

Post Conditions:

On success: The information of fee value updated successfully.

On failure: The information of fee value not updated or updated incorrectly.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Danh sách các mức phí” link button in menu bar.	
2.		Display list of fees.

3.	Choose the fee want to change value and click “Chỉnh sửa” button.	
4.		Display form edit fee for admin can change value of fee.
5.	Admin change some field value and click “Lưu” button.	
6.		Save change new value in database and redirect list of fees page.

Alternative Scenario:

N/A.

Exceptions:

N/A

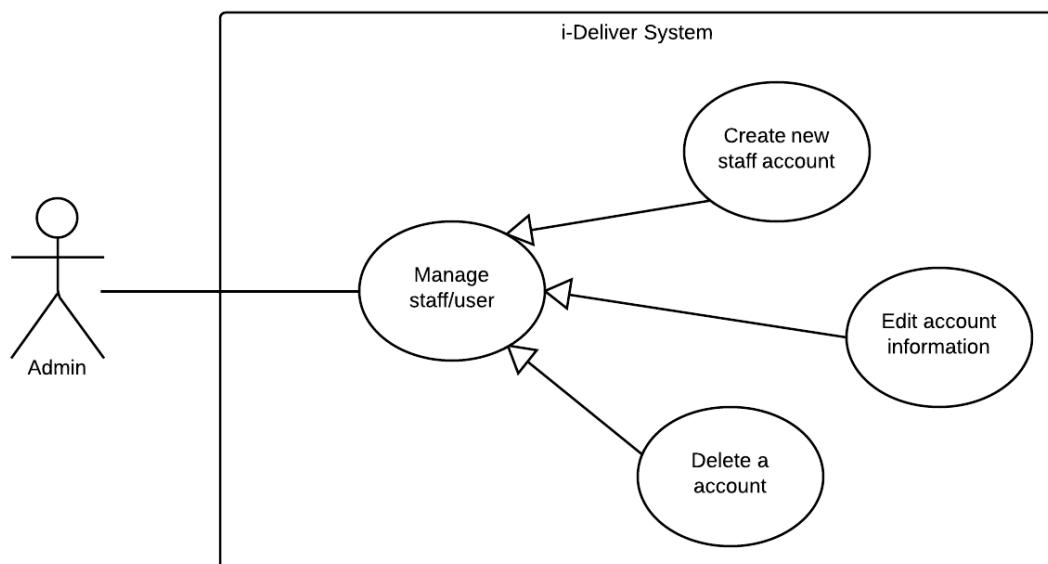
Relationships:

“Add fee value” use case.

Business Rules:

No	Business Rule Description
1.	Login into the system with admin role.

3.2.2.12. « System administrator » Create new staff account



CREATE NEW STAFF ACCOUNT – SPECIFICATION

Use-case No.	UC013	Use-case Version	1.0
Use-case Name	Create new staff account		
Author	Lê Phúc Lữ		

Date	10/02/2014	Priority	Normal
-------------	------------	-----------------	--------

Actor: System administrator (admin)

Summary:

The staffs have an important role in the system when they manage almost the action of system. Admin can create new staff account with username and password but without the basic information as the customer (it is no need on the system). The staff role is authorized to access some specific pages to manage.

Goal:

Create new staff account.

Triggers:

N/A.

Preconditions:

Admin logged in to the system and enter to the manage account page.

Post Conditions:

On success: New account with correct account and password insert successfully.

On failure: No new account added or wrong information added.

Main Success Scenario:

Step	User Action	System Response
1.	Admin click “Tạo tài khoản mới” link button in menu bar	
2.		Display form “Tạo tài khoản mới” on screen
3.	Admin enter all fields required. Click “Tạo mới” button.	
4.		If some fields wrong, system ask admin enter again.
5.	Admin enter all fields required. Click “Tạo mới” button.	
6.		New account is created. Redirect list of account page.

Alternative Scenario:

N/A

Exceptions:

N/A

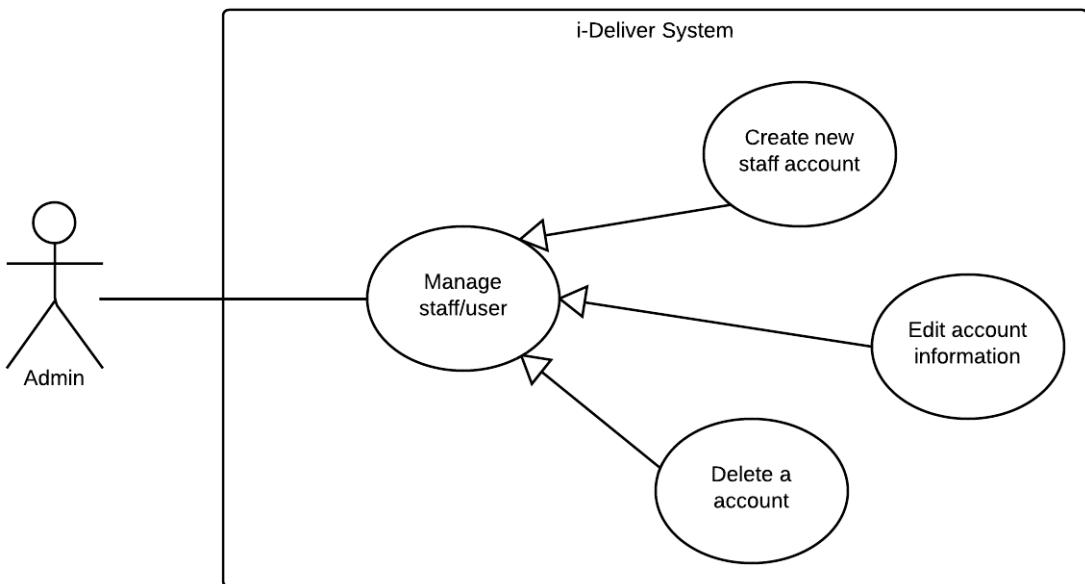
Relationships:

“Add fee value” use case.

Business Rules:

No	Business Rule Description
1.	Login into the system with admin role.

3.2.2.13. « System administrator » Edit account information



EDIT ACCOUNT INFORMATION – SPECIFICATION

Use-case No.	UC014	Use-case Version	1.0
Use-case Name	Edit account information		
Author	Lê Phúc Lữ		
Date	10/02/2014	Priority	Normal

Actor: System administrator (admin)

Summary:

This use case allows admin can update the information of staff or user account due to some special reason.

Goal:

Edit the account (including staff and customer) information.

Triggers:

N/A.

Preconditions:

Admin logged in to the system and enter to the manage account page.

Post Conditions:

On success: The updated information is saved successfully.

On failure: No new update information are saved or information incorrect.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Danh sách nhân viên” link button in menu bar.	
2.		Display list of accounts.
3.	Choose the account want to change information and click “Chỉnh sửa” button.	

4.		Display form edit account for admin can change information.
5.	Admin change information of account and click “Lưu” button.	
6.		Save change in database and redirect list of account page.

Alternative Scenario:

N/A

Exceptions:

N/A

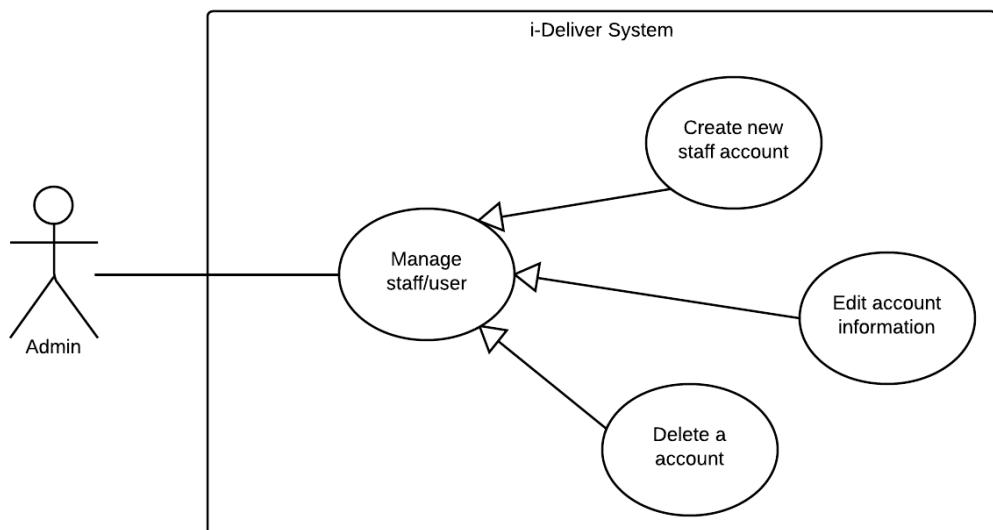
Relationships:

N/A

Business Rules:

No	Business Rule Description
1.	User login into the system with admin role.

3.2.2.14. « System administrator » Delete account



DELETE ACCOUNT – SPECIFICATION			
Use-case No.	UC015	Use-case Version	1.0
Use-case Name	Delete account		
Author	Lê Phúc Lữ		
Date	10/02/2014	Priority	Normal

Actor: System administrator (admin)

Summary:

This use case allows admin can delete account of staff or customer. When some staff don't work for the center anymore so he/she cannot access to the manage page of system. And in the other, if some customer has registered an account at the page for the bad objective, then admin can delete their account.

Goal:

Delete the account of staff or customer.

Triggers:

N/A.

Preconditions:

Admin logged in to the system and enter to the manage account page.

Post Conditions:

On success: The account delete successfully.

On failure: The account still not is deleted.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Danh sách nhân viên” link button in menu bar.	
2.		Display list of accounts.
3.	Choose the account want to change information and click “Xoá” button.	
4.		Display form confirms to delete account.
5.	Admin click “Xác nhận” button	
6.		Account will be delete, redirect list of accounts page

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Click status “Hủy” button	
	2.		Redirect list of accounts page.

Exceptions:

N/A

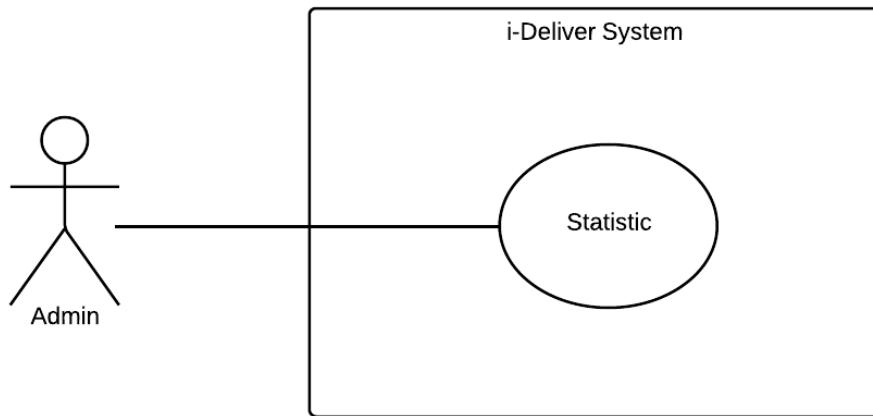
Relationships:

N/A.

Business Rules:

N/A

3.2.2.15. « System administrator » Statistic



STATISTIC – SPECIFICATION

Use-case No.	UC017	Use-case Version	1.0
Use-case Name	Statistic		
Author	Lê Phúc Lữ		
Date	10/02/2014	Priority	Normal

Actor: System administrator (admin)

Summary:

This use case allows admin statistic information of system, especially statistic on customer account.

Goal:

Statistic the information of account.

Triggers:

N/A.

Preconditions:

Admin logged in to the system and enter to the statistic page.

Post Conditions:

On success: The result of statistic information will be displayed.

On failure: The statistic information is not displayed.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Thống kê” button in menu bar.	
2.		Display statistic information of account by pie chart.

Alternative Scenario:

N/A

Exceptions:

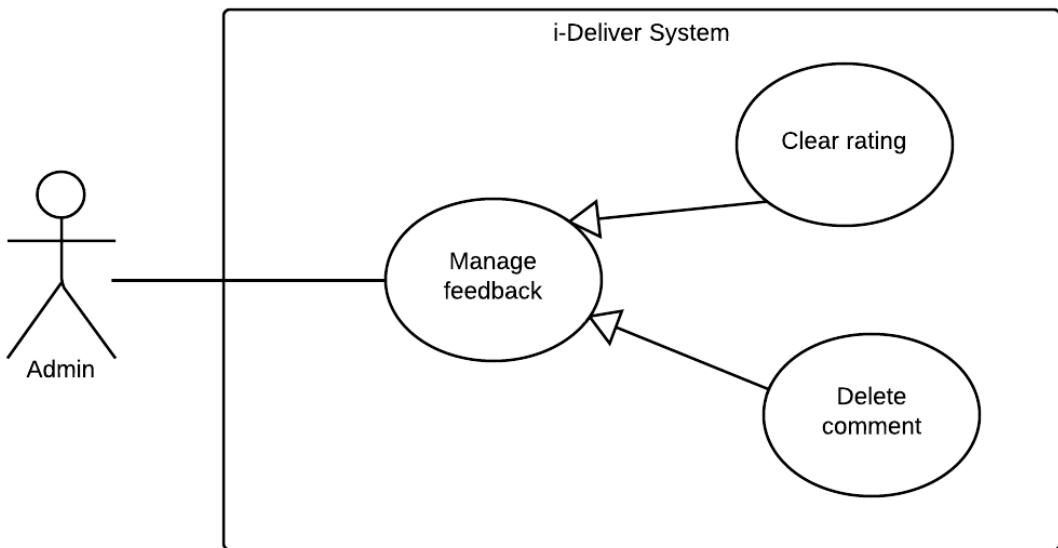
N/A

Relationships:

N/A.

Business Rules: N/A

3.2.2.16. « System administrator » Delete comment



DELETE COMMENT – SPECIFICATION

Use-case No.	UC018	Use-case Version	1.0
Use-case Name	Delete comment		
Author	Nguyễn Tân Đức		
Date	09/02/2014	Priority	Normal

Actor: Admin

Summary:

This use case allows admin physical delete the illegal comment for the route, i.e., use the word are not bright meaning, advertise for external service, ...

Goal:

Delete the specific comment of customer for route.

Triggers:

- Admin click on “Quản lý bình luận” link in the admin page.
- The admin can delete the comment that violates the rule.

Preconditions:

Staff enters to the manage feedback page of the system and choose the comment of route for delete.

Post Conditions:

On success: The comment will be deleted successfully.

On failure: The comment won't be deleted as expected.

Main Success Scenario:

Step	User Action	System Response
1.	Admin chooses the comment of route and clicks the button “Xóa nhận xét”.	
2.		Shows a popup to confirm the action that the staff really wants to delete the

		comment. “Bạn có thực sự muốn xóa nhận xét này không?”
3.	If admin clicks to “Yes” button.	
4.		The comment will be deleted.
5.	If admin clicks to “No” button.	
6.		Closes the popup and doesn’t do anything.

Alternative Scenario:

N/A.

Exceptions:

No	User Action	System Response
1.	Admin chooses the comment of route and clicks the button “Xóa nhận xét”. When the popup showed, he/she clicks to “Yes” button to confirm.	The comment is not deleted.
2.	Admin chooses the list of route that he/she want to reset the rating, then click to button “Clear”. When the popup showed, he/she clicks to “No” button to confirm.	The comment is deleted or the content it changed.
3.	Admin chooses the comment of route and clicks the button “Xóa nhận xét”.	The popup is not showed.

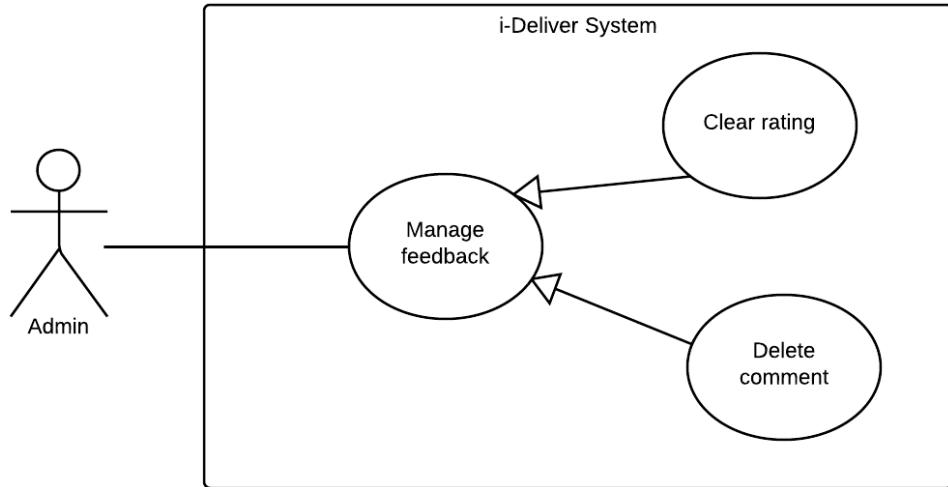
Relationships:

“Comment” use case.

Business Rules:

No	Business Rule Description
1.	The comment is legal if its content is clear and point to the quality of service, not chat, advertise, ... The staff has to read the content of comment carefully to make sure not delete legal comment.

3.2.2.17. « System administrator » Clear rating



CLEAR RATING – SPECIFICATION

Use-case No.	UC019	Use-case Version	1.0
Use-case Name	Clear rating		
Author	Nguyễn Tân Đức		
Date	09/02/2014	Priority	Normal

Actor: Admin

Summary:

This use case allows admin clear the rating of the route due to some reasons. After a period run the system, each route maybe receive a lot of rating for the service, so these rating and the average point of rate stand for the quality of a specific period of time. So to make sure the rating is up-to-date, staff have to reset (or clear) the rating for route.

Goal:

Clear the rating of customer for the route after a period run the system.

Triggers:

Admin can clear the rating after it has been saved or statistic after a long enough period.

Preconditions:

Admin enters to the manage feedback page of the system and choose one or multiple route for clear the rating.

Post Conditions:

On success: The ratings of selected routes will be reset to null.

On failure: The ratings of selected routes are not set to null exactly.

Main Success Scenario:

Step	User Action	System Response
1.	Admin chooses the list of route that he/she want to reset the rating, then click to button “Clear”.	
2.		Shows a popup to confirm the action

		that the staff really wants to clear the rating or not. “Bạn có thực sự muốn reset các đánh giá cũ hay không?”
3.	If admin clicks to “Yes” button.	
4.		All the rating of user rate for this route, also the average rating will be set to null.
5.	If admin clicks to “No” button.	
6.		Closes the popup and doesn't do anything.

Alternative Scenario:

N/A.

Exceptions:

No	User Action	System Response
1.	Admin chooses the list of route that he/she want to reset the rating, then click to button “Clear”. When the popup showed, he/she clicks to “Yes” button to confirm.	The ratings of selected routes are not to null exactly.
2.	Amin chooses the list of route that he/she want to reset the rating, then click to button “Clear”. When the popup showed, he/she clicks to “No” button to confirm.	Some ratings of selected routes are re while user don't expect.
3.	Amin chooses the list of route that he/she want to reset the rating, then click to button “Clear”.	The popup is not showed.

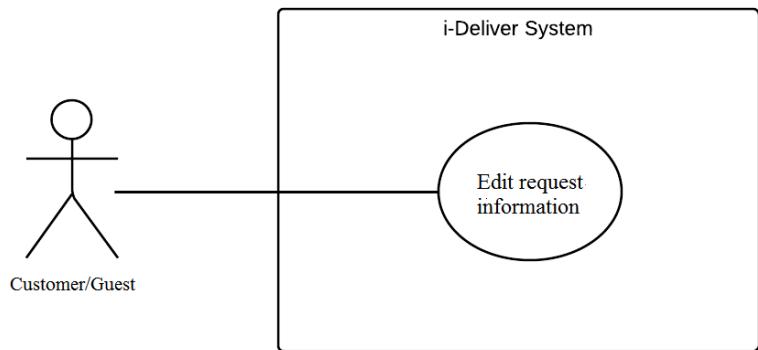
Relationships:

“Rating” use case.

Business Rules:

No	Business Rule Description
1.	The rating of route after clear will become NULL (means there is not any user rate), not become 0.

3.2.2.18. « Customer/Guest » Edit request information



- SPECIFICATION

Use-case No.	UC0018	Use-case Version	1.0
Use-case Name	Edit request information		
Author	Nguyễn Thanh Tùng		
Date	20/5/2014	Priority	Medium

Actor:

Staffs

Summary:

This use case allows customer or guest can update the information of request, which includes sender name, sender address, sender phone, receiver name, receiver's address and receiver phone.

Goal:

To update the request before it will be scheduled by staff.

Triggers:

The customer clicks on button “Chỉnh sửa” on the request list page. With guest, he/she inputs the request code and click on button “Chỉnh sửa”. At that time, the request posted on the system still not processed by staff.

Preconditions:

The request list page is being displayed.

Post Conditions:

On success: The information of request will be update.

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Quản lý yêu cầu” button on the left menu of the customer page. In case guest, he/she should input the request code to find the corresponding request information.	

	[See Exception No. 1]	
2.		Display list of requests that customer had posted into the system.
3.	Click to button “Chỉnh sửa” below the request information box. [See Exception No. 2]	
4.		A page with some editable fields will be showed to customer or guest.
5	Retype the information of the field that he/she want to edit, then click “Lưu lại” button to save.	
6		The system will redirect to the request list page so customer/guest can perform another actions.

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null	Login modal will show
2.	If request not available, and user click on “Xác nhận”.	Redirect to error pages.

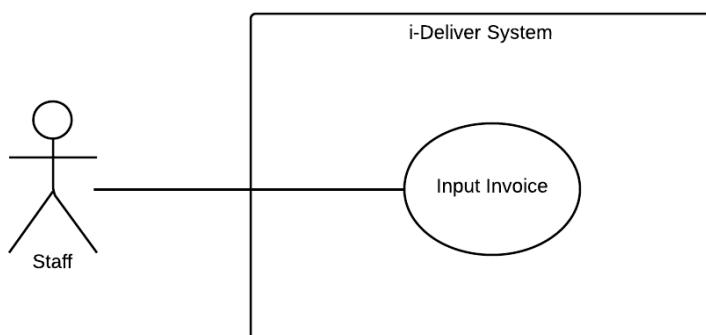
Relationships:

“Post request” use case.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role
2.	Only staff at this station has the right to accept requests on this station.

3.2.2.19. « Staff » Input invoice



– SPECIFICATION

Use-case No.	UC0019	Use-case Version	1.0
Use-case Name	Input invoice		
Author	Nguyễn Thanh Tùng		
Date	20/5/2014	Priority	Medium

Actor:

Staffs

Summary:

This use case allows staffs to input the information of invoice which including: volume, weight and the price.

Goal:

To confirm a request is valid and that request will be processed.

Triggers:

The staffs click on button “Xác nhận thanh toán” on the staff page.

Preconditions:

The staff page is being displayed.

Post Conditions:

On success: The invoice of corresponding request will be added.

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Xác nhận thanh toán” link button in menu bar. [See Exception No. 1]	
2.		Display list of requests that user posted at his/her station.
3.	Click to the button “Nhập hóa đơn” of the request that he/she want to input the invoice. [See Exception No. 2]	
4.		A popup will be showed with the information about weight and volume of request on some editable field.
5	Staff type of exact weight and volume of the invoice of request after they receive the package. Then wait for the suggest price for the invoice and then click to button “Nhập”.	

6		The new invoice of request will be add and the request change status to “Đã thanh toán” so the staff can assign it to some trips.
---	--	---

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null	Login modal will show
2.	If request not available, and user click on “Xác nhận”	Redirect to error pages.

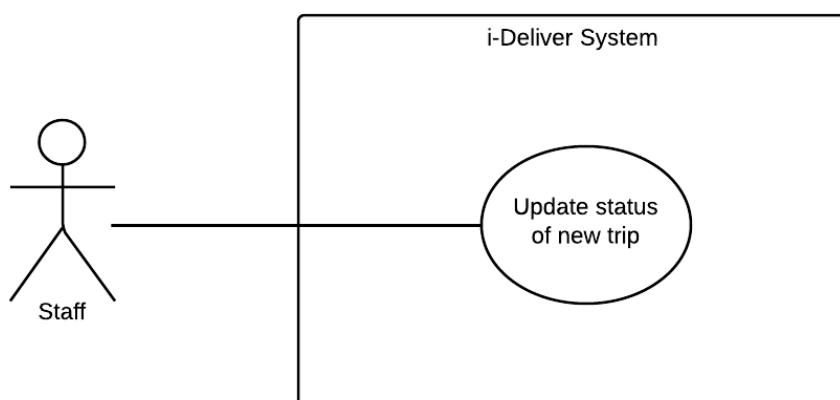
Relationships:

“Post request” use case.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role
2.	Only staff at this station has the right to accept requests on this station.

3.2.2.20. « Staff » Update status of new trip



– SPECIFICATION			
Use-case No.	UC0020	Use-case Version	1.0
Use-case Name	Update status of new trip		
Author	Lê Phúc Lữ		
Date	20/5/2014	Priority	High
Actor:			

Staffs

Summary:

This use case allows staffs to search the trip with given ID and change its status to “Đã khởi hành” which means the created trip departed at that time. Along that action, all request assigned to that trip will be change status to “Đã chuyển đi”. One message will be sent to all customers that have request on the trip.

Goal:

To change the status of trip and all requests that assigned to that trip.

Triggers:

The staffs click on button “Chuyển” on the staff page.

Preconditions:

The staff management page is being displayed.

Post Conditions:

On success: The corresponding trip's status and status of all requests assigned to that trip will be change.

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Chuyển xe đã tạo” link button on the left menu bar. [See Exception No. 1]	
2.		Display list all trips that the staff has created before for the current station and the trips have not departed yet.
3.	Input to the search field the ID of trip that he/she want to change status. [See Exception No. 2]	
4.		System will find out the corresponding trip with the given ID.
5	Click to the link which included in the number plate of the current trip.	
6		A popup contains all the requests that assigned to that trip will be showed. Each request has some information as: request code, from station, to station and date post.
7	Staff views the list of requests and then click to button “Xác nhận”	

	chuyển đi”.	
8		The pop up closes, page will redirect to the trip management before. Then the trip and all requests assigned to it will be changed the status.

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null.	Login modal will show
2.	If the ID of the trip that staff input to the search field doesn't exist.	Page show “Không có dữ liệu” on result.

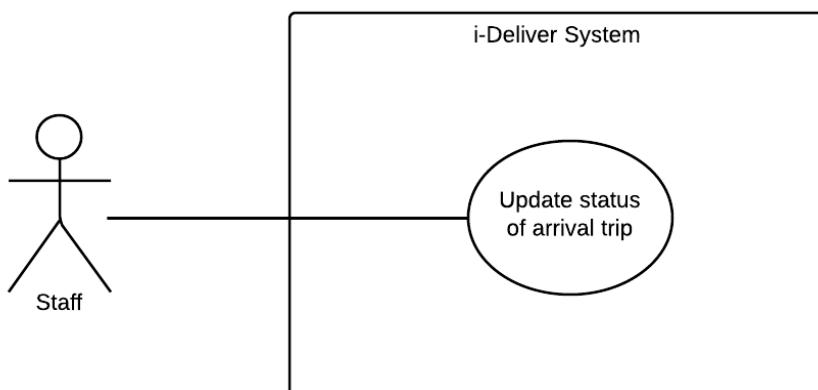
Relationships:

Update status of arrival trip.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role.
2.	Only staff at this station has the right to accept requests on this station.

3.2.2.21. « Staff » Update status of arrival trip



– SPECIFICATION			
Use-case No.	UC0020	Use-case Version	1.0
Use-case Name	Update status of arrival trip		
Author	Lê Phúc Lũ		
Date	20/5/2014	Priority	High
Actor:			

Staffs

Summary:

This use case allows staffs to search the trip with given ID and change its status to “Đã đến trạm” which means the departed trip from other station come to the current station. Along that action, all request assigned to that trip will be updated the status with 2 cases:

- If request sent to that station, its status will be changed status to “Đã đến đích” and a message will be sent to all customers that have request on the trip.
- If request sent to other station, its status will be not changed.

In case the trip comes to that station later than the estimate arrival time, all requests that assigned to trip will be rescheduled.

Goal:

To update the status of the given trip and all requests that assigned to that trip.

Triggers:

The staffs click on button “Chuyển” on the staff page.

Preconditions:

The staff management page is being displayed.

Post Conditions:

On success: The corresponding trip's status and status of all requests assigned to that trip will be change.

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Chuyển từ trạm khác/Xác nhận đến trạm” link button on the menu bar. [See Exception No. 1]	
2.		Display list all trips departed from the other station and cross over the current station.
3.	Input to the search field the ID of trip that he/she want to change status. [See Exception No. 2]	
4.		System will find out the corresponding trip with the given ID.
5	Click to the link which included in the number plate of the current trip.	
6		A popup contains all the requests that assigned to that trip will be showed. Each request has some information as:

		request code, from station, to station and date post.
7	Staff views the list of requests and then click to button “Xác nhận đến trạm”.	
8		The pop up closes, page will redirect to the trip management before. Then the trip and all requests assigned to it will be changed the status.

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null.	Login modal will show
2.	If the ID of the trip that staff input to the search field doesn't exist.	Page show “Không có dữ liệu” on result.

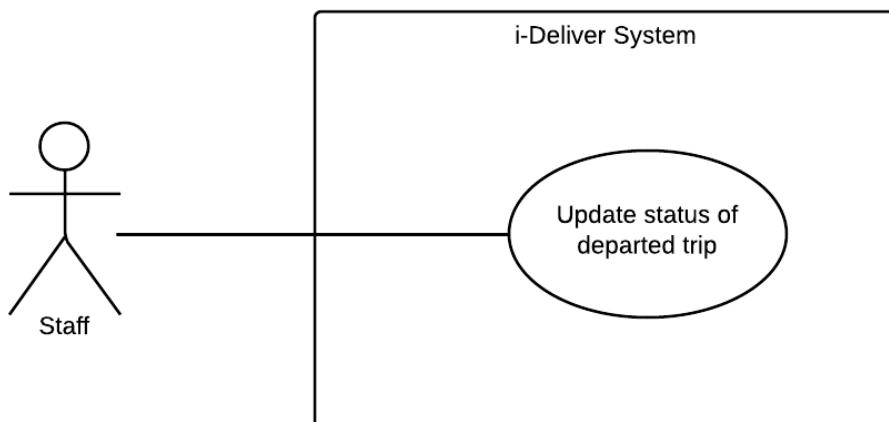
Relationships:

Update status of arrival trip.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role
2.	Only staff at this station has the right to accept requests on this station.

3.2.2.22. « Staff » Update status of departed trip



– SPECIFICATION			
Use-case No.	UC0022	Use-case	1.0

		Version	
Use-case Name	Update status of departed trip		
Author	Nguyễn Tân Đức		
Date	20/5/2014	Priority	High

Actor:

Staffs

Summary:

This use case allows staffs to search the trip with given ID and change its status to “Đã đến trạm” which means the departed trip from other station come to the current station. Along that action, some new request that assigned to that trip will be changed status.

Goal:

To update the status of the given trip and all requests that assigned to that trip.

Triggers:

The staffs click on button “Chuyển” on the staff page.

Preconditions:

The staff management page is being displayed.

Post Conditions:

On success: The corresponding trip's status and status of all requests assigned to that trip will be change.

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Chuyển từ trạm khác/Xác nhận khởi hành” link button on the menu. [See Exception No. 1]	
2.		Display list all trips arrival from the other station and cross over the current station.
3.	Input to the search field the ID of trip that he/she want to change status. [See Exception No. 2]	
4.		System will find out the corresponding trip with the given ID.
5	Click to the link which included in the number plate of the current trip.	
6		A popup contains all the requests that

		assigned to that trip will be showed. Each request has some information as: request code, from station, to station and date post.
7	Staff views the list of requests and then click to button “Xác nhận khởi hành”.	
8		The pop up closes, page will redirect to the trip management before. Then the trip and all requests assigned to it will be changed the status.

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null.	Login modal will show
2.	If the ID of the trip that staff input to the search field doesn't exist.	Page show “Không có dữ liệu” on result.

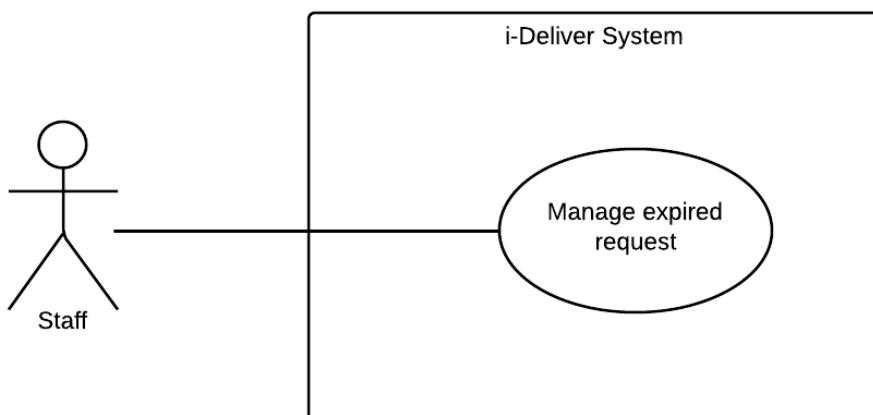
Relationships:

Update status of arrival trip.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role
2.	Only staff at this station has the right to accept requests on this station.

3.2.2.23. « Staff » Manage expired request



- SPECIFICATION																								
Use-case No.	UC0023	Use-case Version	1.0																					
Use-case Name	Manage expired request																							
Author	Nguyễn Tân Đức																							
Date	20/5/2014	Priority	High																					
Actor:	<i>Staffs</i>																							
Summary:	This use case allows staffs view the expired request, which were delivered to the destination but cannot be sent to the user after 5 days.																							
Goal:	To make a decision that cancelling that request or sending back the request to sender.																							
Triggers:	The staffs click on button “Chấp thuận” on the staff page.																							
Preconditions:	The staff page is being displayed.																							
Post Conditions:	<p><i>On success:</i> The request's status will be changed</p> <p><i>On failure:</i> An error message is displayed. The content of the error message will be based on the type of failure.</p>																							
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>User Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1.</td><td>Click “Hàng đã hết hạn giao và chờ trả” link button in menu bar. [See Exception No. 1]</td><td></td></tr> <tr> <td>2.</td><td></td><td>Display list of requests that delivered to the station but have not sent to user after 5 days.</td></tr> <tr> <td>3.</td><td>If customer chooses radio button “Trả hàng” and click “Xác nhận”. [See Exception No. 2]</td><td></td></tr> <tr> <td>4.</td><td></td><td>A new request will be created with the information corresponding to that request but the sender information will be changed to receiver information.</td></tr> <tr> <td>5</td><td>If customer choose radio button “Hủy hàng” and click “Xác nhận”.</td><td></td></tr> <tr> <td>6</td><td></td><td>This request will be cancelled and the</td></tr> </tbody> </table>			Step	User Action	System Response	1.	Click “Hàng đã hết hạn giao và chờ trả” link button in menu bar. [See Exception No. 1]		2.		Display list of requests that delivered to the station but have not sent to user after 5 days.	3.	If customer chooses radio button “Trả hàng” and click “Xác nhận”. [See Exception No. 2]		4.		A new request will be created with the information corresponding to that request but the sender information will be changed to receiver information.	5	If customer choose radio button “Hủy hàng” and click “Xác nhận”.		6		This request will be cancelled and the
Step	User Action	System Response																						
1.	Click “Hàng đã hết hạn giao và chờ trả” link button in menu bar. [See Exception No. 1]																							
2.		Display list of requests that delivered to the station but have not sent to user after 5 days.																						
3.	If customer chooses radio button “Trả hàng” and click “Xác nhận”. [See Exception No. 2]																							
4.		A new request will be created with the information corresponding to that request but the sender information will be changed to receiver information.																						
5	If customer choose radio button “Hủy hàng” and click “Xác nhận”.																							
6		This request will be cancelled and the																						

		system doesn't consider it anymore.
--	--	-------------------------------------

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null	Login modal will show
2.	If request not available, and user click on “Xác nhận”	Redirect to error pages.

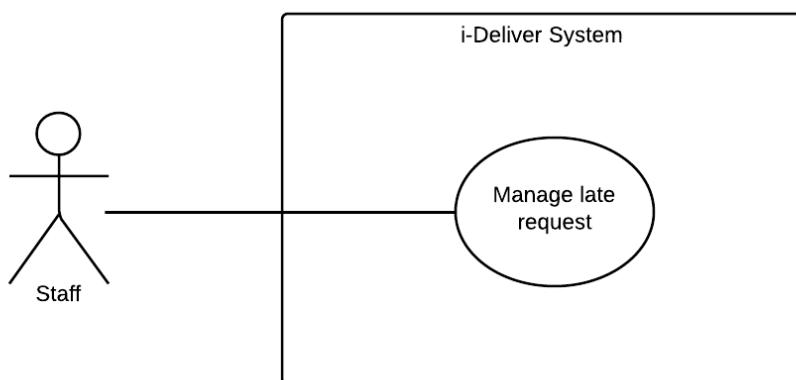
Relationships:

“Post request” use case.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role
2.	Only staff at this station has the right to accept requests on this station.

3.2.2.24. « Staff » Manage late request



– SPECIFICATION			
Use-case No.	UC0024	Use-case Version	1.0
Use-case Name	Manage late request		
Author	Đào Bảo Long		
Date	20/5/2014	Priority	High

Actor:
Staffs

Summary:
This use case allows staffs manage the request posted on the system but after 24h, the customer has not paid for it.

Goal:

To extend the due day of payment of request or not.

Triggers:

The staffs click on button “Trễ hạn thanh toán” on the staff page.

Preconditions:

The staff page is being displayed.

Post Conditions:

On success: The request's status will be changed

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Trễ hạn thanh toán” link button in menu bar. [See Exception No. 1]	
2.		Display list of requests that customer doesn't pay for after 24 hours.
3.	Select the request that staff wants to extend the due day and click button “Gia hạn thanh toán”. [See Exception No. 2]	
4.		The day post of request will be changed to the current time and its status will be changed.
5	Select the request that staff want to cancel and click button “Hủy”. [See Exception No. 2]	
6		This request will be cancelled and the system doesn't consider it anymore.

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null	Login modal will show
2.	If request not available, and user click on “Xác nhận”	Redirect to error pages.

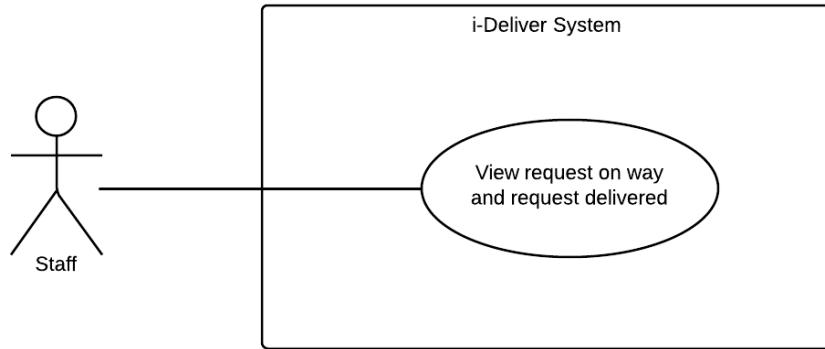
Relationships:

“Post request” use case.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role
2.	Only staff at this station has the right to accept requests on this station.

3.2.2.25. « Staff » View request on way and request delivered



– SPECIFICATION			
Use-case No.	UC0025	Use-case Version	1.0
Use-case Name	View request on way and request delivered		
Author	Đào Bảo Long		
Date	20/5/2014	Priority	High
Actor:	<i>Staffs</i>		
Summary:	This use case allows staffs view the information of request on way and request delivered in two cases as following: - Request of the current station. - Request of the other station.		
Goal:	To view information of request on way and request delivered.		
Triggers:	The staffs click on button “Hàng đang đến trạm/Hàng đã đến đích” on the staff page.		
Preconditions:	The staff page is being displayed.		
Post Conditions:	<i>On success:</i> The request's status will be changed <i>On failure:</i> An error message is displayed. The content of the error message will be based on the type of failure.		
Main Success Scenario:			

Step	User Action	System Response
1.	Click “Hàng đang vận chuyển” link button in menu bar. [See Exception No. 1]	
2.		Display list of requests from the current station which are on way.
3.	Click “Hàng đã được giao” link button in menu bar. [See Exception No. 1]	
4.		Display list of requests from the current station which are delivered to the customer.
5.	Click “Hàng đang đến trạm” link button in menu bar. [See Exception No. 1]	
6.		Display list of requests from the other station and pass over the current which are on way.
7.	Click “Hàng đã đến đích” link button in menu bar. [See Exception No. 1]	
8.		Display list of requests from the other station and came to the current station.

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null	Login modal will show
2.	If request not available, and user click on “Xác nhận”	Redirect to error pages.

Relationships:

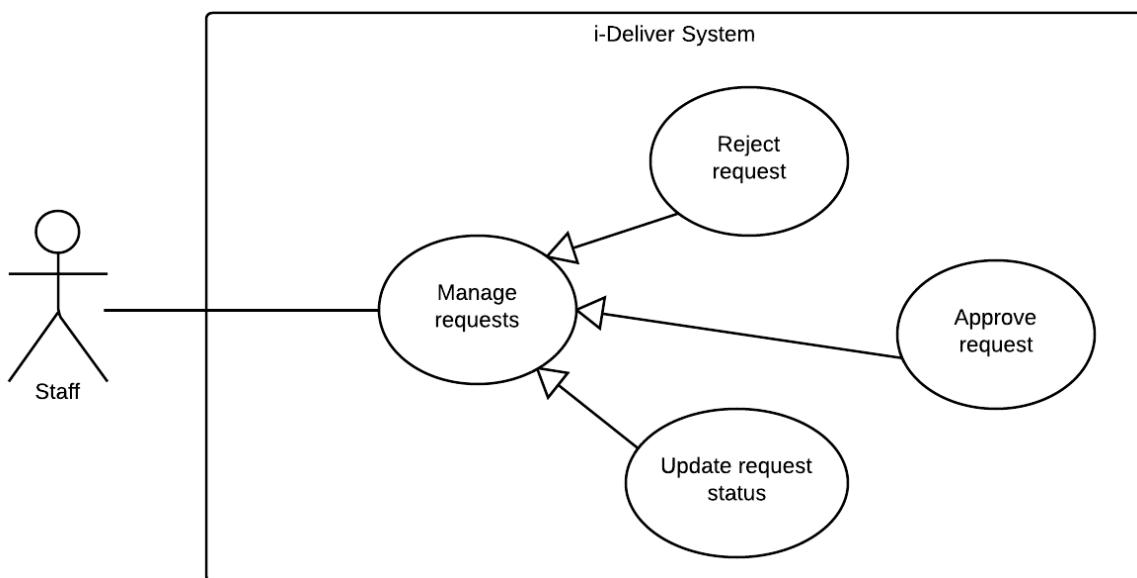
“Post request” use case.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role

- | | |
|----|--|
| 2. | Only staff at this station has the right to accept requests on this station. |
|----|--|

3.2.2.26. « Staff » Approve request



– SPECIFICATION

Use-case No.	UC0020	Use-case Version	1.0
Use-case Name	Approve request		
Author	Nguyễn Thanh Tùng		
Date	10/2/2014	Priority	High

Actor:

Staffs

Summary:

This use case allows staffs to approve a request from customers or in other words the staffs will change the status of request.

Goal:

To confirm a request is valid and that request will be processed.

Triggers:

The staffs click on button “Chấp thuận” on the staff page.

Preconditions:

The staff page is being displayed.

Post Conditions:

On success: The request's status will be changed

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
------	-------------	-----------------

1.	Click “Tiếp nhận yêu cầu” link button in menu bar. [See Exception No. 1]	
2.		Display list of requests that customer request.
3.	Select the request want to approve and click “Xác nhận” button. [See Exception No. 2]	
4.		Request is accepted and system redirect list of requests page.

Alternative Scenario:

N/A

Exceptions:

No	User Action	System Response
1.	If session equals null	Login modal will show
2.	If request not available, and user click on “Xác nhận”	Redirect to error pages.

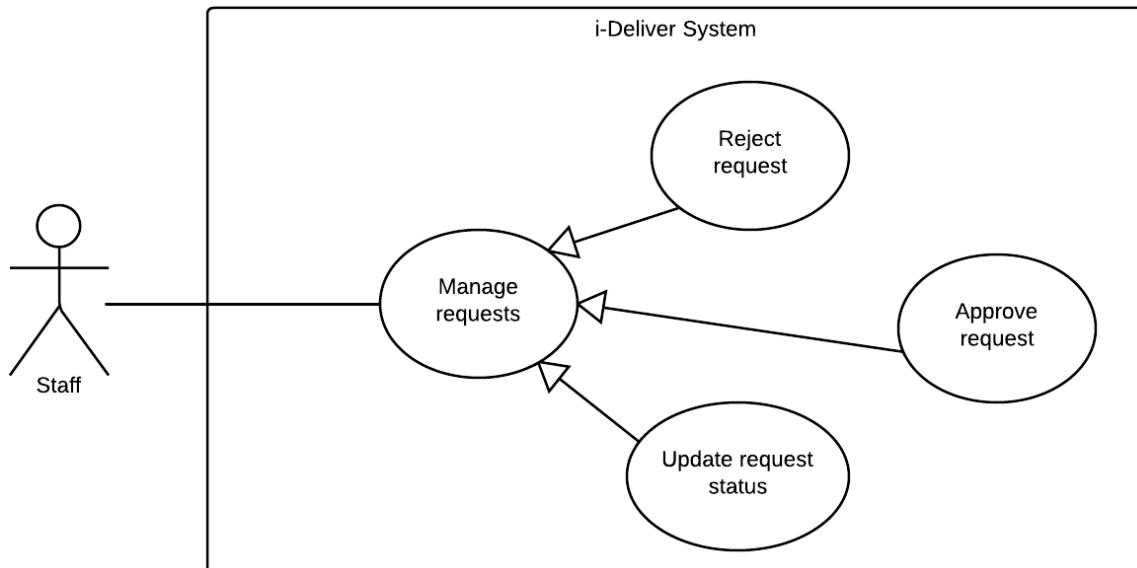
Relationships:

“Post request” use case.

Business Rules:

No	Business Rule Description
1.	User login into the system with staff role
2.	Only staff at this station has the right to accept requests on this station.

3.2.2.27. « Staff » Reject request



– SPECIFICATION			
Use-case No.	UC0021	Use-case Version	1.0

Use-case Name	Reject request		
Author	Nguyễn Thanh Tùng		
Date	10/2/2014	Priority	High

Actor:

Staffs

Summary:

This use case allows staffs to delete a request from manager page.

Goal:

To reject a request because it's may be not valid or in a period time, don't have any response from customers.

Triggers:

The staffs click on button “Xóa” on the staff pages.

Preconditions:

The staff page is being displayed.

Post Conditions:

On success: The request will be removed.

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Click on “Xóa” button on staff pages	
2.		Popup will show and the content is “Bạn có muốn huỷ bỏ yêu cầu này?”
3.	Click on “Yes” button	
4.		Pages will reload and request has been removed

Alternative Scenario:

N/A

Exceptions:

N/A

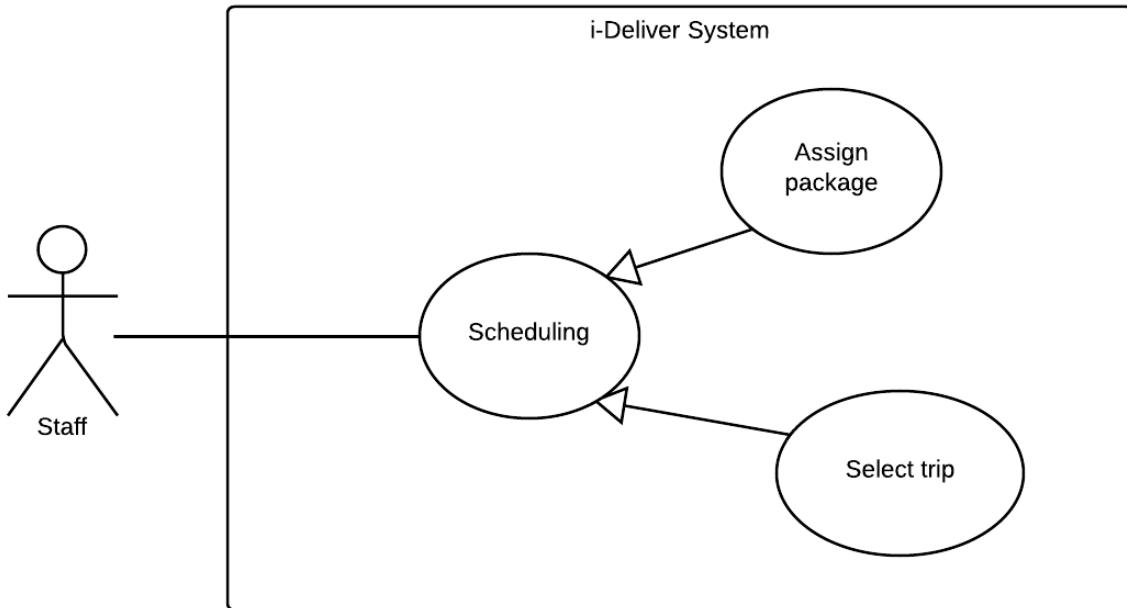
Relationships:

“Check room prices and availability” use case

Business Rules:

No	Business Rule Description
1.	User must be login to system with role Staff
2.	Only staff at this station can do this action

3.2.2.28. « Staff » Assign package



– SPECIFICATION

Use-case No.	UC022	Use-case Version	1.0
Use-case Name	Assign package		
Author	Lê Phúc Lữ		
Date	10/02/2014	Priority	High

Actor:

Staffs

Summary:

This use case allows staffs assign a request to which car, route...follow the content of request, and request will be become a package.

Goal:

To transport package according to customer requirements

Triggers:

N/A

Preconditions:

The request must be approved.

Post Conditions:

On success: Package will be assigned.

On failure: An error message is displayed. The content of the error message will be based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Click “Xếp lịch vận chuyển” link button in menu bar.	
2.		Display list of request have been paid on screen.
3.	Staff can choose one or more requests to assign and click “Xếp	

	lịch”.	
4.		Redirect to list of request has been assigned

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	If request not assign for any trip	
	2.		Popup will show with the content is reason why the request not assign include: volume, time, trip is not available....

Exceptions:

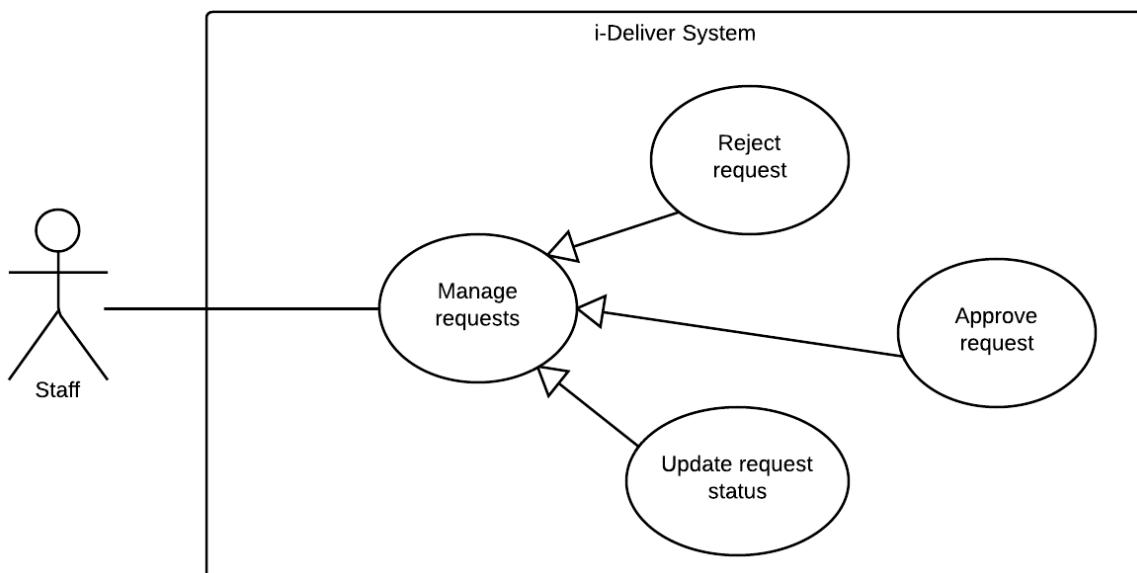
No	User Action	System Response
1.	User not choose any request but click on “Xếp lịch” button	
2.		Redirect to errors pages

Relationships:

Business Rules:

No	Business Rule Description
1.	User must be login to system with role is staff
2.	Only the staff at station can be assign

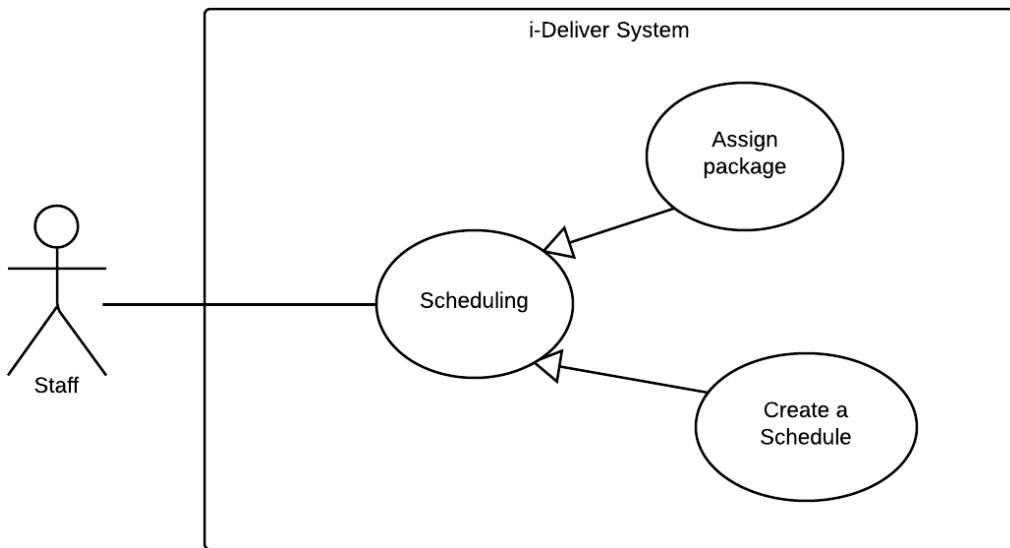
3.2.2.29. « Staff » Update package(s) status



– SPECIFICATION			
Use-case No.	UC023	Use-case Version	1.0
Use-case Name	Update request status		
Author	Nguyễn Thanh Tùng		

Date	10/02/2014	Priority	High																																				
Actor:																																							
Staffs																																							
Summary:																																							
The use case allows staffs update status of a package when it's assigned.																																							
Goal:																																							
To notifies to customers about their package through the status of package: on way or waiting.., help their tracking package.																																							
Triggers:																																							
Staffs click on button “Xác nhận chuyến đi” or “Xác nhận đến trạm” or “Huỷ hàng” or “Trả hàng” on staff’s page.																																							
Preconditions:																																							
.Package must be assigned.																																							
Post Conditions:																																							
<i>On success:</i> Status of package will be changed.																																							
<i>On failure:</i> An error message is displayed. The content of the error message will be based on the type of failure.																																							
Main Success Scenario:																																							
<table border="1"> <thead> <tr> <th>Step</th><th>User Action</th><th colspan="2">System Response</th></tr> </thead> <tbody> <tr> <td>1.</td><td>Click on button “Xác nhận chuyến đi” on trip pages</td><td colspan="2"></td></tr> <tr> <td>2.</td><td></td><td colspan="2">All request in trip will be change status to “Đã chuyến đi”</td></tr> <tr> <td>3.</td><td>Click on button “Xác nhận đến trạm”</td><td colspan="2"></td></tr> <tr> <td>4.</td><td></td><td colspan="2">All request has stop station same with station of staff will be change status to “Đã đến đích”</td></tr> <tr> <td>5.</td><td>Click on button “Huỷ hàng”</td><td colspan="2"></td></tr> <tr> <td>6.</td><td></td><td colspan="2">Request will be change status to “Đã huỷ”</td></tr> <tr> <td>7.</td><td>Click on button “Trả hàng”</td><td colspan="2"></td></tr> <tr> <td>8.</td><td></td><td colspan="2">Request will be change status to “Chờ xếp lịch”</td></tr> </tbody> </table>				Step	User Action	System Response		1.	Click on button “Xác nhận chuyến đi” on trip pages			2.		All request in trip will be change status to “Đã chuyến đi”		3.	Click on button “Xác nhận đến trạm”			4.		All request has stop station same with station of staff will be change status to “Đã đến đích”		5.	Click on button “Huỷ hàng”			6.		Request will be change status to “Đã huỷ”		7.	Click on button “Trả hàng”			8.		Request will be change status to “Chờ xếp lịch”	
Step	User Action	System Response																																					
1.	Click on button “Xác nhận chuyến đi” on trip pages																																						
2.		All request in trip will be change status to “Đã chuyến đi”																																					
3.	Click on button “Xác nhận đến trạm”																																						
4.		All request has stop station same with station of staff will be change status to “Đã đến đích”																																					
5.	Click on button “Huỷ hàng”																																						
6.		Request will be change status to “Đã huỷ”																																					
7.	Click on button “Trả hàng”																																						
8.		Request will be change status to “Chờ xếp lịch”																																					
Alternative Scenario:																																							
N/A																																							
Exceptions:																																							
<table border="1"> <thead> <tr> <th>No</th><th>User Action</th><th colspan="2">System Response</th></tr> </thead> <tbody> <tr> <td>1.</td><td>If don't have any request available, user click on button</td><td colspan="2">Redirect to errors pages.</td></tr> </tbody> </table>				No	User Action	System Response		1.	If don't have any request available, user click on button	Redirect to errors pages.																													
No	User Action	System Response																																					
1.	If don't have any request available, user click on button	Redirect to errors pages.																																					
Relationships:																																							
Business Rules:																																							
<table border="1"> <thead> <tr> <th>No</th><th>Business Rule Description</th></tr> </thead> <tbody> <tr> <td>1.</td><td>User must be login to system with role staff</td></tr> <tr> <td>2.</td><td>Only staff at station can be do this action</td></tr> </tbody> </table>				No	Business Rule Description	1.	User must be login to system with role staff	2.	Only staff at station can be do this action																														
No	Business Rule Description																																						
1.	User must be login to system with role staff																																						
2.	Only staff at station can be do this action																																						

3.2.2.30. « Staff » Create a schedule



CREATE A SCHEDULE – SPECIFICATION

Use-case No.	UC025	Use-case Version	1.0
Use-case Name	Create new schedule		
Author	Đào Bảo Long		
Date	10/02/2014	Priority	Normal

Actor:

Staff

Summary:

This use case allows a staff to create a schedule for specify coach and route.

Goal:

To manually define a new schedule in the i-Deliver system.

Triggers:

The user clicks on the button “Tạo lịch chạy” on the schedule management page.

Preconditions:

The user is already logged in with an account of Staff role.

Post Conditions:

On success: A new schedule was added to system's database. The user is redirected to schedule list page which is now has a section that describes the currently added schedule.

On failure: An error message is displayed. Please refer to the Exception Scenario section the see what the content of the error message is.

Main Success Scenario:

Step	User Action	System Response
1.	Clicks on the button “Tạo lịch chạy”.	
2.	Fills out the form with relevant values. [See Exception No. 1, 2, 3, 4]	

3.	Clicks the button “Tạo mới” [See Alternative No. 1, 2, 3]	
4.		Saves the schedule to the database and redirects the user to the schedule list page which now has a new section that describes the newly added schedule.

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Hủy”.	
	2.		Shows a popup to ask the user to confirm the action.
2.	1.	The user chooses “OK”.	
	2.		If the user confirms to go back, redirects the user to the list page, otherwise, just hide the popup.
3.	1.	Clicks on the button “Tạo mới”.	
	2.		Saves the schedule to the database and redirects the user to the schedule list page.

Exceptions:

No	User Action	System Response
1.	Does not select an item in the select list “Tuyến”.	Shows the message “Vui lòng chọn tuyến” next to the field.
2.	Does not select an item in the select list “Xe”.	Shows the message “Vui lòng chọn xe” next to the field.
3.	Leaves the field “Thời gian đi dự kiến” empty.	Shows the message “Vui lòng nhập thời gian” next to the field.
4.	Leaves the field “Thời gian đến dự kiến” empty.	Shows the message “Vui lòng nhập thời gian” next to the field.

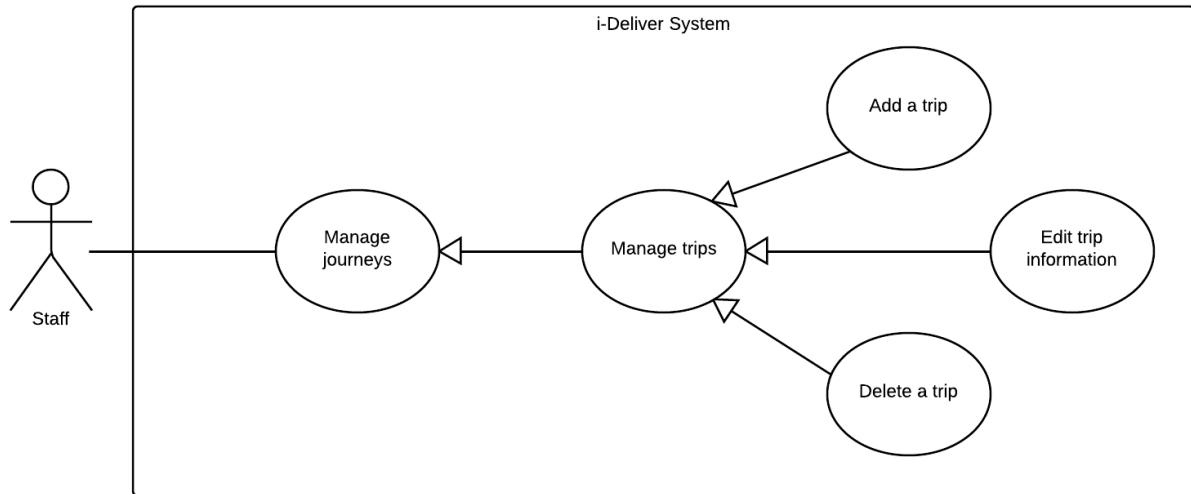
Relationships:

N/A

Business Rules:

No	Business Rule Description
1.	A new schedule can be created for a route if only that route had specified coaches run on.

3.2.2.31. « Staff » Create a trip



ADD A TRIP – SPECIFICATION

Use-case No.	UC026	Use-case Version	1.0
Use-case Name	Create new trip		
Author	Đào Bảo Long		
Date	10/02/2014	Priority	Normal

Actor:

Staff

Summary:

This use case allows a staff to create trips for a route based on defined schedule.

Goal:

To manually open a new trip in the i-Deliver system

Triggers:

The user clicks on the button “Tạo một chuyến mới” on the trip management page.

Preconditions:

The user is already logged in with an account of Staff role.

Post Conditions:

On success: A new trip was added to system's database. The user is redirected to trip list page which is now has a section that describes the currently added trip.

On failure: An error message is displayed. Please refer to the Exception Scenario section to see what the content of the error message is.

Main Success Scenario:

Step	User Action	System Response
1.	Clicks on the button “Tạo một chuyến mới”.	
2.		Browser popup a dialog with create form
3.	Fills out the form with relevant values. [See Exception No. 1, 2]	
4.	Clicks the button “Tạo mới” [See Alternative No. 1, 2, 3]	

5.		Saves the trip to the database and redirects the user to the trip list page which now has a new section that describes the newly added trip.
----	--	--

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Hủy”.	
	2.		Shows a popup to ask the user to confirm the action.
2.	1.	The user chooses “OK”.	
	2.		If the user confirms to go back, redirects the user to the list page, otherwise, just hide the popup.
3.	1.	Clicks on the button “Tạo mới”.	
	2.		Saves the trip to the database and redirects the user to the trip list page.

Exceptions:

No	User Action	System Response
1.	Does not select a date in the date picker “Ngày chạy”.	Shows the message “Vui lòng chọn ngày” next to the field.
2.	Does not select an item in the select list “Tuyến”.	Shows the message “Vui lòng chọn tuyến” next to the field.

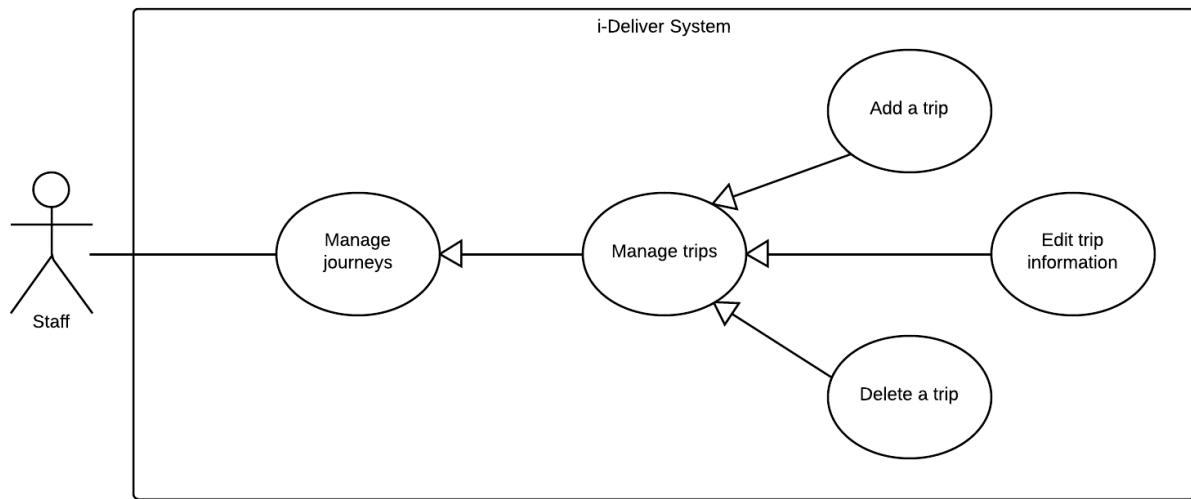
Relationships:

N/A

Business Rules:

No	Business Rule Description
1.	A new trip can be created for a route if only that route had defined schedule.
2.	Before create a new trip, user must select a date to review all trip created in that day in order to avoid duplicate.
3.	Once time user pick a date to create new trip, the i-Deliver system will create trip for 7 days continuous automatically

3.2.2.32. « Staff » Delete a trip



DELETE A TRIP – SPECIFICATION

Use-case No.	UC027	Use-case Version	1.0
Use-case Name	Delete a trip		
Author	Đào Bảo Long		
Date	10/02/2014	Priority	Normal

Actor:

Staff

Summary:

This use case allows a staff to remove an existing trip from the system.

Goal:

To manually remove an existing trip from the system.

Triggers:

The staff clicks on the button “Xóa chuyến” on the section that contains the trip to be deleted in the trip management page.

Preconditions:

The user is already logged in with an account of Staff role.

The trip management page is being displayed.

The trip to be deleted is being selected.

Post Conditions:

On success: The selected trip is deleted from the system database. The row that contains the selected trip is removed from the trip list page.

On failure: An error message is displayed. The content of the error message will be specified in the Exception Scenario section based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Clicks on the button “Xóa” on the row that contains the trip to be deleted.	
2.		Shows a popup to ask the staff to confirm the action.
3.	Chooses “OK” to continue to	Marks the selected trip as deleted in

	delete the trip. [See Alternative No. 1]	the database. Removes the row that contains the selected trip from the trip list page.
--	---	---

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Hủy bỏ”.	
	2.		Hides the popup message box.

Exceptions:

N/A

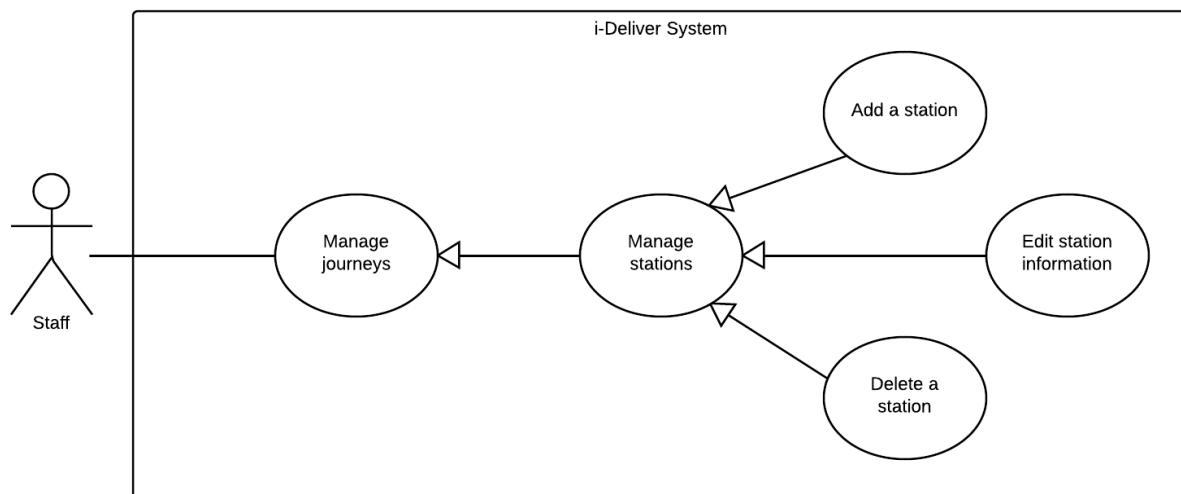
Relationships:

“Add a trip” use case

Business Rules:

No	Business Rule Description
1.	Staff will be notified by requests list which assigned to selected trip before delete that trip.
2.	All of requests has assigned to selected trip will be re-assigned

3.2.2.33. « Staff » Add a station



ADD A STATION – SPECIFICATION

Use-case No.	UC028	Use-case Version	1.0
Use-case Name	Add new station		
Author	Nguyễn Thanh Tùng		
Date	10/02/2014	Priority	Normal
Actor: <i>Staff</i>			
Summary: ...			

Goal:

To manually open a new station in the i-Deliver system

Triggers:

The user clicks on the button “Thêm một trạm mới” on the station management page.

Preconditions:

The user is already logged in with an account of Staff role.

Post Conditions:

On success: A new station was added to system's database. The user is redirected to trip station page which is now has a section that describes the currently added station.

On failure: An error message is displayed. Please refer to the Exception Scenario section to see what the content of the error message is.

Main Success Scenario:

Step	User Action	System Response
1.	Clicks on the button “Tạo trạm chuyên mới”.	
2.	Fills out the form with relevant values.	
3.	Clicks the button “Tạo mới”. [See Alternative No. 1, 2, 3]	
4.		Saves the stations to the database and redirects the user to the station list page which now has a new section that describes the newly added station.

Alternative Scenario:

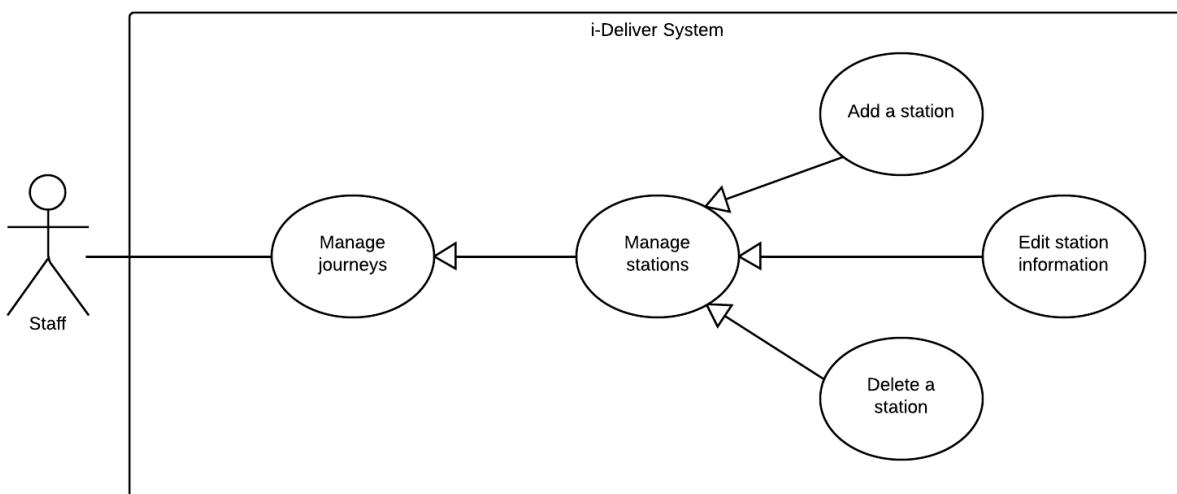
No	Step	User Action	System Response
1.	1.	Clicks on the button “Tạo mới”.	
	2.		Shows a popup to ask the user to confirm the action.
2.	1.	The user chooses “OK”.	
	2.		If the user confirms to go back, redirects the user to the list page, otherwise, just hide the popup.
3.	1.	Clicks on the button “Tạo mới”.	
	2.		Saves the station to the database and redirects the user to the station list page.

Exceptions:

No	User Action	System Response
1.	Leaves the field “Tên trạm” empty.	Shows the message “Vui lòng nhập tên trạm” next to the field.
2.	Leaves the field “Địa chỉ” empty.	Shows the message “Vui lòng nhập địa chỉ trạm” next to the field.
3.	Does not select an item in the select list “Tỉnh thành”.	Shows the message “Vui lòng chọn tỉnh, thành phố” next to the select

		list.
4.	Fills the field “Số điện thoại” with a value that cannot be recognized as a phone number.	Shows the message “Vui lòng nhập số điện thoại hợp lệ” next to the field.
5.	Leaves the field “Địa chỉ” empty.	Shows the message “Vui lòng nhập địa chỉ” next to the field.
Relationships:		
N/A		
Business Rules:		
N/A		

3.2.2.34. « Staff » Edit station information



EDIT STATION INFORMATION – SPECIFICATION			
Use-case No.	UC029	Use-case Version	1.0
Use-case Name	Edit station information		
Author	Nguyễn Thanh Tùng		
Date	10/02/2014	Priority	Normal
Actor:			
<i>Staff</i>			
Summary:			
This use case allows a staff to view and manually change information about an existing station in the database.			
Goal:			
To manually edit information of a station in the system.			
Triggers:			
The user clicks on the button “Thông tin trạm” on station management page.			
Preconditions:			
The user is already logged in with an account of Staff role.			
The station management page is being displayed.			
Post Conditions:			

On success: The information of selected station is updated in the system. The user is redirect to the station management page where information is currently updated.

On failure: An error message is displayed. The content of the error message will be specified in the Exception Scenario section based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Clicks on the button “Chinh sửa thông tin trạm”.	
2.	Fills out the form with relevant values.	
3.	Clicks the button “Lưu thay đổi”. [See Alternative No. 1, 2, 3]	
4.		Saves new station information to the database and redirects the staff to the station list page.

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Quay về trang quản lý”.	
	2.		Shows a popup to ask the user to confirm the action.
2.	1.	The user chooses “OK”.	
	2.		If the user confirms to go back, redirects the user to the list page, otherwise, just hide the popup.
3.	1.	Clicks on the button “Lưu thay đổi”.	
	2.		Saves new station information to the database and redirects the staff to the station list page.

Exceptions:

N/A

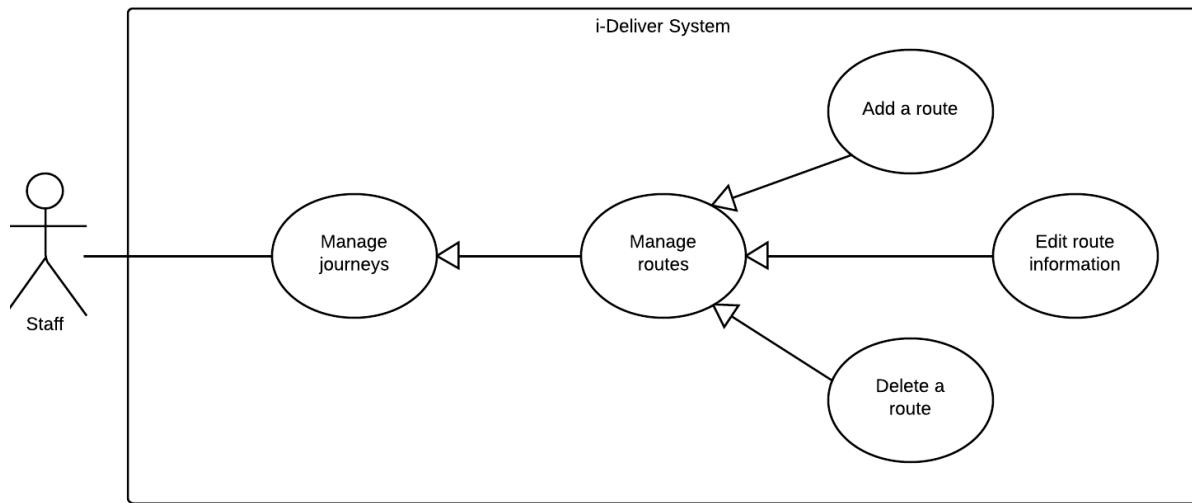
Relationships:

“Add a station” use case

Business Rules:

N/A

3.2.2.35. « Staff » Add a route



ADD A ROUTE – SPECIFICATION

Use-case No.	UC030	Use-case Version	1.0
Use-case Name	Add new route		
Author	Nguyễn Tân Đức		
Date	10/02/2014	Priority	Normal

Actor:

Staff

Summary:

The new route will be created in i-Deliver system. This route will be put into operation for passengers and delivery goods.

Goal:

To manually open a new route in the i-Deliver system

Triggers:

The user clicks on the button “Thêm một tuyến mới” on the routes management page.

Preconditions:

The user is already logged in with an account of Staff role.

Post Conditions:

On success: A new route was added to system's database. The user is redirected to route station page which is now has a section that describes the currently added station.

On failure: An error message is displayed. Please refer to the Exception Scenario section to see what the content of the error message is.

Main Success Scenario:

Step	User Action	System Response
1.	Clicks on the button “Tạo một tuyến mới”.	
2.	Fills out the form with relevant values.	
3.	Clicks the button “Tạo mới”. [See Alternative No. 1, 2, 3]	
4.		Saves the routes to the database and redirects the user to the route list page

		which now has a new section that describes the newly added route.
--	--	---

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Quay về trang quản lý”.	
	2.		Shows a popup to ask the user to confirm the action.
2.	1.	The user chooses “OK”.	
	2.		If the user confirms to go back, redirects the user to the list page, otherwise, just hide the popup.
3.	1.	Clicks on the button “Tạo mới”.	
	2.		Saves the route to the database and redirects the user to the route list page.

Exceptions:

No	User Action	System Response
1.	Not enough enter required fields.	Display message error: "Điền vào những ô yêu cầu"
2.		

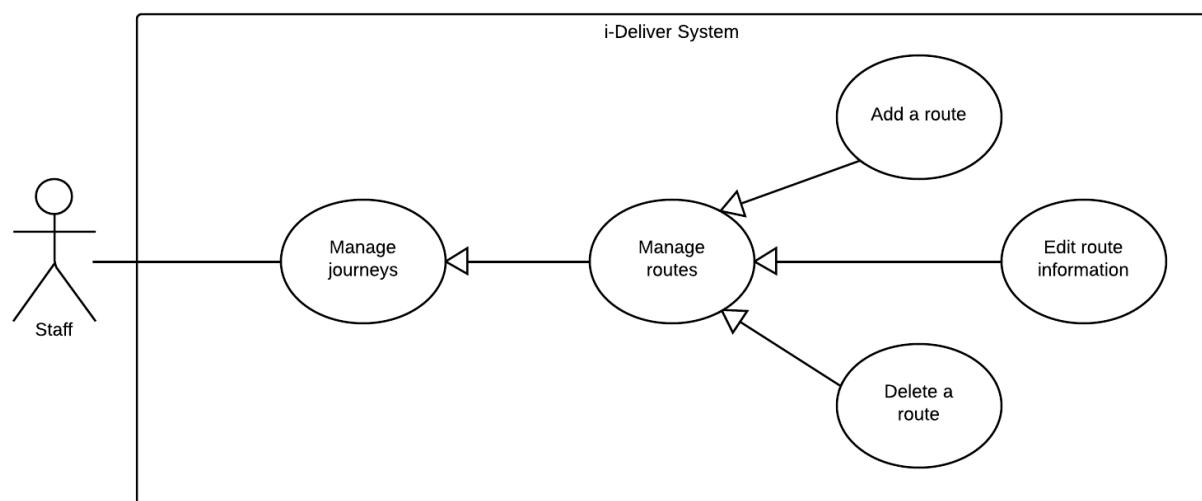
Relationships:

N/A

Business Rules:

No	Business Rule Description
1.	User must login into the system with staff role.
2.	This route is active when added to database.

3.2.2.36. « Staff » Delete a route



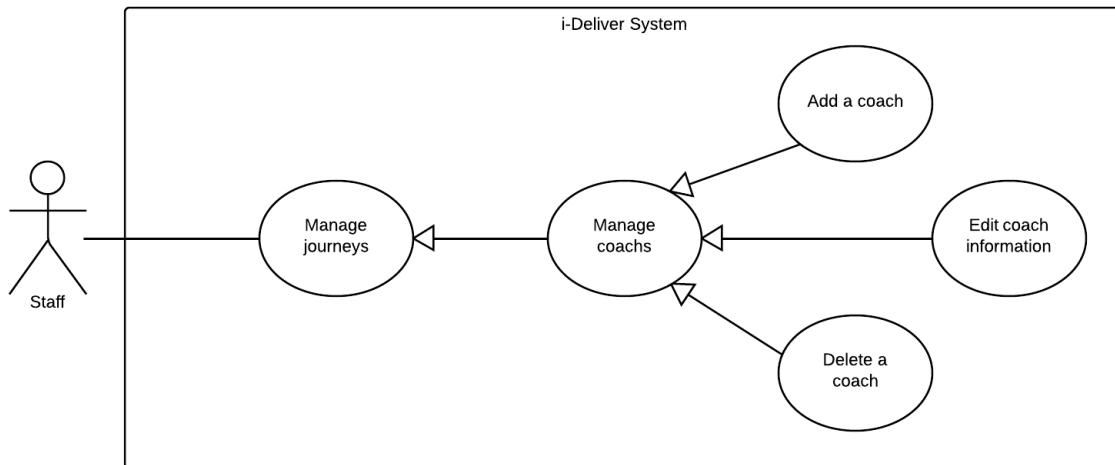
DELETE A ROUTE – SPECIFICATION															
Use-case No.	UC032	Use-case Version	1.0												
Use-case Name	Delete a route														
Author	Nguyễn Tân Đức														
Date	10/02/2014	Priority	Normal												
Actor:	<i>Staff</i>														
Summary:	This use case allows a staff to remove an existing route from the system.														
Goal:	To manually remove an existing route from the system.														
Triggers:	The staff clicks on the button “Xóa tuyến” on the section that contains the route to be deleted in the route management page.														
Preconditions:	<p>The user is already logged in with an account of Staff role.</p> <p>The route management page is being displayed.</p> <p>The route to be deleted is being selected.</p>														
Post Conditions:	<p><i>On success:</i> The selected route is deleted from the system database. The row that contains the selected route is removed from the route list page.</p> <p><i>On failure:</i> An error message is displayed. The content of the error message will be specified in the Exception Scenario section based on the type of failure.</p>														
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>User Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1.</td><td>Clicks on the button “Xóa” on the row that contains the trip to be deleted.</td><td></td></tr> <tr> <td>2.</td><td></td><td>[See Exception No. 1] Shows a popup to ask the staff to confirm the action.</td></tr> <tr> <td>3.</td><td>Chooses “OK” to continue to delete the route. [See Alternative No. 1]</td><td>Marks the selected route as deleted in the database. Removes the row that contains the selected route from the route page. [See Exception No. 2, 3]</td></tr> </tbody> </table>			Step	User Action	System Response	1.	Clicks on the button “Xóa” on the row that contains the trip to be deleted.		2.		[See Exception No. 1] Shows a popup to ask the staff to confirm the action.	3.	Chooses “OK” to continue to delete the route. [See Alternative No. 1]	Marks the selected route as deleted in the database. Removes the row that contains the selected route from the route page. [See Exception No. 2, 3]
Step	User Action	System Response													
1.	Clicks on the button “Xóa” on the row that contains the trip to be deleted.														
2.		[See Exception No. 1] Shows a popup to ask the staff to confirm the action.													
3.	Chooses “OK” to continue to delete the route. [See Alternative No. 1]	Marks the selected route as deleted in the database. Removes the row that contains the selected route from the route page. [See Exception No. 2, 3]													
Alternative Scenario:	<table border="1"> <thead> <tr> <th>No</th><th>Step</th><th>User Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td rowspan="2">1.</td><td>1.</td><td>Clicks on the button “Hủy bỏ”.</td><td></td></tr> <tr> <td>2.</td><td></td><td>Hides the popup message box.</td></tr> </tbody> </table>			No	Step	User Action	System Response	1.	1.	Clicks on the button “Hủy bỏ”.		2.		Hides the popup message box.	
No	Step	User Action	System Response												
1.	1.	Clicks on the button “Hủy bỏ”.													
	2.		Hides the popup message box.												
Exceptions:	N/A														
Relationships:															

Manage route

Business Rules:

No	Business Rule Description
1.	Packages assigned to selected coach must be update and re-assigned.
2.	

3.2.2.37. « Staff » Add new coach



ADD A COACH – SPECIFICATION

Use-case No.	UC033	Use-case Version	1.0
Use-case Name	Add new coach		
Author	Lê Phúc Lữ		
Date	10/02/2014	Priority	Normal

Actor:

Staff

Summary:

...

Goal:

To manually add a new coach to the i-Deliver system

Triggers:

The user clicks on the button “Thêm mới” on the coach management page.

Preconditions:

The user is already logged in with an account of Staff role.

Post Conditions:

On success: A new coach was added to system’s database. The user is redirected to coach station page which is now has a section that describes the currently added coach.

On failure: An error message is displayed. Please refer to the Exception Scenario section to see what the content of the error message is.

Main Success Scenario:

Step	User Action	System Response
1.	Clicks on the button “Thêm mới”	

	xe mới”.	
2.	Fills out the form with relevant values.	
3.	Clicks the button “Tạo mới”. [See Alternative No. 1, 2]	
4.		Saves the coach to the database and redirects the user to the coach list page which now has a new section that describes the newly added coach.

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on the button “Quay về trang quản lý”.	
	2.		Shows a popup to ask the user to confirm the action.
2.	1.	The user chooses “OK”.	
	2.		If the user confirms to go back, redirects the user to the list page, otherwise, just hide the popup.
3.	1.	Clicks on the button “Tạo mới”.	
	2.		Saves the coach to the database and redirects the user to the coach list page.

Exceptions:

N/A

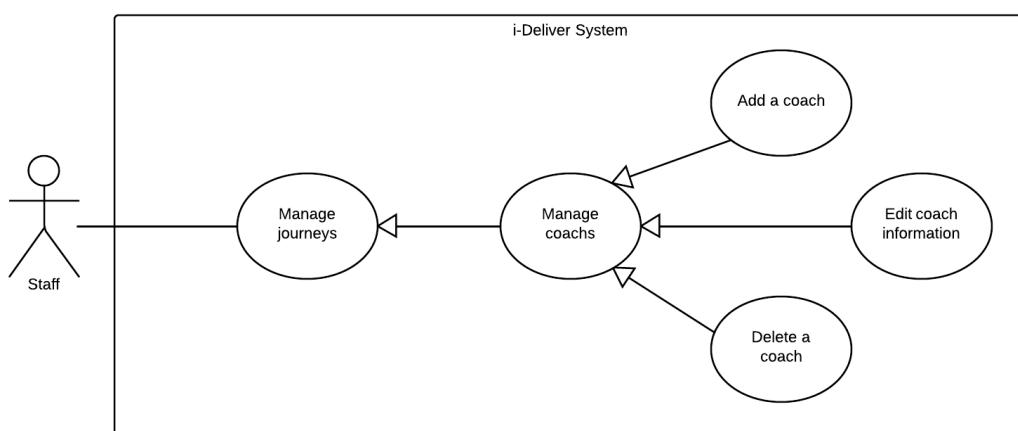
Relationships:

N/A

Business Rules:

N/A

3.2.2.38. « Staff » Delete a coach



DELETE A COACH – SPECIFICATION

Use-case No.	UC035	Use-case	1.0
--------------	-------	----------	-----

		Version	
Use-case Name	Delete a coach		
Author	Lê Phúc Lữ		
Date	10/02/2014	Priority	Normal

Actor:

Staff

Summary:

This use case allows a staff to remove an existing coach from the system.

Goal:

To manually remove an existing coach from the system.

Triggers:

The staff clicks on the button “Xóa xe này” on the section that contains the coach to be deleted in the coach management page.

Preconditions:

The user is already logged in with an account of Staff role.

The coach management page is being displayed.

The coach to be deleted is being selected.

Post Conditions:

On success: The selected coach is deleted from the system database. The row that contains the selected coach is removed from the coach list page.

On failure: An error message is displayed. The content of the error message will be specified in the Exception Scenario section based on the type of failure.

Main Success Scenario:

Step	User Action	System Response
1.	Clicks on the button “Xóa” on the row that contains the trip to be deleted.	
2.		Shows a popup to ask the staff to confirm the action.
3.	Chooses “OK” to continue to delete the coach. [See Alternative No. 1]	Marks the selected coach as deleted in the database. Removes the row that contains the selected coach from the coach list page.

Alternative Scenario:

No	Step	User Action	System Response
1.	1.	Clicks on button “Hủy bỏ”.	
	2.		Hides the popup message box.

Exceptions:

N/A

Relationships:

“Add a coach” use case

Business Rules:

No	Business Rule Description

- | | |
|----|---|
| 1. | Packages assigned to selected coach must be update and re-assigned. |
|----|---|

3.2.3. Non-Functional Requirements

3.2.2.1. Reliability

Information about journeys are provided by staffs in such a way that preserves the most accurate and up-to-date data.

3.2.2.2. Availability

N/A

3.2.2.3. Security

Authentication is required when the user utilizes certain functions in the system. Passwords must be encrypted so they are unrecognizable to even system admins and database admins. The system should also require authorization when the user tries to access an unauthorized function.

3.2.2.4. Maintainability

In the architecture design of the system, the responsibilities of all components should be clear and do no overlap each other. The communications with a web service provider should be packaged in a single module independent from the rest of the system. The module should support plug-and-play, so that, when the web service provider changes the structure of the web service results, or when the system must work with another web service provider, minimal changes to the rest of the system are required.

3.2.2.5. Portability

On the client side, the website should be accessible to anyone with a personal computer that connects to the Internet and has a browser.

3.2.2.6. Performance

N/A

3.3. Entity Relationship Diagram

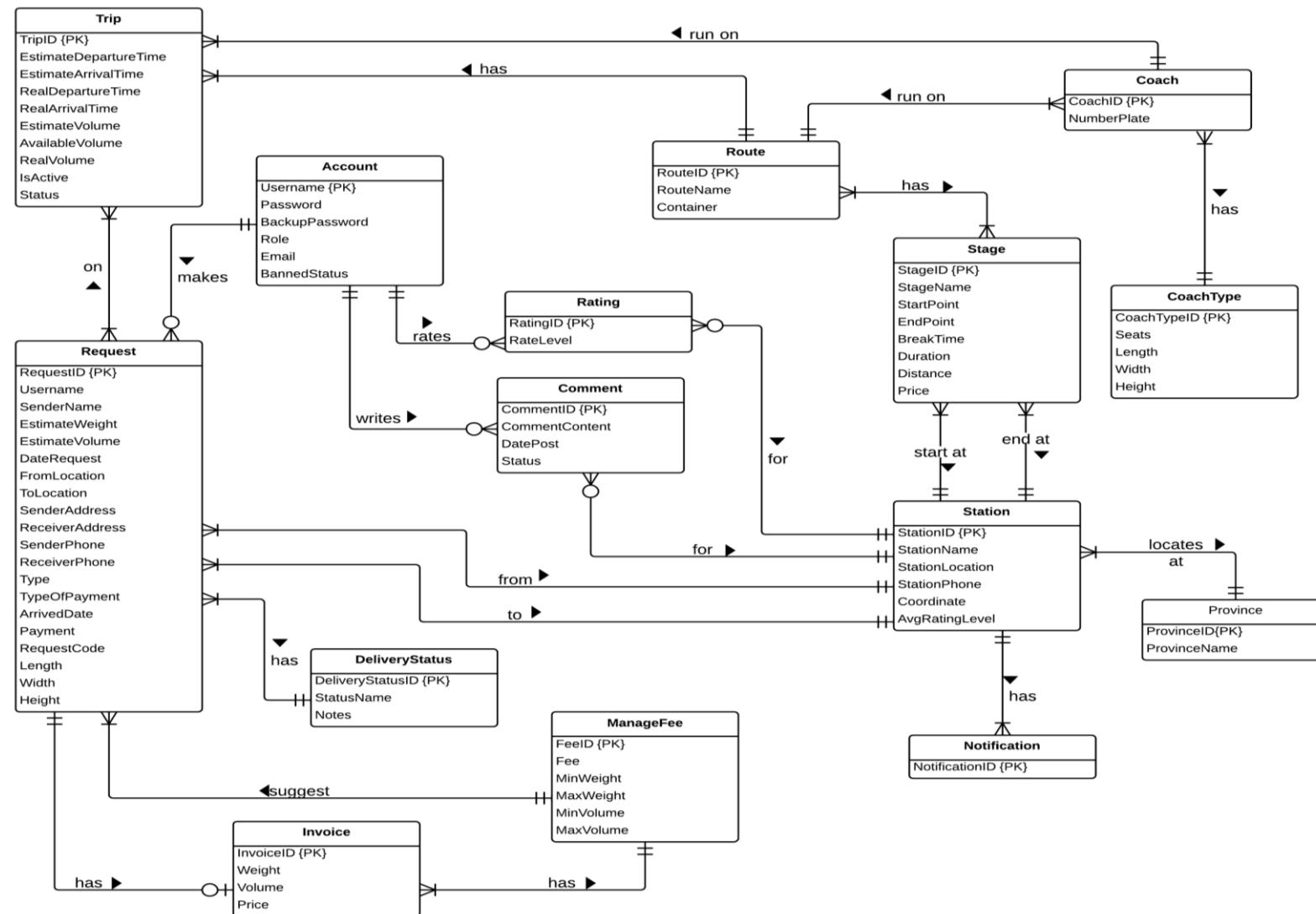


Diagram 2. Entity relationship diagram

4. Software Design Description (SDD)

4.1. 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.

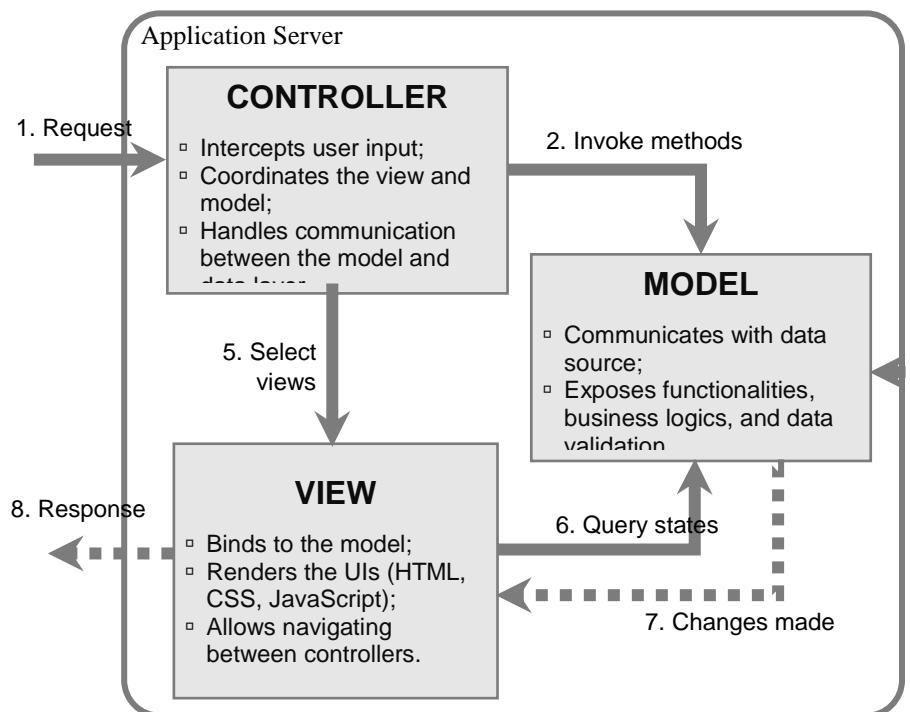
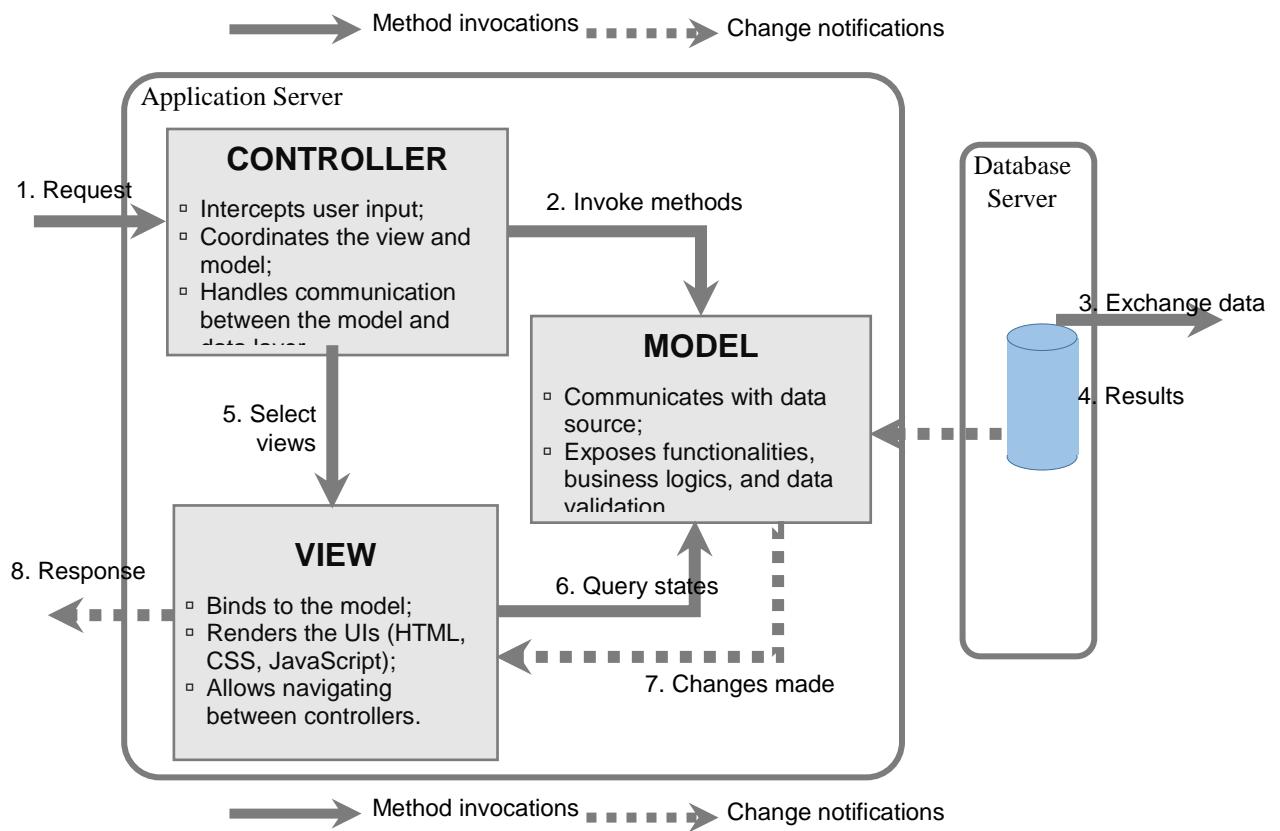
The detailed designs of the system should adopt the basic principles of software design, including “high cohesion, low coupling” and “open to extension, closed to modification” principles.

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.

4.2. 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.



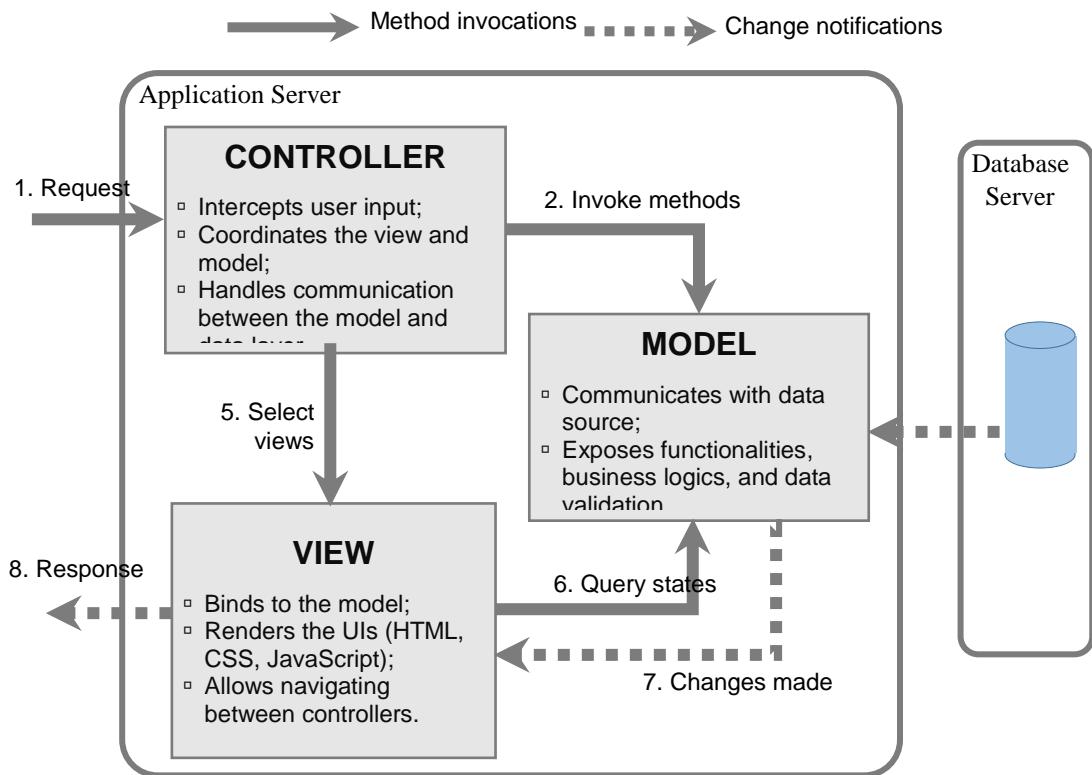


Figure 4. Model-View-Controller Architecture

4.3. Component Diagram

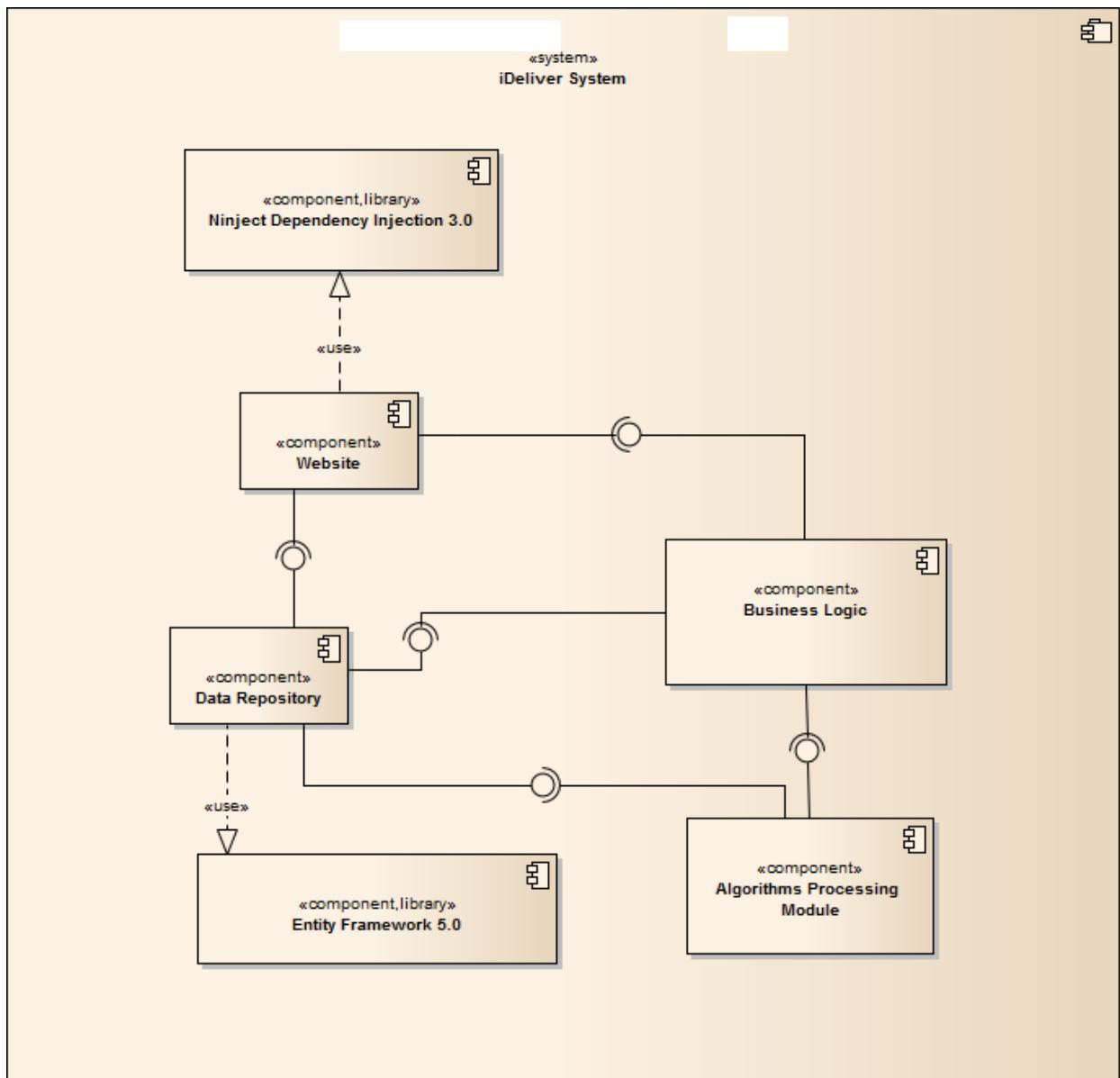


Diagram 3. Component Diagram

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 communicate 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 expose all the functionalities to end-users.

- **Algorithms Processing Module:** A module that uses Interface and DataRepository to execute the business of system by processing through some relative entities and find out the needed connections between them that satisfying the given constraint.

From the point of view of the MVC-III 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.

4.4. Detailed Description of Components

4.4.1. Entities Classes

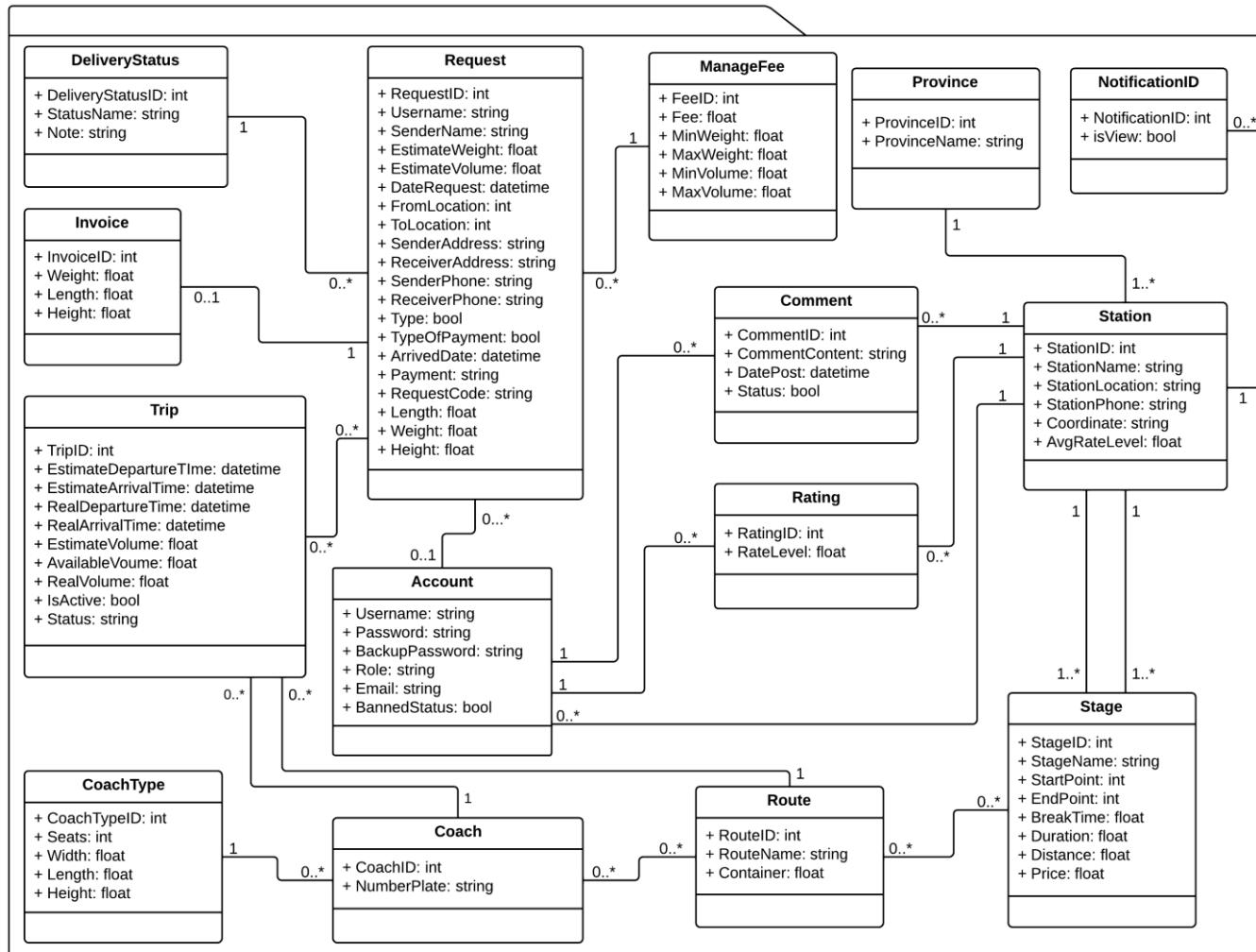


Diagram 4. Entities Class Diagram

4.4.2. Algorithm Processing Classes

4.4.2.1. Class diagram

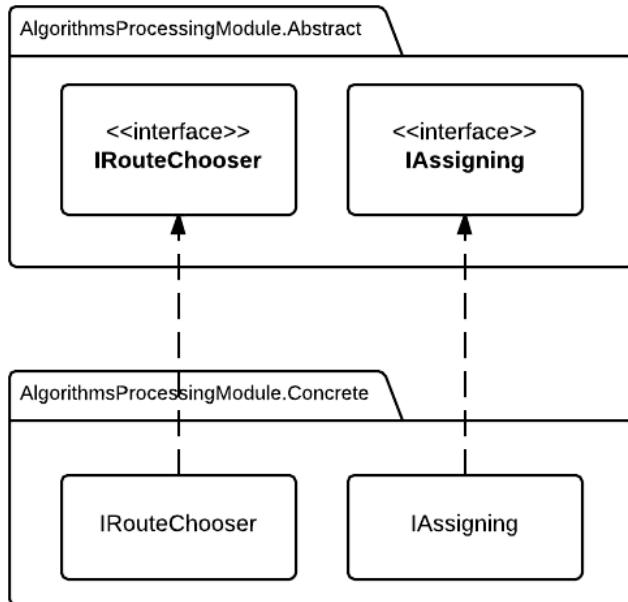


Diagram 5. Algorithm Processing Class Diagram

4.4.2.2. Algorithms description

4.4.2.2.1. Introduction

Scheduling problem is very popular in real life and appears in a lot of scopes, such as: education, human resource, transport, etc. For most of the scheduling problems, it has been shown that they are NP – complete problem, and they cannot be solved in polynomial time using a deterministic algorithm or find out the optimize solution within considered space in short time.

Our problem points to a type of this problem, which is scheduling the vehicles to deliver some package from a specific location to another location. The detail description of the problem:

At the transport center using coaches, they want to do an addition task for each coach as deliver the package that some customer sends to center along to each trip. Customers post request to the system to ask for transport the package with some specific information as:

- The time that user post request to the center.
- Volume of request (in dm³).
- Weight of request (in kilograms).
- Begin and end location of request.

The center has some resource as follow:

- Some stations with given address, exact coordinate with longitude and latitude.

- Some stages and each stage connect exactly two stations.
- Some routes and each route include one or multiple stages with the given order.
- Some coach with specific capacity (about volume can serve).
- Some schedule infers that which coach travels on which route and the coach start at what specific time of day.
- Some trip and each trip is an instant of schedule in a specific day.

The problem turns out is one type of scheduling problem: arrange resource for satisfying the given constraint. For detail, we want to assign each request for some trips such that:

- The total volume of requests assigned to trip not more than this coach' volume.
- The time of delivery for each request not more than 5 days and the time for deliver each request as min as possible.
- In the list of trips, the departure time of each trip not earlier than the time for right previous trip depart to the middle station and the delta time between two point times (arrival time of previous trip and depart time of current trip) is 45 minutes.
- The way of deliver is straight forward, not go backward.

4.4.2.2.2. *Detail solution*

We name the begin station of request is A and the end station of request is B .

(1) Step 1:

First, build an indirect and single graph $G = (V, E)$ in which:

- V is the set of stations of the system.
- E is the set of stage connects two stations.

List all data from database and sort the list of request base on 3 criterions:

- The time of user post request (like the principles: first come, first served of queue).
- The volume of request (follow the greedy rules: to gain the most, first try to process the biggest one).
- The weight of request (the higher weight of request, the more money system can get).

(2) Step 2:

Using Breath First Search algorithm to find some way connect A and B . The original of this algorithm is just finding one satisfied solution and that solution travels on the smallest number of edges of graph. We extend it a little bit to find not only one solution but also multiple solutions. The maximum number of solutions allowed is $k = 50$ (to avoid the loop in graph with thick graph situation that leads to the number can be up to $|E|!$).

In some cases, the direction of the partial of way is not always forward so we should reject the inappropriate way (the path in graph usually fallen into that case). To check the way not go backward, we use the mathematical idea as:

The point T with longitude and latitude are a, b respectively, we have the coordinate of its in space is $I(x, y, z)$ with $x = R \cos a \times \cos b, y = R \sin a \times \cos b, z = R \sin b$ and R is the radius of Earth (we can assume that Earth is a quite sphere).

The angle of two vectors $\overset{\text{u}}{m}(x_1, y_1, z_1), \overset{\text{l}}{n}(x_2, y_2, z_2)$ is

$$\arccos \frac{\overset{\text{u}}{m} \cdot \overset{\text{l}}{n}}{\left\| \overset{\text{u}}{m} \right\| \left\| \overset{\text{l}}{n} \right\|} = \arccos \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \sqrt{x_2^2 + y_2^2 + z_2^2}}.$$

with $\overset{\text{u}}{m} \cdot \overset{\text{l}}{n}$ is the dot product of two vectors and we can find this value easily by the found coordinate as above.

Assume that we have the list of stages:

$$A_1 A_2, A_2 A_3, A_3 A_4, \dots, A_{n-1} A_n$$

with n is the number of stages. So we iterate through the list of stages to check whether two consecutive stages form an obtuse angle with value bigger than the allow angle (in this case, we choose $a = 135^\circ$ as the upper value). If all pairs of consecutive stages are satisfy the condition, we accept this way, otherwise, we reject it and find another way.

(3) Step 3:

For each found way, try to perform 3 greedy ideas as following:

- Check one trip:
 - ✓ If the is such route, find list of trip travel on that route: this can be obtained by iterate through the list of route.
 - ✓ If this trip satisfies the condition, take it as the good solution immediately (because it is easy to deliver and manage a package when it is only in one trip than multiple trips).
- Check multiple trip:
 - ✓ Check to find some routes connect the consecutive stages of each list of found way list and the package hasn't been get off the coach at the middle way, this can be obtained by brute-force way as following:
 - ⊕ Cut the list of stages into 2 parts and find route for each part. If all well, save the current solution to list.
 - ⊕ Cut the list of stages into 3 parts and perform action the same as above.

 Cut the list of stages into 4 parts and perform action the same as above.

- ✓ For each list of found route, find trip and check condition to find some solution and add them to the list.
- ✓ Sort the solution bases on 2 criteria: the duration and the distance. Then choose the first of list mean the best solution for this case and stop the finding method.
- Check middle trips:
 - ✓ Check to find some trips travel on some routes and the package can be gets on and gets off each trip at the middle of its journey, this can be done by the brute-force way as above that try to cut the list of into 1, 2, 3 or 4 (not consider more) parts and find the route.
 - ✓ For each found list, we check the time that previous trip will comes to the break station that earlier than the next trip 45 minutes or more. If all are well, save the current list as one of solution in this case.
 - ✓ Sort the solution bases on 2 criteria: the duration and the distance. Then choose the first of list mean the best solution for this case and stop the finding method.

Notes that the condition including some criterions:

- The total request assigned to trip not more than available volume of trip.
- If the list of trip contains more than one trip, the departure time of each trip is 45 minutes later than the arrival time of previous trip on the list.

4.4.2.2.3. Pros and cons of algorithm

Pros	Cons
Use greedy method for almost steps of algorithm so it is fast to find out the solution and meet the local optimize of the given optimization problem.	The step of algorithm are separately performed so in some case, it cannot meet the absolute optimize solution (it occurs in some tricky or special case).
Some list of needed data listed for the first request also can be used for next requests.	Data list all from database so when the system works in a long time, the amount of item can be very large so reducing the time of action.
It is easy to see the correctness of this solution and because all condition are check step by step so if a solution is found, it also meet all constraint.	Some steps of algorithm can be implemented better to reduce the complexity but still not have done.

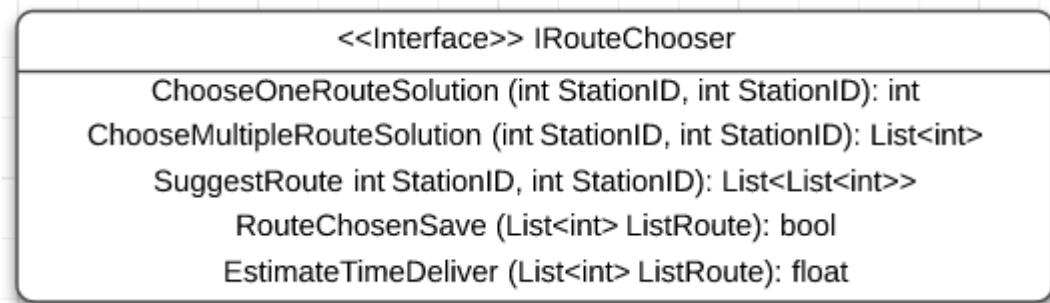
The complexity of algorithm is $O((k|V||E|^2 + |E|^4)|T||R||S||I|)$ with $|T|, |R|, |S|, |I|$ are the number of trip, route, stage and requests, respectively.

4.4.2.2.4. References

[1] <http://www.codeproject.com/Questions/164691/Retrieve-all-simple-paths-in-a-graph>

[2] <http://math.stackexchange.com/questions/441182/how-to-check-if-three-coordinates-form-a-line>

4.4.2.3. IRouteChooser



Properties: N/A

Operations

Signature: int ChooseOneRouteSolution(int StartStation, int EndStation)

Description: return one suitable route to deliver from the start station to the end station, if the route not exists, return -1.

Parameter	Type	Description
StartStation	int	The ID of the start station of the request
EndStation	int	The ID of the end station of the request

Signature: List <int> ChooseMultipleRouteSolution(int StartStation, int EndStation)

Description: return the list of suitable route to deliver from the start station to the end station.

Parameter	Type	Description
StartStation	int	The ID of the start station of the request
EndStation	int	The ID of the end station of the request

Signature: List<int> SuggestRoute(int StartStation, int EndStation)

Description: Suggest for staff the best suitable route or list of routes base on the input start and end station of request.

Parameter	Type	Description
StartStation	int	The ID of the start station of the request
EndStation	int	The ID of the end station of the request

Signature: bool RouteChosenSave (List<int> RouteList)

Description: Retrieves the list of route and save them to the table with exactly indicate order.

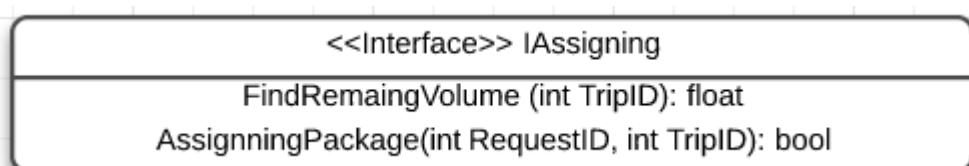
Parameter	Type	Description
RouteList	List<int>	List of route that staff choose or system suggest

Signature: float EstimateTimeDeliver (List<int> RouteList)

Description: Find the estimate time to deliver the package base on the input route list and their relative information.

Parameter	Type	Description
RouteList	List<int>	List of route that staff choose or system suggest

4.4.2.4. IAssigning



Properties: N/A

Operations

Signature: float FindRemaingVolume (int TripID)

Description: Return the remaining volume that the coach on input trip can be serving.

Parameter	Type	Description
TripID	int	The ID of trip that staff wants to estimate.

Signature: List <int> AssignningPackage (int RequestID)

Description: Automatically find the list of trip suitable for deliver the package.

Parameter	Type	Description
RequestID	int	The ID of request that staff processing.

4.4.3. Data Repository Classes

4.4.3.1. Class diagram

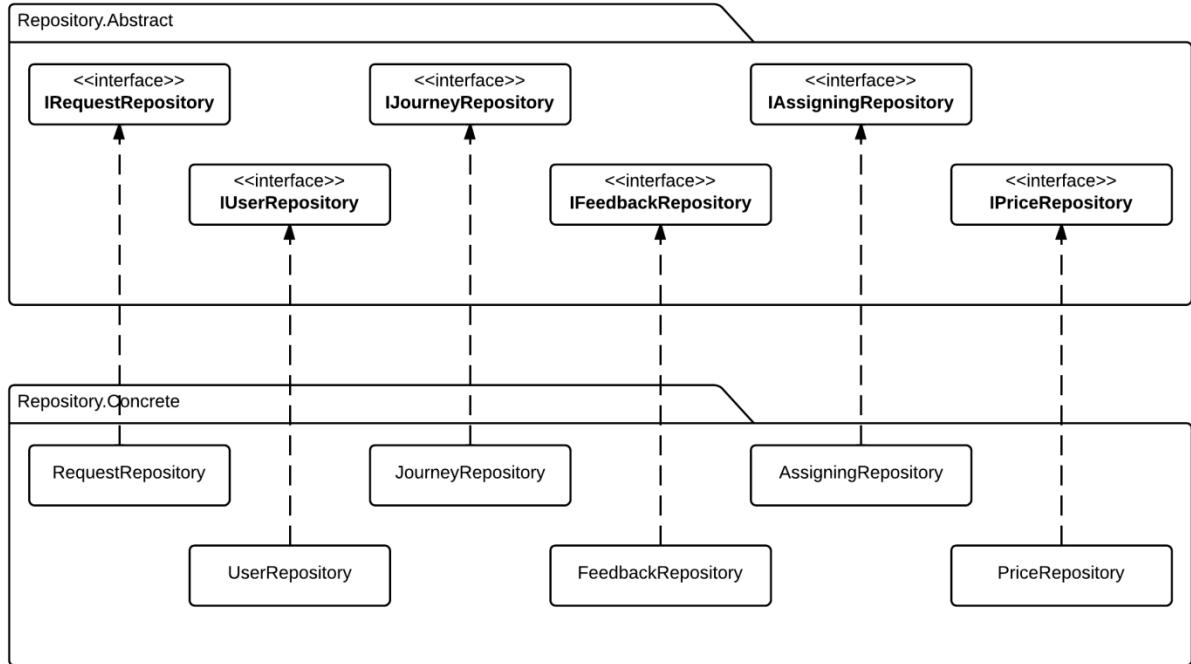
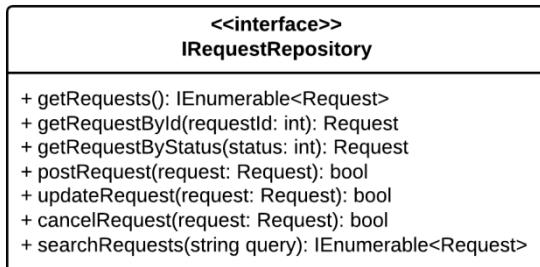


Diagram 6. Data Repository Class Diagram

4.4.3.2. IRequestRepository Interface



Properties: N/A

Operations:

Signature: IEnumerable<Request> Get Requests()
Description: Retrieves all the requests from the database

N/A	N/A	N/A
-----	-----	-----

Signature: Request GetRequestById(int requestId)
Description: Retrieves a request from the database given its id

Parameter	Type	Description
requestId	int	Id of the request to be retrieved

Signature: IEnumerable<Request> GetRequestByStatus(int status)
Description: Retrieves all the request from the database given its status

Parameter	Type	Description
Status	Int	Status of the request to be retrieved

Signature: bool postRequest(Request request)
Description: post new delivery request to system and save to database

Parameter	Type	Description
<i>request</i>	Request	New request to be saved

Signature: bool updateRequest(int requestId)

Description: Update existing request to the database given its id

Parameter	Type	Description
<i>requestId</i>	int	Id of the request to be updated

Signature: bool cancelRequest(int requestId)

Description: Cancel/Remove existing request from the database given its id

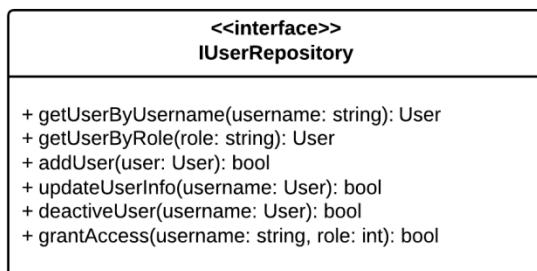
Parameter	Type	Description
<i>requestId</i>	int	Id of the request to be updated

Signature: IEnumerable<Request> searchRequest(string query)

Description: Retrieves all the request from the database given by query string

Parameter	Type	Description
<i>query</i>	String	Query string to find matching request

4.4.3.3. IUserRepository Interface



Properties: N/A

Signature: User GetUserByUsername(String username)

Description: Retrieves a user given its username from the database

Parameter	Type	Description
<i>username</i>	String	Username of user to be retrieved

Signature: User GetUserByRole(String role)

Description: Retrieves a user given its role from the database

Parameter	Type	Description
<i>role</i>	String	Role of user to be retrieved

Signature: bool addUser(User user)

Description: Add new user to the database

Parameter	Type	Description
<i>user</i>	UserProfile	The user to be saved

Signature: bool updateUser(int username)

Description: Update/Save user information to the database

Parameter	Type	Description
<i>username</i>	String	Username of user to be updated

Signature: bool GrantAccess(String username, int roleId)

Description: Grants the selected role the user with the user id		
Parameter	Type	Description
<i>username</i>	String	Username of the user to be granted access
<i>roleId</i>	Role	Role that will be granted to the user

Signature: bool DeactivateUser(String username)		
Description: Marks the user as deactivated in the database		
Parameter	Type	Description
<i>username</i>	String	Username of the user to be marked as deactivated

4.4.3.4. IJourneyRepository Interface

<<interface>> IJourneyRepository	
+ getStations(): IEnumerable<Station>	
+ getStationById(stationId: int): Station	
+ addStation(station: Station): bool	
+ updateStationInfo(stationId: int): bool	
+ deleteStation(stationId: int): bool	
+ getTrips(): IEnumerable<Trip>	
+ getTripById(tripId: int): Trip	
+ addTrip(trip: Trip): bool	
+ updateTripInfo(tripId: int): bool	
+ deleteTrip(tripId: int): bool	
+ getRoutes(): IEnumerable<Route>	
+ getRouteById(routeId: int): Route	
+ addRoute(route: Route): bool	
+ updateRouteInfo(routeId: int): bool	
+ deleteRoute(routeId: int): bool	
+ getCoaches(): IEnumerable<Coaches>	
+ getCoachById(coachId: int): Coach	
+ addCoach(coach: Coach): bool	
+ updateCoachInfo(coachId: int): bool	
+ deleteCoach(coachId: int): bool	
+ getCoachTypes(): IEnumerable<CoachTypes>	
+ getCoachTypeById(coachTypeId: int): CoachType	
+ addCoachType(coachType: CoachType): bool	
+ updateCoachTypeInfo(coachTypeId: int): bool	
+ deleteCoachType(coachTypeId: int): bool	

Properties: N/A

Operations

Signature: IEnumerable<Station> GetStations()		
Description: Retrieves all the stations from the database		
Parameter	Type	Description
N/A	N/A	N/A

Signature: Station GetStationByStationId(int stationId)		
Description: Retrieves the station given its station id from the database		
Parameter	Type	Description
<i>stationId</i>	int	Id of the station to be retrieved

Signature: Station AddStation(Station station)		
Description: Add new station to the database		
Parameter	Type	Description
<i>station</i>	Station	The new station to be added to database

Signature: bool UpdateStation(int stationId)

Description: Update station information from given its id from the database

Parameter	Type	Description
stationId	int	Id of the station to be updated

Signature: bool DeleteStation(int stationId)

Description: Delete the station given its id from the database

Parameter	Type	Description
stationId	int	Id of the station to be deleted

Signature: IEnumerable<Trip> GetTrips()

Description: Retrieves all the trips from the database

Parameter	Type	Description
N/A	N/A	N/A

Signature: Station GetTripByTripId(int tripId)

Description: Retrieves the station given its station id from the database

Parameter	Type	Description
tripId	int	Id of the trip to be retrieved

Signature: Station AddTrip(Trip trip)

Description: Add new trip to the database

Parameter	Type	Description
trip	Trip	The new trip to be added to database

Signature: bool UpdateTrip(int tripId)

Description: Update trip information from given its id from the database

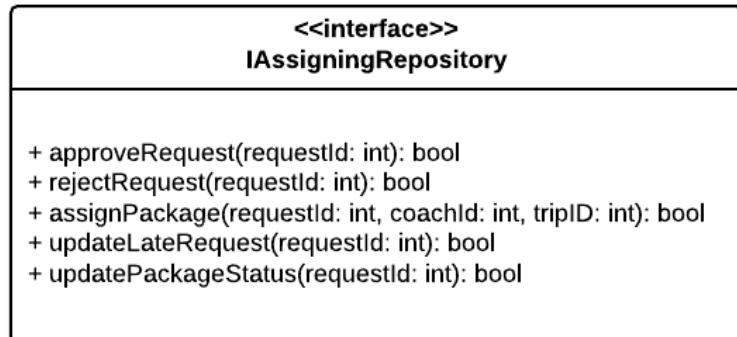
Parameter	Type	Description
tripId	int	Id of the trip to be updated

Signature: bool DeleteTrip(int tripId)

Description: Delete the trip given its id from the database

Parameter	Type	Description
tripId	int	Id of the trip to be deleted

4.4.3.5. IAssigningRepository Interface



Properties: N/A

Operations

Signature: bool approveRequest(int requestId)

Description: Approve pending requests

Parameter	Type	Description
<i>requestId</i>	int	Id of the request to be approved

Signature: bool rejectRequest(int requestId)

Description: Reject pending requests

Parameter	Type	Description
<i>requestId</i>	int	Id of the request to be rejected

Signature: bool assignPackage(int requestId, int coachId, int tripId)

Description: Assign approved package for specified coach

Parameter	Type	Description
<i>requestId</i>	int	Id of the package to be assigned
<i>coachId</i>	int	Id of the coach to assign package
<i>tripId</i>	Int	Id of the trip to assign package

Signature: bool updateLateRequest(int requestId)

Description: Update and re-assign for late arrive package

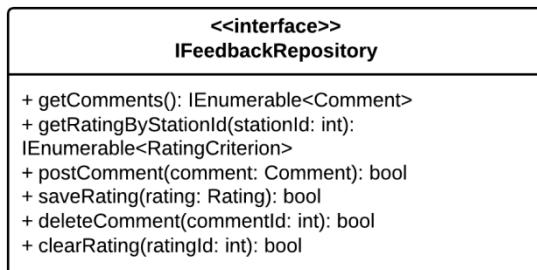
Parameter	Type	Description
<i>requestId</i>	int	Id of the request to be update and re-assigned

Signature: bool updatePackageStatus(int requestId)

Description: Update package status when it departed, arrived, delivered or expired

Parameter	Type	Description
<i>requestId</i>	int	Id of the request to be update and re-assigned

4.4.3.6. IFeedbackRepository Interface



Properties: N/A

Operations

Signature: IEnumerable<Comment> GetComments()

Description: Retrieves all the comments from the database

Parameter	Type	Description
N/A	N/A	N/A

Signature: IEnumerable<RatingCriterion> GetRatingByStationId(int stationId)

Description: Retrieves all the rating that belong to a given station from the database

Parameter	Type	Description
<i>stationId</i>	int	Id of the station that has the rating to be retrieved

Signature: bool postComment(Comment comment)

Description: Post new comment and save to database

Parameter	Type	Description
<i>comment</i>	Comment	The new comment to be saved to database

Signature: bool saveRating(Rating rating)

Description: Save station rating to database

Parameter	Type	Description
<i>rating</i>	Rating	The new station rating to be saved to database

Signature: bool deleteComment(int commentId)

Description: Marks a comment as deleted in the database

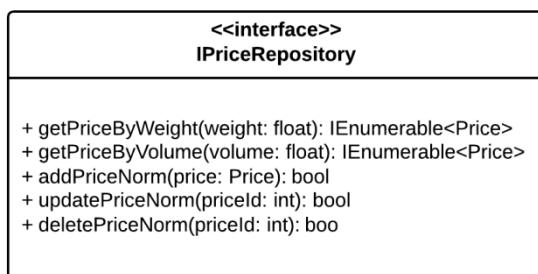
Parameter	Type	Description
<i>commentId</i>	int	Id of the comment to be deleted

Signature: bool clearRating(int ratingId)

Description: Marks a station rating as deleted in the database

Parameter	Type	Description
<i>ratingId</i>	Int	Id of the rating to be deleted

4.4.3.7. IPriceRepository Interface



Properties: N/A

Operations

Signature: IEnumerable<Price> GetPriceByWeight(float weight)

Description: Get delivery price/fee based on package weight

Parameter	Type	Description
<i>weight</i>	float	Weight of delivered package

Signature: IEnumerable<Price> GetPriceByVolume(float volume)

Description: Get delivery price/fee based on package volume

Parameter	Type	Description
<i>volume</i>	float	Volume of delivered package

Signature: bool addPriceNorm(Price price)

Description: Add new price/fee norm to database

Parameter	Type	Description
<i>price</i>	Price	New price norm to be added to database

Signature: bool updatePriceNorm(priceId: int)

Description: Update a price norm to the database

Parameter	Type	Description
<i>priceId</i>	int	The price norm to be saved

Signature: bool updatePriceNorm(priceId: int)

Description: Delete/Remove a price norm from the database

Parameter	Type	Description
priceId	int	The price norm to be deleted

4.4.4. Business Logic Classes

4.4.4.1. Class diagram

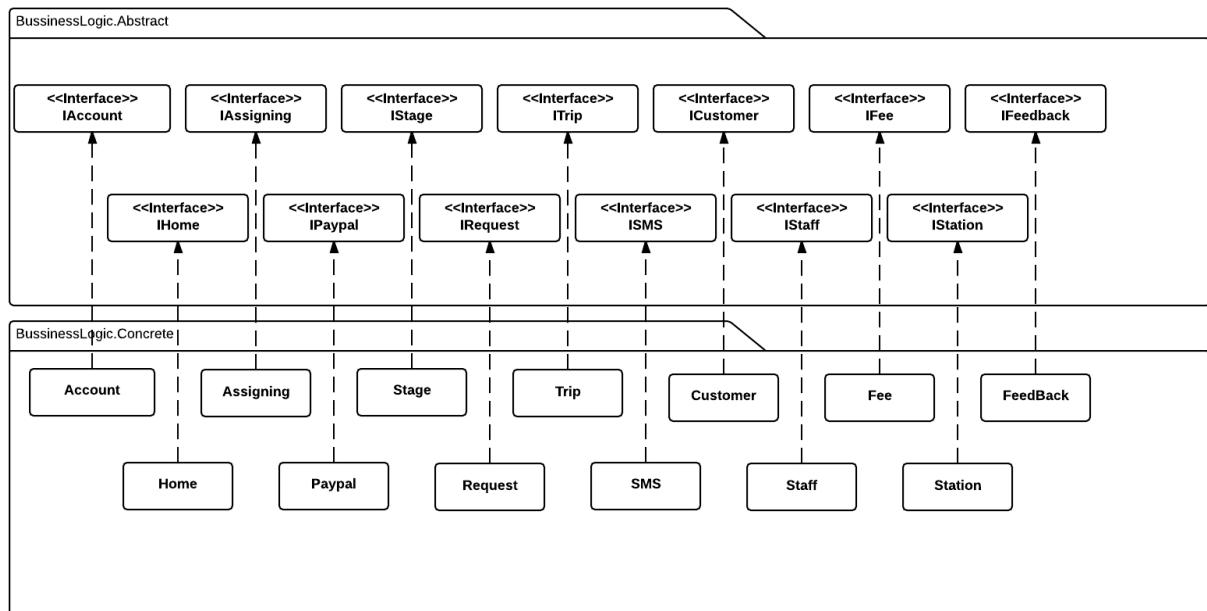
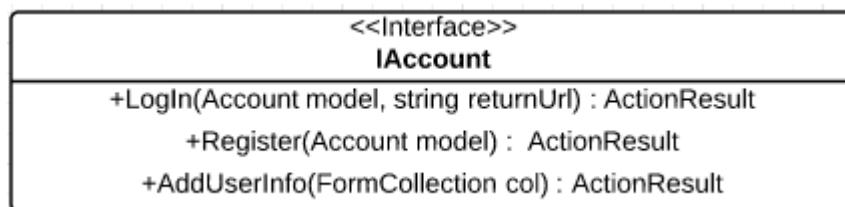


Diagram 7. Business Logic Class Diagram

4.4.4.2. IAccount Interface



Properties: N/A

Operations

Signature: bool LogIn(Account model, string returnUrl)

Description: Login account to system

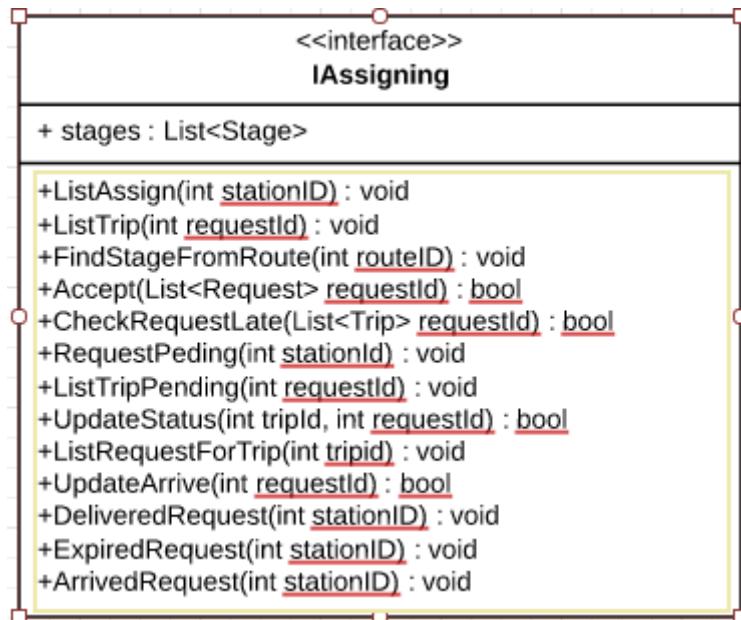
Parameter	Type	Description
<code>model</code>	Account	The data of account model
<code>returnUrl</code>	string	The url of redirect

Signature: bool Register(Account model)

Description: Create a new account for user

Parameter	Type	Description
<code>model</code>	Account	The data of account model

4.4.4.3. IAssigning Interface



Properties

Property name	Type	Visibility	Description
stages	List<Stage>	public	A list of all stage in the stage list

Operations

Signature: void ListAssign(int stationID)

Description: List requests has been assigned at the station of staff

Parameter	Type	Description
stationID	int	Id of stationID of staff

Signature: void ListTrip(int requestId)

Description: List trip that have requestId

Parameter	Type	Description
requestId	int	Id of the request

Signature: void FindStageFromRoute(int routeID)

Description: Find the stage from the route

Parameter	Type	Description
routeID	int	Id of the route

Signature: void Accept(List<Request> requestId)

Description: Accept request that on the ways

Parameter	Type	Description
requestId	List<Request>	Id of the request

Signature: bool CheckRequestLate(List<Trip> requestId)

Description: Saves the booking list to the database

Parameter	Type	Description
requestId	int	Id of list request to check

Signature: void RequestPeding(int stationId)

Description: List request pending at the staff station

Parameter	Type	Description
stationId	int	Id of the station

Signature: void ListTripPending(int requestId)

Description: List trip pending

Parameter	Type	Description
requestId	int	Id of the request

Signature: bool UpdateStatus(int tripId, int requestId)

Description: Update the status of request

Parameter	Type	Description
requestId	int	Id of the request
tripId	int	Id of the trip

Signature: void ListRequestForTrip(int tripid)

Description: List request for trip

Parameter	Type	Description
tripid	int	Id of the trip

Signature: bool UpdateArrive(int requestId)

Description: Update request arrived

Parameter	Type	Description
requestId	int	Id of the request

Signature: void DeliveredRequest(int stationID)

Description: List request have status is delivered

Parameter	Type	Description
stationID	int	Id of the station

Signature: void ExpiredRequest(int stationID)

Description: List request have status is expired

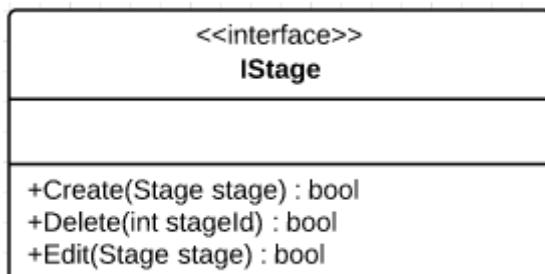
Parameter	Type	Description
stationID	int	Id of the station

Signature: void ArrivedRequest(int stationID)

Description: List request have status is arrived

Parameter	Type	Description
requestId	int	Id of the request

4.4.4.4. IStage Interface



Properties: N/A

Operations

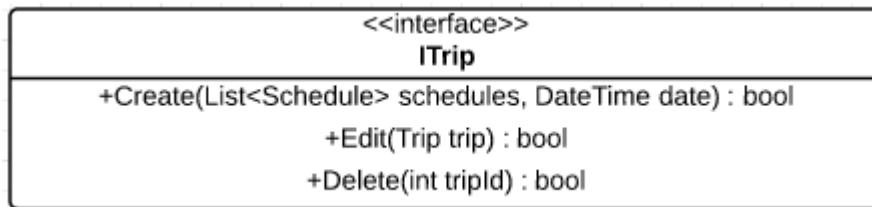
Signature: bool Create(Stage stage)

Description: Create new stage

Parameter	Type	Description

<i>stage</i>	Stage	Stage data model
Signature: bool Delete(int stageId) Description: Delete stage		
Parameter	Type	Description
<i>stageId</i>	Int	Id of the stage to delete
Signature: bool Edit(Stage stage) Description: Edit stage		
Parameter	Type	Description
<i>stage</i>	Stage	Stage data model

4.4.4.5. ITrip Interface



Properties: N/A

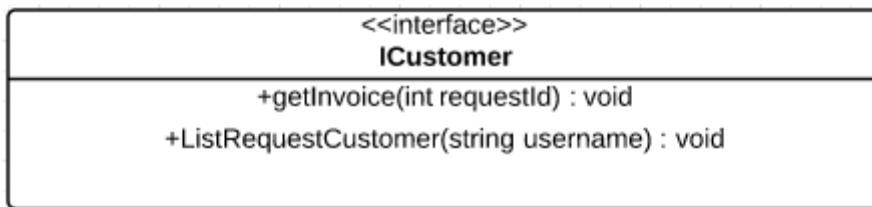
Operations

Signature: bool Create(List<Schedule> schedules, DateTime date) Description: Adds a new trip to database		
Parameter	Type	Description
<i>Schedules</i>	List<Schedule>	List of the schedule
<i>date</i>	DateTime	

Signature: bool Edit(Trip trip) Description: Update existing trip		
Parameter	Type	Description
<i>trip</i>	Trip	Trip to edit

Signature: bool Delete(int tripId) Description: Delete trip from database		
Parameter	Type	Description
<i>tripId</i>	int	Id of trip to delete

4.4.4.6. ICustomer Interface



Properties: N/A

Operations

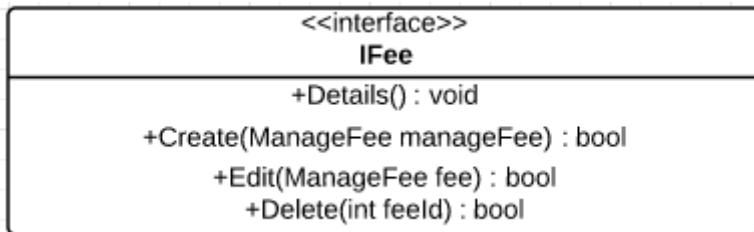
Signature: void getInvoice(int requestId) : void Description: List invoice		
Parameter	Type	Description
<i>requestId</i>	Int	Id of the request to get the invoice

Signature: void ListRequestCustomer(string username)

Description: List of request customer

Parameter	Type	Description
username	string	Username of customer to list the request

4.4.4.7. IFee Interface



Properties: N/A

Operations

Signature: bool Create(ManageFee manageFee)

Description: Add new fee to manage

Parameter	Type	Description
manageFee	ManageFee	ManageFee model need to create new data

Signature: bool Edit(ManageFee fee)

Description: Edit fee from manage

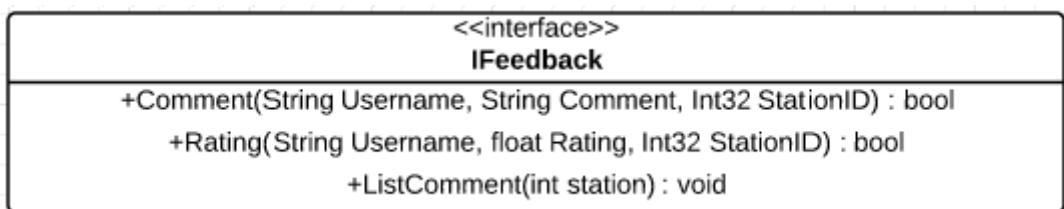
Parameter	Type	Description
fee	ManageFee	ManageFee model need to edit data

Signature: bool Delete(int feedId)

Description: Removes the fee from manage

Parameter	Type	Description
feedId	int	Id of the fee need to removes

4.4.4.8. IFeedback Interface



Properties: N/A

Operations

Signature: bool Comment(String Username, String Comment, Int32 StationID)

Description: Add new comment of customer

Parameter	Type	Description
Username	String	Username of customer to comment
Comment	String	Content of comment
StationID	Int	Id of the station to comment

Signature: bool Rating(String Username, float Rating, Int32 StationID)

Description: Add rating of customer for station

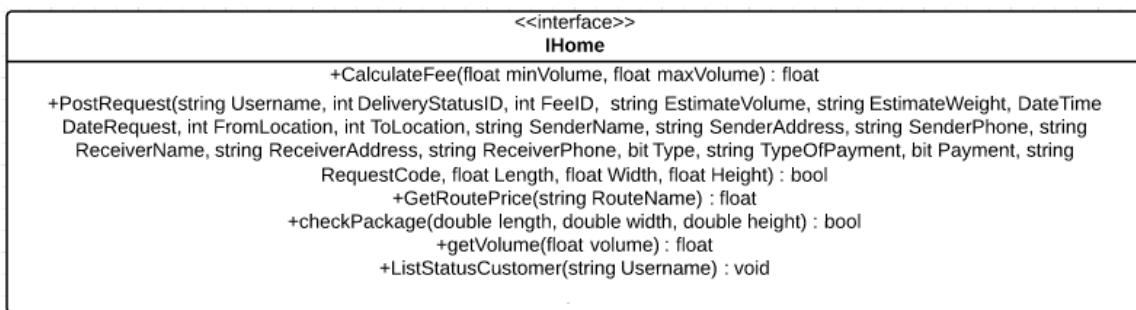
Parameter	Type	Description
Username	String	Username of customer to rating
Rating	Float	Point rating of customer
StationID	Int	Id of station to rating

Signature: void ListComment(int station)

Description: List all the comment of station

Parameter	Type	Description
station	int	Id of the station to list the comment

4.4.4.9. IHome Interface



Properties: N/A

Operations

Signature: float CalculateFee(float minVolume, float maxVolume)

Description: CalculateFee from customer input

Parameter	Type	Description
minVolume	float	Min of the volume to calculate
maxVolume	float	Max of the volume to calculate

Signature: bool PostRequest(string Username, int DeliveryStatusID, int FeeID, string EstimateVolume, string EstimateWeight, DateTime DateRequest, int FromLocation, int ToLocation, string SenderName, string SenderAddress, string SenderPhone, string ReceiverName, string ReceiverAddress, string ReceiverPhone, bit Type, string TypeOfPayment, bit Payment, string RequestCode, float Weight, float Width, float Height)

Description: Insert new request of customer

Parameter	Type	Description
Username	String	Username of customer to post new request
DeliveryStatusID	Int	Id of the delivery to know status of request
FeeID	Int	Id of the fee when customer post request
EstimateVolume	String	The estimate volume customer input
EstimateWeight	String	The estimate weight customer input
DateRequest	DateTime	Date that customer post request
FromLocation	Int	Id of the station that customer sent package
ToLocation	Int	Id of the station that customer receiver package
SenderName	String	Name of sender who post request
SenderAddress	String	Address of sender who post request
SenderPhone	String	Phone of sender who post request

<i>ReceiverName</i>	String	Name of receiver who will receiver package
<i>ReceiverAddress</i>	String	Address of receiver who will receiver package
<i>ReceiverPhone</i>	String	Phone of receiver who will receiver package
<i>Type</i>	Bit	Status of request, return or delete
<i>TypeOfPayment</i>	String	Type of payment when customer post request
<i>Payment</i>	Bit	Status of payment when customer post request
<i>RequestCode</i>	String	Random code for request when customer post request
<i>Width</i>	Float	Width of package request
<i>Height</i>	Float	Height of package request
<i>Length</i>	Float	Length of package request

Signature: float GetRoutePrice(string RouteName)

Description: Get the price of route to calculate total fee

Parameter	Type	Description
<i>RouteName</i>	String	Name of the route to calculate fee

Signature: bool checkPackage(double length, double width, double height)

Description: Check package is valid or not

Parameter	Type	Description
<i>length</i>	Double	Length of the package to check
<i>width</i>	Double	Width of the package to check
<i>height</i>	Double	Height of the package to check

Signature: float getVolume(float volume)

Description: Get min volume, max volume when customer input length, width, height

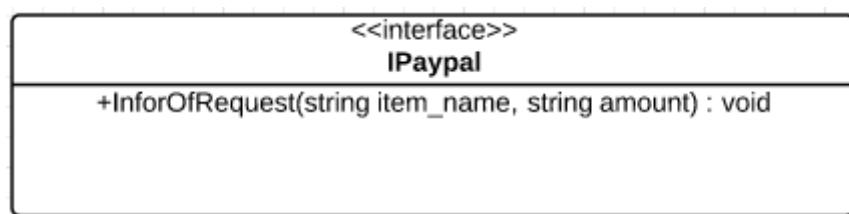
Parameter	Type	Description
<i>volume</i>	float	Volume that calculate base on length, width, height when customer input request

Signature: void ListStatusCustomer(string Username)

Description: List all status of customer to view request that they has been posted

Parameter	Type	Description
<i>Username</i>	String	Username of customer to list request follow the status

4.4.4.10. IPaypal Interface



Properties: N/A

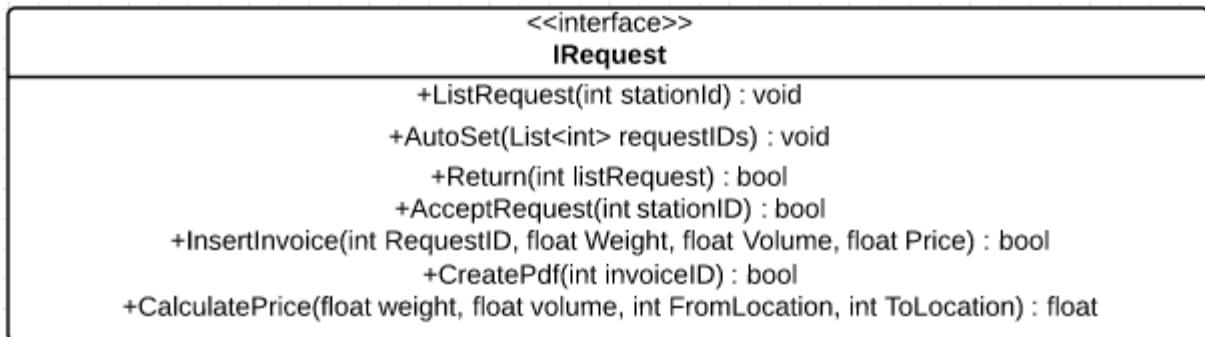
Operations

Signature: void InforOfRequest(string item_name, string amount)

Description: View infor of request on paypal page

Parameter	Type	Description
item_name	String	Name of request
amount	String	Amount of request

4.4.4.11. IRequest Interface



Signature: void ListRequest(int stationId)

Description: List all request of station

Parameter	Type	Description
stationId	Int	Id of station to list request

Signature: void AutoSet(List<int> requestIDs)

Description: Auto set request status to 'Đã hết hạn - 8' if delivery date was late 2 days

Parameter	Type	Description
requestIDs	List<int>	List of request that need to check late

Signature: bool Return(int listRequest)

Description: Prepare late delivery request for returning

Parameter	Type	Description
listRequest	Int	Id of request need to returning

Signature: bool AcceptRequest(int stationID)

Description: The place to receiver new request and the station of staff

Parameter	Type	Description
stationID	Int	Id of the station to view new request

Signature: bool InsertInvoice(int RequestID, float Weight, float Volume, float Price)

Description: Create invoice for request

Parameter	Type	Description
RequestID	Int	Id of request to create invoice
Weight	Float	Weight of request to create invoice
Volume	Float	Volume of request to create invoice

<i>Price</i>	Float	Price of request to create invoice
--------------	-------	------------------------------------

Signature: bool CreatePDF(int invoiceID)

Description: Create PDF for invoice

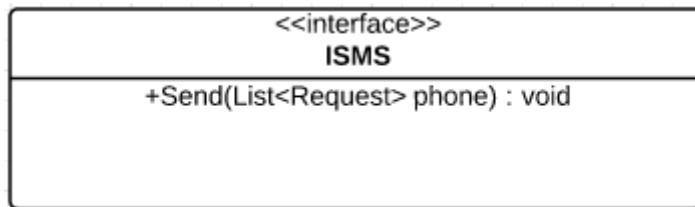
Parameter	Type	Description
<i>invoiceID</i>	Int	Id of the station to view new request

Signature: float CalculatePrice(float weight, float volume, int FromLocation, int ToLocation)

Description: Create invoice for request

Parameter	Type	Description
<i>RequestID</i>	Int	Id of request to create invoice
<i>Weight</i>	Float	Weight of request to create invoice
<i>Volume</i>	Float	Volume of request to create invoice
<i>Price</i>	Float	Price of request to create invoice

4.4.4.12. ISMS Interface



Properties: N/A

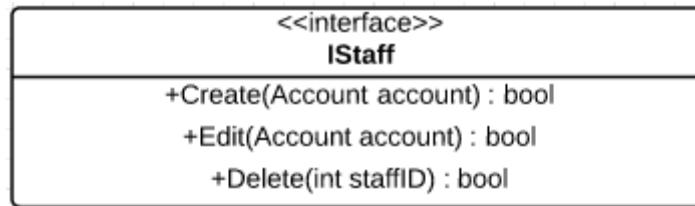
Operations

Signature: void Send(List<Request> phone)

Description: Send notification to customer

Parameter	Type	Description
<i>phone</i>	List<Request>	List of phone customer need to send SMS

4.4.4.13. IStaff Interface



Properties: N/A

Operations

Signature: bool Create(Account account)

Description: Create new account for staff

Parameter	Type	Description
<i>account</i>	Account	Account model to create

Signature: bool Edit(Account account)

Description: Edit account information for staff

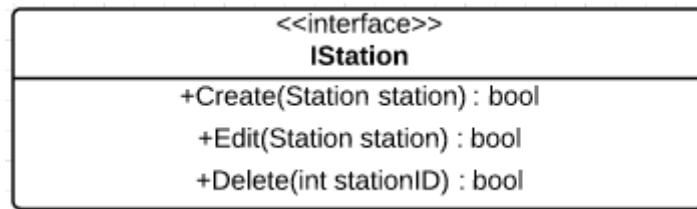
Parameter	Type	Description
account	Account	Account model to edit

Signature: bool Delete(int staffID)

Description: Remove staff

Parameter	Type	Description
staffID	Int	Id of staff to remove

4.4.4.14. IStation Interface



Properties: N/A

Operations

Signature: bool Create(Station station)

Description: Create new station

Parameter	Type	Description
station	Station	Station model that need to create

Signature: bool Edit(Station station)

Description: Edit station information

Parameter	Type	Description
station	Station	Station model that need to edit

Signature: bool Delete(int stationID)

Description: Remove station

Parameter	Type	Description
stationID	Int	Id of the station to remove

4.5. Behavioral Diagrams

4.5.1. State Machine Diagram: Request Status Transition

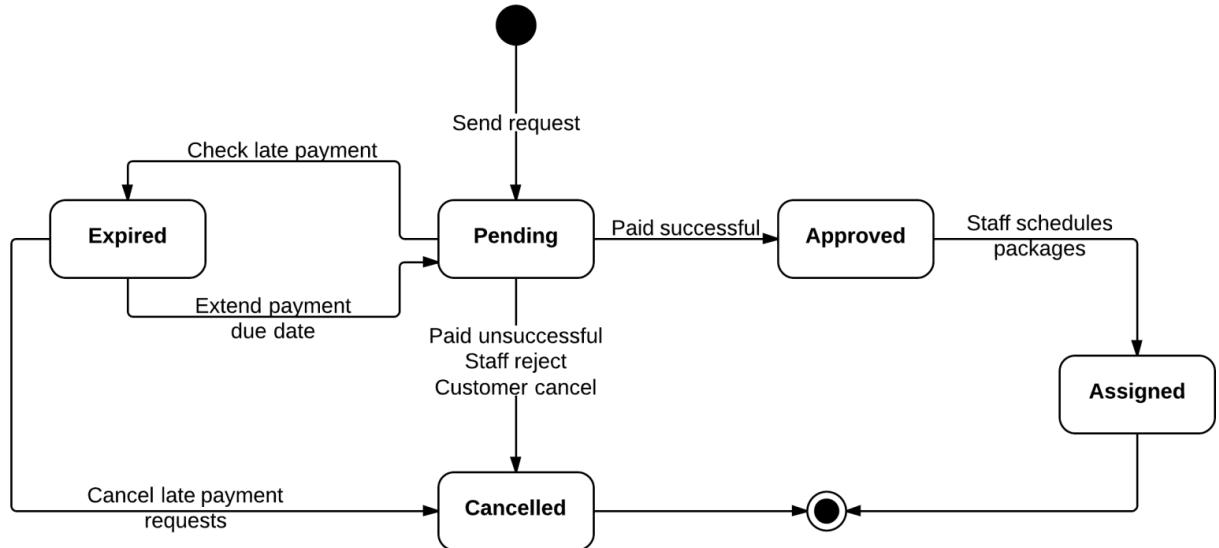


Diagram 8. State machine diagram: Request status transition

4.5.2. State Machine Diagram: Package Status Transition

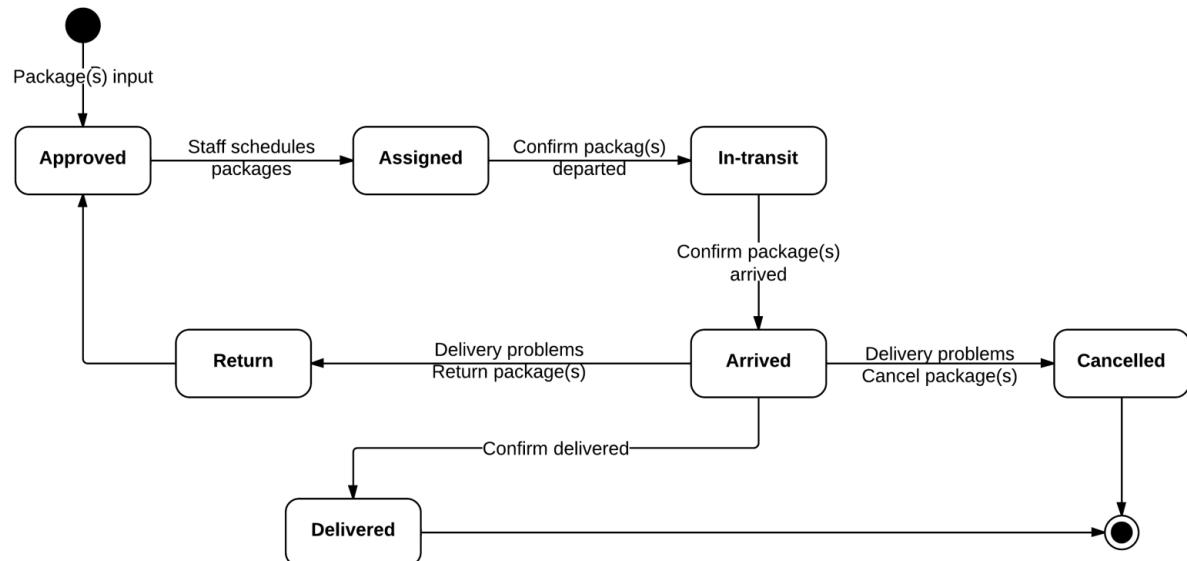


Diagram 9. State machine diagram: Package status transition

4.5.3. Sequence Diagram: Send Delivery Requests

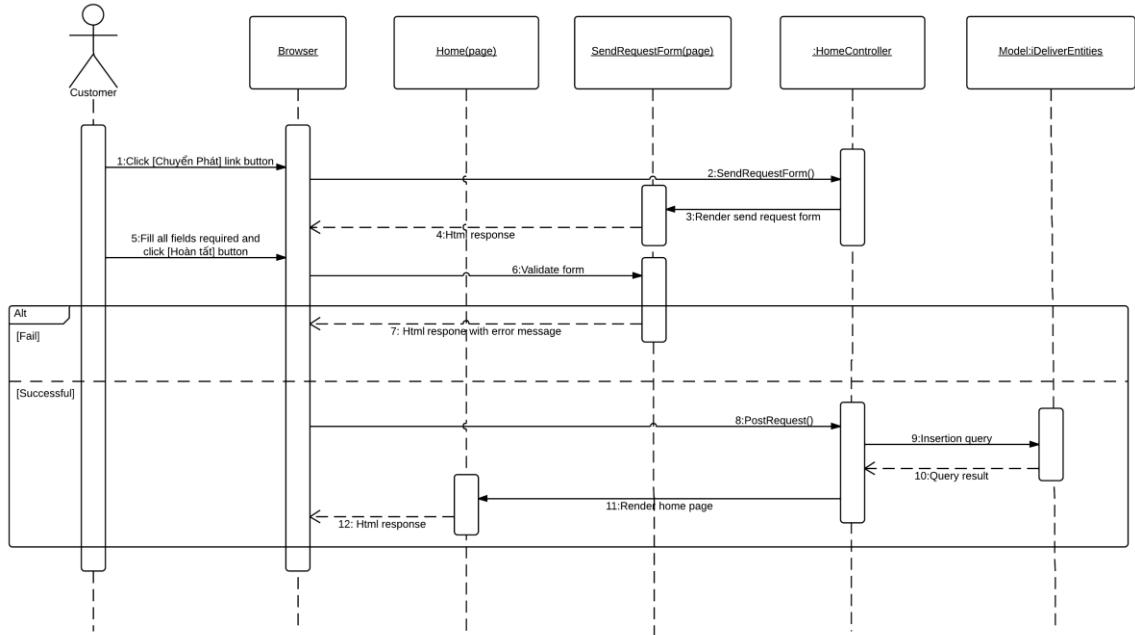


Diagram 10. Sequence diagram: Send delivery request

4.5.4. Sequence Diagram: Tracking

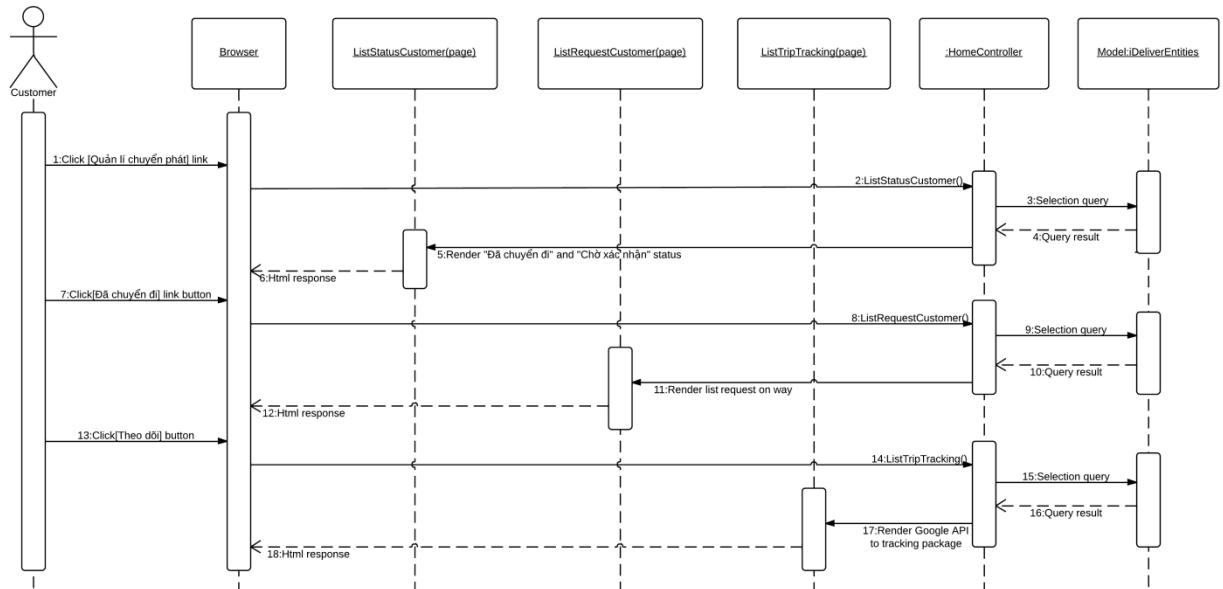


Diagram 11. Sequence diagram: Tracking package

4.5.5. Sequence Diagram: Payment

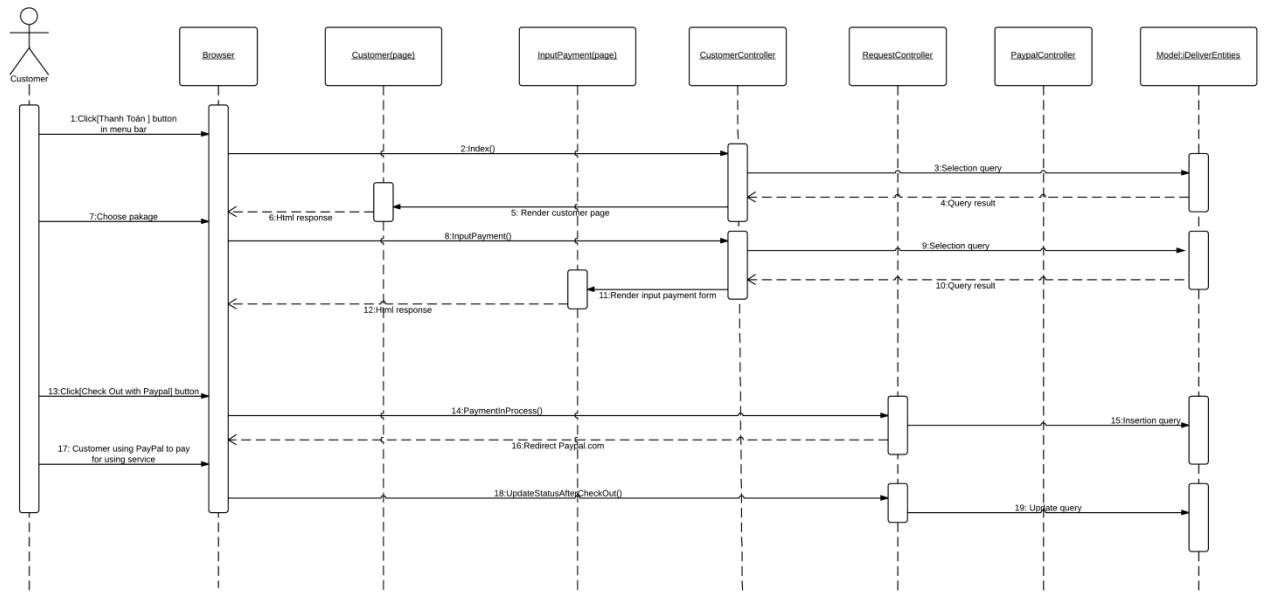


Diagram 12. Sequence diagram: Payment

4.5.6. Sequence Diagram: Feedback

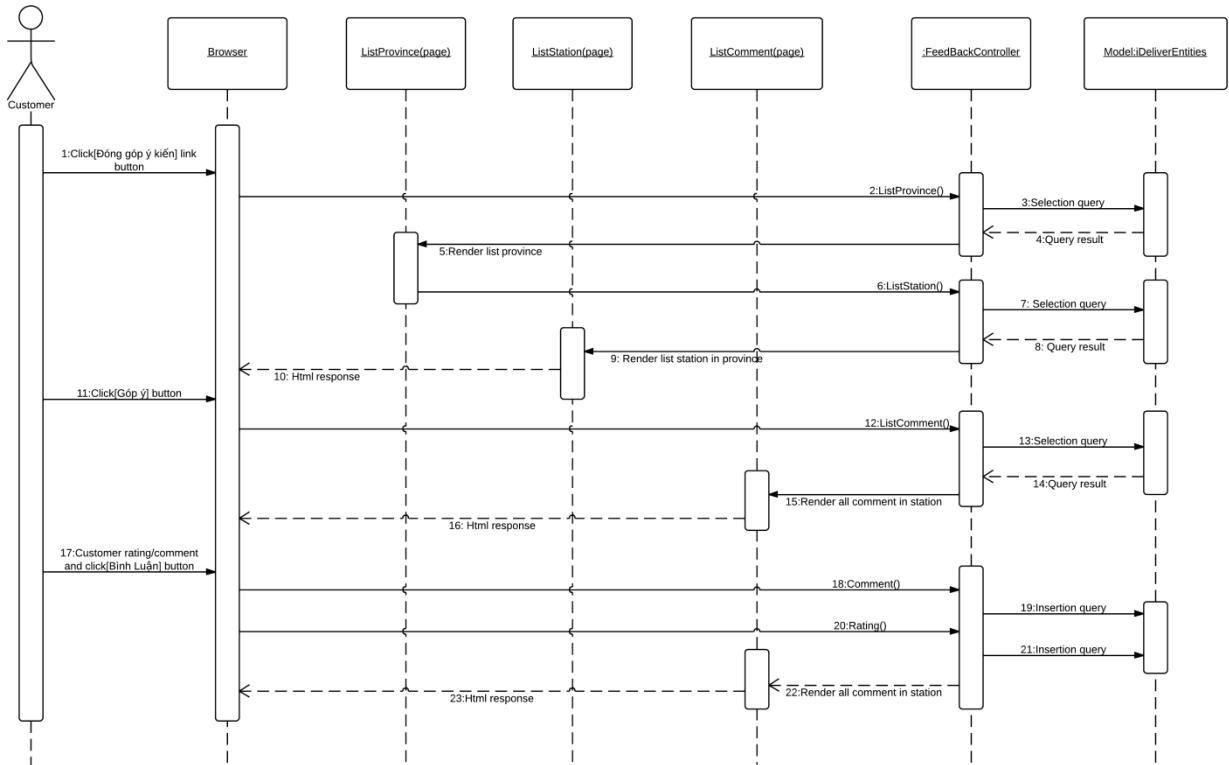


Diagram 13. Sequence diagram: Feedback

4.5.7. Sequence Diagram: Approve Requests

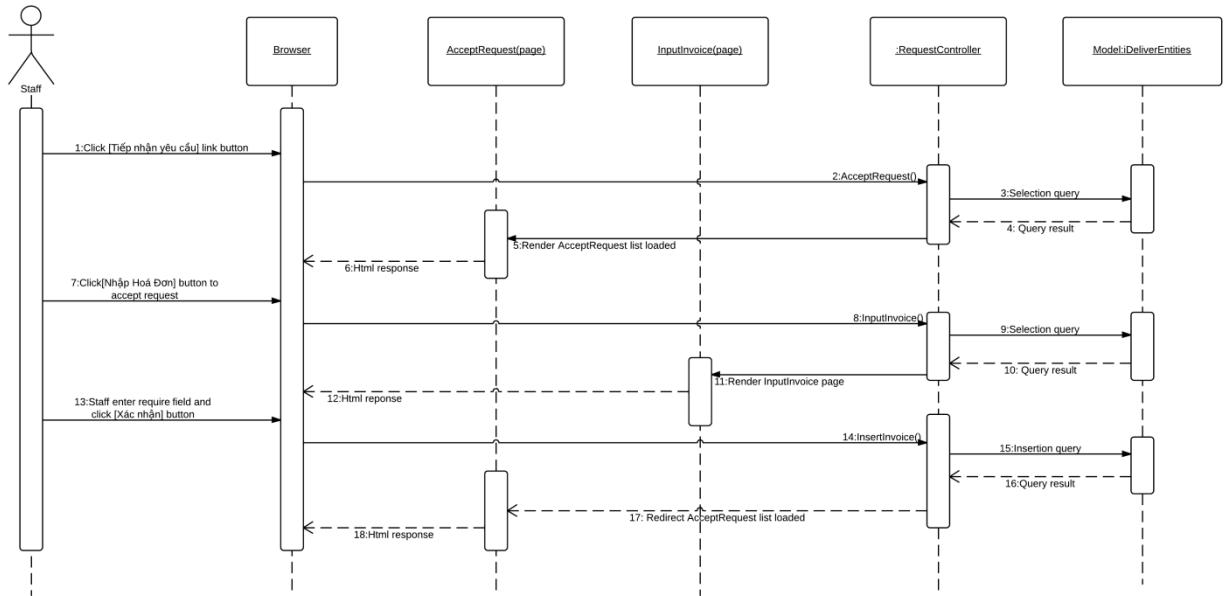


Diagram 14. Sequence diagram: Approve Requests

4.5.8. Sequence Diagram: Assign Package

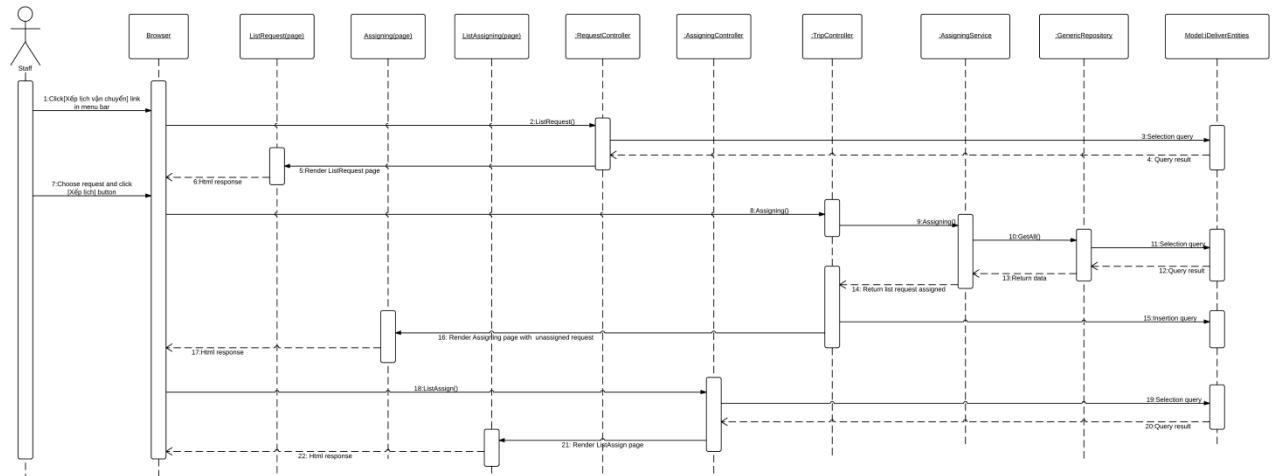


Diagram 15. Sequence diagram: Assign packages

4.5.9. Sequence Diagram: Package in-transit

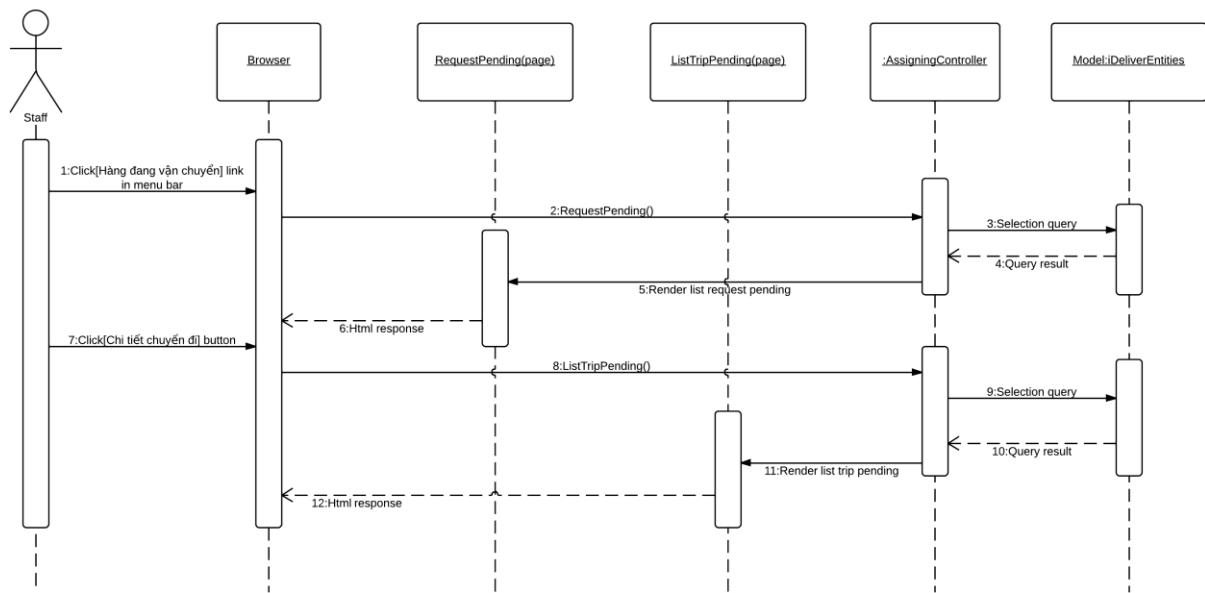


Diagram 16. Sequence diagram: Packages in-transit

4.5.10. Sequence Diagram: Package arrived

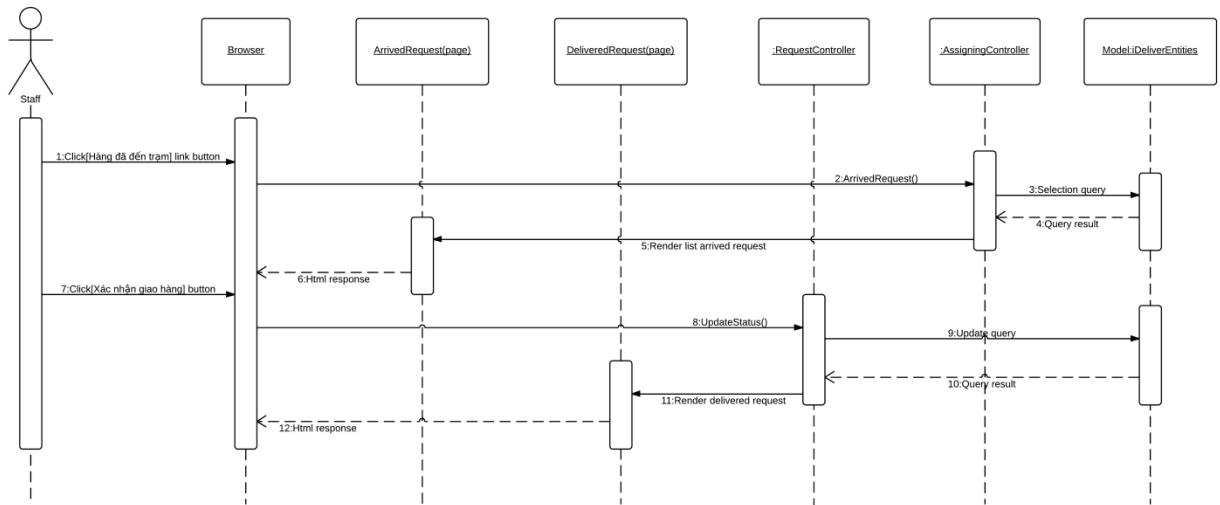


Diagram 17. Sequence diagram: Package arrived

4.5.11. Sequence Diagram: Package delivered

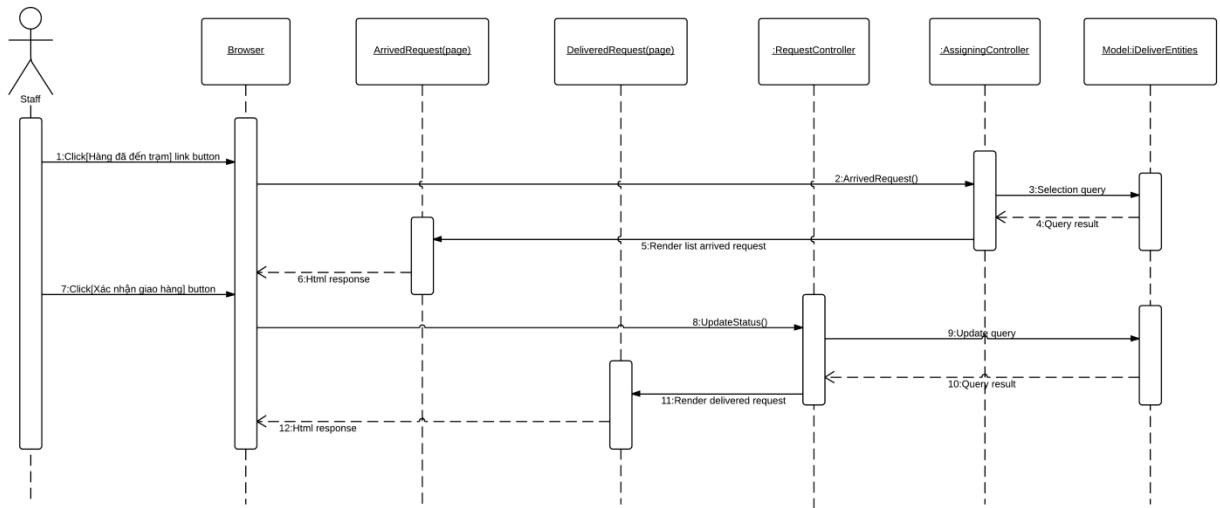


Diagram 18. Sequence diagram: Package delivered

4.5.12. Sequence Diagram: Package return

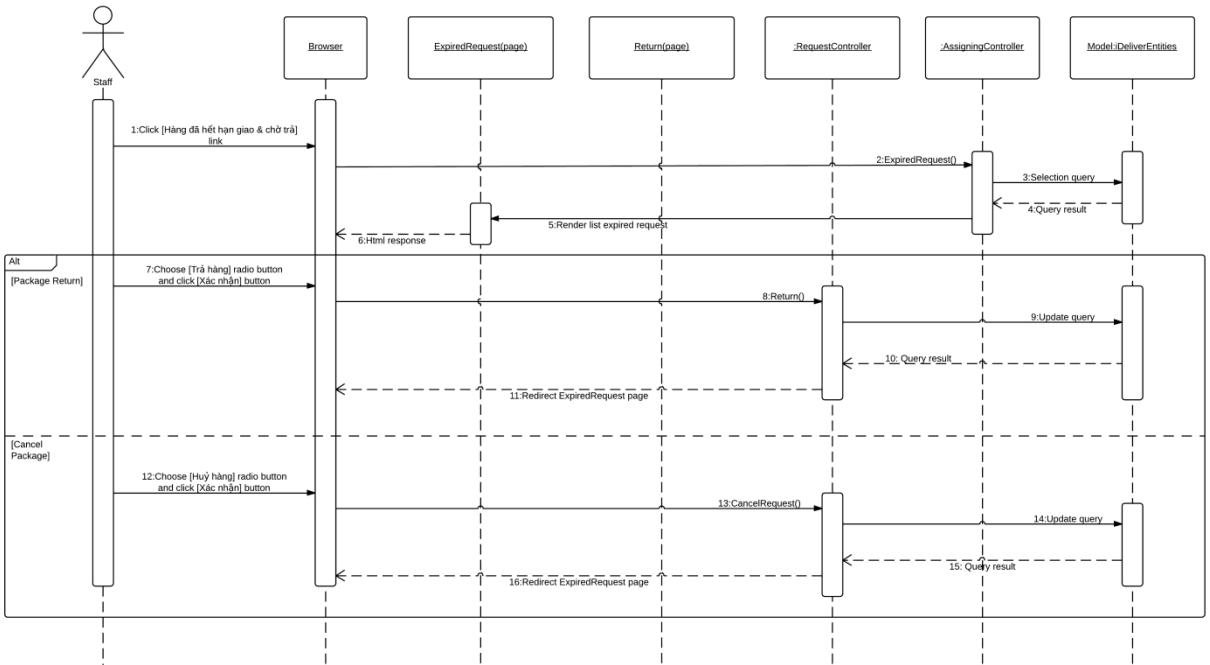


Diagram 19. Sequence diagram: Package return

4.5.13. Sequence Diagram: Manage late payment requests

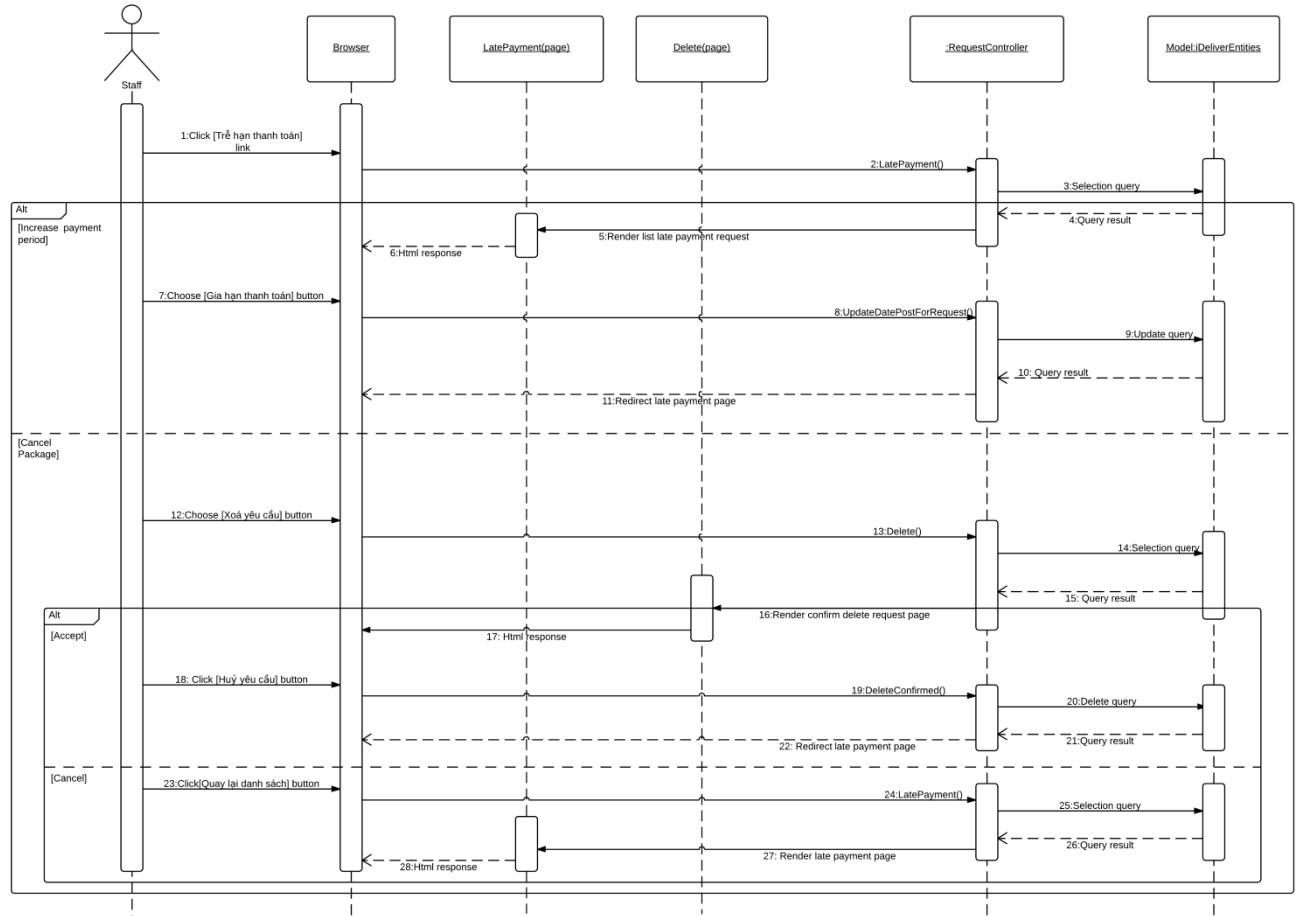


Diagram 20. Sequence diagram: Manage late payment requests

4.5.14. Sequence Diagram: Create schedule

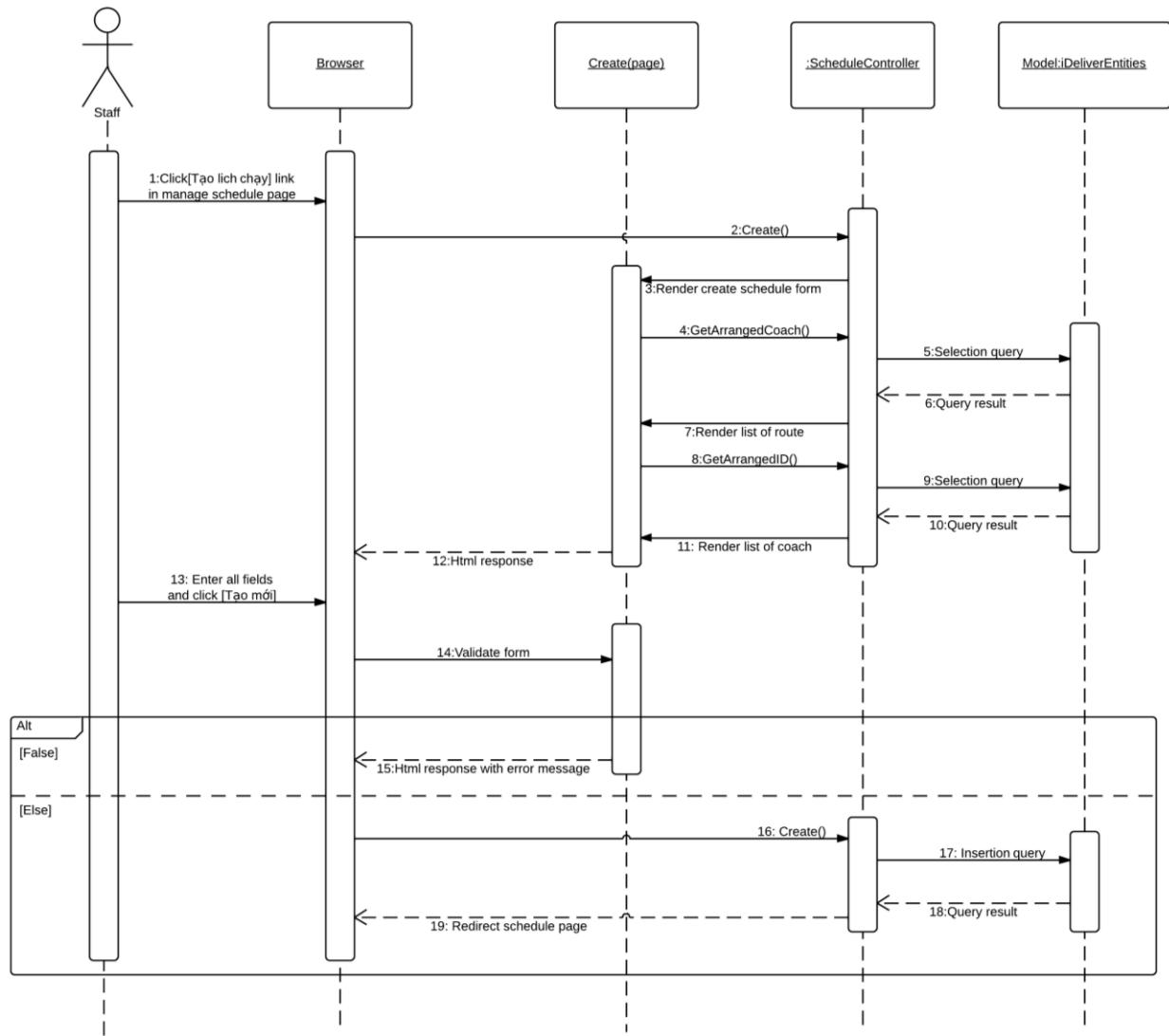


Diagram 21. Sequence diagram: Create schedule

4.5.15. Sequence Diagram: Create trip

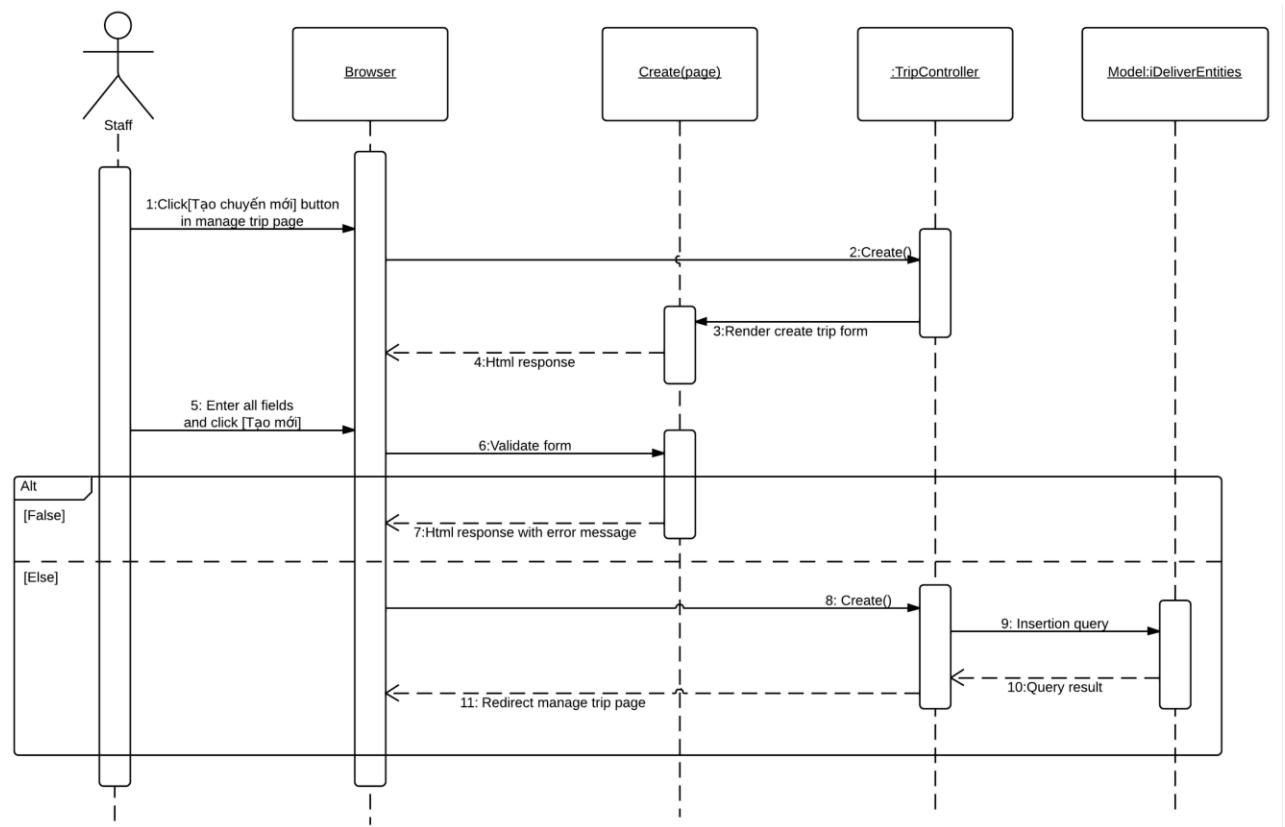


Diagram 22. Sequence diagram: Create trip

4.5.16. Sequence Diagram: Create staff account

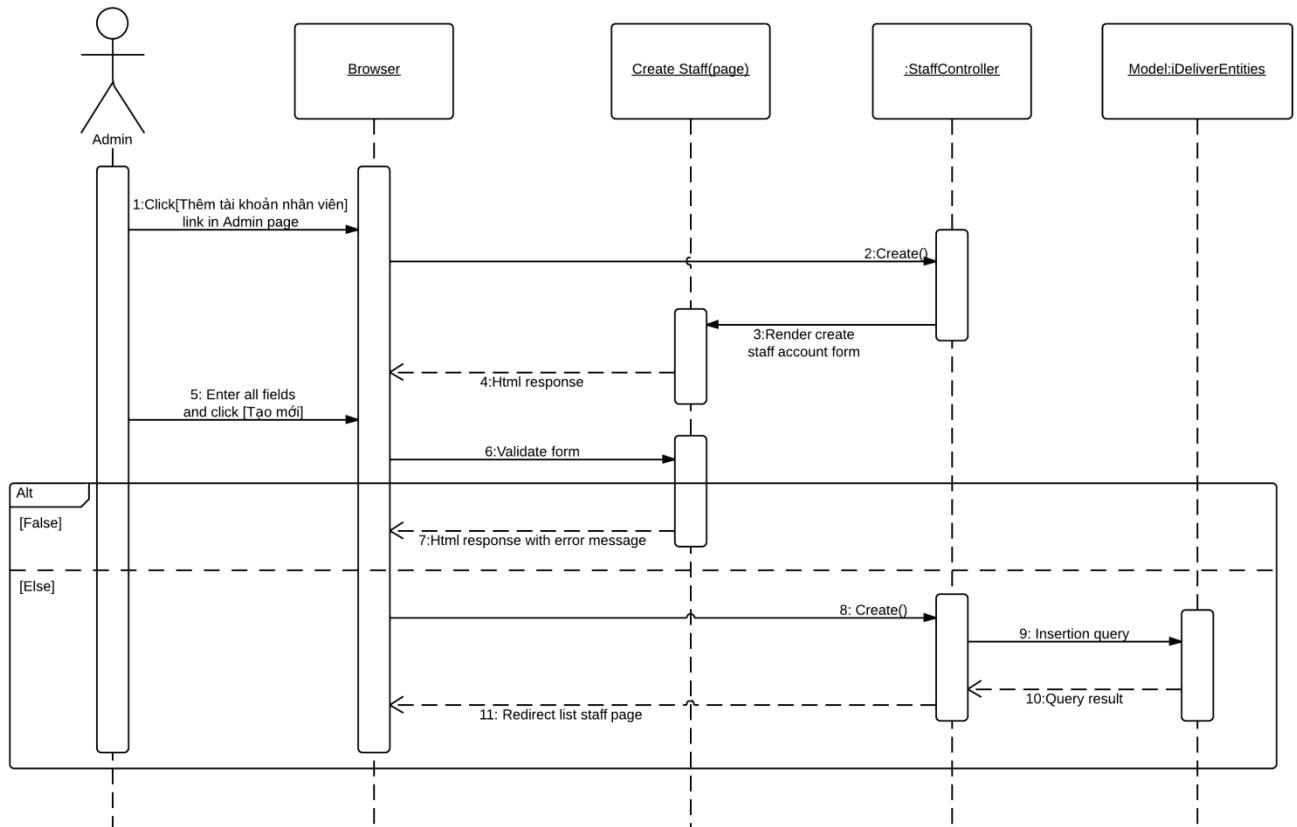


Diagram 23. Sequence diagram: Create staff account

4.6. User Interface Design

Below are some important screen designs for the i-Deliver systems.

4.6.1. Home Page

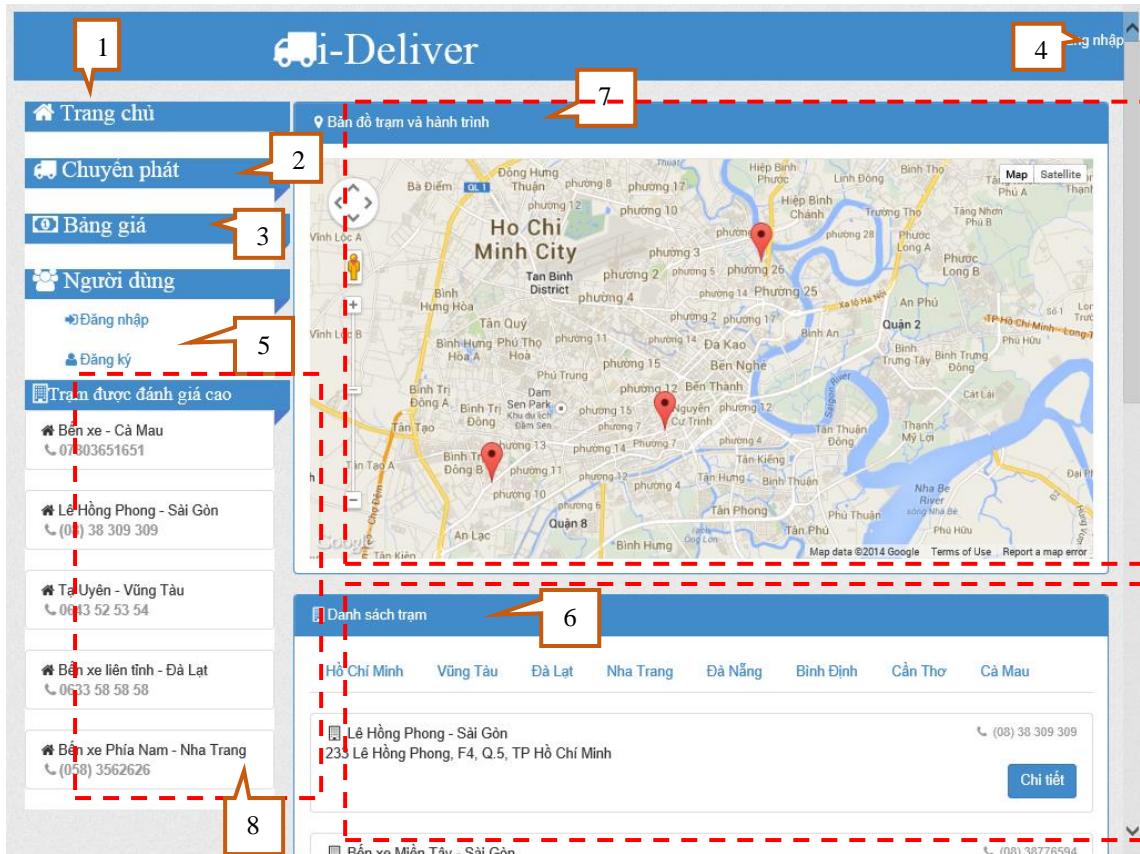


Figure 5. Home page

No.	Element	Description	Read Only	Mandatory	Control Type	Data type
1.	"Trang chủ" link	A link to the home page of the i-Deliver website	N/A	N/A	Link	N/A
2.	"Chuyển phát" link	A link to send delivery request	N/A	N/A	Link	N/A
3.	"Bảng giá" link	A link to the price/fee page	N/A	N/A	Button	N/A
4.	"Đăng nhập" button	A button that displays the log in popup	N/A	N/A	Button	N/A
5.	"Đăng ký" button	A button that displays the register popup	N/A	N/A	Button	N/A
6.	"Danh sách trạm" section	Provinces and stations located in that province	N/A	N/A	N/A	Text
7.	"Bản đồ trạm và hành trình" maps	The marker indicate station location on Google maps	N/A	N/A	N/A	Image
8.	"Trạm được đánh giá cao" section	List of high level rating stations	N/A	N/A	N/A	Text

4.6.2. Send Delivery Request Page

The screenshot shows a delivery request interface. At the top right is a map from Minh City to Vũng Tàu, indicating a route length of 476 km and a travel time of 5 hours. Below the map, travel details are listed: Bến xe - Cà Mau, 07803651651; Lê Hồng Phong - Sài Gòn, (08) 38 309 309; Tà Uyên - Vũng Tàu, 0643 52 53 54; Bến xe liên tỉnh - Đà Lạt, 0633 58 58 58; and Bến xe Phía Nam - Nha Trang, (058) 3562626. The main form area contains the following fields:

- Thông tin cơ bản** (Basic information):
 - Tên Người Gửi: Nguyễn Thanh Tùng
 - Địa chỉ Người Gửi: Trần Nhân Tông, Đà Lạt
 - Số Điện Thoại Người Gửi: 0906425539
 - Tên Người Nhận: Nguyễn Tân Đức
 - Địa chỉ Người Nhận: Hưng Đạo, Vũng Tàu
 - Số Điện Thoại Người Nhận: 0909319921
- Lịch trình dự kiến** (Estimated schedule):
 - Thời gian: 5 giờ
 - Khoảng cách: 476 km
- Thông tin sản phẩm** (Product information):
 - Khối lượng ước tính (kg): 10 - 18
 - Rộng (dcm): 2
 - Cao (dcm): 5
 - Dài (dcm): 7.5
 - Tính ước tính (dcm³): 7 - 75
 - Giá ước tính: 90000VND
 - Hình thức thanh toán: Tiền mặt
- Hoàn Tất** (Finish) button

Numbered callouts point to specific fields and controls:

- “Tên người gửi” textbox
- “Địa chỉ người gửi” textbox
- “Số điện thoại người gửi” textbox
- “Tên người nhận” textbox
- “Địa chỉ người nhận” textbox
- “Số điện thoại người nhận” textbox
- “Khối lượng ước tính” dropdown list
- “Rộng”, “Cao”, “Dài” textbox
- “Thể tích ước tính” dropdown list
- “Hoàn Tất” button
- Map and travel details area

Figure 6. Send delivery request page

No.	Element	Description	Read Only	Mandatory	Control Type	Data type
1.	“Tên người gửi” textbox	Full name of the sender	No	Yes	Textbox	Text
2.	“Địa chỉ người gửi” textbox	Address of the sender	No	Yes	Textbox	Text
3.	“Số điện thoại người gửi” textbox	Phone number of the sender	No	Yes	Textbox	Text
4.	“Tên người nhận” textbox	Full name of the receiver	No	Yes	Textbox	Text
5.	“Địa chỉ người nhận” textbox	Address of the receiver	No	Yes	Textbox	Text
6.	“Số điện thoại người nhận” textbox	Phone number of the receiver	No	Yes	Textbox	Text
7.	“Khối lượng ước tính” dropdown list	Estimate weight of package	Yes	Yes	Textbox	Text
8.	“Rộng”, “Cao”, “Dài” textbox	Length, width and height of the package	No	Yes	Textbox	Text
9.	“Thể tích ước tính” dropdown list	Estimate volume of package	Yes	Yes	Textbox	Text

10.	“Phương thức thanh toán” dropdown list	Type of payment for delivery request	Yes	Yes	Textbox	Text
11.	“Hoàn tất” button	A button to submit delivery request	N/A	N/A	Button	N/A

4.6.3. Online Payment Page

DANH SÁCH YÊU CẦU

Tuyến đường: Bến xe - Cà Mau -----> Bến Xe Trung Tâm - Đà Nẵng
Tên người nhận: Nguyễn Thanh Tùng

Secure shopping made faster.
Shop without retying payment details.

Figure 7. Select new request to make a payment

THÔNG TIN HÓA ĐƠN

Mã giao dịch
9Nu2DOF

Khách hàng
lulp

Khối lượng món hàng (kg)
5

Thể tích món hàng
4

Giá món hàng (VND)
20000

Ngày thanh toán
4/16/2014 12:00:00 AM

Figure 8. Make an online payment via Paypal

4.6.4. Approve Request Page

Figure 9. Approve request page

No.	Element	Description	Read Only	Mandatory	Control Type	Data type
1.	“Tìm kiếm” textbox	Request information to search	N/A	N/A	Textbox	Text
2.	Request information section	All of requests waiting for approved	N/A	N/A	N/A	Text
3.	“Nhập hóa đơn” button	A button to input invoice for paid request	N/A	N/A	Button	N/A

4.6.5. Assign Package Page

Figure 10. Assign packages page

4.6.6. Tracking Page

	Dự kiến đến	Ước tính
Cà Mau - Sài Gòn	4/16/2014 6:00:00 PM	9 h
Sài Gòn - Nha Trang	4/18/2014 5:00:00 PM	8 h
Nha Trang - Đà Nẵng	4/20/2014 7:00:00 PM	10 h

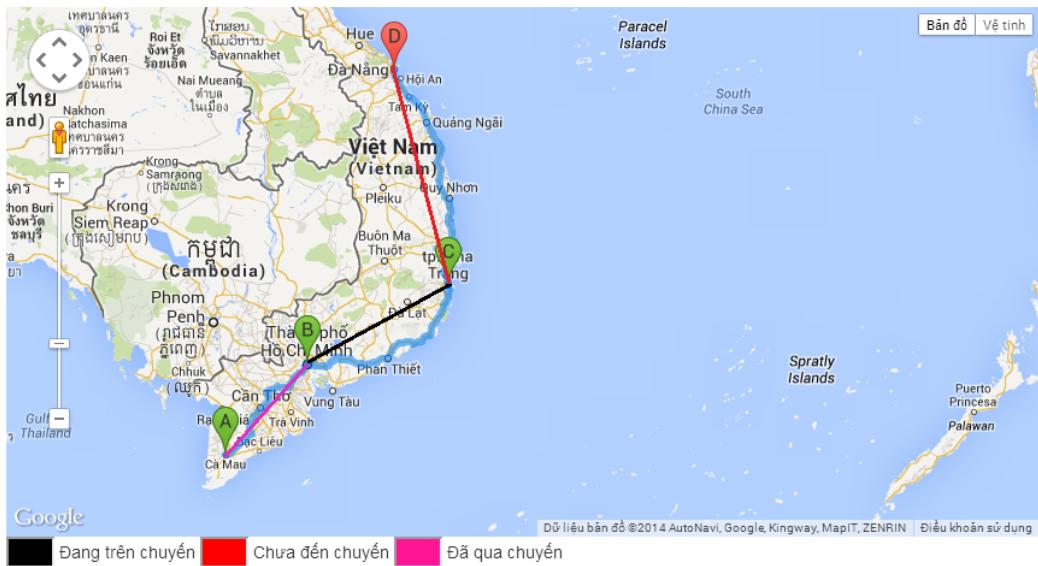


Figure 11. Tracking package page

4.6.7. Update Package Status Page

STAFF10 | Đăng xuất
Nhân viên trạm Bến xe - Cà Mau

Trang chủ

Chuyển Phát

Người dùng

Thông tin cá nhân

Đổi mật khẩu

Đăng xuất

Quản lý hành trình

Xe

Trạm

Chuyến

Tuyến

Lịch chạy

Quản lý yêu cầu vận chuyển

Tiếp nhận yêu cầu 0

Xác nhận thanh toán

Xếp lịch vận chuyển

Hàng đã xếp lịch

Hàng đang vận chuyển

Hàng đã đến trạm

Hàng đã được giao

Hàng đã hết hạn giao & chờ trả

Trễ hạn thanh toán

Quản lý chuyến

Danh sách chuyến Chuyển xe vừa tạo Chuyển đã khởi hành Chuyển đã đến trạm

Xe đến từ trạm khác Tạo chuyến mới

Tìm kiếm

Tuyến	Xe	Giờ đi	Giờ đến dự kiến	Quản lý	Xác nhận chuyến đi	Xóa chuyến này
Cà Mau - Sài Gòn	51B-34578	4/16/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/17/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/18/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/19/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/20/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/21/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/22/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34579	4/16/2014 3:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34579	4/17/2014 3:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34579	4/18/2014 3:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	

Figure 12. Confirm package departed

STAFF5 | Đăng xuất
Nhân viên trạm Bến xe Miền Tây - Sài Gòn

Trang chủ

Chuyển Phát

Người dùng

Thông tin cá nhân

Đổi mật khẩu

Đăng xuất

Quản lý hành trình

Xe

Trạm

Chuyến

Quản lý chuyến

Danh sách chuyến Chuyển xe vừa tạo Chuyển đã khởi hành Chuyển đã đến trạm

Xe đến từ trạm khác Tạo chuyến mới

Tìm kiếm

Tuyến	Xe	Giờ đi	Giờ đến dự kiến		
Cà Mau - Sài Gòn	51B-34580	4/16/2014 7:45:58 AM	4/16/2014 6:00:00 PM	Hàng trên chuyến	Xác nhận đến trạm

Figure 13. Confirm package arrived

STAFF8 | Đăng xuất
Nhân viên trạm Bến Xe Trung Tâm - Đà Nẵng

Các gói hàng đã đến trạm & chờ giao

Tìm kiếm						
Người gửi	Người nhận	Điện thoại người nhận	Ngày gửi	Trạm gửi	Ngày đến	<input type="checkbox"/>
ducnt	Đức	841657577422	4/16/2014	Bến xe - Cà Mau	4/17/2014	<input type="checkbox"/>

Xác nhận giao hàng

Figure 14. Confirm package delivered

STAFF8 | Đăng xuất
Nhân viên trạm Bến Xe Trung Tâm - Đà Nẵng

Các gói hàng đã hết hạn chờ giao

Tìm kiếm								
Người gửi	Người nhận	Trạm gửi	Ngày gửi	Trạm nhận	Ngày đến	Mã hàng	Trà hàng	Hủy hàng
Long	Đức	Bến xe - Cà Mau	4/16/2014 12:42:29 AM	Bến Xe Trung Tâm - Đà Nẵng	4/17/2014 12:00:00 AM	adcae	<input checked="" type="radio"/>	<input checked="" type="radio"/>

Xác nhận

Figure 15. Prepare package to return

STAFF10 | Đăng xuất
Nhân viên trạm Bến xe - Cà Mau

Các yêu cầu đã trễ hạn thanh toán

Tìm kiếm						
Người gửi	Người nhận	Địa chỉ nhận	Điện thoại người gửi	Ngày gửi yêu cầu	Hạn chót thanh toán	Quản lý
ducnt	Đức	Đà Nẵng	0123456789	4/16/2014 12:42:29 AM	4/17/2014 12:42:29 AM	

Figure 16. Extend due-date for late payment requests

4.6.8. Create Trip Page

The screenshot shows a software interface for creating a new trip. At the top, there's a header 'Thông tin chuyến mới'. Below it, a date input field 'Ngày chạy' contains '2014/04/16'. A dropdown menu 'Tuyến' is set to 'Cà Mau - Vũng Tàu'. At the bottom right of the form area are two buttons: 'Tạo mới' (highlighted by callout 4) and 'Hủy'. Callout 2 points to a dashed red rectangular selection box around a list of schedules. Callout 3 points to the first item in this list. The list is titled 'Thời gian ước tính xuất bến' and contains the following data:

Tuyến	Biển số	Thời gian ước tính xuất bến
<input type="checkbox"/> Cà Mau - Vũng Tàu	51B-34584	05:00:00
<input type="checkbox"/> Cà Mau - Vũng Tàu	51B-34585	08:00:00
<input checked="" type="checkbox"/> Cà Mau - Vũng Tàu	51B-34586	11:00:00
<input checked="" type="checkbox"/> Cà Mau - Vũng Tàu	51B-34587	14:00:00
<input type="checkbox"/> Cà Mau - Vũng Tàu	51B-34588	17:00:00
<input checked="" type="checkbox"/> Cà Mau - Vũng Tàu	51B-34589	20:00:00

Figure 17. Create trip page

No.	Element	Description	Read Only	Mandatory	Control Type	Data type
1.	“Ngày chạy” dropdown list	Specific date for trip	N/A	Yes	Textbox	Text
2.	“Tuyến” dropdown list	Specific route for trip	N/A	Yes	Textbox	Text
3.	Schedules checkbox	Defined schedules for selected route	N/A	Yes	Checkbox	Text
4.	“Tạo mới” button	A button to submit create trip command	N/A	N/A	Button	N/A

4.6.9. Feedback and Review Page

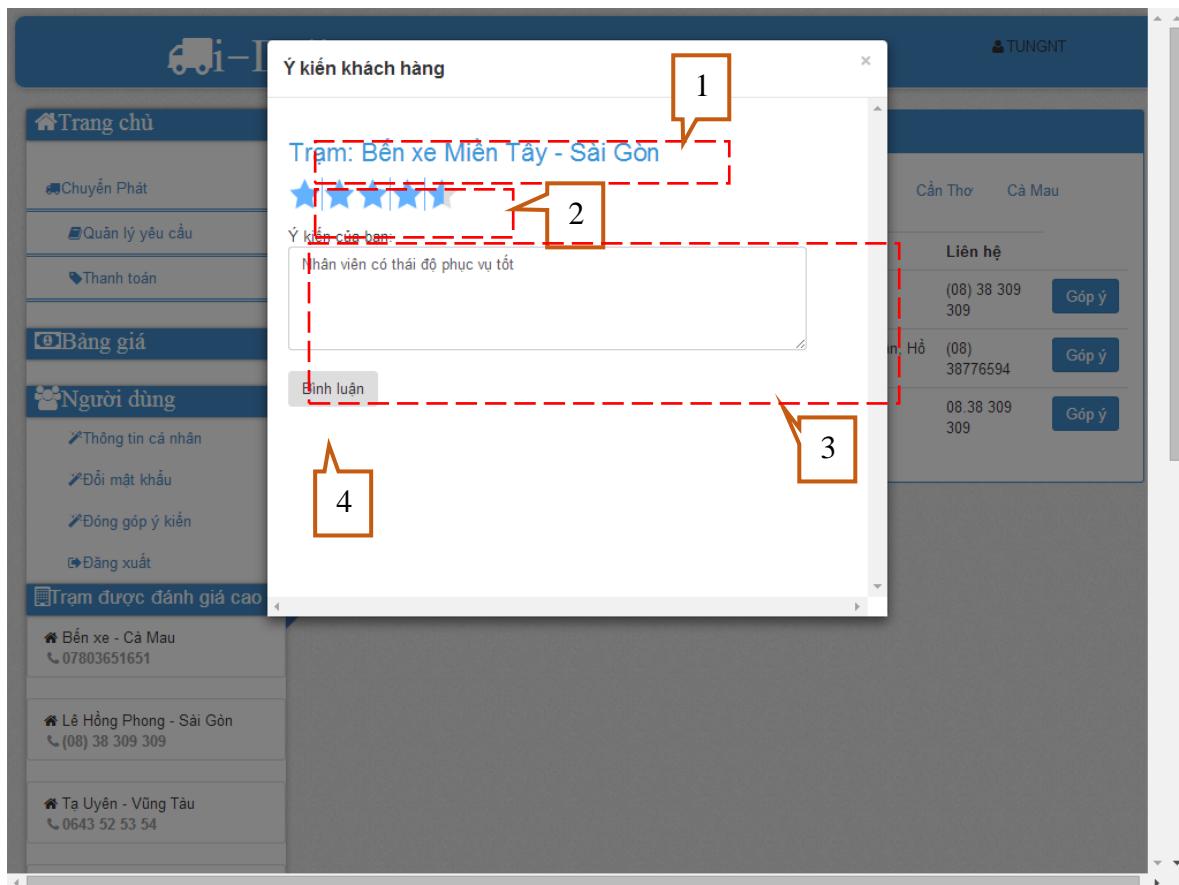


Figure 18. Feedback and review page

No.	Element	Description	Read Only	Mandatory	Control Type	Data type
1.	“Trạm: ...”	Station name to review/feedback	N/A	N/A	N/A	Text
2.	Stars rating bar	Drag able stars bar for rating	N/A	Yes	Adjustable bar	Image
3.	“Ý kiến của bạn” textbox	Textbox for comment	N/A	Yes	Textbox	Text
4.	“Bình luận” button	A button to submit review/feedback	N/A	N/A	Button	N/A

4.6.10. Manage Fee Page

Figure 19. Manage fee page

No.	Element	Description	Read Only	Mandatory	Control Type	Data type
1.	“Tạo mới phí mới” link	A link to define new fee value	N/A	N/A	Link	Text
2.	Fee, min to max weight and min to max volume	Fee correlative to weight and volume	N/A	N/A	N/A	Text
3.	“Thao tác” links	Links to edit, detail and delete page of each fee level	N/A	N/A	Link	Text

4.7. Database Design

4.7.1. 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.

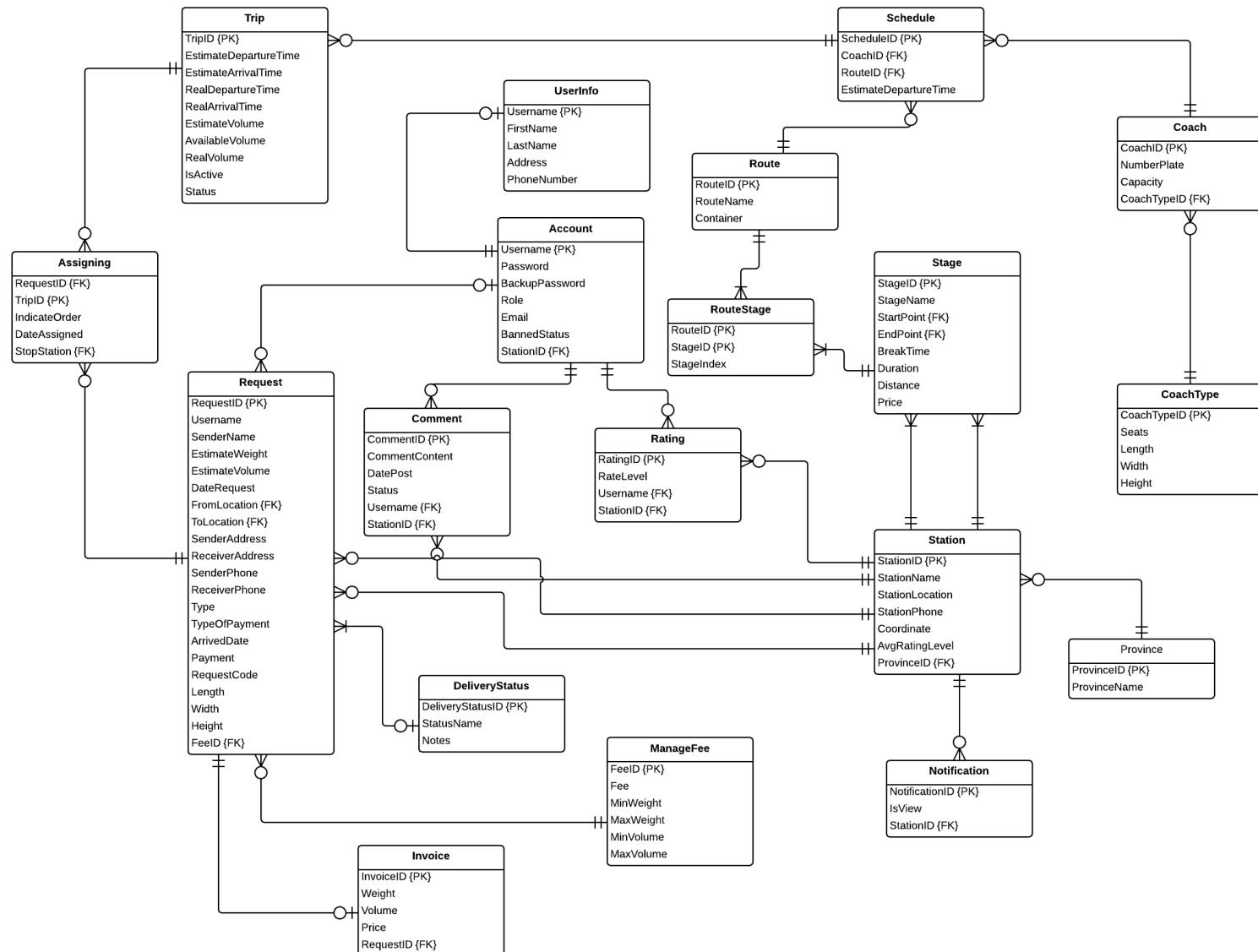


Figure 20. Logical database design

The details of each relation are specified below

4.7.1.1. User

Column Name	Type	Allow Null	Default Value	P/F Key	Description
Username	int		-	PK	Username of user
Password	nvarchar(50)		-		Password of user
BackupPassword	nvarchar(50)		-		Backup password for user
Role	nvarchar(50)		-		Role of user in the system: customer, admin or staff
Email	nvarchar(50)		-		Email of user
Phone	nvarchar(50)		-		Phone of user
BannedStatus	bit		-		User is banned or not
StationID	int		11	FK	The station of the staff belong to (just for staff account)

Unique: Username.

Foreign key: StationID.

4.7.1.2. UserInfo

Column Name	Type	Allow Null	Default Value	P/F Key	Description
Username	nvarchar(50)		-	FK	Username of user registered to the system or the staff
Firstname	nvarchar(50)		-		Firstname of user registered to the system or the staff
Lastname	nvarchar(50)		-		Lastname of user registered to the system or the staff
Address	nvarchar(500)		-		Address of user registered to the system or the staff

Unique: Username.

Foreign key: N/A.

4.7.1.3. Coach

Column Name	Type	Allow Null	Default Value	P/F Key	Description
CoachID	int		Auto generated	PK	Uniquely identifies of the coach
NumberPlate	nvarchar(50)		-		Number plate of coach
CoachTypeID	int		-	FK	The type of coach

Unique: CoachID, NumberPlate.

Foreign key: CoachTypeID (table CoachType).

4.7.1.4. CoachType

Column Name	Type	Allow Null	Default Value	P/F Key	Description
CoachTypeID	int		Auto generated	PK	Uniquely identifies of the coach
Seats	int		-		The number of seats of the coach, that define the type of coach
Capacity	float				The percentage of the cabin that coach can be serving.

Unique: Seats.

Foreign key: N/A.

4.7.1.5. Route

Column Name	Type	Allow Null	Default Value	P/F Key	Description
RouteID	int		Auto generated	PK	Uniquely identifies a route
RouteName	nvarchar(50)		-		The name of route

Unique: RouteID.

Foreign key: StartPoint (table Station), EndPoint (table Station).

4.7.1.6. Trip

Column Name	Type	Allow Null	Default Value	P/F Key	Description
TripID	int		Auto	PK	Uniquely identifies a

			generated		route
EstimateDepartureTime	time		-		The estimate time for the trip departures in each day
RealDepartureTime	time		-		The real time that the trip departures in a specific day
EstimateArrivalTime	time		-		The estimate time for the trip departures in each day
RealArrivalTime	float				The real time for the trip arrival to the station in a specific day
AvailableVolume	float				The volume that the coach on this trip can deliver
Date	date				The date that trip start.
RouteID				FK	The route that the trip traveling on
CoachID				FK	The coach of this trip.

Unique: RouteID.

Foreign key: RouteID (table Route), CoachID (table Coach).

4.7.1.7. Station

Column Name	Type	Allow Null	Default Value	P/F Key	Description
StationID	int		Auto generated	PK	Uniquely identifies a route
StationName	nvarchar(50)		-		The name of the station
StationLocation	nvarchar(50)		-		The address of location
BreakTime	float		-		The default break time that for each trip come to this station
StationPhone	nvarchar(50)		-		The telephone number of this station
Description	nvarchar(500)		-		The description for this station

Longitude	float		-		The longitude of the station location
Latitude	float		-		The latitude of the station location
Province	nvarchar		-		The province of the station location

Unique: StationID.

Foreign key: N/A.

4.7.1.8. Schedule

Column Name	Type	Allow Null	Default Value	P/F Key	Description
ScheduleID	int		Auto generated	PK	Uniquely identifies the volume
CoachID	Int		-		Uniquely identifies the coach used to scheduling
RouteID	Int		-		Uniquely identifies the route to be scheduled
EstimateDepartureTime	time		-		The minimum volume the coach can be deliver on the route
EstimateArrivalTime	time		-	FK	The route in this volume define

Unique: N/A.

Foreign key: CoachID (table Coach), RouteID (table Route).

4.7.1.9. Request

Column Name	Type	Allow Null	Default Value	P/F Key	Description
ScheduleID	int		Auto generated	PK	Uniquely identifies the volume
ArrangeID	float		-		The maximum volume the coach can be deliver on the route
EstimateDepartureTime	time		-		The minimum volume the coach can be deliver on the route

EstimateArrivalTime	time		-	FK	The route in this volume define
---------------------	------	--	---	----	---------------------------------

Unique: N/A.

Foreign key: ArrangeID (table CoachArrangement).

4.7.1.10. DeliveryStatus

Column Name	Type	Allow Null	Default Value	P/F Key	Description
DeliveryStatusID	Int		Auto generated	PK	Uniquely identifies the delivery status
StatusName	nvarchar(50)		-		The name of delivery status of the package
Note	nvarchar(500)	X	-		Some more description for the status

Unique: DeliveryStatusID.

Foreign key: N/A.

4.7.1.11. Invoice

Column Name	Type	Allow Null	Default Value	P/F Key	Description
InvoiceID	Int		Auto generated	PK	Uniquely identifies an invoice
Weight	float	X	-		Weight of package
Volume	int	X	-		Volume of package
Price	float		-		Price of delivery service
RequestID	int		-	FK	The request that the invoice point to

Unique: InvoiceID.

Foreign key: RequestID (table Request).

4.7.1.12. ManageFee

Column Name	Type	Allow Null	Default Value	P/F Key	Description
FeeID	int		Auto generated	PK	Uniquely identifies a fee rule
Fee	float		-		The fee of this weight and volume range

MinWeight	float		-		The minimum weight of the weight range
MaxWeight	float		-		The maximum weight of the weight range
MinVolume	float		-		The minimum volume of the volume range
MaxVolume	float		-		The maximum volume of the volume range

Unique: FeeID.

Foreign key: N/A.

4.7.1.13. Comment

Column Name	Type	Allow Null	Default Value	P/F Key	Description
CommentID	int		Auto generated	PK	Uniquely identifies the comment
Username	nvarchar(50)		-	FK	The user post comment
CommentContent	nvarchar(4000)		-		Content of comment
DatePost	date		-		Date that user post the comment
Status	bit		-		Approve or reject the comment
StationID	int		-	FK	The station that customer comment for

Unique: CommentID.

Foreign key: Username (table User), StationID (table Station).

4.7.1.14. Rating

Column Name	Type	Allow Null	Default Value	P/F Key	Description
RatingID	int		Auto generated	PK	Uniquely identifies the rating
Username	nvarchar(50)		-	FK	The user post comment
RateLevel	int		-		The level of rate: from 1 to 5
StationID	int		-	FK	The station that user rating for

Unique: RatingID.

Foreign key: Username (table User), StationID (table Station).

4.7.1.15. Assigning

Column Name	Type	Allow Null	Default Value	P/F Key	Description
CoachID	int		-	PK	The ID of coach for the specific request
RequestID	int		-		The ID of request
IndicateOrder	int		-		The indicated order of the assigning the package of this request

Unique: (CoachID, RequestID).

Foreign key: N/A.

4.7.1.16. Stage

Column Name	Type	Allow Null	Default Value	P/F Key	Description
CoachID	int		-	PK	The ID of coach
StartPoint	int		-	FK	The start position (station) of the stage
EndPoint	int		-	FK	The end position (station) of the stage
Duration	float		-		The estimate duration (in hours) for moving on this route
Distance	float		-		The estimate distance (in km) between start and end location
Container	float		-		The percentage of coach that allow to travel on the stage
Price	float		-		The fee for travel on the stage

Unique: N/A.

Foreign key: StartPoint, EndPoint.

4.7.1.17. RouteStage

Column Name	Type	Allow Null	Default Value	P/F Key	Description

RouteID	int		-	PK	The ID of route
StageID	int		-		The ID of stage
StageIndex	int		-		The index of stage in the list stage of the route

Unique: N/A.

Foreign key: N/A.

4.7.1.18. Notification

Column Name	Type	Allow Null	Default Value	P/F Key	Description
ID	int		-	PK	The index of notification
Username	Int		-		Username of user
StationID	int		-		The ID of station the user mange's
isView	bit		-		Status of notification, it is seen by user or not.

Unique: N/A.

Foreign key: StationID, Username.

4.7.1.19. Province

Column Name	Type	Allow Null	Default Value	P/F Key	Description
ProvinceID	int		-	PK	The index of province
ProvinceName	nvarchar(50)		-		Name of province

Unique: N/A.

Foreign key: N/A.

4.7.2. Physical database design

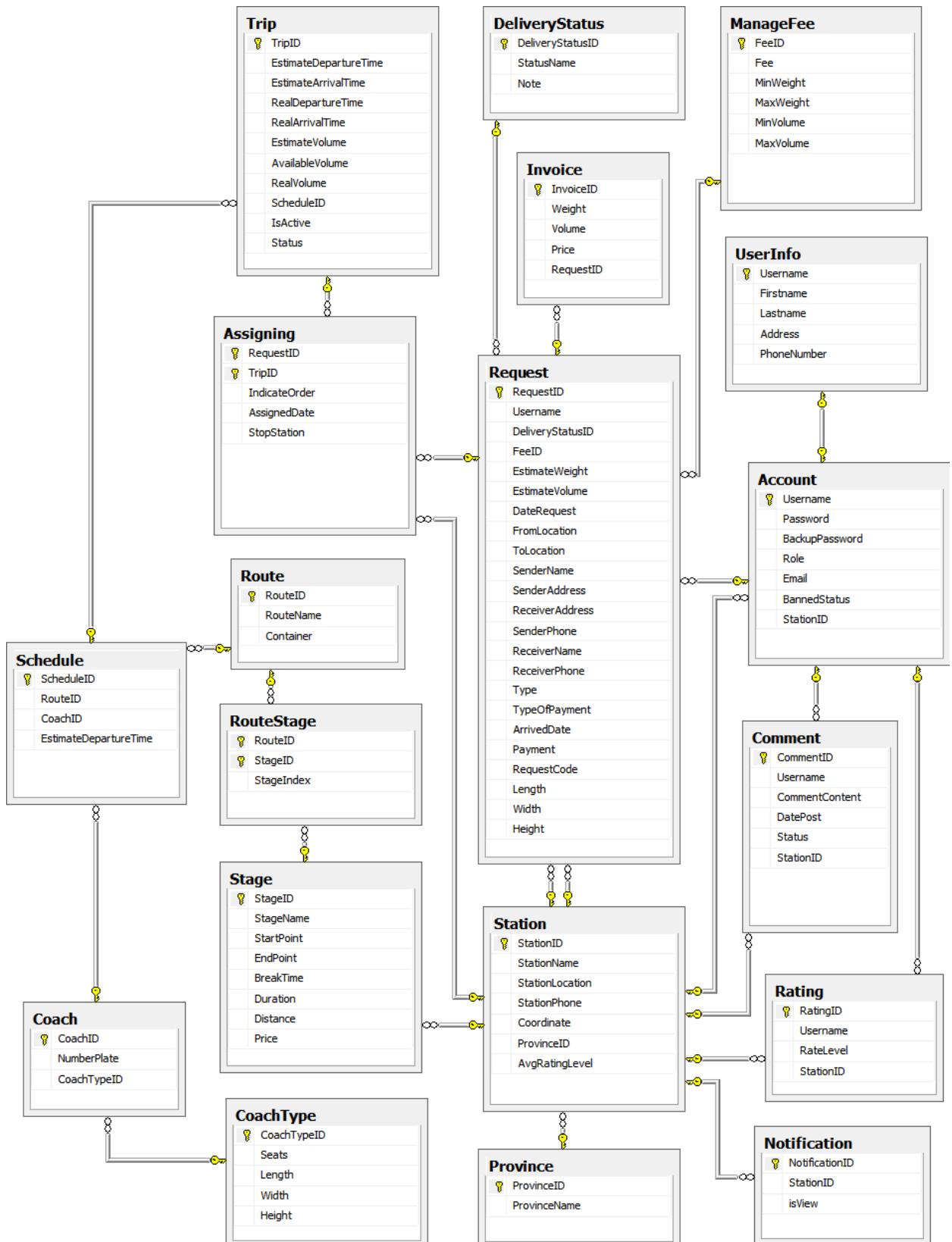
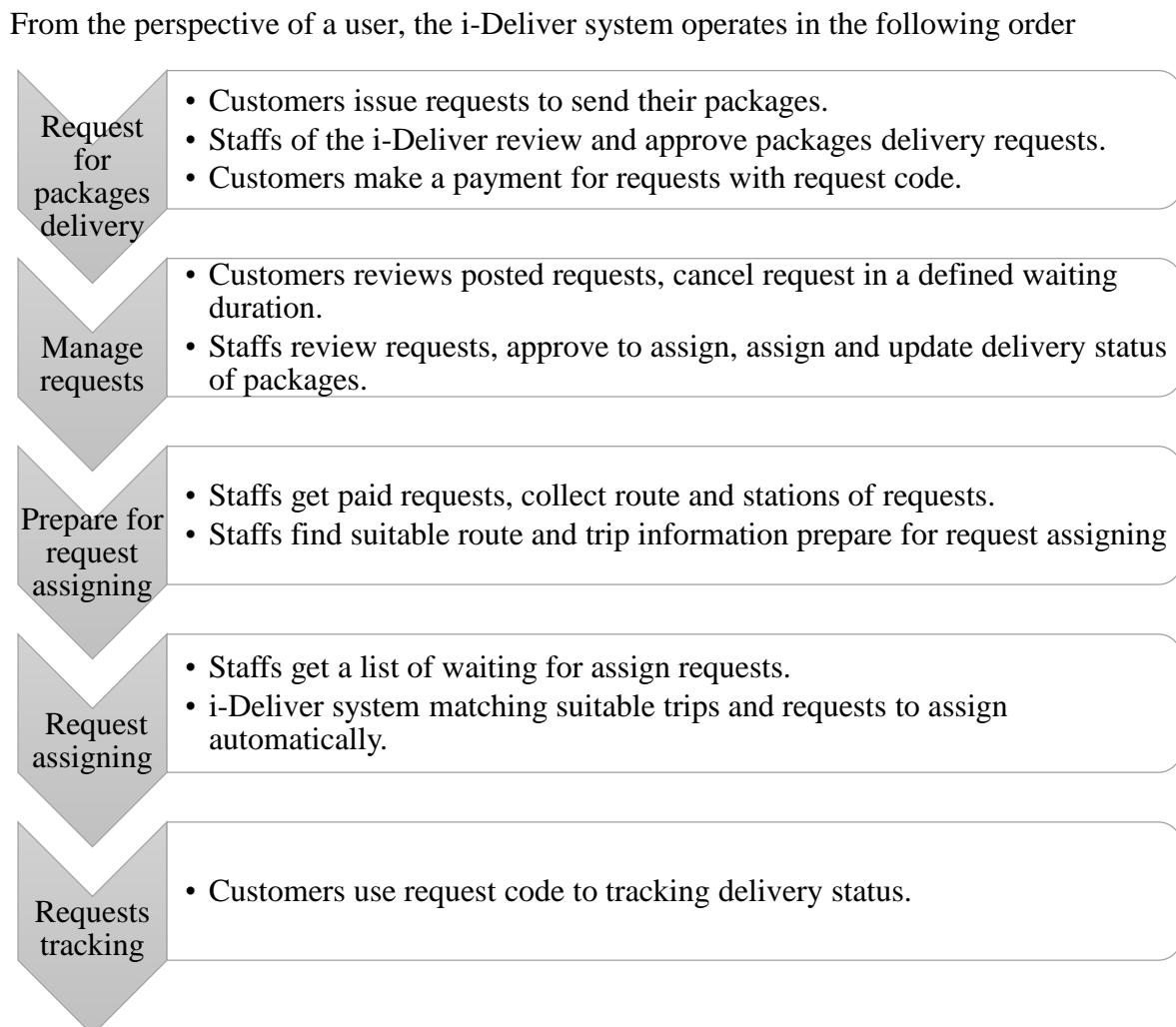


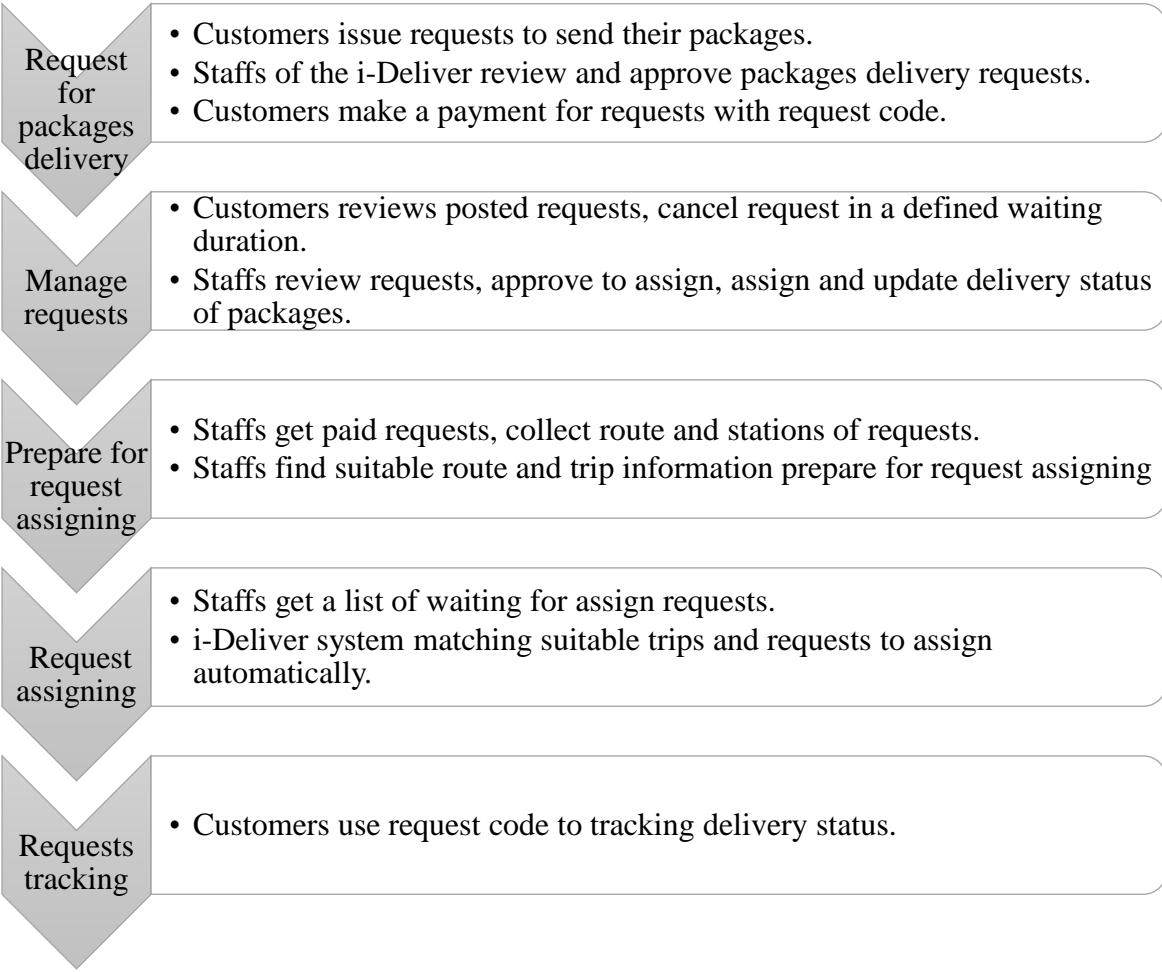
Figure 21. Physical database design

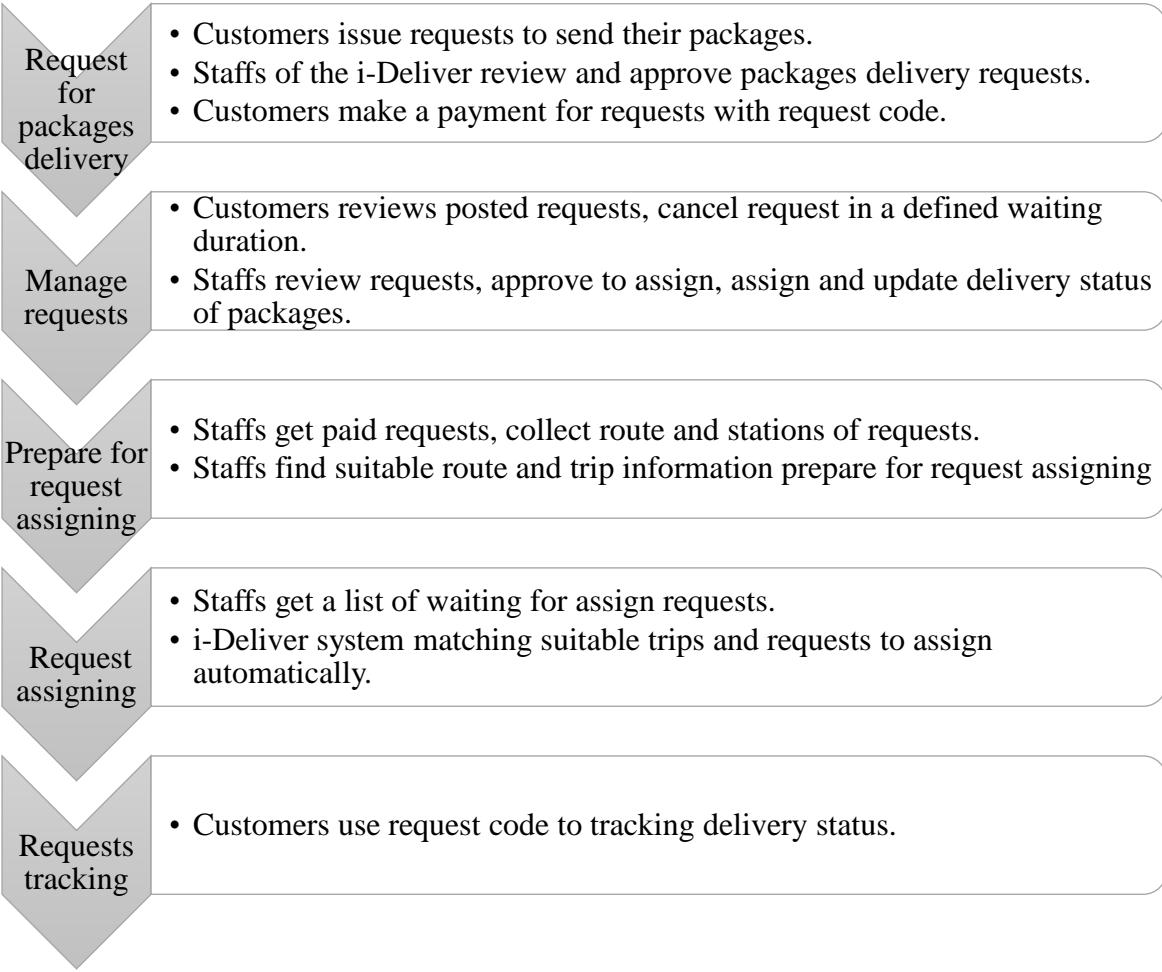
5. Software Test Document

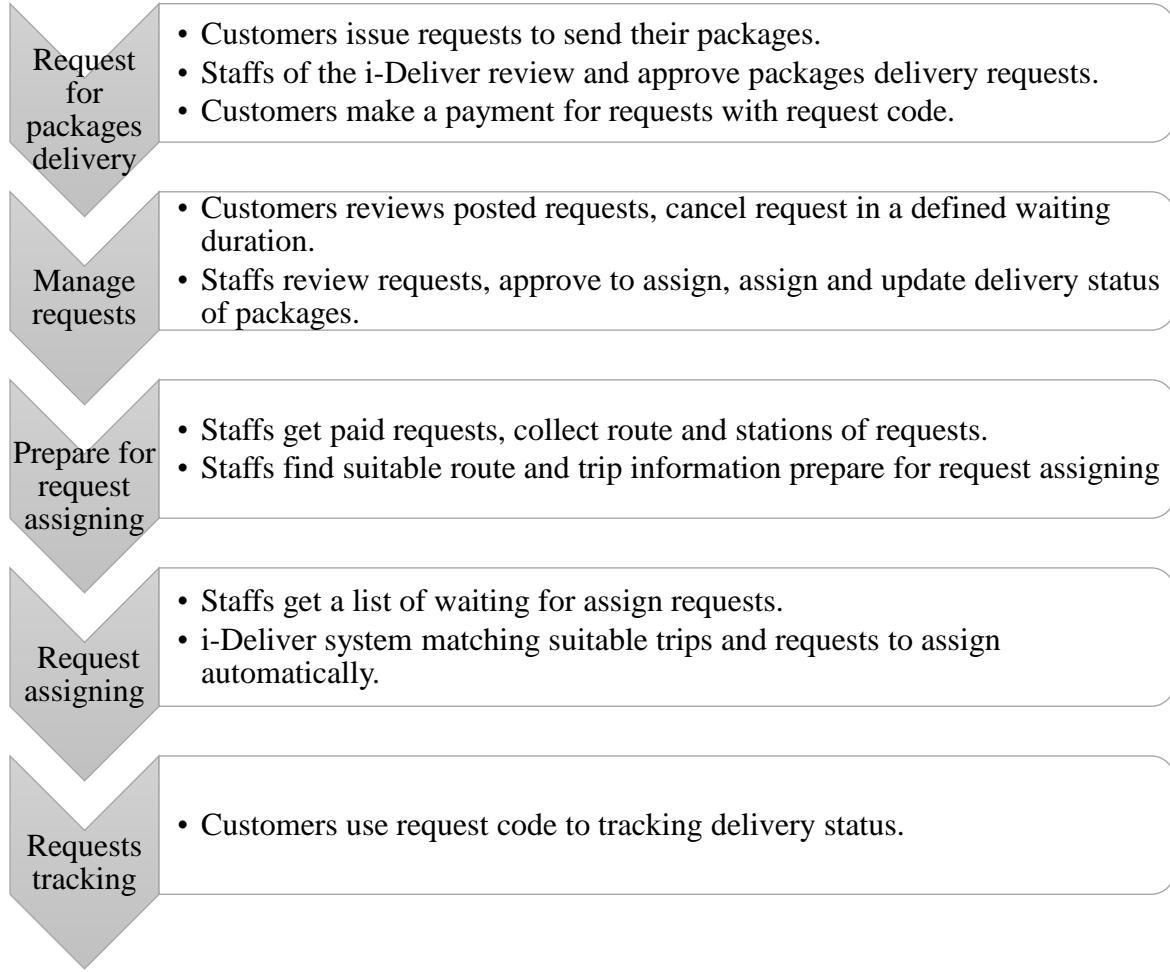
5.1. Introduction

5.1.1. System Overview









5.1.2. Test Approach

The system will be tested following the normal testing approach for a software project, including for major stages: Unit testing, Integration testing, System testing, and Acceptance testing.

Firstly, each member will be responsible for *unit testing* all the modules he or she developed. The other team members will later approve of the modules developed by verifying that they comply with the detailed designs.

Secondly, the team leader will perform *integration testing* for the whole application with the assistance of the other team members.

Next, all of the team members will conduct *system testing* to make sure that the system is consistent with the requirements initially specified.

Finally, the supervisor of the project will be accountable for *acceptance testing*, with the assistance of all team members.

5.2. Test Plan

5.2.1. Features to be tested

The following features will be system-tested and the results are specified in this test document. The order of the tested features is listed in the system main flow.

No.	Feature	Description	Assigned
1.	Send new package delivery request	Test if the customers can post new package delivery request easily	TungNT
2.	Online payment	Test if the dashboard page for staffs and administrators is displayed correctly	LongDB
3.	Tracking packages	Test if the customers can track delivery status of packages	TungNT
4.	Feedback and review	Test if the customers can write reviews for stations	DucNT
5.	Approve package delivery request	Test if the staffs can approve package delivery request issued by a customer	TungNT
6.	Assign packages		
7.	Prepare for packages return	Test if the customers can search routes information.	TungNT
8.	Create a trip		
9.	Create a schedule	Test if packages can be returned if it can't be delivered to receiver	LongDB
10.	Monitor price/fee	Test if the delivery fee can be easily and accurately monitored	LuLP

5.2.2. Features not to be tested

The following features will not be tested.

- For guests and travelers: Change password, Update account information, View past requests.
- For staffs and system admins: Add a coach, Edit a coach, Delete a coach, Add a coach type, Edit a coach type, Delete a coach type, Add a station, Edit a station, Delete a station, Add a route, Edit a route, Delete a route, Add a schedule, Edit a schedule, Delete a schedule, Add a stage, Edit a stage, Delete a stage, Delete a comment, Clear station rating.
- For system administrators: Manage staff accounts

5.3. System Test Cases

5.3.1. Send new delivery request

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Log in the system with Customer role 2. Click on the “Chuyển phát” link in the menu bar 3. Provide the necessary information about the receiver and packages 4. Click the “Hoàn tất” button 						
[SR-1]	Try to post a request with erroneous input	<ul style="list-style-type: none"> ▫ Make one or more validation errors when entering the information for the request. ▫ Click the “Hoàn tất” button. 	The corresponding message will be shown next to the fields that have erroneous input.		Passed	10/04/14	
[SR-2]	Post a request with validated input	<ul style="list-style-type: none"> ▫ Fill in all fields with validated input; ▫ Click the “Hoàn tất” button. 	The request is saved into the database. A delivery request is issued waiting for a Staff to approve of it. The customer is redirected to the request management page.		Passed	10/04/14	

5.3.2. Online payment

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Clicks on the “Thanh toán” link in the menu bar. 2. Select request need to pay on request list, user is redirected to new page with payment information. 3. Clicks on “Check out with PayPal” button to direct to PayPal payment gate. 						
[OP-1]	Payment was successful	<ul style="list-style-type: none"> ▫ User complete payment information at Paypal site; 	<p>The user completed payment information with validate credit card;</p> <p>Paypal returns successful payment.</p>		Passed	10/04/14	
[OP-2]	Payment was not completed	<ul style="list-style-type: none"> ▫ User complete payment information at Paypal site; 	<p>The user completed payment information with invalidate credit card;</p> <p>Paypal cannot charge money and return unsuccessful payment result.</p>		Passed	10/04/14	
[OP-3]	User cancelled payment action	<ul style="list-style-type: none"> ▫ User click on “Hủy bỏ” button 	<p>The user cancelled payment action then they is redirected to request list page</p>		Passed	10/04/14	

5.3.3. Tracking package

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Log in the system with Customer role. 2. Click “Quản lý yêu cầu” link on the left menu bar of Customer main page. 3. Click on “Đã chuyển đi” tab on requests section below Google maps. 						
[TP-1]	Track package successfully	<ul style="list-style-type: none"> ▫ Click on “Theo dõi” button at the end of each request record. 	The corresponding route of request will display on above Google maps where passed road is highlighted with pink, on-route road is highlighted with black and left distance with red.		Passed	13/04/13	
[TP-2]	Track package unsuccessfully	<ul style="list-style-type: none"> ▫ Click on “Theo dõi” button at the end of each request record. 	The Internet connection has problems so Google maps cannot render route of package.		Passed	13/04/13	
[TP-3]	Track package unsuccessfully	<ul style="list-style-type: none"> ▫ Requests list have no record to tracking 	The logged in account has unapproved requests, late payment requests or no posted request		Passed	13/04/13	

5.3.4. Feedback and review

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Log in the system with Customer role. 2. Click “Đóng góp ý kiến” link on the left menu bar of Customer main page. 3. Click on “Góp ý” button of each station in stations section. 						
[FR-1]	Rating and write a review successfully	<ul style="list-style-type: none"> ▫ Rating for station services by checking the number of stars on rating bar. ▫ Fill the content of the review (more than 50 characters in length) in the “Ý kiến của bạn” text area. ▫ Click the “Hoàn tất” button. 	The rating and review is saved and is added at the top of the review list.		Passed	12/04/14	
[FR-2]	Fail to validate the review content	<ul style="list-style-type: none"> ▫ Rating for station services by checking the number of stars on rating bar. ▫ Leave the box “Ý kiến của bạn” empty. ▫ Click the “Hoàn tất” button. 	<p>The message “Bình luận phải có độ dài ít nhất 50 ký tự” is displayed.</p> <p>The rating is not saved to database for selected station</p>		Passed	12/04/14	

5.3.5. Approve delivery requests

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Log in with account granted Staff role. 2. Click on the “Quản lý yêu cầu vận chuyển” link in the menu bar if the log in form does not redirect you to this page. 3. Click on the “Tiếp nhận yêu cầu” link in the menu bar <p>All of new requests which waiting for approved is on the view.</p>						
[AD-1]	Approve requests successfully with validated input	<ul style="list-style-type: none"> ▫ Click on “Nhập hóa đơn” button at the end of each row on the requests table. ▫ Fills in all fields with validated input. ▫ Click on “Xác nhận” button 	<p>The selected request status is updated in database with corresponding invoice.</p> <p>The staff is redirected to assign packages page.</p>		Passed	11/04/14	
[AD-2]	Try to approve a request with erroneous input	<ul style="list-style-type: none"> ▫ Click on “Nhập hóa đơn” button at the end of each row on the requests table. ▫ Make one or more validation errors when entering the information for the invoice. ▫ Click on “Xác nhận” button 	<p>The corresponding message will be shown next to the fields that have erroneous input.</p>		Passed	11/04/14	
[AD-3]	Approve requests unsuccessfully with validated input	<ul style="list-style-type: none"> ▫ Click on “Nhập hóa đơn” button at the end of each row on the requests table. ▫ Fills in all fields with validated input. ▫ Click on “Hủy bỏ” button 	<p>Nothing has changed in database.</p> <p>The current page still keeps on.</p>		Passed	11/04/14	

5.3.6. Assign packages

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Log in with account granted Staff role. 2. Click on the “Quản lý yêu cầu vận chuyển” link in the menu bar if the log in form does not redirect you to this page. 3. Click on the “Xếp lịch vận chuyển” link in the menu bar <p>All of approved requests which waiting for assigned is on the view.</p>						
[AP-1]	Assign packages successfully with validated input	<ul style="list-style-type: none"> ▫ Check box at the end of each row to select which package to assign or check box on the top to select all packages ▫ Click on “Xếp lịch” button to assign selected package(s) 	<ul style="list-style-type: none"> ▫ The current page is redirected to a new page with a list of assigned package(s) and detail of trip that package was assigned to 		Passed	11/04/14	
[AP-2]	Assign packages unsuccessfully	<ul style="list-style-type: none"> ▫ Check box at the end of each row to select which package to assign or check box on the top to select all packages ▫ Click on “Xếp lịch” button to assign selected package(s) 	The current page is redirected to a new page with nothing because assign procedure does not meet condition (trips/schedules is not available)		Passed	11/04/14	
[AP-3]	Assign packages unsuccessfully	<ul style="list-style-type: none"> ▫ Click on “Xếp lịch” button without select package to assign 	The current page is redirected to a new page with nothing because none of packages were selected.		Passed	11/04/14	

5.3.7. Prepare for packages return

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Log in to the system with System Admin role. 2. Click on the “Hàng đã đến trạm” link in the left menu bar. <p>All of requests have arrived in end-station but late for delivery now waiting for return or cancel.</p>						
[PR-1]	Successfully prepare for packages return	<ul style="list-style-type: none"> ▫ Click on “Trả hàng” button at the end of each row on the packages table. ▫ A popup dialog is displayed with confirm message “Bạn muốn gửi trả lại món hàng này?” ▫ Click on “Xác nhận” button of confirm popup dialog. 	<p>The package has selected for return is updated in database.</p> <p>The staff is redirected to requests waiting for assign page.</p>		Passed	11/04/14	
[PR-2]	Fail to prepare for packages return	<ul style="list-style-type: none"> ▫ Click on “Trả hàng” button at the end of each row on the packages table. ▫ A popup dialog is displayed with confirm message “Bạn muốn gửi trả lại món hàng này?” ▫ Click on “Hủy bỏ” button of confirm popup dialog. 	<p>Nothing has changed in database.</p> <p>The current page still keeps on.</p>		Passed	11/04/14	

5.3.8. Create the trip

Test case	Test Case	Test Case Procedure	Expected Output	Inter-Test Case	Result	Test Date	Note
-----------	-----------	---------------------	-----------------	-----------------	--------	-----------	------

ID	Description			Dependence				
	Main Steps: 1. Log in the system with Staff role. 2. Click on the “Quản lý chuyến xe” link in the left menu bar							
[CT-1]	Create trip successfully with validated input	<ul style="list-style-type: none"> ▫ Select a date by date picker ▫ Select which route to create trip by click on dropdown list under date picker. ▫ Check box at the start of each row in the schedules table to choose schedule for trip or check box on top header of schedules table to select all displayed schedules. ▫ Click on “Tạo chuyến mới” button 	<p>The trip is saved to database. The current page is redirected to trip list page with new trip records on the top of list.</p>		Passed	11/04/14		
[CT-2]	Try to create a trip with erroneous input	<ul style="list-style-type: none"> ▫ Make one or more validation errors when click the “Tạo chuyến mới” button without select date or route or schedule 	<p>The corresponding message will be shown next to the fields that have erroneous input.</p>		Passed	11/04/14		
[CT-3]	Create trip unsuccessfully	<ul style="list-style-type: none"> ▫ Select a date by date picker ▫ Select which route to create trip by click on dropdown list under date picker. ▫ Check box at the start of each row in the schedules table to choose schedule for trip or check box on top header of 	<p>The trip is not saved to database. The current page is back to trip list page.</p>		Passed	11/04/14		

		schedules table to select all displayed schedules. ▫ Click on “Hủy bỏ” button					
--	--	--	--	--	--	--	--

5.3.9. Create schedules

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	Main Steps: 1. Log in the system with Staff role. 2. Click on the “Quản lý lịch trình” link in the left menu bar						
[CT-1]	Create schedule successfully with validated input	<ul style="list-style-type: none"> ▫ Select a route from dropdown list ▫ Select a coach from dropdown list ▫ Fills in textbox of “Giờ đi dự kiến” and “Giờ đến dự kiến” with valid timespan value ▫ Click on “Tạo mới” button 	<p>The schedule is saved to database.</p> <p>The current page is redirected to schedule list page with new schedule records on the top of list.</p>		Passed	11/04/14	
[CT-2]	Try to create a schedule with erroneous input	<ul style="list-style-type: none"> ▫ Make one or more validation errors when click the “Tạo mới” button without select route/coach or both 	The corresponding message will be shown next to the fields that have erroneous input.		Passed	11/04/14	
[CT-3]	Try to create a schedule with erroneous input	<ul style="list-style-type: none"> ▫ Make one or more validation errors when click the “Tạo mới” button without fills in “Giờ đi dự kiến” or “Giờ đến dự kiến” textbox 	The corresponding message will be shown next to the fields that have erroneous input.		Passed	11/04/14	

[CT-4]	Create schedule unsuccessfully	<ul style="list-style-type: none"> ▫ Select a route from dropdown list ▫ Select a coach from dropdown list ▫ Fills in textbox of “Giờ đi dự kiến” and “Giờ đến dự kiến” with valid timespan value ▫ Click on “Hủy bỏ” button 	The current page is back to schedules list page.		Passed	11/04/14	
--------	--------------------------------	--	--	--	--------	----------	--

5.3.10. Monitor price/fee

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
		<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Go the the “Hủy đặt phòng” link. 2. Choose a booking you have made or enter your booking code in the “Mã xác nhận” box. 3. Click the “Gửi yêu cầu hủy phòng” button. 					
[RR-1]	Successfully request for a refund	<ul style="list-style-type: none"> ▫ Fill the generated code of booking that has been approved, not expired and not refunded before in the “Mã xác nhận” box. ▫ Click the “Gửi yêu cầu hủy phòng” button. 	The message “Bạn đã gửi yêu cầu thành công, chúng tôi sẽ xử lý và thông báo trong email cho bạn trong thời gian sớm nhất. Vui lòng kiểm tra e-mail để xác nhận.” is displayed.		Passed	19/11/2013	
[RR-2]	Fail to request for a refund	<ul style="list-style-type: none"> ▫ Fill the generated code of a booking that has been refunded before or expired in the “Mã xác nhận” box. ▫ Click the “Gửi yêu cầu hủy 	The message “Thông tin bạn nhập không chính xác, đã tồn tại hoặc đã quá hạn. Vui lòng thử lại.” is displayed.		Passed	19/11/2013	

		phòng” button.					
[RR-3]	Fail to request for a refund	<ul style="list-style-type: none"> ▫ Leave the “Mã xác nhận” textbox empty. ▫ Click the “Gửi yêu cầu hủy phòng” button. 	The message box “Mã xác nhận phải là kí tự chữ hoặc số có 6 kí tự” is displayed.		Passed	19/11/2013	

5.3.11. Monitor price/fee

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	Main Steps:	1. Log in to the system with System Admin role. 2. Clicks on the “Phí vận chuyển” link in the menu bar. 3. Change price/fee, weight and volume value to updates it information.					
[MP-1]	Modify/update price, volume or weight with validated input	<ul style="list-style-type: none"> ▫ Fills in all textboxes with valid price, volume or weight values. ▫ Click the “Hoàn tất” button. 	Modified/updated information is saved to database. The system admin is redirected to list of fee levels page.		Passed	24/11/2013	
[MP-2]	Try to modify/update price, volume or weight with erroneous input	<ul style="list-style-type: none"> ▫ Make one or more validation errors when entering the information for price, volume or weight. ▫ Click the “Hoàn tất” button. 	The corresponding message will be shown next to the fields that have erroneous input.		Passed	10/04/14	

5.3.12. Create staff account

Test case ID	Test Case Description	Test Case Procedure	Expected Output	Inter-Test Case Dependence	Result	Test Date	Note
	<p>Main Steps:</p> <ol style="list-style-type: none"> 1. Log in the system with System Admin role. 2. Click on the “Quản lý nhân viên” link in the left menu bar 						
[CT-1]	Create staff account successfully with validated input	<ul style="list-style-type: none"> ▫ Fill in all fields with validated input; ▫ Click the “Tạo mới” button. 	<p>The new staff account is saved to database.</p> <p>The system admin is redirected to staff accounts list page with new account records on the top of list.</p>		Passed	11/04/14	
[CT-2]	Try to create staff account with erroneous input	<ul style="list-style-type: none"> ▫ Make one or more validation errors when click the “Tạo mới” button without fill in all fields or fill in all fields with erroneous input 	<p>The corresponding message will be shown next to the fields that have erroneous input.</p>		Passed	11/04/14	
[CT-3]	Create staff account unsuccessfully	<ul style="list-style-type: none"> ▫ Fill in all fields with validated input; ▫ Click the “Hủy bỏ” button. 	<p>Nothing has changed in database.</p> <p>The current page is back to staff accounts list page.</p>		Passed	11/04/14	

5.4. Checklists

5.4.1. Checklist of Validation

	Question	Yes	No	N/A
1	Does a failure of validation on every field cause a sensible user error message?	<input checked="" type="checkbox"/>		
2	Is the user required to fix entries which have failed validation tests?	<input checked="" type="checkbox"/>		
3	Have any fields got multiple validation rules and if so are all rules being applied?	<input checked="" type="checkbox"/>		
4	If the user enters an invalid value and clicks on the OK button (i.e. does not TAB off the field) is the invalid entry identified and highlighted correctly with an error message?		<input checked="" type="checkbox"/>	
5	Is validation consistently applied at screen level unless specifically required at field level?		<input checked="" type="checkbox"/>	
6	For all numeric fields check whether negative numbers can and should be able to be entered.	<input checked="" type="checkbox"/>		
7	For all numeric fields check the minimum and maximum values and also some mid-range values allowable?	<input checked="" type="checkbox"/>		
8	For all character/alphanumeric fields check the field to ensure that there is a character limit specified and that this limit is exactly correct for the specified database size?		<input checked="" type="checkbox"/>	
9	Do all mandatory fields require user input?	<input checked="" type="checkbox"/>		
10	If any of the database columns don't allow null values then the corresponding screen fields must be mandatory. (If any field which initially was mandatory has become optional then check whether null values are allowed in this field.)	<input checked="" type="checkbox"/>		

5.4.2. Submission Checklist

	Question	Yes	No	N/A
TABLES AND FIGURES				
1	Does every table column, including the stub column, have a heading?	<input checked="" type="checkbox"/>		
2	Are all tables referred to in text?		<input checked="" type="checkbox"/>	
3	Are the elements in the figures large enough to remain legible after the figure has been reduced to the width of a journal column or page?		<input checked="" type="checkbox"/>	
4	Is lettering in a figure no smaller than 8 points and no larger than 14 points?		<input checked="" type="checkbox"/>	
5	Are the figures being submitted in a file format acceptable to the publisher?		<input checked="" type="checkbox"/>	
6	Has the figure been prepared at a resolution sufficient to produce a high-quality image?		<input checked="" type="checkbox"/>	

7	Are all figures numbered consecutively with Arabic numerals?	<input checked="" type="checkbox"/>		
8	Are all figures and tables mentioned in the text and numbered in the order in which they are mentioned?	<input checked="" type="checkbox"/>		

REFERENCES

1	Are references cited both in text and in the reference list?	<input checked="" type="checkbox"/>		
2	Do the text citations and reference list entries agree both in spelling and in date?		<input checked="" type="checkbox"/>	
3	Are journal titles in the reference list spelled out fully?	<input checked="" type="checkbox"/>		
4	Are the references (both in the parenthetical text citations and in the reference list) ordered alphabetically by the authors' surnames?		<input checked="" type="checkbox"/>	
5	Are inclusive page numbers for all articles or chapters in books provided in the reference list?		<input checked="" type="checkbox"/>	
6	Are references to studies included in your meta-analysis preceded by an asterisk?		<input checked="" type="checkbox"/>	

FORMAT

1	Have you checked the journal's website for instructions to authors regarding specific formatting requirements for submission?	<input checked="" type="checkbox"/>		
2	Is the entire manuscript—including quotations, references, author note, content footnotes, and figure captions—double-spaced? Is the manuscript neatly prepared?		<input checked="" type="checkbox"/>	
3	Are the margins at least 1 in. (2.54 cm)?		<input checked="" type="checkbox"/>	
4	Are the title pages, abstracts, references, appendices, content footnotes, tables, and figures on separate pages (with only one table or figure per page)? Are the figure captions on the same page as the figures? Are manuscript elements ordered in sequence, with the text pages between the abstract and the references?	<input checked="" type="checkbox"/>		
5	Are all pages numbered in sequence, starting with the title page?	<input checked="" type="checkbox"/>		

PARAGRAPHS AND HEADINGS

1	Is each paragraph longer than a single sentence but not longer than one manuscript page?	<input checked="" type="checkbox"/>		
2	Do the levels of headings accurately reflect the organization of the paper?	<input checked="" type="checkbox"/>		
3	Do all headings of the same level appear in the same format?	<input checked="" type="checkbox"/>		

6. Software User's Manual

6.1. Installation Guide

6.1.1. Prerequisites

Before we install the application and throw it into operation, the preconditions about the environment and tools specified below must be satisfied.

Firstly, system requirements for the i-Deliver application to run on top of are specified below:

- Personal computers for the minimum configuration: 2 Gb of RAM, 50Gb of hard disk, Core 2 Duo 2.0 Ghz;
- Microsoft Windows 7 or higher.
- At least 100MB for the application resources and another 100MB for data storage.

Secondly, these tools must be installed in advance and run without errors:

- IIS Express 7.0 or higher
- SQL Server 2008 or higher,
- .NET Framework 4.5,
- WebDeploy 3.5

6.1.2. Installation procedures

Step 1. Prepare the deployment package

Firstly, you need to publish your application and you'll get folder to add to IIS

Panes	Layout	Current view	Show/hide
Computer > Inconsolable (D:) > iDeliver			
Subversion Videos	bin Content ImageRating Models Scripts Views Global.asax packages.config Web.config	4/16/2014 15:23 PM File folder 4/16/2014 15:23 PM File folder 4/6/2014 17:21 PM ASAX File 1 KB 4/6/2014 17:21 PM CONFIG File 4 KB 4/16/2014 15:26 PM CONFIG File 8 KB	

Figure 22. i-Deliver web application release folder

Step 2. Create the database

Open the file “iDeliver.sql” in the folder “Document” of the deployment package by Microsoft SQL Server Management Studio (MSSQL).

The screenshot shows the SQL Server Management Studio interface. The Object Explorer on the left lists the database structure for 'TUNGNTE60609\SQLEXPRESS'. The 'Tables' node under the 'iDeliver' database contains numerous tables such as 'Account', 'Assigning', 'Coach', 'Comment', 'DeliveryStatus', 'Invoice', 'ManageFee', 'Notification', 'Province', 'Rating', 'Request', 'Route', 'RouteStage', 'Schedule', 'Stage', 'Station', 'Trip', and 'UserInfo'. The 'Script' tab at the top has the following SQL script:

```

USE [master]
GO
/****** Object: Database [iDeliver] Script Date: 04/02/2014 14:21:21 *****/
CREATE DATABASE [iDeliver] ON PRIMARY
( NAME = N'iDeliver', FILENAME = N'c:\Program Files (x86)\Microsoft SQL Server\MSSQL10_50.SQLEXPRESS\MSSQL\DATA\iDeliver.mdf' )
LOG ON
( NAME = N'iDeliver_log', FILENAME = N'c:\Program Files (x86)\Microsoft SQL Server\MSSQL10_50.SQLEXPRESS\MSSQL\DATA\iDeliver.ldf' )
GO
ALTER DATABASE [iDeliver] SET COMPATIBILITY_LEVEL = 100
GO
IF ( 1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled') )
begin
EXEC [iDeliver].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [iDeliver] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [iDeliver] SET ANSI_NULLS OFF
GO
ALTER DATABASE [iDeliver] SET ANSI_PADDING OFF
GO
ALTER DATABASE [iDeliver] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [iDeliver] SET ARITHABORT OFF
GO
ALTER DATABASE [iDeliver] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [iDeliver] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [iDeliver] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [iDeliver] SET AUTO_UPDATE_STATISTICS ON
GO

```

The status bar at the bottom indicates 'Connected. (1/1)', 'TUNGNTE60609\SQLEXPRESS (1...) sa (56) iDeliver 00:00:00', 'Ln 1 Col 1 Ch 1', and 'Ready'.

Figure 23. Create i-Deliver database using SQL Query

Step 3. Install the application

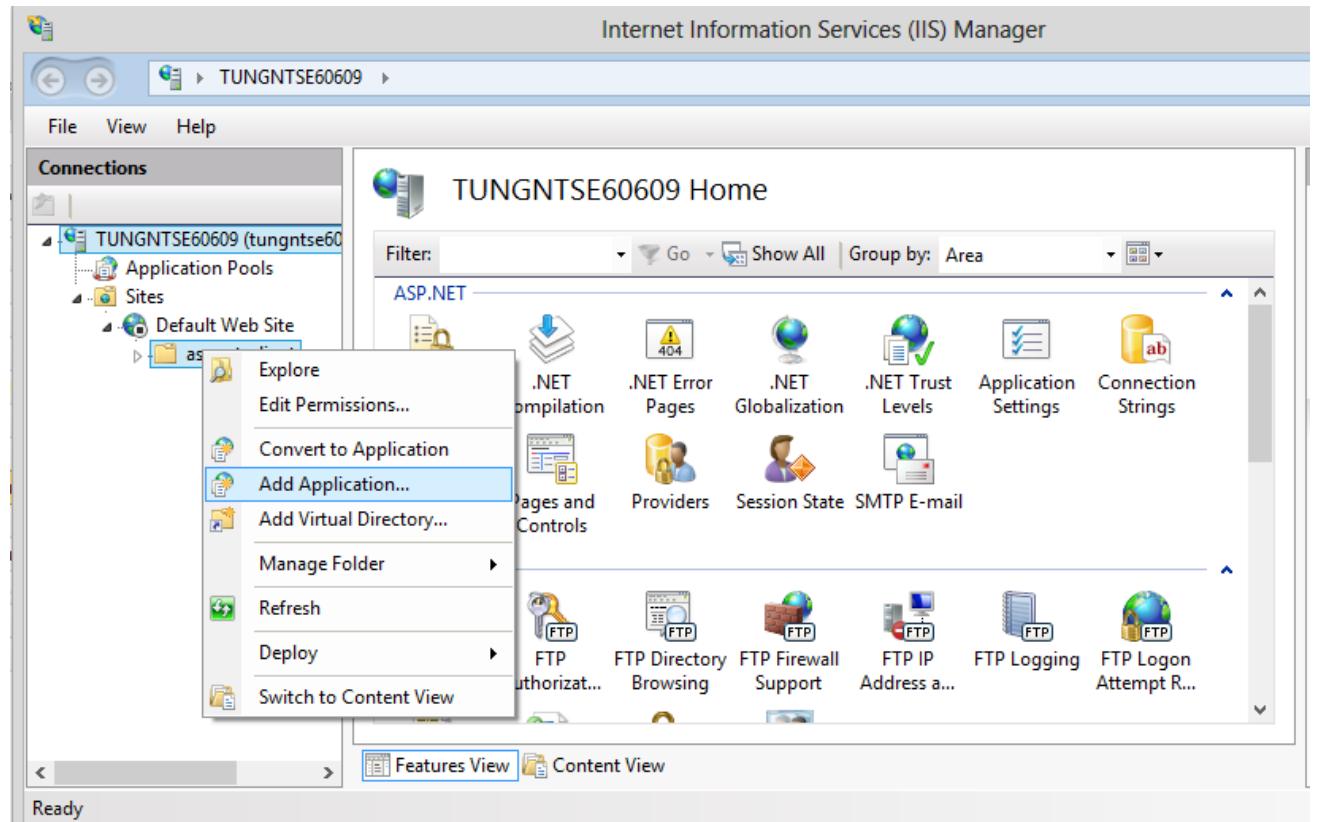
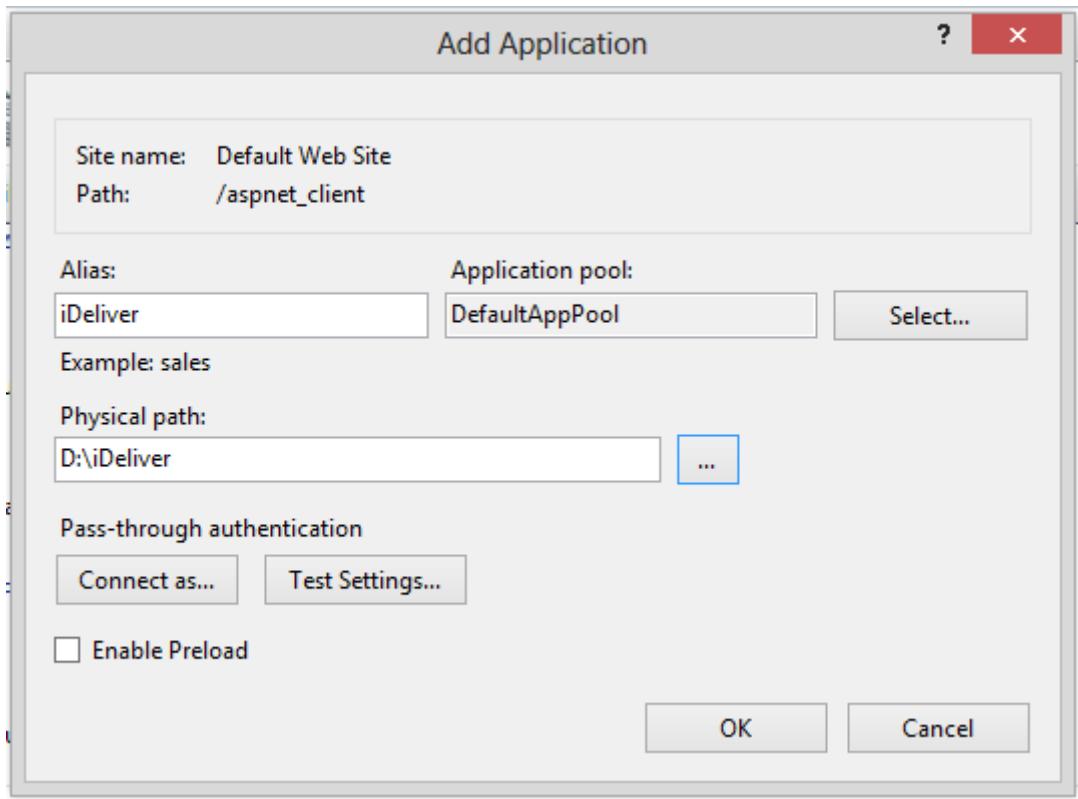


Figure 24. Add new web application

Expand the 'Sites' folder and then right click on 'Default Web Site'. Select "Add Application..." to open the "Add application" window. In the "Add application" window, fill the field "Alias" with the name of the website.



Click the “...” button to browse to the folder that contains the deployment package. Select the “Website” folder in the “iDeliver” folder, in the “Application” folder and click OK.

Verify that the iDeliver Website has been added to “Default Web Site” of IIS Express by right-clicking the website’s name, choosing “Manage Application” and then “Browse”.

Step 4. Configure database connection string

Folder of the deployment package, find and open the file Web.config using Notepad or any other text editor.

```

<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkId=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=5.0.0.0, Culture=neutral, PublicKeyToken=b71d1fba2e7e8c37" />
  </configSections>
  <connectionStrings>
    <add name="DefaultConnection" providerName="System.Data.SqlClient" connectionString="Data Source=TUNGNTSE60609\SQLEXPRESS;Initial Catalog=iDeliver;user id=iDeliver;password=123456" />
    <add name="iDeliverEntities" connectionString="metadata=res://*/Models.iDeliver.csdl|res://*/Models.iDeliver.ssdl|res://*/Models.iDeliver.msl;provider=System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="webpages:Version" value="2.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="PreserveLoginUrl" value="true" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="business" value="lephuclu-facilitator@gmail.com" />
    <add key="IsSandbox" value="true" />
    <add key="currency_code" value="VND" />
    <add key="return" value="http://localhost:5434/PayPal/RedirectFromPaypal" />
    <add key="cancel_return" value="http://localhost:5434/PayPal/CancelFromPaypal" />
    <add key="notify_url" value="http://localhost:5434/PayPal/NotifyFromPaypal" />
    <add key="test_url" value="https://www.sandbox.paypal.com/cgi-bin/webscr" />
    <add key="Prod_url" value="https://www.sandbox.paypal.com/cgi-bin/webscr" />
  </appSettings>

```

Figure 25. Configure database connection string

In the connectionStrings tag, change the Data Source of the connection strings to the SQLServer instance where you installed the database. Also, update the password to access the SQLServer instance in the connection strings.

Now go to http://localhost/aspnet_client/iDeliver and enjoy the website.

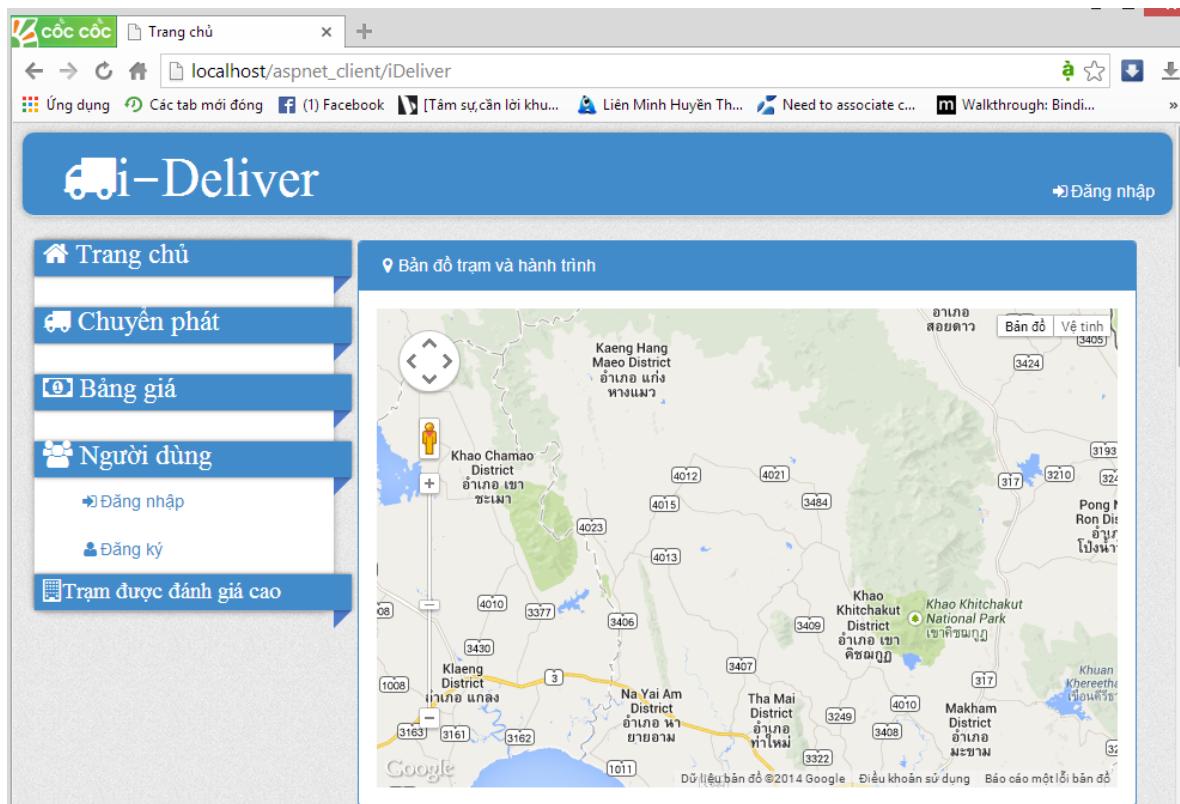


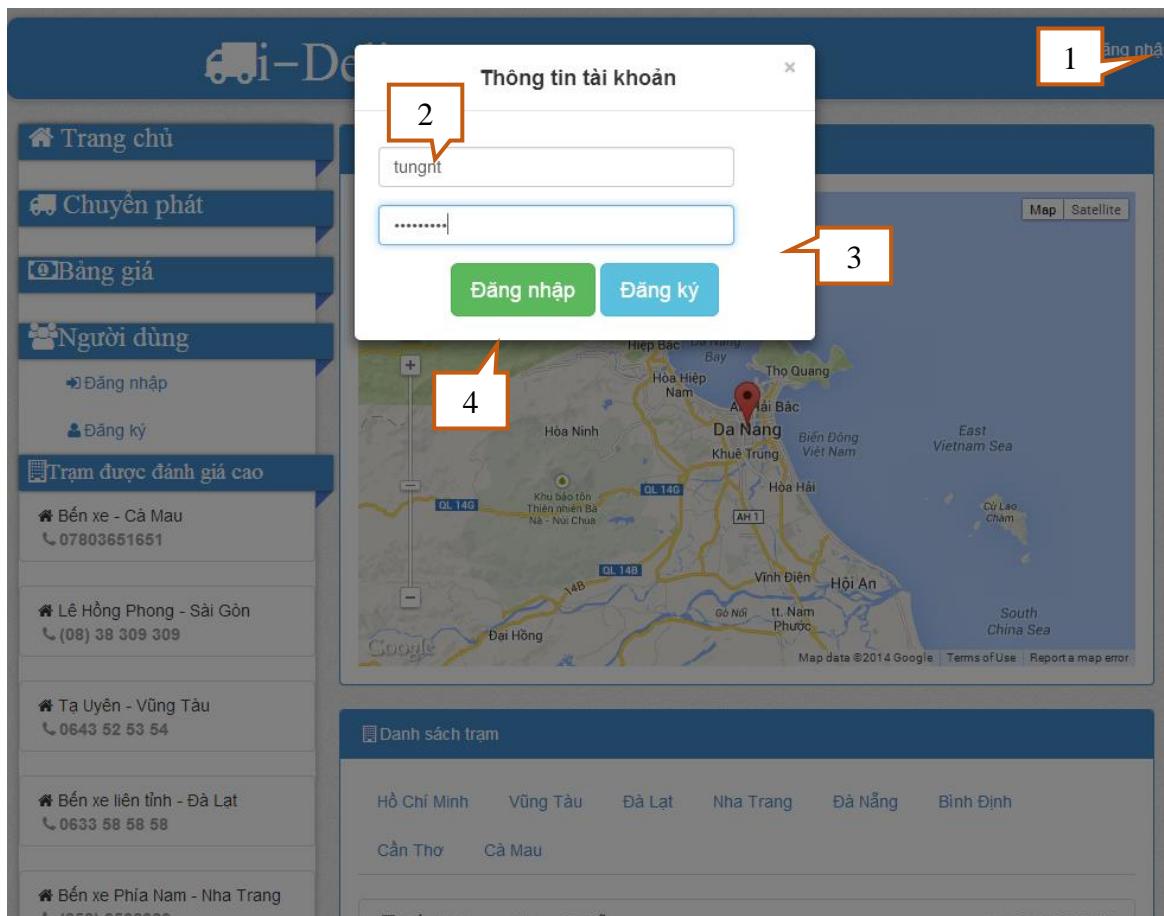
Figure 26. Run i-Deliver website

6.2. User's Guide

6.2.1. Guides for Customer

6.2.1.1. Log in to the system

This function allows a user with a registered account to log in to the i-Deliver system.

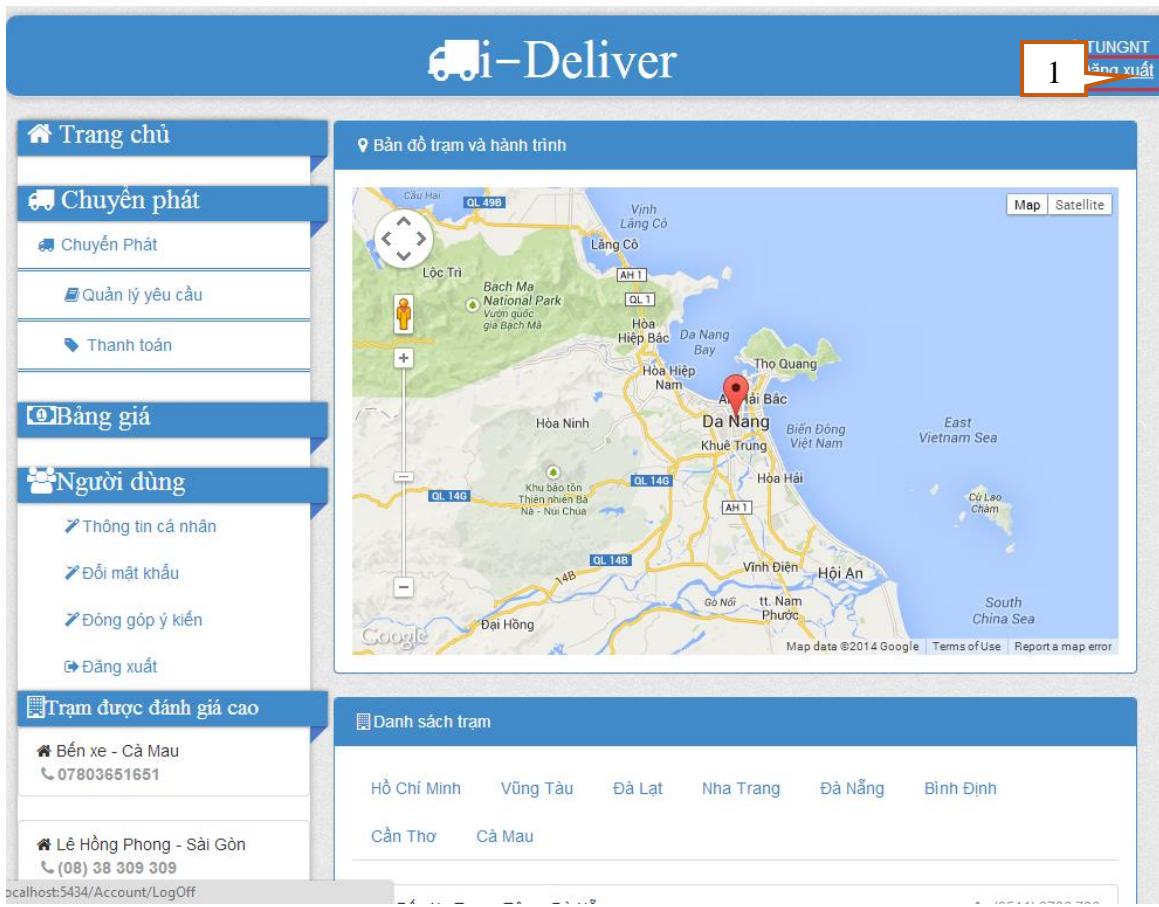


Screen 1. Login screen

Step	Instruction
1.	Click the “Đăng nhập” button on the menu bar of the “Trang chủ” page.
2.	Fill the registered account in the “Tên đăng nhập” field
3.	Fill the password in the “Mật khẩu” field
4.	Click the “Đăng nhập” button to log in to the system. You will be redirected to the page you are viewing

6.2.1.2. Log out of the system

This function allows the user to log out of the system after he or she finishes browsing the website.

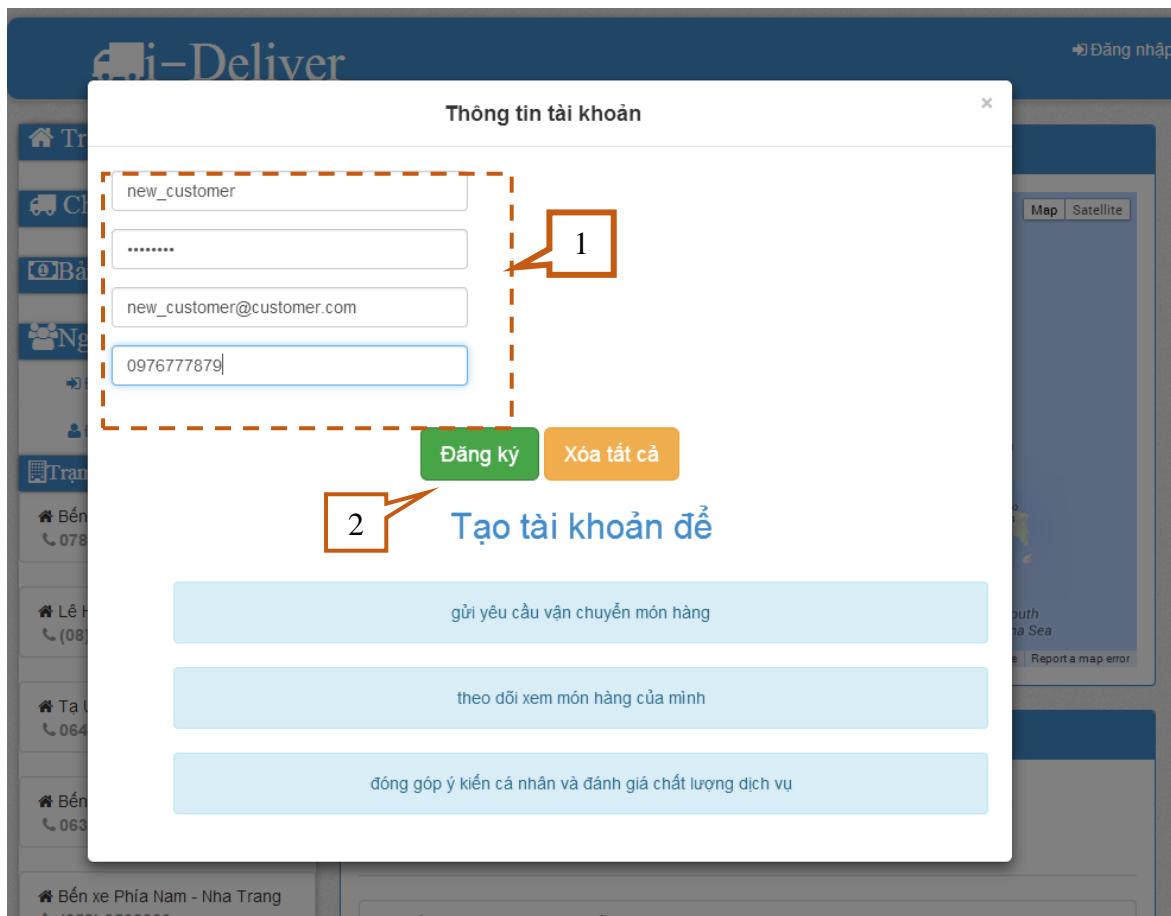


Screen 2. Logout screen

Step	Instruction
1.	Click the “Đăng xuất” link on top right of page you are viewing. You will be redirected to the page you are viewing if this page allows guests to access or the “Trang chủ” page otherwise.

6.2.1.3. Register an account with Customer role

This function allows the user to register a new account with Customer role in the i-Deliver system.



Screen 3. Register screen

Step	Instruction
1.	Fill in the fields “Tên đăng nhập”, “Mật khẩu”, “Email” “Điện thoại” with your username, password, email address, phone number. Your input will be validated as you type. Please make sure that all the values that you provide are validated.
5.	Click the button “Đăng ký” to proceed to register your account with Customer role. If registered successfully, you will be redirected to the “Thông tin cá nhân” page to input profile information with the newly created account.

6.2.1.4. Send a delivery request

This function allows the user to send a delivery request to the i-Deliver system.

Screen 4. Customer screen

The screenshot shows the customer interface of the i-Deliver system. On the left, there's a sidebar with links like 'Trang chủ', 'Chuyên phát' (highlighted by a red box labeled '1'), 'Quản lý yêu cầu', 'Thanh toán', 'Bảng giá', 'Người dùng', and 'Đăng xuất'. The main area features a map of Ho Chi Minh City and its surroundings, with various stations marked. Below the map is a table titled 'Danh sách trạm' (Station list) with columns for city names and phone numbers.

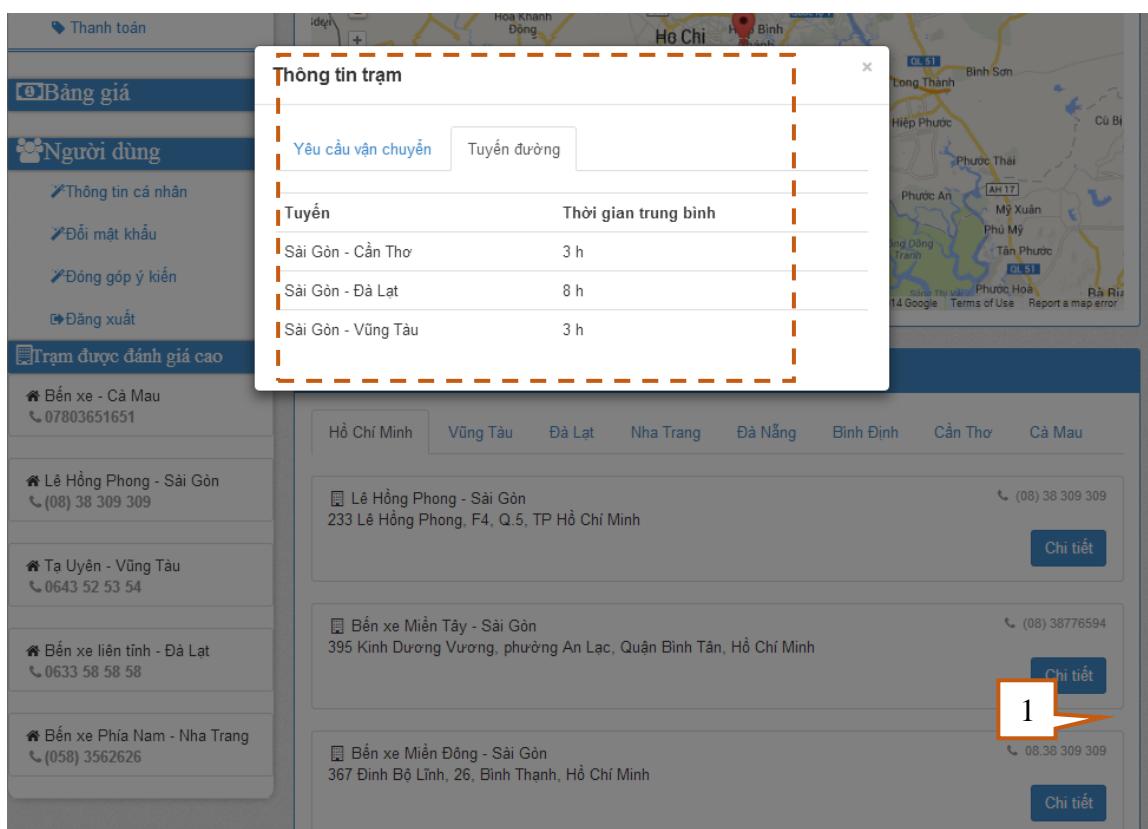
Screen 5. Send delivery request screen

This screenshot shows the delivery request form. It includes a map of Vung Tau and its surroundings. The form is divided into sections: 'Thông tin cơ bản' (Basic information), 'Thông tin sản phẩm' (Product information), and 'Phương thức thanh toán' (Payment method). Fields include 'Tên Người Gửi' (Name of sender), 'Địa Chỉ Người Gửi' (Address of sender), 'Số Điện Thoại Người Gửi' (Phone number of sender), 'Tên Người Nhận' (Name of receiver), 'Địa Chỉ Người Nhận' (Address of receiver), and 'Số Điện Thoại Người Nhận' (Phone number of receiver). A yellow box labeled 'Giá ước tính: 90000VND' (Estimated price: 90000VND) is shown. The 'Phương thức thanh toán' (Payment method) dropdown is set to 'Tiền mặt' (Cash). A green button labeled 'Hoàn Tất' (Complete) is at the bottom right.

Step	Instruction
1.	Click “Chuyển phát” link on the menu bar when you logged in with Customer role
2.	Fill in all fields. Your input will be validated as you type. Please make sure that all the values that you provide are validated.
3.	Click the button “Hoàn tất”. You will be redirected to payment page.

6.2.1.5. View station details and review

This function allows the customer to view details of station information, review, and rate the station.



Screen 6. Station information screen

Trang chủ

Chuyên phát

- Chuyển Phát
- Quản lý yêu cầu
- Thanh toán

Bảng giá

Người dùng

- Thông tin cá nhân
- Đổi mật khẩu
- Đóng góp ý kiến
- LĐang xuất** (highlighted with a red box)

Trạm được đánh giá cho

- Bến xe - Cà Mau
07803651651
- Lê Hồng Phong - Sài Gòn
(08) 38 309 309
- Ta Uyên - Vũng Tàu
0643 52 53 54

Bản đồ trạm và hành trình

Danh sách trạm

Hồ Chí Minh	Vũng Tàu	Đà Lạt	Nha Trang	Đà Nẵng	Bình Định	Cần Thơ	Cà Mau
Lê Hồng Phong - Sài Gòn 233 Lê Hồng Phong, F4, Q.5, TP Hồ Chí Minh Chi tiết							
Rắn và Miền Tây - Sài Gòn Chi tiết							

Screen 7. Customer main screen

Trang chủ

Chuyên Phát

- Quản lý yêu cầu
- Thanh toán

Bảng giá

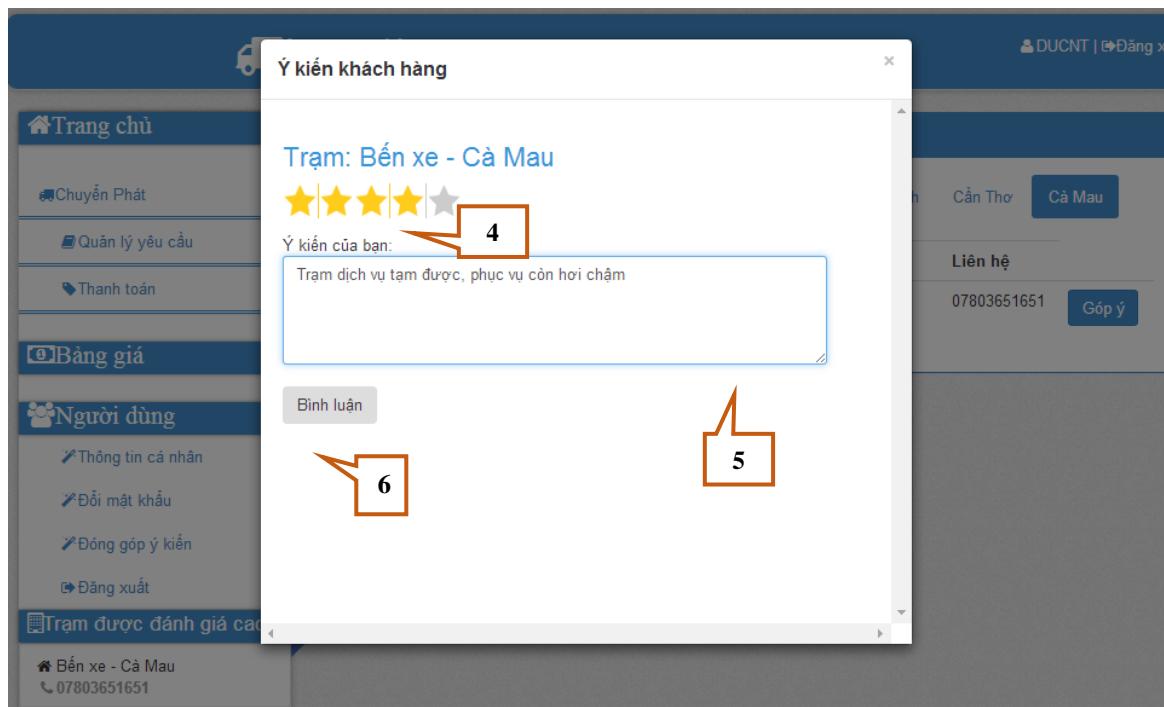
Người dùng

- Thông tin cá nhân
- Đổi mật khẩu

Đóng góp ý kiến

Hồ Chí Minh	Vũng Tàu	Đà Lạt	Nha Trang	Đà Nẵng	Bình Định	Cần Thơ	Cà Mau
Lê Hồng Phong - Sài Gòn 233 Lê Hồng Phong, F4, Q.5, TP Hồ Chí Minh Góp ý							
Bến xe Miền Tây - Sài Gòn 395 Kinh Dương Vương, phường An Lạc, Quận Bình Tân, Hồ Chí Minh Góp ý							
Bến xe Miền Đông - Sài Gòn 367 Đinh Bộ Lĩnh, 26, Bình Thạnh, Hồ Chí Minh Góp ý							

Screen 8. Station list for review screen

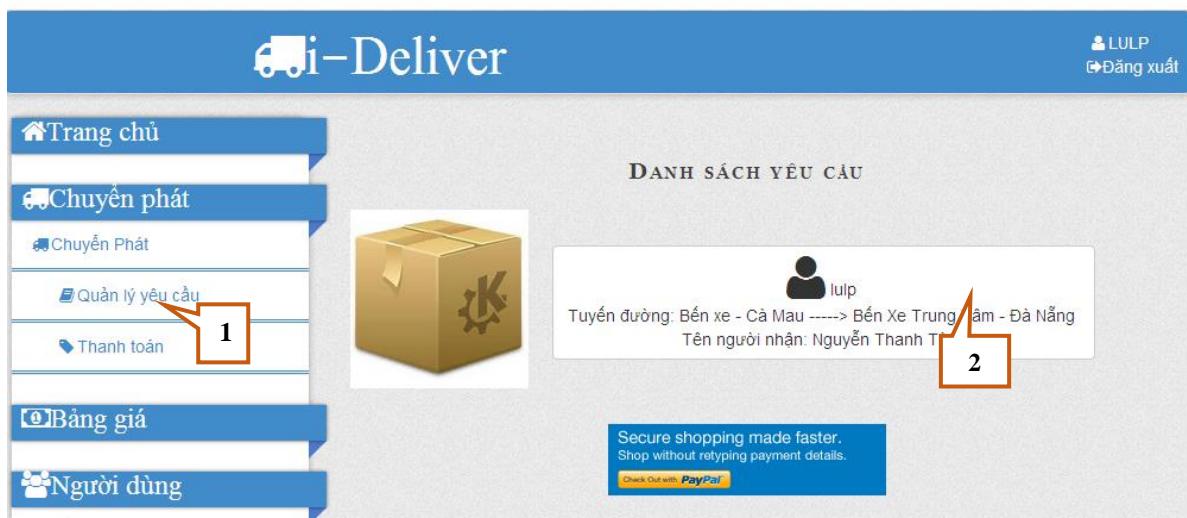


Screen 9. Review screen

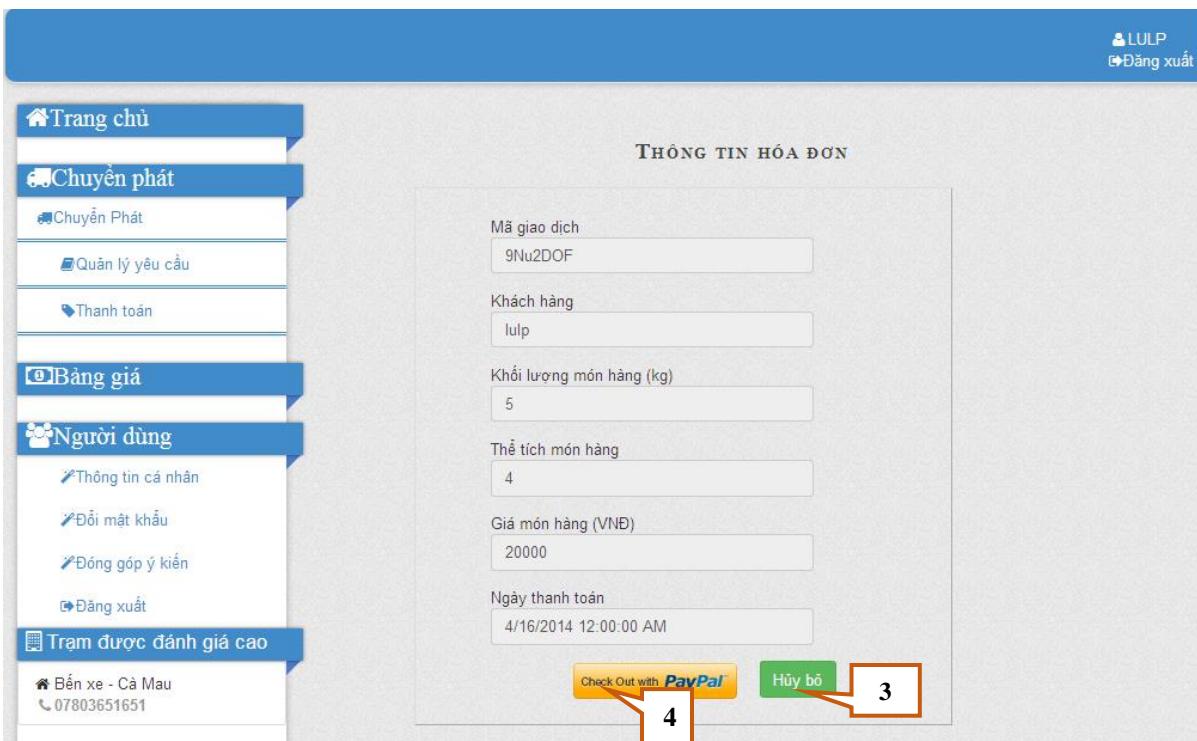
Step	Instruction
1.	Click on “Chi tiết” button in the result list of stations sections. A popup will be displayed to show route/stage information.
2.	On Customer’s main screen, click “Đóng góp ý kiến” link to direct to stations list for review page
3.	On stations list for review page, click on “Góp ý” button to show review popup dialog
4.	Rate the station about station services by checking the number of stars on rating bar.
5.	Fill your review about the station in the box “Ý kiến của bạn”. The content of the review can be your opinion on the station services, your personal experience, and what needs to be improved.
6.	Press the button “Bình luận” in order to submit a review for the station.

6.2.1.1. Online payment

This function allows the customer make a payment online via PayPal.



Screen 10. Request view before make a payment screen

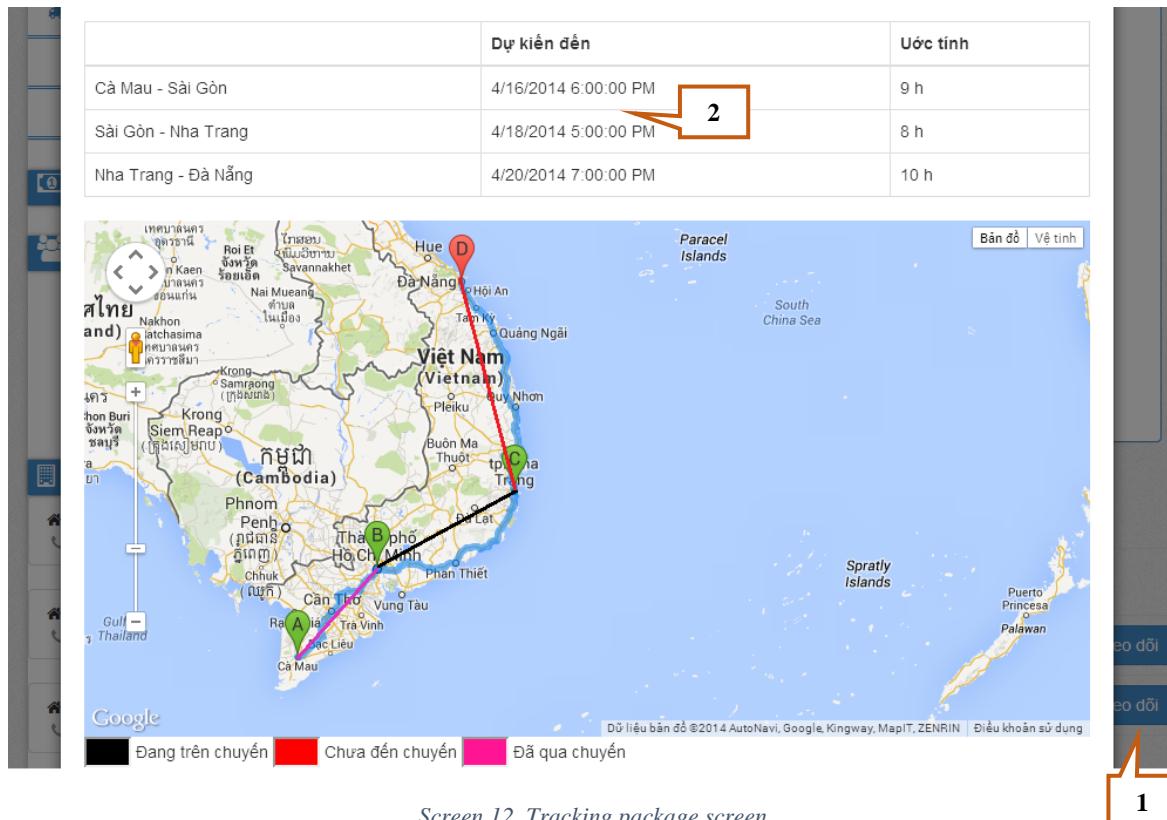


Screen 11. Invoice review before checkout screen

Step	Instruction
1.	Click “Thanh toán” link on the left menu bar when you logged in with Customer role.
2.	With the list of request are showed on the body of page, click on the box of request you want to pay. The website will redirect to the new page.
3.	Click button “Hủy bỏ” if you don’t want to page anymore and the website will redirect to the previous page.
4	Click Check out with “PayPal” and

6.2.1.2. Tracking package(s)

This function allow customer view the delivery status of assigned request with 3 types: not depart, departed and ongoing.



Screen 12. Tracking package screen

Step	Instruction
1.	On the index page of customer, click to the button “Theo dõi” to tracking the delivery status of request
2.	See the pop up with estimate time deliver, end station and map.

6.2.2. Guides for Staffs and System Admins

6.2.2.1. Approve pending requests

This function allows the staff to view all pending requests, input invoice for each request by click “Nhập hóa đơn” button then approve it after confirm payment for request by click “Xác nhận thanh toán” button.

STAFF10 | Đăng xuất
Nhân viên trạm Bến xe - Cà Mau

Trang chủ

Chuyển Phát

Người dùng

Thông tin cá nhân

Đổi mật khẩu

Đăng xuất

Quản lý hành trình

- Xe
- Trạm
- Chuyển
- Tuyến
- Lịch chạy

Quản lý yêu cầu vận chuyển

- Tiếp nhận yêu cầu **0**
- Xác nhận thanh toán
- Xếp lịch vận chuyển **1**
- Hàng đã xếp lịch

Các yêu cầu chuyển phát mới

Tìm kiếm

Tên	Ngày gửi	Trạm gửi	Trạm nhận	
Long	4/16/2014	Bến xe - Cà Mau	Bến xe Phía Nam - Nha Trang	Nhập hóa đơn
lulp	4/16/2014	Bến xe - Cà Mau	Bến Xe Trung Tâm - Đà Nẵng	Nhập hóa đơn

Screen 13. Requests review screen

STAFF10 | Đăng xuất
Nhân viên trạm Bến xe - Cà Mau

Trang chủ

Chuyển Phát

Người dùng

Thông tin cá nhân

Đổi mật khẩu

Đăng xuất

Quản lý hành trình

- Xe
- Trạm
- Chuyển
- Tuyến
- Lịch chạy

Quản lý yêu cầu vận chuyển

Hoá đơn

Khối lượng thực: **5**

Thể tích thực: **7**

Thành tiền: **28000** **4**

Đóng Xác nhận **5**

Nhập hóa đơn

Nhập hóa đơn

Screen 14. Input invoice screen

STAFF10 | Đăng xuất
Nhân viên trạm Bến xe - Cà Mau

Tên	Trạm gửi	Trạm nhận	Ngày gửi	Hình thức	
Long	Bến xe - Cà Mau	Bến xe Phía Nam - Nha Trang	4/16/2014	Tiền mặt	In hóa đơn
Đức	Bến xe - Cà Mau	Bến Xe Trung Tâm - Đà Nẵng	4/14/2014	Tiền mặt	In hóa đơn

6

Screen 15. Print invoice screen

7

Section	Description
1.	Click to button “Tiếp nhận yêu cầu” on the left menu after user logged in to the system with account staff.
2.	See the list of request on the body page with some information: sender name, receiver name, phone of receiver, date post, from location and date arrive.
3.	Click to button “Nhập hóa đơn” for show the pop up for input information of invoice.
4.	Input the exact value of weight and volume of the request after customer bring to center.
5.	Click to button “Xác nhận” to save the information of invoice.
6.	Click to button “Xác nhận thanh toán” on the left menu.
7.	Click to button “In hóa đơn” to print the invoice as the PDF format for each request.

6.2.2.2. Create a trip

This function allows the staff to create new trip and save to the system database.

Ngày đi	Tuyến	Xe chạy	Giờ đi dự kiến	Giờ đến dự kiến
15/04/2014	Cà Mau - Sài Gòn	51B-34578	15/04/2014 5:00:00 AM	15/04/2014 12:00:00 PM
16/04/2014	Cà Mau - Sài Gòn	51B-34578	16/04/2014 5:00:00 AM	16/04/2014 12:00:00 PM
17/04/2014	Cà Mau - Sài Gòn	51B-34578	17/04/2014 5:00:00 AM	17/04/2014 12:00:00 PM
18/04/2014	Cà Mau - Sài Gòn	51B-34578	18/04/2014 5:00:00 AM	18/04/2014 12:00:00 PM
19/04/2014	Cà Mau - Sài Gòn	51B-34578	19/04/2014 5:00:00 AM	19/04/2014 12:00:00 PM
20/04/2014	Cà Mau - Sài Gòn	51B-34578	20/04/2014 5:00:00 AM	20/04/2014 12:00:00 PM
21/04/2014	Cà Mau - Sài Gòn	51B-34578	21/04/2014 5:00:00 AM	21/04/2014 12:00:00 PM
15/04/2014	Cà Mau - Sài Gòn	51B-34579	15/04/2014 8:00:00 AM	15/04/2014 3:00:00 PM
16/04/2014	Cà Mau - Sài Gòn	51B-34579	16/04/2014 8:00:00 AM	16/04/2014 3:00:00 PM

Screen 15. Trip list screen

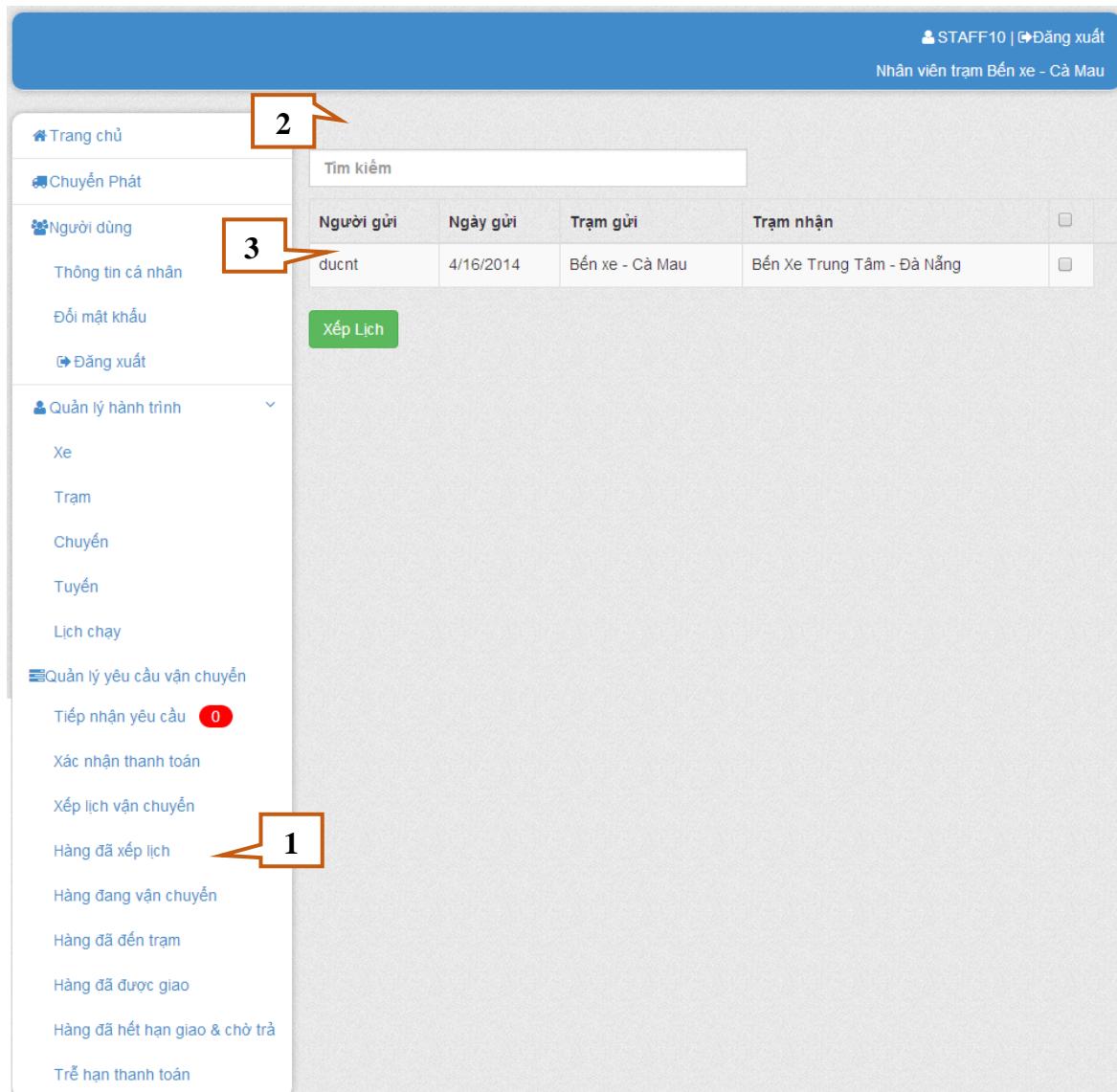
Tuyến	Biển số	Thời gian ước tính xuất bến
Cà Mau - Vũng Tàu	51B-34584	05:00:00
Cà Mau - Vũng Tàu	51B-34585	08:00:00
Cà Mau - Vũng Tàu	51B-34586	11:00:00
Cà Mau - Vũng Tàu	51B-34587	14:00:00
Cà Mau - Vũng Tàu	51B-34588	17:00:00
Cà Mau - Vũng Tàu	51B-34589	20:00:00

Screen 16. Create trip screen

Step	Instruction
1.	Click the “Chuyến” link on the menu bar when already logged in with Staff role account.
2.	Click on “Tạo chuyến mới” button to display a popup dialog.
3.	Provide necessary information for new trip. <ul style="list-style-type: none"> ▫ Choose a date from date picker “Ngày chạy”; ▫ Choose a route from “Tuyến” dropdown list; ▫ Check box to select defined schedule(s) of selected route Your input will be validated as you type.
4.	Click the “Tạo mới” button. The system saves the new trip, close popup dialog and return to trip list page.

6.2.2.3. Assign package(s)

This function allow staff account can choose one or multiple request that send at the current station and schedule them by assigning each of them to some trip to deliver them to their final destination.



Screen 17. Assign package(s) screen

Step	Instruction
1.	Click to button “Xếp lịch vận chuyển” on the left menu after user logged in to the system with account staff.
2.	See the list of request on the body page with some information: sender name, receiver name, phone of receiver, date post, from location and date arrive. Choose one or more request you want to schedule.
3.	Click to button “Xếp lịch” to run automatically function of the system to schedule for each of them for delivery. The assigning information are list of trip with the specific indicate order.

6.2.2.4. Confirm package(s) departed from start-station

This function allow staff to change status of the trip that also change status of all request assigned to that trip to “Đã chuyến đi” mean the package departed from its start station.

STAFF10 | Đăng xuất
Nhân viên trạm Bến xe - Cà Mau

1 Quản lý hành trình

- Xe**
- Trạm**
- Chuyển**
- Tuyến** **2**
- Lịch chạy**
- Quản lý yêu cầu vận chuyển**
- Tiếp nhận yêu cầu** **0**
- Xác nhận thanh toán**
- Xếp lịch vận chuyển**
- Hàng đã xếp lịch**
- Hàng đang vận chuyển**
- Hàng đã đến trạm**
- Hàng đã được giao**
- Hàng đã hết hạn giao & chờ trả**
- Trễ hạn thanh toán**

Quản lý chuyến

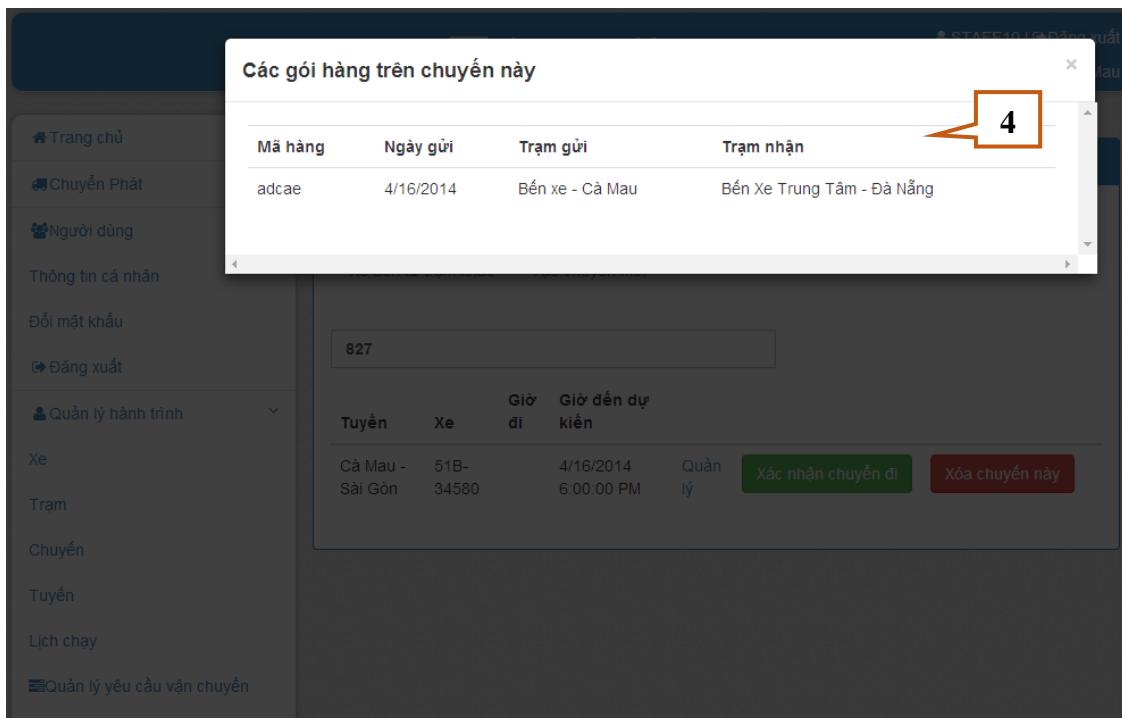
Danh sách **2** Chuyến xe vừa tạo

Xe đến từ trạm khác Tạo chuyến mới

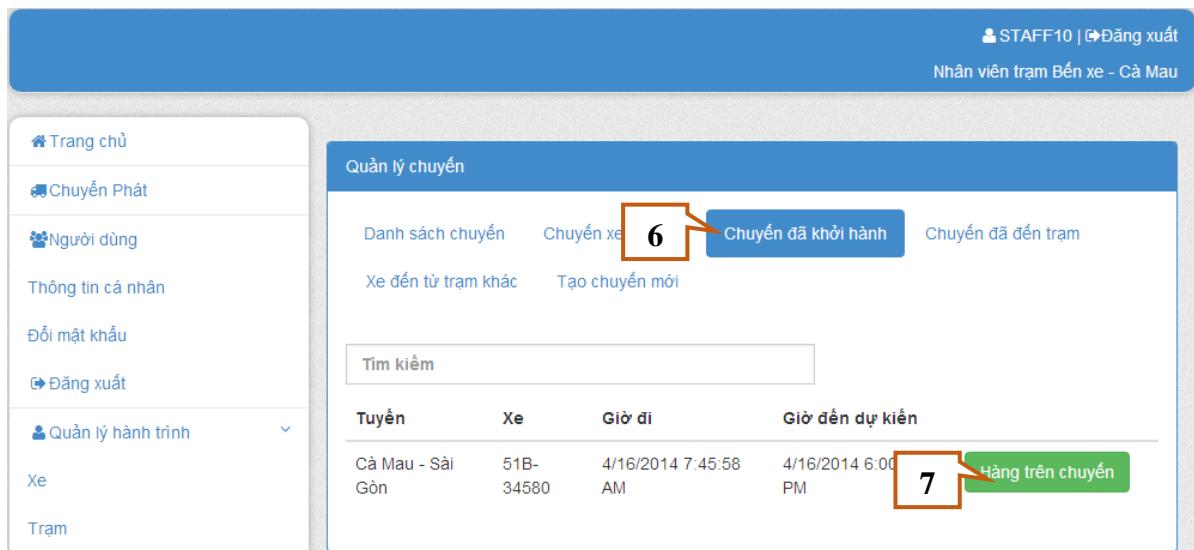
Tìm kiếm

Tuyến	Xe	Giờ đi	Giờ đến dự kiến	Quản lý	Xác nhận chuyến đi	Xóa chuyến này
Cà Mau - Sài Gòn	51B-34578	4/16/2014 12:00:00 PM	Quản lý	3 Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/17/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	5 Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/18/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/19/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/20/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/21/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34578	4/22/2014 12:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34579	4/16/2014 3:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34579	4/17/2014 3:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	
Cà Mau - Sài Gòn	51B-34579	4/18/2014 3:00:00 PM	Quản lý	Xác nhận chuyến đi	Xóa chuyến này	

Screen 18. Confirm departed package(s) screen



Screen 19. List request assigned to trip screen



Screen 20. List departed trip screen

Step	Instruction
1.	Click to button “Chuyển” on the left menu of page when user logs in with the account “Staff” of the some station.
2.	Click to button “Chuyển xe vừa tạo” and assume that staff have created some trip from this station and assigned some request to one of those trips.
3.	Click to button “Quản lý” to view the list of request assigned to that trip.
4.	View the information of list requests were assgined to that trip, each of them including Request Code, Date Post, From Station and To Station.
5.	Clik to button “Xác nhận chuyến đi” to confirm that trip departs from the start station. After that action, the station of trip will be changed to “Đang chạy” and the real departure time set as the time staff clicks button.

6.	Click to button “Chuyến đã khởi hành” on the menu bar to view the list of trip have departed from that station.
7.	Click to button “Hàng trên chuyến” to view the list request again.

6.2.2.5. Confirm package(s) arrived in end-station

This function allow staff to change status of the trip that also change status of all request assigned to that trip and have the destination station the same to the current station to “Đã đến đích” mean the package arrived in its end station.

Tuyến	Xe	Giờ đi	Giờ đến dự kiến
Cà Mau - Sài Gòn	51B-34580	4/16/2014 7:45:58 AM	4/16/2014 6:00:00 PM

Screen 19. List trip from the other station screen

Step	Instruction
1.	Click to button “Chuyển” on the left menu of page when user logs in with the account “Staff” of the some station.
2.	Click to button “Xe đến từ trạm khác” to view the list of trip travel cross that station.
3.	Click to button “Hàng trên chuyến” to view the list of request assigned to each trip, the information is the same to the information show on the previous list request.
4.	Click to button “Xác nhận đến trạm” to confirm the corresponding trip arrived to

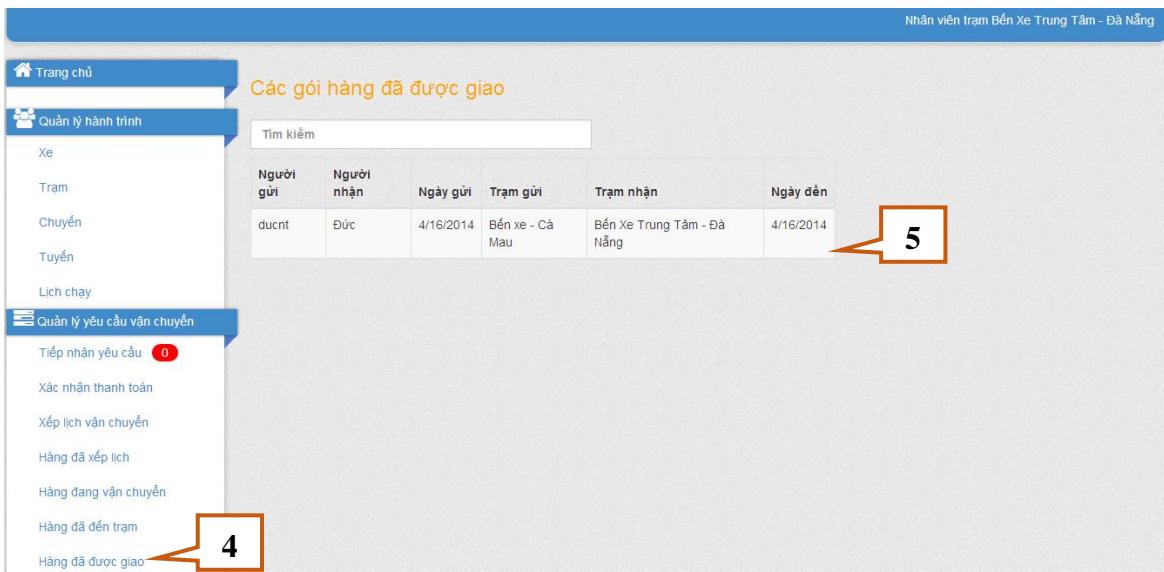
the station and the status of that trip change to “Đã đến đích” and all request with the destination is the same to this station will be changed status to “Đã đến đích”.

6.2.2.6. Confirm package(s) delivered

This function allows manage the package at the end station. Each package are sent to the receiver should be changed status to “Đã giao hàng” by click the button “Xác nhận giao hàng” on screen.

Người gửi	Người nhận	Điện thoại người nhận	Ngày gửi	Trạm gửi	Ngày đến	
ducnt	Đức	841657577422	4/16/2014	Bến xe - Cà Mau	4/17/2014	<input type="checkbox"/>

Screen 22. List request that delivered to end station screen



Screen 20. List request that delivered to customer screen

Step	Instruction
1.	Click to button “Hàng đã đến trạm” on the left menu after user logged in to the system with account staff.
2.	See the list of request on the body page with some information: sender name, receiver name, phone of receiver, date post, from location and date arrive. Choose which request you want to change status.
3.	Click to button “Xác nhận giao hàng” to change status of request to “Đã giao hàng” mean the request are sent to the receiver.
4.	Click to button “Hàng đã được giao” on the left menu.
5.	View the list of delivered requests on the body page.

6.2.2.7. Prepare package(s) for return

This function allows manage the package at the end station. Each package is not sent to the receiver and overdue the time of delivery, staff will consider return that package to the sender or cancel this package.

STAFF8 | Đăng xuất
Nhân viên trạm Bến Xe Trung Tâm - Đà Nẵng

Trang chủ
 Chuyển Phát
 Người dùng
Thông tin cá nhân
Đổi mật khẩu
Đăng xuất

Quản lý hành trình
Xe
Trạm
Chuyển
Tuyến
Lịch chạy

Quản lý yêu cầu vận chuyển
Tiếp nhận yêu cầu 0
Xác nhận thanh toán
Xếp lịch vận chuyển
Hàng đã xếp lịch
Hàng đang vận chuyển
Hàng đã đến trạm
Hàng đã được giao
Hàng đã hết hạn giao & chờ trả
Trễ hạn thanh toán

Xác nhận

Người gửi	Người nhận	Trạm gửi	Ngày gửi	Trạm nhận	Ngày đến	Mã hàng	Trả hàng	Hủy hàng
Long	Đức	Bến xe - Cà Mau	4/16/2014 12:42:29 AM	Bến Xe Trung Tâm - Đà Nẵng	4/17/2014 12:00:00 AM	adcae	<input type="radio"/>	<input checked="" type="radio"/>

Screen 21. List late deliver request screen

Step	Instruction
1.	Click to button “Hàng đã hết hạn & chờ giao” on the left menu after user logged in to the system with account staff.
2.	See the list of request on the body page with some information: sender name, receiver name, phone of receiver, date post, from location and date arrive. Choose which request you want to change status.
3.	If you choose the radio button “Trả hàng” and click to button “Xác nhận”, the package will consider as the new request and will return to sender. If you choose the radio button “Hủy hàng” and click to button “Xác nhận”, the package will be cancelled.

6.2.2.8. Extend due-date for online payment

This function allow staff to extend the due date of request that post to the system but after 24 hours, the customer still not pay for this request.

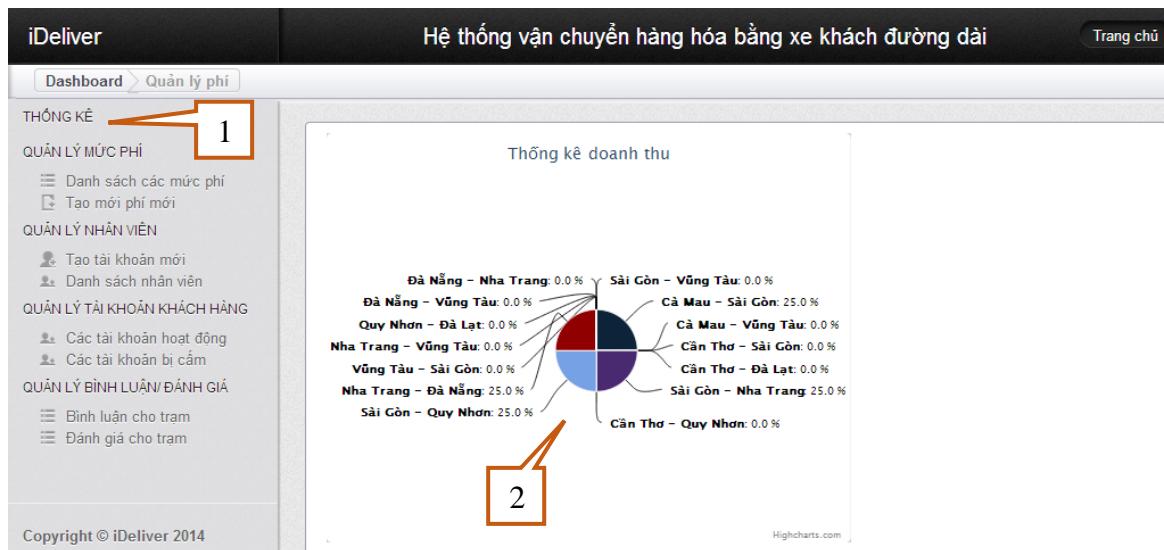
Người gửi	Người nhận	Địa chỉ nhận	Điện thoại người gửi	Ngày gửi yêu cầu	Hạn chót thanh toán	Quản lý
ducent	Đức	Đà Nẵng	0123456789	4/16/2014 12:42:29 AM	4/17/2014 12:42:29 AM	

Screen 22. List late payment requests screen

Step	Instruction
1.	Click to button “Trễ hạn thanh toán” on the left menu after user logged in to the system with account staff.
2.	See the list of request on the body page with some information: sender name, receiver name, phone of receiver, date post, from location and date arrive. For each request, choose which action you want to perform.
3.	If you click to the green button, the corresponding package will be extended the due day to 24 hour more. If you click to the orange button, the corresponding package will be cancelled.

6.2.2.9. Statistics

This function allows a system admin to view statistic of routes.



Screen 23. Statistic screen

Step	Instruction
1.	Click “Thống kê” link on menu bar when already logged in which System admin role account
2.	Revenues of each route will display on pie chart.

6.2.2.10. Add a fee

This function allows a system admin to add a new fee for delivery service.

Screen 24. Add a fee screen

Step	Instruction
1.	Click “Tạo mức phí mới” link on menu bar when already logged in which System admin role account
2.	Fills in all fields with validated input. Your input will be validated as you type.
3.	Click the “Tạo mới” button. The fee will be saved to database.

6.2.2.11. Monitor fees

This function allows a system admin to monitor the fees/prices of delivery service (includes edit, view detail and delete defined fee level).

Screen 25. Monitor fees screen

ID	Mức phí	Khối lượng (kg)	Thẻ tích (lit)	Thao tác
Từ	Đến	Từ	Đến	
1	20000	0	7.5	
2	27000	7.5	15	
3	35000	15	22.5	
4	42000	22.5	30	
5	50000	30	37.5	
6	60000	37.5	45	
7	72000	45	52.5	
8	90000	52.5	60	
9	100000	60	67.5	
10	115000	67.5	75	

Section	Description
1.	Click “Danh sách các mức phí” link on menu bar when already logged in which System admin role account
2.	A view of fees list will be display with monitoring actions.
3.	Click “pen” icon to edit correlative fee record.
4.	Click “i” icon to view fee detail.
5.	Click “recycle bin” icon to delete correlative fee record.

Appendix

Reference

<http://en.wikipedia.org/wiki/Model-view-controller>

<http://www.visual-paradigm.com/product/vpuml/tutorials/>

<http://www.packtpub.com/mastering-ninject-for-dependency-injection/book>

http://en.wikipedia.org/wiki/Waterfall_model

<http://www.asp.net/mvc/tutorials/mvc-4>

<http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>

<http://www.csharpfriends.com/articles/getarticle.aspx?articleid=336>

<http://www.uml-diagrams.org/>

http://en.wikipedia.org/wiki/Database_design