|  |  |
| --- | --- |
| http://muabanraovat.com/images/picture/mien-phi-modem-lap-dat-internet-(2955708).jpg | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Long Travel Bus Reservation |
|  |
| |  |  | | --- | --- | | **Group 4** | | | **Group Members** | Nguyễn Sơn Trường – Team Leader – 60466  Nguyễn Lương Hải – Member – 60335  Nguyễn Thị Bích Trâm – Member – 60156  Nguyễn Ngọc Sơn – Member – 60409 | | **Supervisor** | Kiều Trọng Khánh | | **Ext Supervisor** | N/A | | **Capstone Project code** | BRES | |
|  |

- Ho Chi Minh City, 04/2013 –

# Table of Contents

[1 Table of Contents 2](#_Toc354354418)

[2 Introduction 5](#_Toc354354419)

[2.1 Current travelling service system 5](#_Toc354354420)

[2.1.1 Vehicle managing 5](#_Toc354354421)

[2.1.2 Ticket managing 5](#_Toc354354422)

[2.2 Problem Definition 5](#_Toc354354423)

[2.3 Proposed Solution 6](#_Toc354354424)

[2.3.1 Non-Financial 6](#_Toc354354425)

[2.3.2 Financial 6](#_Toc354354426)

[2.4 Functional Requirements 6](#_Toc354354427)

[2.4.1 Search for journey 6](#_Toc354354428)

[2.4.2 Booking 6](#_Toc354354429)

[2.4.3 Scheduling 7](#_Toc354354430)

[2.4.4 Vehicles Management 7](#_Toc354354431)

[2.4.5 Fare 7](#_Toc354354432)

[2.4.6 User management 7](#_Toc354354433)

[3 Software Project Management Plan (SPMP) 8](#_Toc354354434)

[3.1 Project Overview 8](#_Toc354354435)

[3.2 Problem Abstract 8](#_Toc354354436)

[3.2.1 About the System 8](#_Toc354354437)

[3.3 Project organization 9](#_Toc354354438)

[3.3.1 Software Process Model 9](#_Toc354354439)

[3.3.2 Roles and Responsibilities 9](#_Toc354354440)

[3.3.3 Tools and Techniques 10](#_Toc354354441)

[3.3.4 Project management plan 10](#_Toc354354442)

[3.3.5 Product backlog 10](#_Toc354354443)

[3.4 Coding Convention 12](#_Toc354354444)

[4 Software Requirements Specifications (SRS) 13](#_Toc354354445)

[4.1 User Requirement Specification 13](#_Toc354354446)

[4.1.1 Guest requirements 13](#_Toc354354447)

[4.1.2 Customer requirements 13](#_Toc354354448)

[4.1.3 Staff requirements 13](#_Toc354354449)

[4.2 System Requirement Specification 14](#_Toc354354450)

[4.2.1 System Features 14](#_Toc354354451)

[4.2.2 Software System Attributes 76](#_Toc354354452)

[5 Software Design Description 78](#_Toc354354453)

[5.1 Design Overview 78](#_Toc354354454)

[5.2 System Architectural Design 79](#_Toc354354455)

[5.2.1 Choice of System Architecture 79](#_Toc354354456)

[5.3 Component Diagram 79](#_Toc354354457)

[5.3.1 Presentation Layer’s Components 80](#_Toc354354458)

[5.3.2 Business Logic Layer’s Components 80](#_Toc354354459)

[5.3.3 Data Access Layer’s Components 80](#_Toc354354460)

[5.3.4 Common Component 81](#_Toc354354461)

[5.4 Detailed Description of Components 81](#_Toc354354462)

[5.4.1 Search for journeys 81](#_Toc354354463)

[5.5 Database Design or Data Structures 88](#_Toc354354464)

[5.5.1 Entity Relationship Diagram 88](#_Toc354354465)

[5.5.2 Database diagram 95](#_Toc354354466)

[6 Software Test Documentation 96](#_Toc354354467)

[6.1 Introduction 96](#_Toc354354468)

[6.1.1 System Overview 96](#_Toc354354469)

[6.1.2 Test Approach 96](#_Toc354354470)

[6.2 Test Plan 96](#_Toc354354471)

[6.2.1 Features to be tested 96](#_Toc354354472)

[6.2.2 Features not to be tested 97](#_Toc354354473)

[6.3 Test Cases 98](#_Toc354354474)

[6.3.1 Manage bus 98](#_Toc354354475)

[6.3.2 Manage segment 99](#_Toc354354476)

[6.3.3 Manage route and schedule 100](#_Toc354354477)

[6.3.4 Create reservation 107](#_Toc354354478)

[6.4 Checklists 109](#_Toc354354479)

[6.4.1 Checklist of Validation 109](#_Toc354354480)

[6.4.2 Submission Checklist 109](#_Toc354354481)

[7 Software User’s Manual 112](#_Toc354354482)

[7.1 Installation Guide 112](#_Toc354354483)

[7.1.1 Setting up environment at server side 112](#_Toc354354484)

[7.1.2 Deployment at server sites 112](#_Toc354354485)

[7.1.3 Setting up environment at client side 112](#_Toc354354486)

[7.2 User’s Guide 133](#_Toc354354487)

[7.2.1 Guest Guide 133](#_Toc354354488)

[7.2.2 Customer Guide 133](#_Toc354354489)

[7.2.3 Staff Guide 133](#_Toc354354490)

[8 Appendix 133](#_Toc354354491)

## Table list

[Table 4‑1 Actor description 16](#_Toc354354582)

[Table 4‑2 UC001 Search journeys specification 21](#_Toc354354583)

[Table 4‑3 Search journey screen description 22](#_Toc354354584)

[Table 4‑4 Search result screen description 23](#_Toc354354585)

[Table 4‑5 Journey details screen description 24](#_Toc354354586)

[Table 4‑6 Map screen description 25](#_Toc354354587)

[Table 4‑7 UC002 Choose journeys use case specification 26](#_Toc354354588)

[Table 4‑8 UC0003 Choose seat use case specification 29](#_Toc354354589)

[Table 4‑9 Choose seat screen description 30](#_Toc354354590)

[Table 4‑10 UC004 Provide booker’s details user case specification 33](#_Toc354354591)

[Table 4‑11 UC005 Pay with Paypal Use Case Specification 36](#_Toc354354592)

[Table 4‑12 Review reservation details screen description 37](#_Toc354354593)

[Table 4‑13 Paypal site screen’s Controls 39](#_Toc354354594)

[Table 4‑14 UC006 View Reservation List Use Case Specification 41](#_Toc354354595)

[Table 4‑15: Reservation List screen’s Controls 41](#_Toc354354596)

[Table 4‑16 UC007 View Reservation Details Use Case Specification 44](#_Toc354354597)

[Table 4‑17 Search Reservation screen’s controls 44](#_Toc354354598)

[Table 4‑18 Reservation Details screen’s Control 46](#_Toc354354599)

[Table 4‑19 UC008 Cancel Reservation Use Case Specification 48](#_Toc354354600)

[Table 4‑20 Cancel Reservation popup description 48](#_Toc354354601)

[Table 4‑21 UC009 View route details Use case Specification 50](#_Toc354354602)

[Table 4‑22 Manage route screen description 51](#_Toc354354603)

[Table 4‑23 Route detail screen description 52](#_Toc354354604)

[Table 4‑24 UC010 Add route Use case Specification 54](#_Toc354354605)

[Table 4‑25 Add new route screen description 55](#_Toc354354606)

[Table 4‑26 UC011 Add fare to route Use case Specification 56](#_Toc354354607)

[Table 4‑27 Edit route price screen description 57](#_Toc354354608)

[Table 4‑28 UC012 Assign bus to route Use case Specification 59](#_Toc354354609)

[Table 4‑29 List bus in route screen description 60](#_Toc354354610)

[Table 4‑30 UC013 Deactivate route use case Specification 61](#_Toc354354611)

[Table 4‑31 UC014 View bus list Use case Specification 63](#_Toc354354612)

[Table 4‑32 Manage bus screen description 63](#_Toc354354613)

[Table 4‑33 UC015 Add new bus Use case Specification 65](#_Toc354354614)

[Table 4‑34 Add new bus screen description 65](#_Toc354354615)

[Table 4‑35 UC016 Delete bus Use case specification 67](#_Toc354354616)

[Table 4‑36 Delete bus confirmation screen description 67](#_Toc354354617)

[Table 4‑37 UC017 Manage segment Use case Specification 69](#_Toc354354618)

[Table 4‑38 Manage segment screen description 70](#_Toc354354619)

[Table 4‑39 Edit segment screen description 71](#_Toc354354620)

[Table 4‑40 Add new schedule Use case Specification 73](#_Toc354354621)

[Table 4‑41 Add new schedule screen description 73](#_Toc354354622)

[Table 4‑42 UC019 View schedule list Use case Specification 75](#_Toc354354623)

[Table 4‑43 Manage schedule screen description 75](#_Toc354354624)

[Table 4‑44 Schedule list in route details screen description 76](#_Toc354354625)

[Table 4‑45 Cancel schedule Use case Specification 77](#_Toc354354626)

[Table 4‑46 Cancel schedule screen description 78](#_Toc354354627)

[Table 5‑1 SearchResultAction class explanation 84](#_Toc354354628)

[Table 5‑2 CityDAO class explanation 85](#_Toc354354629)

[Table 5‑3 SearchResultLogic class explanation 85](#_Toc354354630)

[Table 5‑4 TripDAO class explanation 86](#_Toc354354631)

[Table 5‑5 SearchTripDetailsJSONAction class explanation 87](#_Toc354354632)

[Table 5‑6 SearchLatitudeJSONAction class explanation 88](#_Toc354354633)

[Table 5‑7 Entity description 94](#_Toc354354634)

[Table 5‑8 Attribute description 96](#_Toc354354635)

[Table 5‑9 Database status description 96](#_Toc354354636)

[Table 6‑1 Manage bus test case 101](#_Toc354354637)

[Table 6‑2 Manage segment test case 102](#_Toc354354638)

[Table 6‑3 Manage route and schedule test case 109](#_Toc354354639)

[Table 6‑4: Checklist of Validation 111](#_Toc354354640)

[Table 6‑5: Submission Checklist 112](#_Toc354354641)

[Table 7‑1 Register steps 135](#_Toc354354642)

[Table 7‑2 Search steps 135](#_Toc354354643)

[Table 7‑3 Choose trip steps 135](#_Toc354354644)

[Table 7‑4 Choose seat steps 135](#_Toc354354645)

[Table 7‑5 Fill booker’s information steps 135](#_Toc354354646)

[Table 7‑6 Pay with PayPal steps 135](#_Toc354354647)

[Table 7‑7 Search reservation steps 135](#_Toc354354648)

[Table 7‑8 View reservation steps 135](#_Toc354354649)

[Table 7‑9: View manage bus screen step 135](#_Toc354354650)

[Table 7‑10: Add bus step. 135](#_Toc354354651)

[Table 7‑11: Delete bus step. 135](#_Toc354354652)

[Table 7‑12: View manage route screen step. 135](#_Toc354354653)

[Table 7‑13: Add route step. 135](#_Toc354354654)

[Table 7‑14: Deactivate route step. 135](#_Toc354354655)

[Table 7‑15 View manage route details screen step 135](#_Toc354354656)

[Table 7‑16: Edit Price step. 135](#_Toc354354657)

[Table 7‑17: Assign bus to route step. 135](#_Toc354354658)

[Table 7‑18 Add new schedule step. 135](#_Toc354354659)

[Table 7‑19: Cancel Schedule step. 135](#_Toc354354660)

[Table 7‑20 View manage schedule step. 135](#_Toc354354661)

[Table 7‑21: Delete Schedule step. 135](#_Toc354354662)

[Table 7‑22 View schedule details step. 135](#_Toc354354663)

[Table 7‑23 View manage segment step. 135](#_Toc354354664)

[Table 7‑24 Edit segment step 135](#_Toc354354665)

## Figure list

[Figure 3‑1 Scrum model 13](#_Toc354354745)

[Figure 4‑1 System use case 18](#_Toc354354746)

[Figure 4‑2 Flow for booking journeys 19](#_Toc354354747)

[Figure 4‑3 Flow for scheduling journey 19](#_Toc354354748)

[Figure 4‑4 Choose journeys use case 20](#_Toc354354749)

[Figure 4‑5 Search journeys screen 24](#_Toc354354750)

[Figure 4‑6 Search result screen 25](#_Toc354354751)

[Figure 4‑7 Journey details screen 26](#_Toc354354752)

[Figure 4‑8 Map screen 27](#_Toc354354753)

[Figure 4‑9 Choose seat use case 29](#_Toc354354754)

[Figure 4‑10 Choose seat screen 31](#_Toc354354755)

[Figure 4‑11 Provide booker’s details use case 32](#_Toc354354756)

[Figure 4‑12 Pay use case 36](#_Toc354354757)

[Figure 4‑13- Review reservation details screen 38](#_Toc354354758)

[Figure 4‑14- Payment details screen 40](#_Toc354354759)

[Figure 4‑15- View Reservation List use case diagram 41](#_Toc354354760)

[Figure 4‑16 Reservation List screen description 43](#_Toc354354761)

[Figure 4‑17 Search Reservation screen 46](#_Toc354354762)

[Figure 4‑18 Reservation Details screen 47](#_Toc354354763)

[Figure 4‑19 Cancel Reservation Popup 50](#_Toc354354764)

[Figure 4‑20 Manage route use case 50](#_Toc354354765)

[Figure 4‑21 Manage route (Route list) screen 53](#_Toc354354766)

[Figure 4‑22 Route details screen 53](#_Toc354354767)

[Figure 4‑23 Add new route screen 56](#_Toc354354768)

[Figure 4‑24 Edit route price screen 59](#_Toc354354769)

[Figure 4‑25 List bus in route screen 62](#_Toc354354770)

[Figure 4‑26 Manage bus use case 64](#_Toc354354771)

[Figure 4‑27 Manage bus screen 65](#_Toc354354772)

[Figure 4‑28 Add new bus screen 67](#_Toc354354773)

[Figure 4‑29 Delete bus confirm screen 69](#_Toc354354774)

[Figure 4‑30 Manage segment use case 69](#_Toc354354775)

[Figure 4‑31 Manage segment screen 72](#_Toc354354776)

[Figure 4‑32 Edit segment screen 72](#_Toc354354777)

[Figure 4‑33 Manage schedule use case 73](#_Toc354354778)

[Figure 4‑34 Add new schedule screen 75](#_Toc354354779)

[Figure 4‑35 Manage schedule screen 77](#_Toc354354780)

[Figure 4‑36 Schedule list in route details screen 77](#_Toc354354781)

[Figure 4‑37 Cancel schedule screen 79](#_Toc354354782)

[Figure 4‑38 Register and login Use case 80](#_Toc354354783)

[Figure 4‑39 Account management Use case 80](#_Toc354354784)

[Figure 5‑1 3-Layer Architecture 83](#_Toc354354785)

[Figure 5‑2: Component Diagram 83](#_Toc354354786)

[Figure 5‑3: Presentation Layer's Components 84](#_Toc354354787)

[Figure 5‑4: Business Layer's Components 84](#_Toc354354788)

[Figure 5‑5: Data Access Layer's Components 84](#_Toc354354789)

[Figure 5‑6 Common Components 85](#_Toc354354790)

[Figure 5‑7 Search function class diagram 85](#_Toc354354791)

[Figure 5‑8 Algorithm of search journeys 91](#_Toc354354792)

[Figure 5‑9 Sequence diagram of search function 92](#_Toc354354793)

[Figure 5‑10 Entity relationship diagram (with entity only) 93](#_Toc354354794)

[Figure 5‑11 Attributes of entities 94](#_Toc354354795)

[Figure 5‑12 Relational database diagram 99](#_Toc354354796)

[Figure 7‑1: Register button 137](#_Toc354354797)

[Figure 7‑2 Register page 137](#_Toc354354798)

[Figure 7‑3: Input search criteria 137](#_Toc354354799)

[Figure 7‑4: Choose trip 137](#_Toc354354800)

[Figure 7‑5: Choose seat 137](#_Toc354354801)

[Figure 7‑6: Fill booker’s information 137](#_Toc354354802)

[Figure 7‑7: Pay with PayPal 137](#_Toc354354803)

[Figure 7‑8: Search reservation 137](#_Toc354354804)

[Figure 7‑9: Reservation list 137](#_Toc354354805)

[Figure 7‑10 View manage bus screen. 137](#_Toc354354806)

[Figure 7‑11 Add bus popup 137](#_Toc354354807)

[Figure 7‑12 Delete bus popup 137](#_Toc354354808)

[Figure 7‑13 View manage route screen. 137](#_Toc354354809)

[Figure 7‑14 Add route popup. 137](#_Toc354354810)

[Figure 7‑15 Deactivate route popup. 137](#_Toc354354811)

[Figure 7‑16 View manage route details screen. 137](#_Toc354354812)

[Figure 7‑17 Edit Price popup. 137](#_Toc354354813)

[Figure 7‑18 Assign bus to route popup. 137](#_Toc354354814)

[Figure 7‑19 Add new schedule popup. 137](#_Toc354354815)

[Figure 7‑20 Cancel Schedule popup. 137](#_Toc354354816)

[Figure 7‑21 View manage schedule Screen. 137](#_Toc354354817)

[Figure 7‑22 Delete Schedule popup. 137](#_Toc354354818)

[Figure 7‑23 View schedule Details Screen. 137](#_Toc354354819)

[Figure 7‑24 Manage Segment Screen. 137](#_Toc354354820)

[Figure 7‑25 Edit segment popup. 137](#_Toc354354821)

## List of abbreviation and acronym

|  |  |  |
| --- | --- | --- |
| **No.** | **Abbreviation& Acronym** | **Definition** |
| 1 | BRES | Long Travel Bus Reservation |
| 2 | Route | A line of travel that buses run on |
| 3 | Route Details | Detailed parts of a route which contain segments that a trip run on |
| 4 | Segment | A direct track that connects two stations of two cities |
| 5 | Station | A place where buses load or unload passengers |
| 6 | Tariff | A schedule of fares apply on a segment for a bus type in a period of time |
| 7 | Trip | A trip going on a route detail which is scheduled before. |
| 8 | Schedule | A plan for bus to run on a route in a specific time. |
| 9 | Journey | A list of trips in a schedule which customers reserved tickets on. |
| 10 | Ticket | An indicator for booked seats of a journey. |

|  |
| --- |
|  |

# Introduction

Every year in holidays, travel needs are rising, many people move out the city for relaxing or home town visiting. Especially in big cities like Hanoi or Ho Chi Minh City, it is always overloaded for the ticket booking system in all the ticket offices. Customers have to wait for hours to have a chance to buy a ticket.

## Current travelling service system

### Vehicle managing

* + The buses are grouped by route. Each bus has only one onward and one return route (For example: Hà Nội – Hồ Chí Minh City and Hồ Chí Minh City – Hà Nội). So they are always scheduled to run on those route.
  + Staff of the company will manage buses and assign buses to reasonable routes. Then they will create journey by scheduling for each bus, for instance: which time that a bus can depart from a station…
  + If all journeys of a route are always in full or lacking seat, staff will assign more buses to that route. Otherwise, when the journeys usually remains too many seats, staff will remove some buses from that route and extend the time between departure dates.
  + If too few or no customers book tickets on a route, the company may remove that route and all the buses which are assigned to the route will be available for assigning to other routes.

### Ticket managing

* + The entire scheduled journeys will be recorded and managed. Each journey will have a fitted fare defined by the policies of the company mostly based on the length of the journey and facilities of the buses. Fares are changed frequently over the time.
  + Once the journeys are scheduled, ticket officers will receive the journeys information and plan for a tickets selling strategy.
  + If the sold tickets less than the minimum of the planned tickets, then the route will be cancelled and customer will be return all the tickets fee.
  + Customers will purchase for a ticket directly at the ticket offices, staffs at the ticket offices will note down customers’ information and booking detail.

## Problem Definition

* On holidays, limited number of box offices makes it difficult to serve a large number of customers at a time
* It is a hard job for staff to manage all the bus, route, fare and ticket balancing.
* The number of tickets for each type (full journey and constituents of a journey) is fixed which make it inflexible. There’re cases where it lacking of tickets for a segment whiles others types of tickets are excess.
* When customer need to return or change tickets (change type of tickets, departure date, seats), only a small request would be approved.

## Proposed Solution

A system needs to be established which takes care of booking system, journey scheduling and managing vehicles to reduce human effort on operating and managing.  
The system will be hosted on the web, which provides easy access for customer to use the booking services. Customer can have their own account for viewing their transaction history. Scheduling and managing services are available for authorized staffs only.  
In detail, the system will enable the staffs and customers to do the following:

### Non-Financial

* Customer will be able to search by departure/arrival location, date of departure, vehicle type... and view all available journey on the system, which is scheduled by the staffs through a scheduling service.
* Customer can easily registered for an account in the system. This account will maintain the history of the transaction they made, their information including full name, address, and phone number and so on.
* The system will provide a method to manage all of the vehicles and their status to make it easier to track and make the schedule.

### Financial

* Payment for each booking transaction has to be done by the system and credited to the customer account. The system should be able to provide the customer a method of online payment like PayPal.
* A refund method should be provided for the customers in case they want to cancel and return their tickets or the journey is cancelled by the company.
* Fare and promotion plans is also managed by the system according to the policies of the company.

## Functional Requirements

Functional requirements of the proposed system are listed as below:

### Search for journey

* Users search by departure/arrival, date, coach type and number of passengers
* Display search result
  + Information of the journey: bus type, date-time, number of remaining seats, fare. The journeys will be sorted by departure time.

### Booking

* Users select journey based on search results
* Users select seats
  + Display seat map for corresponding bus type
  + User can select position on map
  + Registered users can switch seat position for others
* Payment using PayPal
* Output ticket code
  + User can view journey info by searching using ticket code and their emails
* Cancel and refund

### Scheduling

* Create route and schedule
* Delete route
* Cancel a schedule
  + Inform and refund for customers if the journey is cancelled

### Vehicles Management

* Add new bus
* Assign bus to a route
* Remove bus from a route
* Delete bus

### Fare

* Fares are based on route, bus type and departure time (for example: fares on holidays are higher than usual)

### User management

* Registration of members (customers) to the site
* User can view their transaction history
* User can cancel reservation

# Software Project Management Plan (SPMP)

## Project Overview

The name of this project is “Long Travel Bus Reservation” or “BRES”. It is a website for a transportation company where their staff can manage vehicles, build up schedule and journeys, and provide their customers a fastest way to get tickets on-line.

## Problem Abstract

As described in section 1, the needs of travelling is raising these days. Based on statistics from Mien Dong Bus Station of Ho Chi Minh City website ([http://www.benxemiendong.com.vn/](http://www.benxemiendong.com.vn/article/qua-trinh-phat-trien-14.html) ), in 2010:

* Average number of bus departures in a day is about 1,124. On holidays it can raise to 2,000.
* Average number of passengers in a day is 23,000 and 62,000 on holidays.

With a large amount of passengers and journeys in a day like that, apparently bus operators will be overloaded by setting schedule, managing their vehicles and selling tickets. Their customers will be frustrated by having to wait for hours to buy a ticket to get home.

### About the System

#### Boundaries of the system

The system provides utilities for both bus operators and their customers. There will be a site for customers to book for their trips and pay for the tickets online, the customers may have their own accounts which help them access to transaction history and modify their information. There will also be a separated site for operators’ staff to managing the vehicles, scheduling for new trips or cancel trips.

Detailed functional requirements are mentioned in section 1.4

#### Development Environment

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

**Hardware requirements:**

* Computer with minimum of 3.5GB hard drive for environment configuration and data storage.
* Internet connection.

**Software Requirements:**

* Operating system: Windows 7
* IDE: Eclipse Indigo.
* DBMS: MySQL Server 5.5
* Source control: Google Code.
* Subversion client: Tortoise SVN.

## Project organization

### Software Process Model

The model used for developing this project is Scrum model.

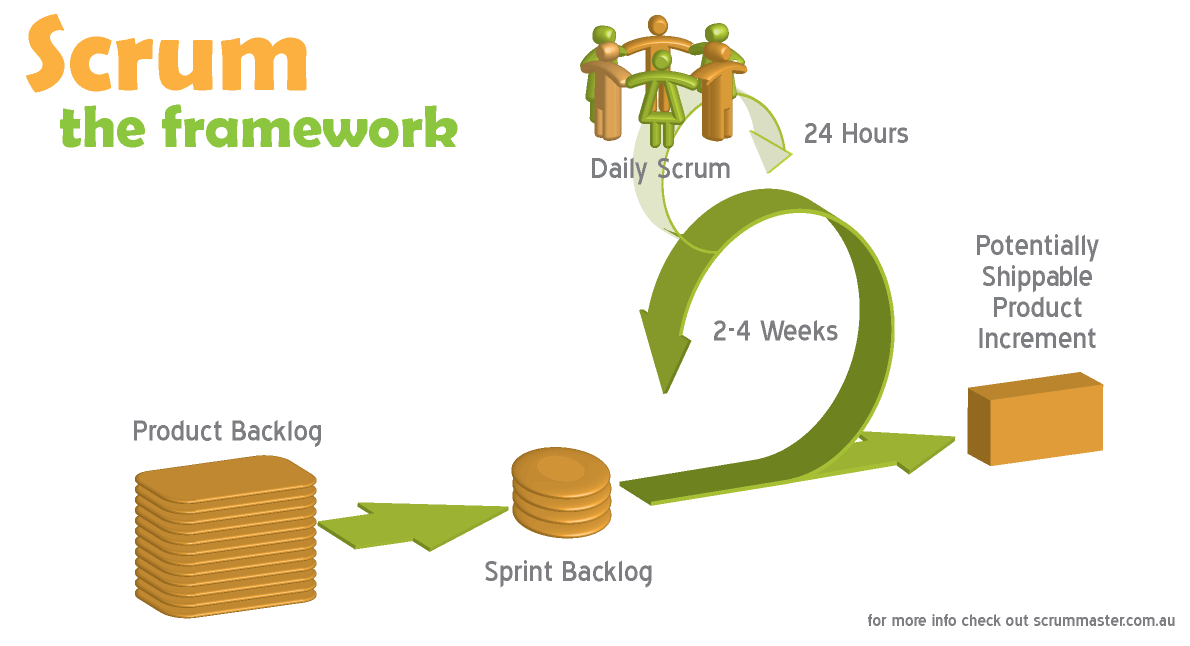


Figure 3‑1 Scrum model

(Image source: http://www.scrummaster.com.au)

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Full name** | **Role** | **Responsibilities** |
| 1 | Kiều Trọng Khánh | Project Owner | * Specify user requirements * Control the development process * Support in technique and business analysis |
| 2 | Nguyễn Sơn Trường | Scrum Master  Developer  Tester | * Manage process * Design database * Develop and analyze requirement * Prepare documents * Create test plan * Support technique * Code * Test |
| 3 | Nguyễn Lương Hải | Developer  Tester | * Design database * Develop and analyze requirement * Configure developing environment * Support technique * Code * Test |
| 4 | Nguyễn Thị Bích Trâm | Developer  Tester | * Develop and analyze requirement * Prepare documents * Create system test cases * Code * Test |
| 5 | Nguyễn Ngọc Sơn | Developer  Tester | * Develop and analyze requirement * Create prototype and design GUI * Create system test cases * Code * Test |

### Tools and Techniques

* Front-end: HTML, CSS, JavaScript, jQuery 1.9.1, JSON, AJAX, Google Map API
* Back-end: Struts 2, Spring 2, Hibernate 3, log4j, PayPal SDK, Java Mail, Apache FOP
* Server: Apache Tomcat 6.0.26

### Project management plan

* This project will finished in 7 sprints, each sprint lasts for 2 weeks.
* Daily scrum meeting is held every day at 22:00 via Skype
* Sprint review meeting is at 8:30 on last Saturday of every sprint. Plan for the next sprint is also created in this meeting.

### Product backlog

| **Story ID** | **Story name** | **Acceptance Criteria** | **Task Id** | **Task description** | **Sprint** | **Owner** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Create project introduction document | Introduction of overall project is recorded. | 1.1 | Research information about the current system | 1 | Tram | Done |
| 1.2 | Analyze the problem of the current system, and propose a solution | 1 | Truong | Done |
| 2 | Create task list | Product backlog is recorded. | 1.3 | Identify main requirements of the system and create task list base on them | 1 | Truong | Done |
| 3 | Study PayPal SDK | Can apply payment by PayPal in the project | 1.4 | Study PayPal SDK | 1 | Hai | Done |
| 4 | Create prototype | Master page is created. | 1.5 | Create UI for master page and search page | 1 | Son | Done |
| 5 | Create Software requirement specification | All the requirements are cleared and recorded. | 2.1 | Analyze and clarify requirements and create the document. | 2 | Team | Done |
| 6 | Design Database | Database is created with sample data | 2.2 | Create entity-relationship diagram | 2 | Truong | Done |
| 2.3 | Design and create database based on entity-relationship diagram | 2 | Hai | Done |
| 2.4 | Test database | 2 | Team | Done |
| 7 | As a staff, I can manage bus, route, segments and schedules | Can I add, edit, or remove bus, route, segments and schedules? | 3.1 | Identify features of manage component and how to implement. | 3 | Team | Done |
| 3.2 | Create detail design for manage component. | 3 | Team | Done |
| 3.3 | Implement function to manage bus, route, schedule and segment. | 3 | Tram | Done |
| 3.4 | Implement GUI of manage components | 3 | Tram | Done |
| 3.5 | Test manage component | 3 | Team | Done |
| 3.6 | Add manual guide | 3 | Team | Done |
| 8 | As a customer, I can search for a journey. | Can I search for a journey by providing departure and arrival information? | 4.1 | Identify features of search function and how to implement | 4 | Team | Done |
| 4.2 | Create detail design for search function | 4 | Team | Done |
| 4.3 | Implement search function | 4 | Truong | Done |
| 4.4 | Implement GUI for search function | 4 | Truong | Done |
| 4.5 | Test search function | 4 | Team | Done |
| 4.6 | Add manual guide | 4 | Team | Done |
| 9 | As a customer, I can book for a ticket on a journey | Can I choose seats on a journey and book for them? | 5.1 | Identify features of booking function and how to implement | 5 | Team | Done |
| 5.2 | Create detail design for booking function | 5 | Team | Done |
| 5.3 | Implement booking function | 5 | Son | Done |
| 5.4 | Generate seat map and implement GUI | 5 | Son | Done |
| 5.5 | Test booking function | 5 | Team | Done |
| 5.6 | Add manual guide | 5 | Team | Done |
| 10 | As a customer, I can pay for my reservation and view reservation history | Can I pay for my reservation using PayPal?  Can I view my reservation history?  Can I cancel my reservation? | 6.1 | Identify features of payment function and how to implement | 6 | Team | Done |
| 6.2 | Create detail design for payment function | 6 | Team | Done |
| 6.3 | Implement payment function | 6 | Hai | Done |
| 6.4 | Implement GUI for payment function | 6 | Hai | Done |
| 6.5 | Test payment function | 6 | Team | Done |
| 6.7 | Add manual guide | 6 | Team | Done |
| 11 | Add a specific user, I can access to specific function of the system. | As a guest, can I create a reservation?  As a customer, can I create a reservation and view my reservation history?  As a staff, can I manage bus, route, segment and schedule?  As an admin, can I manage user? | 7.1 | Add authentication and authorization to the system with minimum affect to the system | 7 | Hai | Done |
| 7.2 | Add manage user function for admin | 7 | Son | Done |
| 7.3 | Test system behavior and performance | 7 | Team | Done |
| 7.4 | Add manual guide | 7 | Team | Done |

### All Meeting Minutes

All meeting minutes are stored in the meeting minute folder.

## Coding Convention

Follow Code Conventions for the Java TMProgramming Language, by Sun Microsystems

# Software Requirements Specifications

## User Requirement Specification

Long Travel Bus Reservation provides public users (guests) an easy mean to search and book for journeys, view their reservation and cancel reservation. It also allows staff to manage buses, routes, route segments and journey schedules. In details, for each user there are respective features the system would provide:

### Guest requirements

1. Register for a customer account.
2. Login to the system as a customer or staff.
3. Search for journeys.
4. Choose seats on the buses, create reservations and pay for them using PayPal.
5. View past reservations by reservation code and email.
6. Cancel guests’ reservations.

### Customer requirements

1. Search for journeys.
2. Choose seats on the buses, create reservations and pay for them using PayPal.
3. View list of reservations and details of each reservation that they made.
4. Cancel their own reservations.

### Staff requirements

1. Search for journeys.
2. Choose seats on the buses and create reservations
3. View list of reservations and details of each reservation that they made.
4. Cancel their own reservations.
5. View list of buses.
6. Add a new bus.
7. Delete a bus.
8. View list of routes.
9. Add a new route.
10. Delete a route
11. Assign bus to route and remove bus from route.
12. Add fare to route.
13. View list of schedules.
14. Add a new schedule.
15. Cancel a schedule.
16. View list of segments.
17. Edit travel time of segment.

## System Requirement Specification

### System Features



Figure 4‑1 System use case

|  |  |  |
| --- | --- | --- |
| No. | Actor | Description |
| 1 | Guest | User who does not have an account or has not logged in the system. |
| 2 | Customer | User who is the customer of the transportation company and uses the services provided by the company. |
| 3 | Staff | Employee of the transportation company. |
| 4 | Admin | Administrator of the system. |

Table 4‑1 Actor description

#### Program flow

1. **Booking journeys**

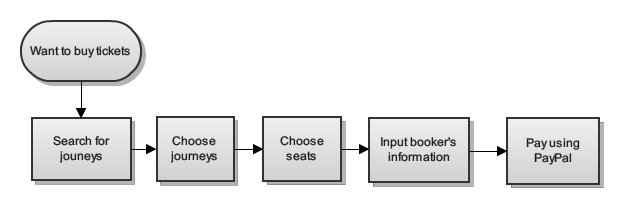


Figure 4‑2 Flow for booking journeys

1. **Scheduling journeys for staff**

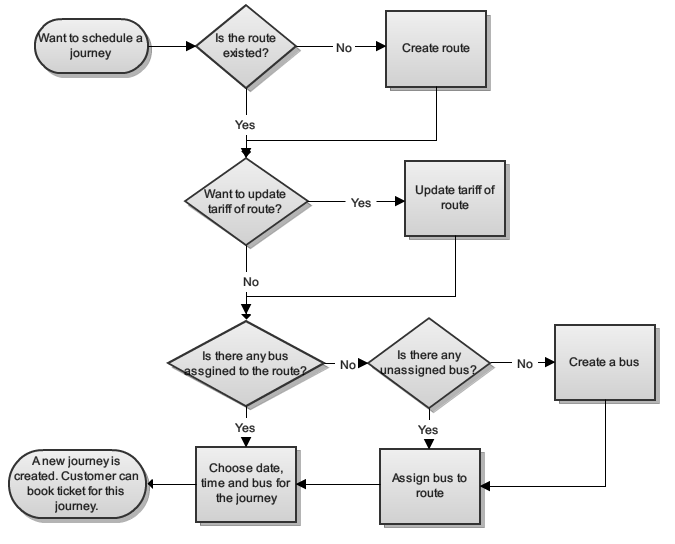
****

Figure 4‑3 Flow for scheduling journey

#### Choose journeys use case



Figure 4‑4 Choose journeys use case

* 1. **Search journeys**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC001 | **Use-case Version** | 0.1 |
| **Use-case Name** | Search journeys | | |
| **Author** | Nguyễn Sơn Trường | | |
| **Date** | 2013/01/29 | **Priority** | High |
| **Actor:** Guest  **Summary:** User can view list of available journeys  **Goal:** Helps user view all the available journeys satisfying their needs.  **Triggers:** User wants to search for journeys. (User provides search conditions and click button “Đặt vé”)  **Pre-conditions:** N/A  **Post-conditions:**  *Success*: Search result is displayed  *Failure*: Show message “Hiện tại chuyến đi không có hoặc đã hết vé. Xin quý khách vui lòng thử tìm chuyến vào ngày khác.”  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User access the search page  3. User chooses type of journeys, departure/arrival cities. [Alternative 1][Alternative 2]  5. User chooses number of passengers, departure date, arrival date and clicks on button “Đặt vé”.  7. User clicks on “Hiện bản đồ” button [Alternative 4][Alternative 5][Alternative 6] | 1. Display search page include search criteria as below:    * One way [Vé một chiều: Radio button, initial: checked]    * Round trip [Vé khứ hồi: Radio button]    * Departure city [Điểm đi: Drop down list]    * Arrival city [Điểm đến: Drop down list]    * Depart date [Ngày đi: Date time picker, min: current date, max: 3 months after current date]    * Return date [Ngày về: Date time picker, min: departure date, max: 3 months after current date]]    * Book tickets [Đặt vé: Button]   4. Display list of arrival cities that have journeys matching departure cities  6. Display search result page with details of journeys matching search condition. [Alternative 3]   * Onward journey [Chuyến đi: string, label] * Return journey [Chuyến về: string, label] * Departure city [Điểm đi: string, label] * Arrival city [Điểm đến: string, label] * Show map [Hiện bản đồ: button] * Hide map [Ẩn bản đồ: button, initial: hide] * Bus type [Loại xe: string, label] * Departure date [Ngày khởi hành: Date, tab] * Departure time [Giờ đi: DateTime, label] * Arrival time [Giờ đến: DateTime, label] * View details [Xem chi tiết: button] * Fare [Giá vé: double, label] * Remained seats [Số ghế trống: integer, label] * Select [Chọn chuyến: checkbox] * Continue [Tiếp tục: button] * Return [Quay lại: button]     8. Display the map and show route of selected journey. Hide “Hiện bản đồ” button and show “Ẩn bản đồ” button |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Alternative 1]  1. User chooses “Khứ hồi” radio button.  [Alternative 2]  1. User chooses “Một chiều” radio button.  [Alternative 3]  1. There are no available journeys satisfied the conditions.  [Alternative 4]  1. User clicks on “Ẩn bản đồ” button  [Alternative 5]  1. User clicks on “Xem chi tiết” link  [Alternative 6]  1. User clicks on “Quay lại” button | 2. Display “Ngày về” control.  2. Hide “Ngày về” control.  2. Show message: “Hiện tại chuyến đi không có hoặc đã hết vé. Xin quý khách vui lòng thử tìm chuyến vào ngày khác”  2. Hide the map, hide “Ẩn bản đồ” button and show “Hiện bản đồ” button.  2. Show a modal that lists all cities and stations in the current journey.   * Departure time [Giờ khởi hành: DateTime, label] * Arrival time [Giờ dừng nghỉ: DateTime, label] * Departure station [Trạm khởi hành: String, label] * Arrival station [Trạm dừng nghỉ: String, label]   2. Redirect to home page. |   **Exceptions:** N/A  **Relationships:**  Choose journeys  **Business Rules:**   * User can only reserve for a journey for maximum of 3 months and minimum of 30 minutes after current date. * Maximum number of passengers for each reservation is 5. * System will search and display journeys which have departure date >= 3 days before selected date by user and <= 3 days after selected date. | | | |

Table 4‑2 UC001 Search journeys specification



Figure 4‑5 Search journeys screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | One way [Vé một chiều: radio button] | Booking for one way journey |
| **2** | Round trip [Vé khứ hồi: radio button] | Booking for round trip |
| **3** | Departure [Điểm đi: drop down list] | Departure location of the journey |
| **4** | Arrival [Điểm đến: drop down list] | Arrival location of the journey |
| **5** | Depart date [Ngày đi: date time picker] | Depart date, min: current date, max: 3 months after current date. |
| **6** | Return date [Ngày về: date time picker] | Return date, min: departure date, max: 3 months after current date. |
| **7** | Book tickets [Đặt vé: button] | Booking button |

Table 4‑3 Search journey screen description

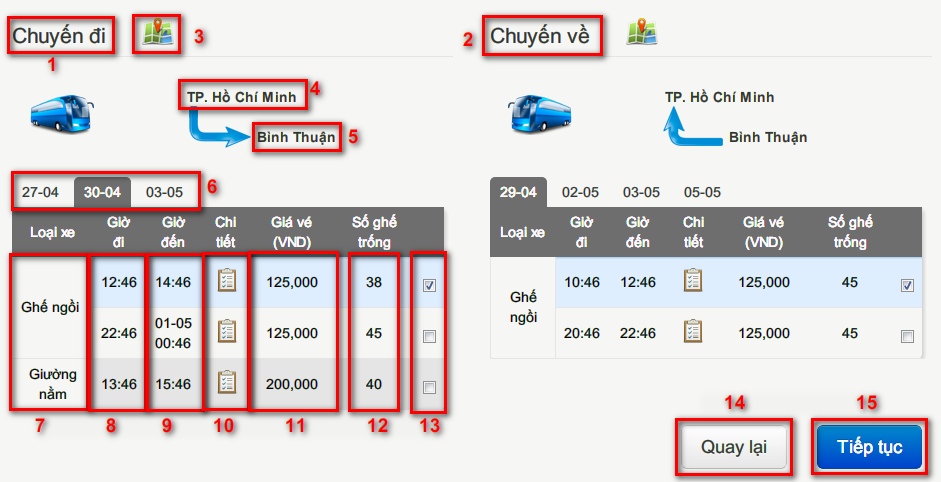


Figure 4‑6 Search result screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Onward [Chuyến đi: label] | Onward label |
| **2** | Return [Chuyến về: label] | Return label |
| **3** | Show map [Hiện bản đồ: button] | Show route of the journey on map |
| **4** | Departure [Điểm đi: label] | Departure location of the journey |
| **5** | Arrival [Điểm đến: label] | Arrival location of the journey |
| **6** | Depart date [Ngày đi: tab] | Tabs of depart date in 7 days |
| **7** | Bus type [Loại xe: label] | Type of the bus that runs on a journey |
| **8** | Depart time [Giờ đi: label] | Depart time of the journey |
| **9** | Arrive time [Giờ đến: label] | Arrive time of the journey |
| **10** | View details [Xem chi tiết: link] | View start and end stations of the journey |
| **11** | Fare [Giá vé: label] | Fare of the journey in VND |
| **12** | Remained seats [Số ghế trống: label] | Number of remained seats |
| **13** | Select [Chọn chuyến: check box] | Check on to choose the journey |
| **14** | Return [Quay lại: button] | Return to home page |
| **15** | Continue [Tiếp tục: button] | Forward to select seat page |

Table 4‑4 Search result screen description

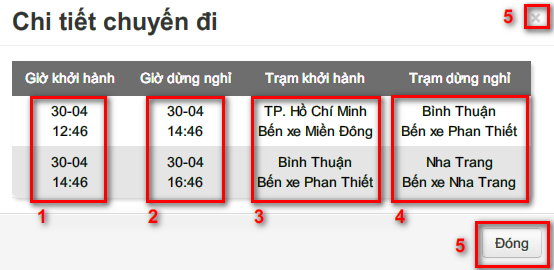


Figure 4‑7 Journey details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Departure time [Giờ khởi hành: label] | Departure time at a station |
| **2** | Arrival time [Giờ dừng nghỉ: label] | Arrival time at a station |
| **3** | Departure station [Trạm khởi hành: label] | Departure location on each segment of the journey |
| **4** | Arrival station [Trạm dừng nghỉ: label] | Arrival location on each segment of the journey |
| **5** | Close [Đóng: button] | Click to close this modal |

Table 4‑5 Journey details screen description

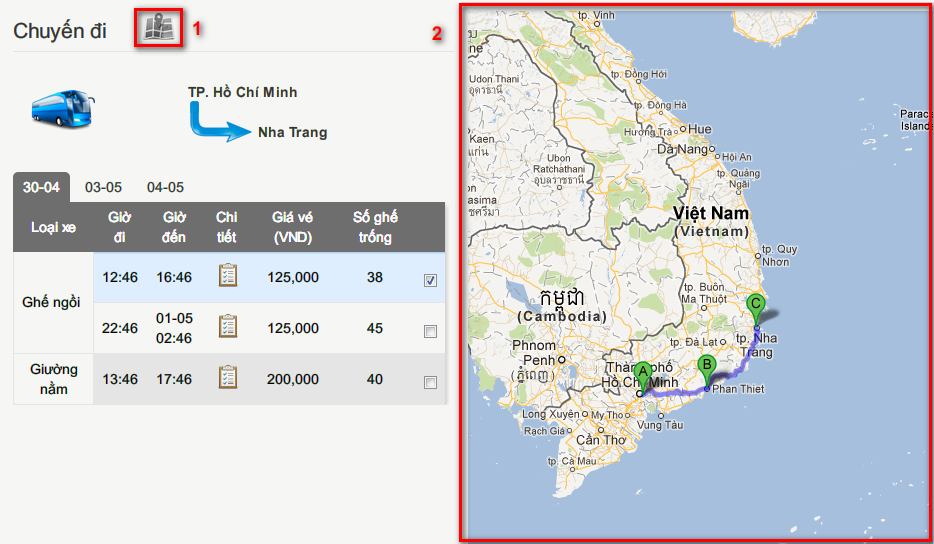


Figure 4‑8 Map screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Hide map [Ẩn bản đồ: button] | Click button to hide the map |
| **2** | Map | Show route of selected journey on map |

Table 4‑6 Map screen description

* 1. **Choose journeys**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC002 | **Use-case Version** | 0.1 |
| **Use-case Name** | Choose journeys | | |
| **Author** | Nguyễn Sơn Trường | | |
| **Date** | 2013/01/30 | **Priority** | High |
| **Actor:** Guest  **Summary:** User can choose one or two journeys that they preferred.  **Goal:** When user decided to choose journeys, guide them to select seats page  **Triggers:** User decided to choose journeys. (User choose journeys in search result page or in home page)  **Pre-conditions:**  User searched for journeys or user is at homepage and choose suggested journeys.  **Post-conditions:**  *Success*: Journeys are selected and select seats page is shown.  *Failure*: Show message “Chuyến đi bạn chọn đã khởi hành, vui lòng thử chọn chuyến vào ngày khác”  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. In search result page, user choose journeys and Click “Tiếp tục” button or user choose a suggested journey at homepage and click “Đặt vé ngay” | 2. Display seat selection page. [Exception 1][Exception 2][Exception 3] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Exception 1]  1. User chooses return journey that departs before the onward journey arrives.  [Exception 2]  1. User does not choose any journeys.  [Exception 3]  1. User chooses departed journey. | 2. Show message: “Vui lòng chọn chuyến về có thời gian khởi hành sau khi chuyến đi kết thúc.”  2. Disable button “Tiếp tục”.  2. Show message: “Chuyến đi bạn chọn đã khởi hành, vui lòng chọn chuyến đi khác” |   **Relationships:**  Search journey  **Business Rules:**   * User can only reserve for a journey for maximum of 3 months and minimum of 30 minutes after current date. * Maximum number of passengers for each reservation is 5. * System will search and display journeys which have departure date >= 3 days before selected date by user and <= 3 days after selected date. | | | |

Table 4‑7 UC002 Choose journeys use case specification

#### Choose seat use case



Figure 4‑9 Choose seat use case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC003 | **Use-case Version** | 0.1 |
| **Use-case Name** | Choose Seat | | |
| **Author** | Nguyễn Ngọc Sơn, Nguyễn Sơn Trường | | |
| **Date** | 31/1/2013 | **Priority** | High |
| **Actor:** Guest  **Summary:** User choose seats after search journey  **Goal:** Use can book seats on the buses for their journeys.  **Triggers:** User select one bus after search.  **Preconditions:**  User chose one journey or two journeys (one onward journey and one return journey).  **Post Conditions:**  *Success*: Seats are chosen and seat numbers are displayed in provide booker’s information page.  *Failure*: Show error message “Ghế đã có người đặt”.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User select journey in search result page and click button “Tiếp tục”  3. Click seats.  5. Click “Tiếp Tục” button to submit view [Alternative 1] | 2. Display a seats map   * Green seats : selected seats * Orange seats : not available seats * White seats : available seats * Gray seats: disabled seats * Submit [“Tiếp Tục”: button] * Back [“Quay Lại”: button]   4. Select seats [Alternative 2], [Alternative 3], [Alternative 4], [Alternative 5] , [Alternative 6] , [Alternative 7] , [Alternative 8]  6. Redirect to page provide booker’s information [Exception 1] |   **Alternative Scenario:**   |  |  | | --- | --- | | Ator Actions | System Response | | [Alternative 1]  1. User clicks on “Quay Lại” button.  [Alternative 2]  1. User selects available seats.  [Alternative 3]  1. User selects sold seats.  [Alternative 4]  1. User deselects a selected seat.  [Alternative 5]  1. Number of selected seats equals number of passengers.  [Alternative 6]  1. User deselects a selected seat after all available seats change color to gray.  [Alternative 7]  1. User chose return seat-map tab.  [Alternative 8]  1. User chose onward seat-map tab. | 2. Back to search result page.  2. Available seats change color to green  2. Show error message “Ghế đã có người đặt”  2. Selected seats change color to white.  2. All available seat change color to gray.  2. All available seat change color to white.  2. Display the seat-map for return journey.  2. Display the seat-map for onward journey. |   **Exceptions:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Exception 1]  1. User do not select any seat and submit page | 2. Display error message: “Bạn phải chọn ít nhất 1 ghế để tiếp tục.” |   **Relationships:**  Create reservation  **Business Rules:**   * User must choose at least one seat and maximum of 5 seats * If ticket type is round-trip, user must choose at least one seat for onward and one seat for return journey. | | | |

Table 4‑8 UC0003 Choose seat use case specification

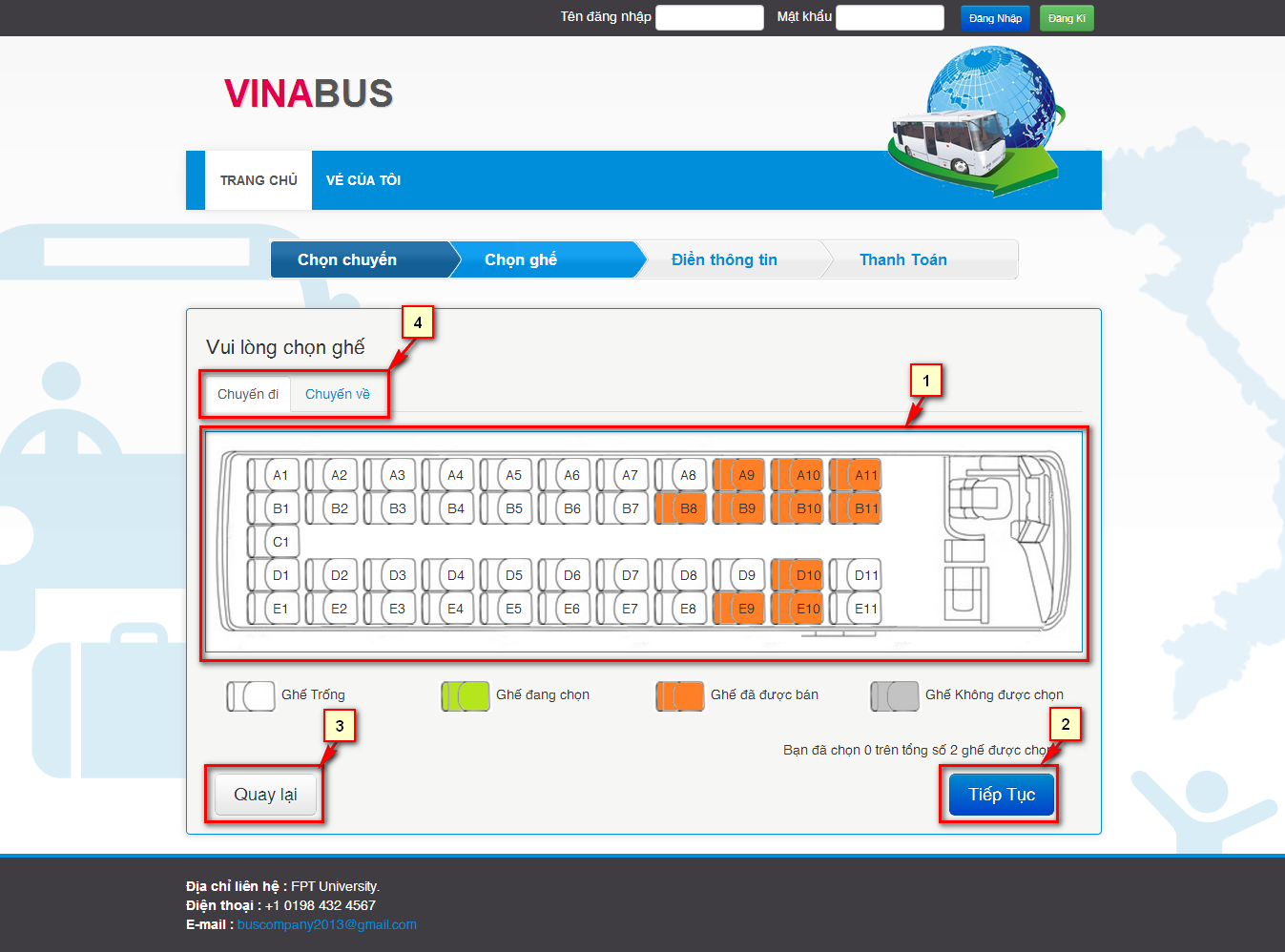


Figure 4‑10 Choose seat screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Seat Map | Display the all seats status in bus selected |
| **2** | Submit – button (Tiếp Tục) | Continue to provide booker’s info page |
| **3** | Back – button (Quay lại) | Cancel booking and back to page search result |
| **4** | Select seat map tabs | Select seat map for one-way or round-trip |

Table 4‑9 Choose seat screen description

#### Provide booker’s information use case



Figure 4‑11 Provide booker’s details use case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC004 | **Use-case Version** | 0.1 |
| **Use-case Name** | Provide booker’s details | | |
| **Author** | Nguyễn Ngọc Sơn, Nguyễn Sơn Trường | | |
| **Date** | 31/1/2013 | **Priority** | High |
| **Actor:** Guest  **Summary:** User provide their information after select seats to create a reservation  **Goal:** Get booker’s details  **Triggers:** User selects seats and clicks button “Tiếp Tục”.  **Preconditions:**  User selected seats in select seat screen.  **Post Conditions:**  Success:   * + Reservation and user’s information is saved.   + Redirect to payment page   Failure: Reservation is not saved. System displays error message. Content of error message will be specified in Exception section.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. Choose seats then click button “Tiếp Tục”  3. Fill information in the form and click “Thanh toán” button. [Alternative 2], [Alternative 3] | 2. Display a form to provide booker’s details [Alternative 1]   * First Name [“Tên”: textbox, min length: 1, max length: 30] * Last Name [“Họ”: textbox, min length: 1, max length: 30] * Phone [“Số điện thoại”: textbox, regular expression: ^[0-9]{10,11}$] * E-mail [“E-mail”: textbox, regular expression: ^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$] * Pay [“Thanh toán”: button] * Back [“Quay Lại”: button]   4. Save reservation and redirect to page payment. [Exception 1][Exception 2][Exception 3][Exception 4][Exception 5] |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Alternative 1]  1. If the user is logged in.  [Alternative 2]  1. User clicks on “Quay lại” button.  [Alternative 3]  1. User clicks on check on seats ‘s checkbox and click button ”Bỏ ghế” | 2. Use the information of that user fulfill all field in form.  2. Redirect to choose seat page.  2. Remove all seats that user checked. Update price of the table reservation info. Show message "Bỏ ghế thành công." [Exception 7] [Exception 8] [Excaption 9] |   **Exceptions:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Exception 1]  1. Set blank on First name field.  [Exception 2]  1. Set blank on Last name field.  [Exception 3]  1. Set blank on Email field.  [Exception 4]  1. Wrong phone number format.  [Exception 5]  1. Wrong email format.  [Exception 6]  1. User don’t check any seats and click button “Bỏ ghế”  [Exception 7]  1. User select all seats and click button “Bỏ ghế”  [Exception 8]  1. User select all seats in onward journey and click button “Bỏ ghế”  [Exception 9]  1. User select all seats in return journey and click button “Bỏ ghế” | 2. Show message "Vui lòng nhập tên."  2. Show message "Vui lòng nhập họ."  2. Show message "Vui lòng nhập địa chỉ email".  2. Show message "Số điện thoại không hợp lệ".  2. Show message "Địa chỉ email không hợp lệ".  2. Show message "Bạn chưa chọn ghế để bỏ."  2. Show message "Không được phép bỏ chọn tất cả các ghế."  2. Show message "Bạn không được phép bỏ chọn tất cả các ghế chuyến đi."  2. Show message "Bạn không được phép bỏ chọn tất cả các ghế chuyến khứ hồi." |   **Relationships:**  Create reservation, Login  **Business Rules:**   * User must fulfill the all field of the form except phone number. * User cannot remove all seats in a journey. | | | |

Table 4‑10 UC004 Provide booker’s details user case specification



Figure 4‑12 Provide information screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Customer’s information – Editable  Lastname (Họ: textbox)  Firstname (Tên: textbox)  Phone (Di động: texbox)  Email (Email: textbox) | Get information of customer. |
| **2** | Payment method (Thanh Toán , Dropdown list) | Show payment method that system support. |
| **3** | User’s selected seats  Table checkbox | Let user delete selected seats. |
| **4** | User reservation info | Show the reservation info. |
| **5** | Submit Reservation(Thanh toán: button) | Redirect to payment page. |
| **6** | Back (Chọn lại ghế button) | Return back to booking page. |

Table 4‑11 Provide customer information screen description

#### Pay use case



Figure 4‑13 Pay use case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC005 | **Use-case Version** | 0.1 |
| **Use-case Name** | Pay | | |
| **Author** | Nguyễn Lương Hải, Nguyễn Sơn Trường | | |
| **Date** | 31/01/2013 | **Priority** | High |
| **Actor:** Guest  **Summary:** User can pay for reservation with Paypal  **Goal:** Guides user through the payment process  **Triggers:** User want to pay for a reservation with Paypal. (User choose payment method “Paypal” and click button “Thanh toán”)  **Pre-conditions:**  Reservation of user is created.  User is not logging in, or is logging in with role customer  **Post-conditions:**  *Success*: Reservation is paid and reservation code is generated for user  *Failure*: Reservation is not paid and show message “Thông tin đặt vé của quý khách đã bị huỷ”.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User choose payment method “Paypal” and clicks on button “Thanh toán”    3. Complete payment process in paypal site[Alternative 1] | 2. Direct user to Paypal site. Display payment details   * Ticket Price [Price: number, label] * Ticket Amount [Quantity: number, label] * Transaction fee [Fee: number, label]     4. Redirect to bus reservation site. Send mail to inform user about the reservation. Show message “Quy trình đặt vé hoàn tất” and reservation details. [Exception 1][Alternative 2]   * Booker’s name [Người đặt: string, label] * Reservation code [Mã đặt vé: string, label] * Phone number [Mã đặt vé: string, label] * Email [Email: string, label] * Departure date [Ngày giờ khởi hành: string, label] * Status [Trạng thái: string, label] * Ticket’s departure station [Từ: string, label] * Ticket’s departure time [Ngày đi: string, label] * Ticket’s arrival station [Đến: string, label] * Ticket’s arrival time [Ngày đến: string, label] * Ticket’s seat numbers [Ghế: string, label] * Ticket’s bus type [Loại xe: string, label] |   **Alternative Scenario:**   | Ator Actions | System Response | | --- | --- | | [Alternative 1]  1. User cancel payment process and return to bus reservation site  [Alternative 2]  1. User takes longer than 15 minutes to complete payment process on paypal site. | 2. Show message “Thông tin đặt vé của quý khách đã được huỷ”  2. Show message “Vì đã quá 15 phút, thông tin đặt vé của quý khách đã bị huỷ”. |   **Exceptions:**  N/A  **Relationships:**  Login, Provide booker’s details  **Business Rules:**  Payment process must be completed within 15 minutes after reservation is created. | | | |

Table 4‑12 UC005 Pay with Paypal Use Case Specification

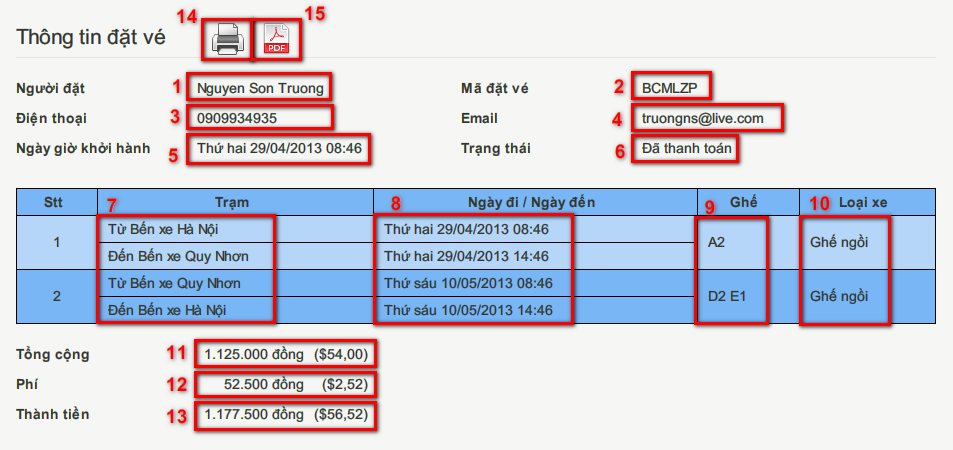


Figure 4‑14- Review reservation details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Booker – read-only text (Người đặt) | Display the name of the booker |
| **2** | Reservation code – read-only text (Mã đặt vé) | Display the unique code representing the reservation |
| **3** | Phone number – read-only text (Điện thoại) | Display the phone number to contact the booker |
| **4** | Email – read-only text (Email) | Display the email address to contact the booker |
| **5** | Departure Date – read-only text (Ngày giờ khởi hành) | Display the departure date and time of the first journey |
| **6** | Status – read-only text (Trạng thái) | Display the status of the reservation |
| **7** | Station – read-only text (Trạm) | Display the departure and arrival stations of the journey |
| **8** | Departure date/Arrival date – read-only text (Ngày đi / Ngày đến) | Display the departure and arrival time of the journey |
| **9** | Seat numbers – read-only text (Số ghế) | Display the booked seat numbers in the journey |
| **10** | Bus type – read-only text (Loại xe) | Display the type of the bus that runs on the journey |
| **11** | Ticket price – read-only text (Giá vé) | Display the price of each ticket |
| **12** | Fee – read-only text (Phí) | Display the fee for online booking |
| **13** | Total amount – read-only text (Tổng cộng) | Display the total amount of the reservation |
| **14** | Print – button | Print the reservation details |
| **15** | Print to pdf – button | Print the reservation details to a pdf file |

Table 4‑13 Review reservation details screen description

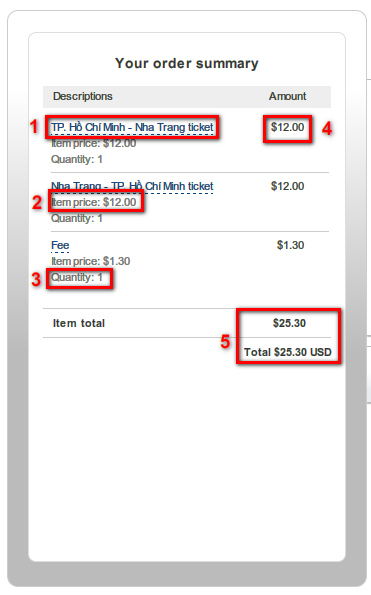


Figure 4‑15- Payment details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Item name – Read-only field | Display name of the item (example: ticket from Ho Chi Minh City to Nha Trang or transaction fee) |
| **2** | Item price - Read-only field (Item price) | Display the price of item |
| **3** | Quantity (Quantity) - Read-only field | Display the quantity of items |
| **4** | Amount (Amount) – Read-only field | Display amount of an item |
| **5** | Total amount (Total amount) - Read-only field | Display total amount that must be paid |

Table 4‑14 Paypal site screen’s Controls

#### View reservation history use case



Figure 4‑16- View Reservation List use case diagram

* 1. **View reservation history**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC006 | **Use-case Version** | 0.1 |
| **Use-case Name** | View Reservation List | | |
| **Author** | Nguyễn Lương Hải, Nguyễn Sơn Trường | | |
| **Date** | 31/01/2013 | **Priority** | High |
| **Actor:** Customer, Staff  **Summary:** User views list of past reservations.  **Goal:** Show all past reservations made by the user  **Triggers:** User click “Vé của tôi” link in page header  **Pre-conditions:**  User has at least one reservation.  User logins with “Customer” or “Staff” role  **Post-conditions:**  *Success*: The reservations of customer are listed  *Failure*: Show message “Quý khách chưa đặt vé lần nào”  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User click “Vé của tôi” link | 2. Display the list of reservations made by user[Alternative 1]   * Number [Stt: number, label] * Journey [Chặng: string, label] * Book time [Ngày đặt: string, label] * Departure date [Ngày giờ khởi hành: string, label] * Details [Chi tiết: string, link] * Cancel [Huỷ: string, link] [Alternative 2] [Alternative 3] [Alternative 4] [Alternative 5] [Alternative 6] |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Alternative 1]  1. User hasn’t made any reservation  [Alternative 2]  1. Departure date is in the past  [Alternative 3]  1. Reservation has been cancelled  [Alternate 4]  1. Reservation is locked  [Alternate 5]  1. Reservation has not been paid for  [Alternate 6]  1. Reservation has been invalidated | 2. Display message “Quý khách chưa đặt vé lần nào”    2. Display Departed [Đã đi: string, label]    2. Display Cancelled [Đã huỷ: string, label]  2. Display Pending [Chờ xuất phát: string, label]  2. Display Unpaid [Chưa thanh toán: string, label]  2. Display Invalid [Hết hiệu lực: string, label] |   **Exceptions:**  N/A  **Relationships:**  Login  **Business Rules:**   * A reservation is invalidated if it is not paid within 15 minutes. * A reservation is locked 5 days before departure date. Locked reservations can’t be cancelled. | | | |

Table 4‑15 UC006 View Reservation List Use Case Specification

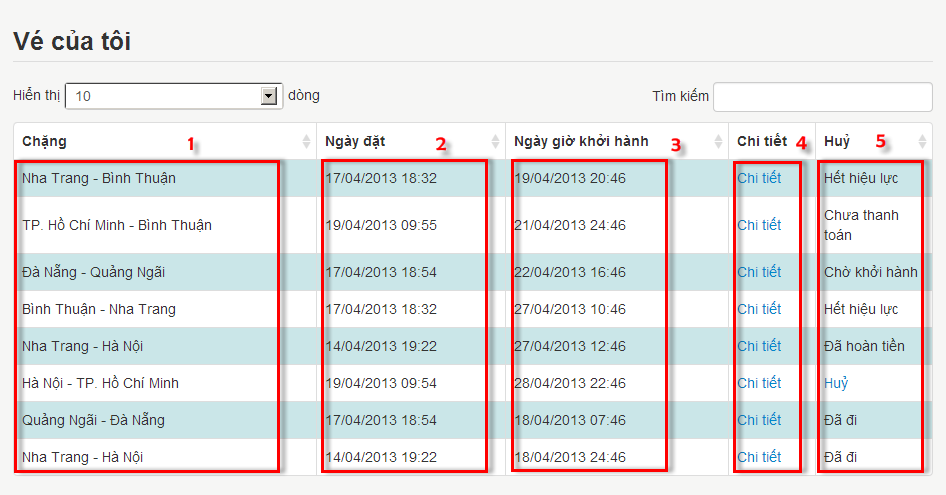


Figure 4‑17 Reservation List screen description

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Journey - Read-only field (Chặng) | Display the starting point and destination of the journey |
| **2** | Book Date – read-only field (Ngày đặt) | Display the book time of the reservation |
| **3** | Departure Date - Read-only field (Ngày giờ khởi hành) | Display the departure date and time of the journey |
| **4** | Details - Link (Chi tiết) | Display reservation details |
| **5** | Cancel - Link (Hủy) | Cancel reservation |

Table 4‑16: Reservation List screen’s Controls

* 1. **View reservation details**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC007 | **Use-case Version** | v0.1 |
| **Use-case Name** | View reservation details | | |
| **Author** | Nguyễn Lương Hải, Nguyễn Sơn Trường | | |
| **Date** | 31/01/2013 | **Priority** | High |
| **Actor:** Guest  **Summary:** User views detailed information of a reservation  **Goal:** Show user necessary information about the reservation  **Triggers:** User wants to view reservation details. (Customer click “Chi tiết” in Reservation List screen, or user input reservation code in Search Reservation)  **Pre-conditions:**  User has a reservation.  **Post Conditions:**  *Success*: Displays reservation details  *Failure*: Display message: “Không tìm thấy thông tin đặt vé. Xin vui lòng kiểm lại mã đặt vé và địa chỉ email.”  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. Click “Chi tiết” in Reservation List screen  Or enter reservation code and click “Hiển thị” button in the Search Reservation screeen | 2. Display reservation details[Exception 1] [Exception 2] [Exception 3] [Exception 4]   * Booker’s name [Người đặt: string, label] * Reservation code [Mã đặt vé: string, label] * Phone number [Mã đặt vé: string, label] * Email [Email: string, label] * Departure date [Ngày giờ khởi hành: string, label] * Status [Trạng thái: string, label] * Ticket’s departure station [Từ: string, label] * Ticket’s departure time [Ngày đi: string, label] * Ticket’s arrival station [Đến: string, label] * Ticket’s arrival time [Ngày đến: string, label] * Ticket’s seat numbers [Ghế: string, label] * Ticket’s bus type [Loại xe: string, label]   Display button to cancel reservation [Alternative 1] [Alternative 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Actor Actions | System Response | | | [Alternative 1]  1. Reservation is locked, or one of the journey has departed  [Alternative 2]  1. The reservation was cancelled | 2. Does not display cancel button  2. Does not display cancel button. Display refunded amount |   **Exceptions:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Exception 1]  1. Reservation code and email does not match any reservation  [Exception 2]  1. Reservation was made by a registered user, but user has not logged in    [Exception 3]  1. User has logged in, but reservation was made by a different registered user    [Exception 4]  1. User has not entered a reservation code, or the reservation code is a string of 6 alpha-numeric characters | 2. Display message “Không tìm thấy thông tin đặt vé. Xin vui lòng kiểm lại mã đặt vé và địa chỉ email.”    2. Display message “Quý khách cần đăng nhập để xem thông tin đặt vé này.”    2. Display message “Quý khách không có quyền truy cập thông tin đặt vé này.”    2. Display message “Xin vui lòng nhập mã đặt vé hợp lệ.” |   **Relationships:**  Login, Create Reservation and View Reservation List.  **Business Rules:**  For reservations made by registered users, only the booker can access the reservation | | | |

Table 4‑17 UC007 View Reservation Details Use Case Specification

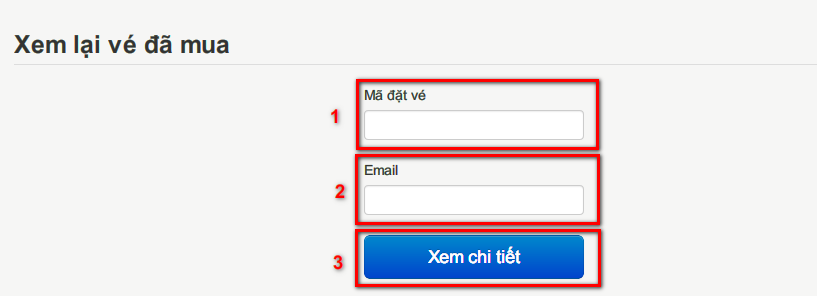


Figure 4‑18 Search Reservation screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Reservation code - Textbox (Mã đặt vé) | Get code of the reservation to be viewed |
| **2** | Email – Textbox (Email) | Get the email of the booker |
| **2** | Display - button (Xem chi tiết) | Display reservation details |

Table 4‑18 Search Reservation screen’s controls

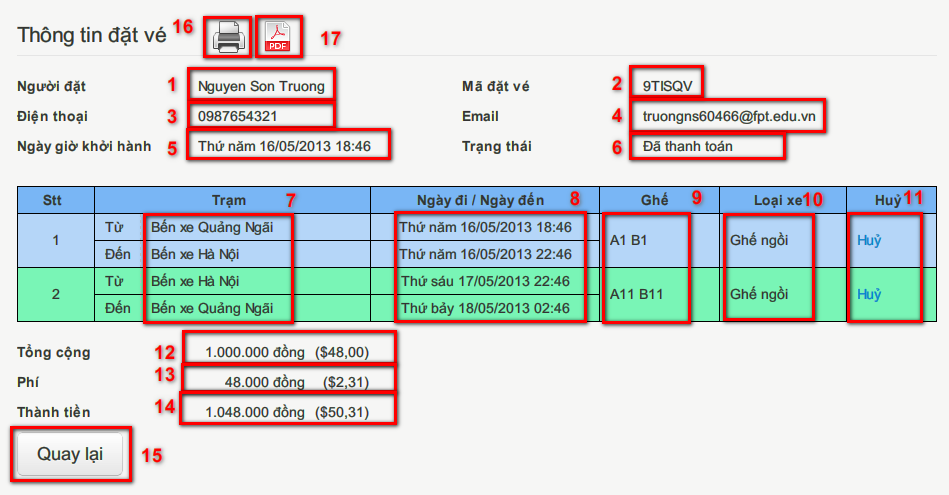


Figure 4‑19 Reservation Details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Booker – read-only text (Người đặt) | Display the name of the booker |
| **2** | Reservation code – read-only text (Mã đặt vé) | Display the unique code representing the reservation |
| **3** | Phone number – read-only text (Điện thoại) | Display the phone number to contact the booker |
| **4** | Email – read-only text (Email) | Display the email address to contact the booker |
| **5** | Departure Date – read-only text (Ngày giờ khởi hành) | Display the departure date and time of the first journey |
| **6** | Status – read-only text (Trạng thái) | Display the status of the reservation |
| **7** | Station – read-only text (Trạm) | Display the departure and arrival stations of the journey |
| **8** | Departure date/Arrival date – read-only text (Ngày đi / Ngày đến) | Display the departure and arrival time of the journey |
| **9** | Seat numbers – read-only text (Số ghế) | Display the booked seat numbers in the journey |
| **10** | Bus type – read-only text (Loại xe) | Display type of the bus that runs on the journey |
| **11** | Cancel – link (Hủy) | Click to cancel the journey |
| **12** | Ticket price – read-only text (Giá vé) | Display the price of each ticket |
| **13** | Fee – read-only text (Phí) | Display the fee for online booking |
| **14** | Total amount – read-only text (Tổng cộng) | Display the total amount of the reservation |
| **15** | Back – button (Quay lại) | Return to last screen |
| **16** | Print – button | Print this reservation information |
| **17** | Export to PDF – button | Export this reservation information to PDF file |

Table 4‑19 Reservation Details screen’s Control

* 1. **Cancel reservation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | | |
| **Use-case No.** | UC008 | **Use-case Version** | | v0.1 |
| **Use-case Name** | Cancel Reservation | | | |
| **Author** | Nguyễn Lương Hải, Nguyễn Sơn Trường | | | |
| **Date** | 31/01/2013 | **Priority** | Medium | |
| **Actor:** Guest, Staff, Customer  **Summary:** User cancels reservation.  **Goal:** Guide user through cancel reservation process  **Triggers:** User want to cancel reservation. (User click “Huỷ” button in Reservation List screen or Reservation Details screen).  **Pre-conditions:**  User has a reservation and cancel day is at least 5 days before departure day.  **Post-conditions:**  *Success*: The reservation is cancelled. The status of the reservation is updated to “Đã hoàn tiền” and cancelling reason is saved as “Khách hàng hủy”. An informing e-mail is sent to user.  *Failure*: The reservation is not cancelled.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User click link “Huỷ”    3. Click button “Hủy” [Alternative 1]    5. Click button “Đóng” or click outside the popup | 2. Display confirm popup with refund details [Exception 1] [Exception 2]   * Refund amount [Số tiền hoàn lại: string, label]   4. Cancel the reservation, display message “Chuyến đi đã được huỷ thành công.” Hide button “Hủy”.  6. Close popup and refresh current page |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Alternative 1]  1. Click “Đóng” button. | 2. Close popup. |   **Exceptions:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Exception 1]  1. Reservation was made by a registered user, but user has not logged in    [Exception 2]  1. User has logged in, but reservation was made by a different registered user | 2. Display message “Quý khách cần đăng nhập để hủy thông tin đặt vé này.”  Return to homepage    2. Display message “Quý khách không có quyền truy cập thông tin đặt vé này.”  Return to homepage |   **Relationships:**  Login, Create Reservation, View Reservation List, View Reservation Details  **Business Rules:**   * If the reservation is cancelled more than 10 days before the first journey departure date, 70% of the paid amount will be refunded. * Else, if the reservation is cancelled more than 5 days before the first journey departure date, 30% of the paid amount will be refunded. * The reservation cannot be cancelled otherwise. | | | | |

Table 4‑20 UC008 Cancel Reservation Use Case Specification

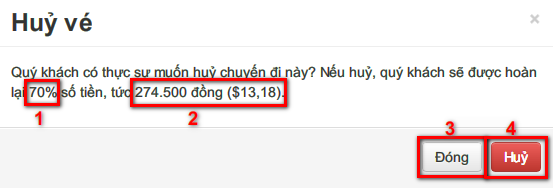


Figure 4‑20 Cancel Reservation Popup

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Refund percent – read-only text (Phần trăm hoàn lại) | Percent of the amount to be refunded |
| **2** | Refund amount – read-only text (Số tiền hoàn lại) | Display the amount to be refunded |
| **3** | Confirm – button (Hủy) | Cancel the reservation |
| **4** | Close – button (Đóng) | Cancel the process |

Table 4‑21 Cancel Reservation popup description

#### Manage route use case

Figure 4‑21 Manage route use case

1. **View route details**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC009 | **Use-case Version** | 0.1 |
| **Use-case Name** | View route details | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** Usercan view the entire available route and its detail.  **Goal:** Help user control the entire routes.  **Triggers:** User wants to view the all the current route. (User clicks “Tuyến đường” in “Quản lý” list in menu bar)  **Preconditions:**  User logins into system with role Staff.  There are at least one available route.  **PostConditions:**  *Success*: List of all available route is shown  *Failure*: Display message: “There is no available route”.  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. At menu bar, user clicks on “Tuyến đường” in “Quản lý” list.   1. User click on button “View Detail”   5. User choose another bus type in drop down list [Alternative 2]  9. User click on button “Show the reverse route” [Alternative 3] | 2. Route list is shown. [Alternative 1]   * Add new route [button] * Route name [label, String] * Total duration [label, String] * View Details [button] * Deactivate [buton]  1. Route details is shown.  * Route name [label, String] * Edit route price [button] * Assign bus to route [button] * Add new schedule [button] * Bus type [drop down list] * Date [date time picker, initial value: current date, min: current date, max: current date + 3 months] * Segment name [label, String] * Travel time [label, String] * Price (VND) [label, double] * Return to route list [button] * Show the reverse route [button]   6. Price of each segment and total is shown by selected bus type  10. Route details screen of reverse route is shown. (Example: reverse route of Hồ Chí Minh City – Hà Nội is Hà Nội – Hồ Chí Minh City) |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. There is no available route.  [Alternative 2]  1. User choose another date in date time picker.  [Alternative 3]  1. User click “Return to route list” | 2. Display message: “There is no available route”.  2. Travel time, fare of each segment and total on selected date are shown.  2. Return to route list screen. |   **Exceptions:** N/A  **Relationships:**  Login, Manage route.  **Business Rules:**  Each route always have a reverse route. In reverse route, all segments are reversed. Example: Route A – B has 2 segments: A – C and C – B, reverse route of route A – B has 2 segment: B – C and C – A. | | | |

Table 4‑22 UC009 View route details Use case Specification

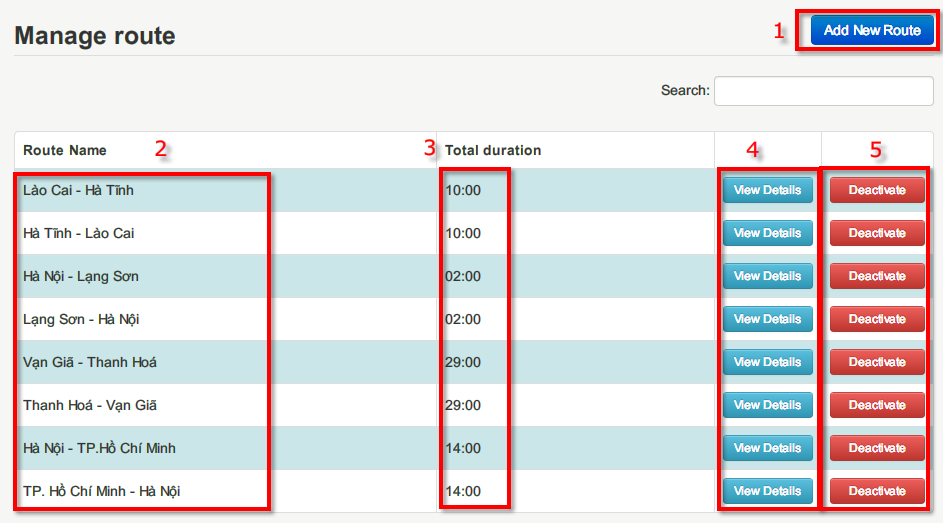


Figure 4‑22 Manage route (Route list) screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Add new route – button | Add a new route to list |
| **2** | Route name – label | Name of the route |
| **3** | Total duration – label | Estimated travel time of the route |
| **4** | View details – button | View route details |
| **5** | Deactivate – button | Deactivate current route |

Table 4‑23 Manage route screen description

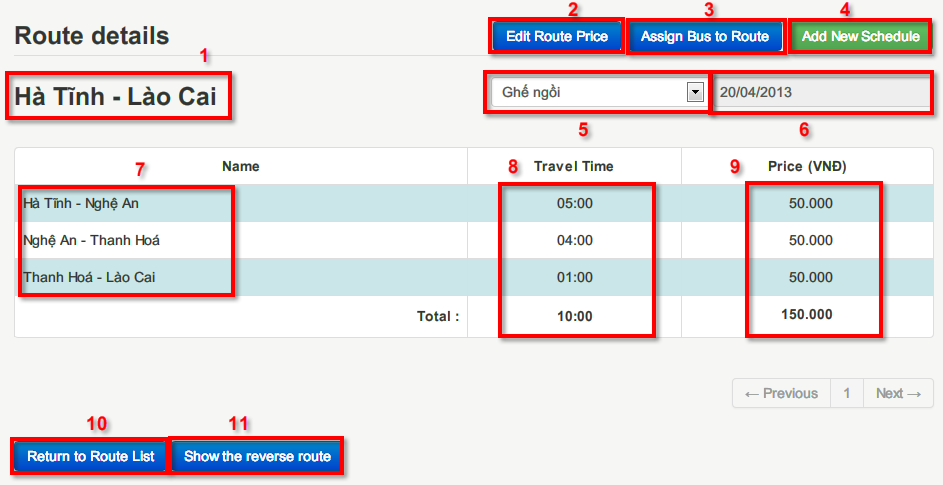


Figure 4‑23 Route details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Route name – label | Name of the route |
| **2** | Edit route price – button | Edit fare of the route |
| **3** | Assign bus to route – button | Assign bus to the route |
| **4** | Add new schedule – button | Scheduling for the route |
| **5** | Bus type – drop down list | List of bus type, choose a type to display fare for that type. |
| **6** | Date – date time picker | Choose date to display fare and travel time in the date |

Table 4‑24 Route detail screen description

1. **Add route**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC010 | **Use-case Version** | 0.1 |
| **Use-case Name** | Add route | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can add new route.  **Goal:** Help user to add new route to the system.  **Triggers:** User want to add a new route. (User click “Add new route” in Manage route page.)  **Preconditions:**  User login into system with role Staff.  **PostConditions:**  *Success*: New route, its reverse route and segments are created.  *Failure*: Display message: “Route is existed. Please verify.”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click on “Add new route” button in manage route page  3. User choose a start location.  5. User choose start station and end location.  7. User choose end station  9. User input duration and click button Add  11. User click button Save. [Alternative 3] | 2. Modal “Add new route” is shown   * Start location [drop down list] * Start station [drop down list] * End location [drop down list] * End station [drop down list] * Duration [textbox – hh:mm, default value: 01:00] * Add [buton - disabled] * Save [button - disabled] * Cancel [button]   4. List of start stations and end location is updated matching with start location.  6. List of end stations is updated matching with end location.  8. Enable button Add. [Alternative 1]  10. Segment is added and shown in the same modal. Set start location = end location, start station = end station. Disable drop down list start location and start station. Enable save button [Alternative 2]  12. Save this route and its segments. Create reverse route then save it and its segments. [Exception 1] |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. Segment which has selected start station and end station is existed.  [Alternative 2]  1. Number of segment added is already 5.  [Alternative 3]  1. User click “Cancel” | 2. Duration of segment is displayed and duration textbox is disabled.  2. Show message: “Sorry, you cannot add more than 5 segments per route.”  2. Close modal, return to route list screen. |   **Exceptions:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. A same route with same segments is existed | 2. Show message: “Route existed. Please verify.” |   **Relationships:**  Login, Manage route, View route list  **Business Rules:**   * Minimum number of segments per route is 1. Maximum is 5. * When a route is added, its reverse route is also added. * If segment is not existed yet, its fare is set: 50,000 VND. | | | |

Table 4‑25 UC010 Add route Use case Specification

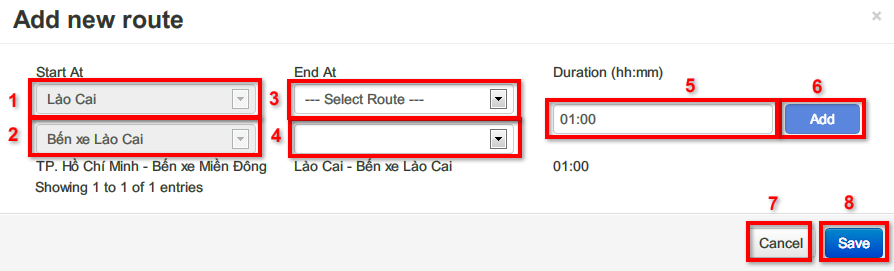


Figure 4‑24 Add new route screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Start location – drop down list | Start location of a segment in route (Example route A – B has segments A – C and C – B, then A and C are two start location of route A – B) |
| **2** | Start station – drop down list | Start station of a segment in route |
| **3** | End location – drop down list | End location of a segment in route |
| **4** | End station – drop down list | End station of a segment in route |
| **5** | Duration – text box | Estimated travel time of a segment |
| **6** | Add – button | Click to add a segment into new route |
| **7** | Save – button | Click to save new route and its reverse route |
| **8** | Cancel – button | Click to close Add new route modal |

Table 4‑26 Add new route screen description

1. **Add fare to route**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC011 | **Use-case Version** | 0.1 |
| **Use-case Name** | Add fare to route | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can add new fare for a route to create tariff.  **Goal:** Help user to add new fare for route.  **Triggers:** User want to add fare for route. (User click “Edit route price” in Route details page.)  **Preconditions:**   * User login into system with role Staff. * The route user want to add fare is existed and active.   **Post Conditions:**  *Success*: Fare of each segment in route is saved.  *Failure*: N/A  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click on “Edit route price” button in route details page  3. User choose a date, bus type, input price and click Save. [Alternative 2][Alternative 3] | 2. Modal “Edit route price” is shown   * Valid date [date time picker – min: current date, max: current date + 3 months] * Bus type [drop down list] * Start location [label] * End location [label] * Price [textbox – number][Alternative 1] * Add price for reverse route [checkbox, default: unchecked] * Save [button] * Close [button]     4. Fare of the route with selected bus type on selected date is saved. |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. Price of segment is existed before.  [Alternative 2]  1. User choose Add price for reverse route and click Save button  [Alternative 3]  1. User click “Close”. | 2. Price of segment is shown on Price textbox.  2. Fare of the route and its reverse route with selected bus type on selected date is saved.  2. Close modal, return to route details screen. |   **Exceptions:** N/A  **Relationships:**  Login, Manage route, View route list, View route details  **Business Rules:**  Minimum price for each segment: 50,000 VND. Maximum: 9,999,000 VND. | | | |

Table 4‑27 UC011 Add fare to route Use case Specification

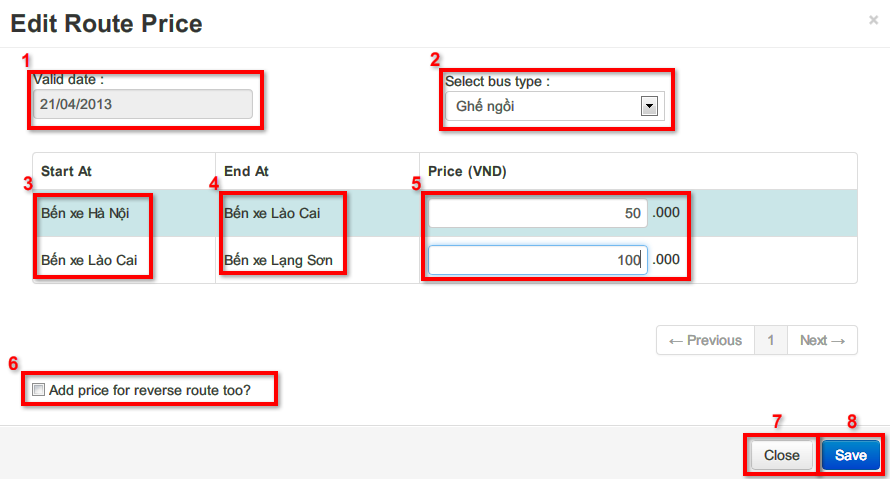


Figure 4‑25 Edit route price screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Valid date – date time picker | The date when the fare is applied. (Example: Route A – B has fare of 500 VND and valid date is 20/04/2013 and 600 VND with valid date is 20/05/2013, so from 20/04 to 20/5, fare of route A – B is 500VND and from 20/5 to the future, fare of route A – B is 600VND) |
| **2** | Bus type – drop down list | Bus type which fare is applied |
| **3** | Start location – label | Start location of segment which fare is applied |
| **4** | End location – label | End location of segment which fare is applied |
| **5** | Price – text box | Price of each segment |
| **6** | Add price for reverse route – checkbox | Confirm to add the same price for reverse route. |
| **7** | Save – button | Click to save route price. |
| **8** | Close – button | Click to close Edit route price modal |

Table 4‑28 Edit route price screen description

1. **Assign bus to route**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC012 | **Use-case Version** | 0.1 |
| **Use-case Name** | Assign bus to route | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can assign buses to route.  **Goal:** Help user to assign buses to route so they can manage buses by route.  **Triggers:** User want to assign buses to route. (User click “Assign bus to route” in Route details page.)  **Preconditions:**   * User login into system with role Staff. * The route user want to add bus is existed and active. * There is at least one bus which is not assigned to any route.   **Post Conditions:**  *Success*: Buses are assigned to the route.  *Failure*: Display message “Bus has been assigned to another route.”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click on “Assign bus to route” button in route details page  3. User choose bus type   1. User choose bus in selected bus list and click button Add. [Alternative 3]   7. User click button Save [Alternative 4] | 2. Modal “List Bus In Route” is shown   * Bus type [drop down list] * Select bus [drop down list] * Add [button] * Plate number [label] * Delete [button] [Alternative 1] * Close [button] * Save [button]   4. List of buses which have selected bus type is displayed in select bus drop down list [Alternative 2]     1. Bus plate number is displayed in table.   8. Changes in bus list is saved (Delete bus or add new bus to route) [Exception 1] |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. The bus has been assigned to a schedule in the route.  [Alternative 1]  1. There is no available bus.  [Alternative 2]  1. User click button delete  [Alternative 3]  1. User click “Close”. | 2. Button Delete is disabled.  2. Display message “There is no free bus at the moment” and disable button Add.  2. The bus is removed from the list.  2. Close modal, return to route details screen. |   **Exceptions:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Exception 1]  1. The bus has been assigned to another route | 2. Show message “Bus has been assigned to another route” |   **Relationships:**  Login, Manage route, View route list, View route details  **Business Rules:**   * Cannot remove a bus from a route if that bus has been assigned to run on a journey for that route in the future. * When user assign a bus to one route, it is also assigned to reverse route. * A bus can be assign to only one route and its reverse route. | | | |

Table 4‑29 UC012 Assign bus to route Use case Specification

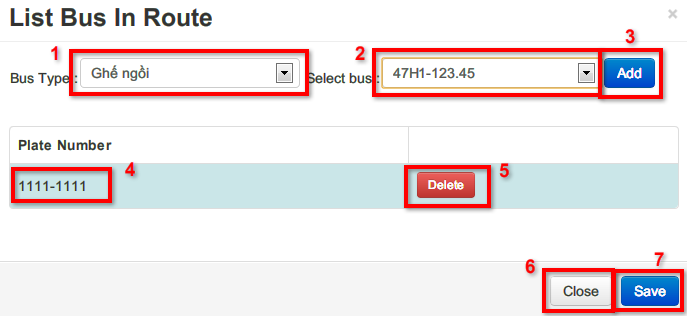


Figure 4‑26 List bus in route screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Bus type – drop down list | All bus types which have free buses |
| **2** | Select bus – drop down list | List of all free buses that haven’t been assigned to any route. |
| **3** | Add – button | Add the bus to current route. |
| **4** | Plate number – label | Plate number of buses assigned to current route. |
| **5** | Delete – button | Remove a bus from current route |
| **6** | Save – button | Click to save changes (Add new bus to route or remove bus from route) |
| **7** | Close – button | Click to close current modal |

Table 4‑30 List bus in route screen description

1. **Deactivate route**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC013 | **Use-case Version** | 0.1 |
| **Use-case Name** | Deactivate route | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can deactivate a route.  **Goal:** Help user to deactivate a route.  **Triggers:** User want to deactivate. (User click “Deactivate” in route list page.)  **Preconditions:**   * User login into system with role Staff. * The route user want to deactivate is existed and active. * The route has no bus and schedule.   **Post Conditions:**  *Success*: Route is deactivated and cannot be edited (Cannot add route, price or schedule)  *Failure*: Display message “Cannot deactivate this route.”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click on “Deactivate” button in route list page | 2. Route is deactivated and cannot be edited. Button Deactivate is hidden and button Activate is shown. [Alternative 1] |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. The route has assigned buses and schedules in the future. | 2. Display message “Cannot deactivate this route”. |   **Exceptions:** N/A  **Relationships:**  Login, Manage route, View route list  **Business Rules:**   * Cannot deactivate a route if it has assigned buses and schedules in the future. | | | |

Table 4‑31 UC013 Deactivate route use case Specification

#### Manage bus use case



Figure 4‑27 Manage bus use case

1. **View bus list**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC014 | **Use-case Version** | 0.1 |
| **Use-case Name** | View bus list | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can view bus list.  **Goal:** Help user to view bus list and manage bus.  **Triggers:** User want to view bus list. (User click “Xe bus” in “Quản lý” list.)  **Preconditions:**   * User login into system with role Staff. * There is at least one bus.   **Post Conditions:**  *Success*: List of buses is shown  *Failure*: Display message “There is no bus.”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click on “Xe bus” in “Quản lý” list. | 2. List of bus is displayed [Alternative 1]   * Add new bus [button] * Bus type [label: String] * Plate number [label: String] * Assigned route [label: String] * Status [label: String] * Delete [button] |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. There is no bus in the system. | 2. Display message “There is no bus”. |   **Exceptions:** N/A  **Relationships:**  Login, Manage bus  **Business Rules:** N/A | | | |

Table 4‑32 UC014 View bus list Use case Specification

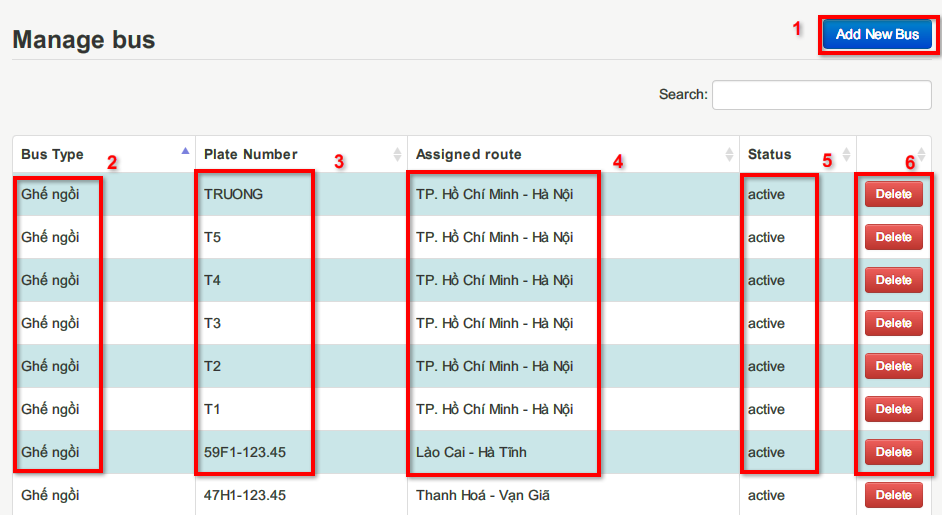


Figure 4‑28 Manage bus screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Add new bus – button | Add new bus to the system |
| **2** | Bus type – label | Bus type of the bus |
| **3** | Plate number – label | Plate number of the bus |
| **4** | Assigned route – label | The route that the bus is assigned |
| **5** | Status – label | Status of the bus: active or inactive |
| **6** | Delete – button | Remove a bus from the system |

Table 4‑33 Manage bus screen description

1. **Add new bus**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC015 | **Use-case Version** | 0.1 |
| **Use-case Name** | Add new bus | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can add new bus to the system.  **Goal:** Help user to add a new bus.  **Triggers:** User want to add a new bus. (User click “Add new bus” in Manage bus screen.)  **Preconditions:**   * User login into system with role Staff.   **Post Conditions:**  *Success*: Bus is added to the system.  *Failure*: Bus is not added. Display message “Bus plate number is existed!”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click “Add new bus” in Manage bus screen.  3. User enter plate number and click button Save [Alternative 1] | 2. Add new bus modal is shown.   * Plate number [textbox – maxlength:10] * Bus type [drop down list] * Save [button] * Close [button]   4. Bus is saved into the system. [Exception 1][Exception 2] |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. User click button Close. | 2. Close current modal. Return to bus list. |   **Exceptions:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Exception 1]  1. Bus plate text box is blank.  [Exception 2]  1. Bus plate number is existed in system | 2. Display message: “Please enter bus plate number”.  2. Display message: “Bus plate number is existed!” |   **Relationships:**  Login, Manage bus, View route list  **Business Rules:**  Every bus has a unique bus plate number. | | | |

Table 4‑34 UC015 Add new bus Use case Specification



Figure 4‑29 Add new bus screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Plate number – text box | Plate number of the bus |
| **2** | Bus type – drop down list | Bus type of the bus |
| **3** | Close – button | Close the modal |
| **4** | Save – button | Save bus to system. |

Table 4‑35 Add new bus screen description

1. **Delete bus**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC016 | **Use-case Version** | 0.1 |
| **Use-case Name** | Delete bus | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can delete bus from the system.  **Goal:** Help user to delete a bus.  **Triggers:** User want to delete bus. (User click “Delete” in Manage bus screen.)  **Preconditions:**   * User login into system with role Staff. * There is at least one bus.   **Post Conditions:**  *Success*: Bus is deleted from the system.  *Failure*: Bus is not deleted. Display message “Cannot delete this bus because it has been scheduled.”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click “Delete” in Manage bus screen.  3. User click button Delete [Alternative 1] | 2. Confirm Delete modal is shown.   * Delete [button] * Close [button]   4. Bus is deleted from the system. [Exception 1][Exception 2] |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. User click button Close. | 2. Close current modal. Return to bus list. |   **Exceptions:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Exception 1]  1. Bus is assigned to a route.  [Exception 2]  1. Bus is assigned to a schedule | 2. Display message: “Cannot delete this bus because it has been assigned to a route”.  2. Display message: “Cannot delete this bus because it has been scheduled!” |   **Relationships:**  Login, Manage bus, View route list  **Business Rules:**   * Cannot delete bus that has been assigned to a route. * Cannot delete bus that has been scheduled. | | | |

Table 4‑36 UC016 Delete bus Use case specification

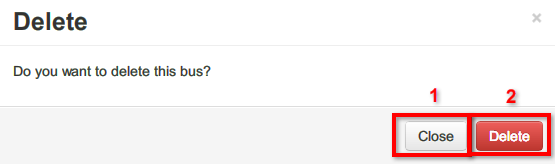


Figure 4‑30 Delete bus confirm screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Close – button | Close this modal and not delete bus |
| **2** | Delete – button | Confirm to delete the bus |

Table 4‑37 Delete bus confirmation screen description

#### Manage segment use case



Figure 4‑31 Manage segment use case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC017 | **Use-case Version** | 0.1 |
| **Use-case Name** | Manage segment | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can view list of segments and edit travel time of segment.  **Goal:** Help user to manage segment.  **Triggers:** User want to view list of segments. (User click “Chặng đi” in “Quản lý” list.)  **Preconditions:**   * User login into system with role Staff. * There is at least one segment.   **Post Conditions:**  *Success*: Show segment list.  *Failure*: Display message “There is no segment.”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click “Chặng đi” in “Quản lý” list.  3. User choose bus type.  5. User choose different valid date.   1. User click on button Edit   9. User choose valid date, input travel time and click button Save [Alternative 2] | 2. Segment list is shown. [Alternative 1]   * Bus type [drop down list] * Valid date [date time picker – min: current date, max: current date + 3 months] * Start at [label - String] * End at [label - String] * Duration [label - String] * Price [label – double] * Edit [button]   4. Segment price is displayed based on selected bus type.  6. Segment price and duration is displayed based on valid date.   1. Edit segment modal is displayed  * Valid date [date time picker – min: current date, max: current date + 3 months] * Duration [textbox – hh:mm] * Close [button] * Save [button]   10. Travel time of segment which is applied from valid date is saved. [Exception 1] |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. There is no segments.  [Alternative 2]  1. User click button Close. | 2. Display message: “There is no segment”  2. Close current modal and return to segment list. Travel time is not saved. |   **Exceptions:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Exception 1]  1. User did not input travel time or travel time is < 1 hour | 2. Display message: “Please input correct travel time” |   **Relationships:**  Login  **Business Rules:**   * Valid date is the date which from that date the price or duration of the segment is applied. * Travel time must be at least 1 hour. | | | |

Table 4‑38 UC017 Manage segment Use case Specification

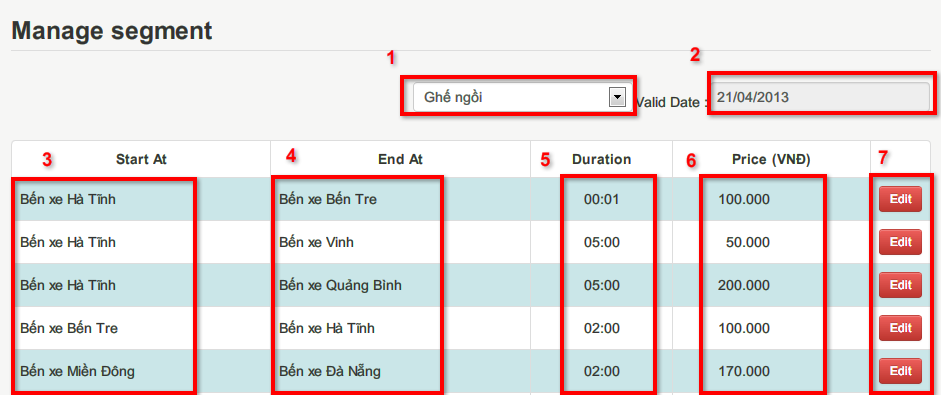


Figure 4‑32 Manage segment screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Bus type – drop down list | List of all bus type |
| **2** | Valid date – date time picker | The date when the price or duration of the segment is applied. |
| **3** | Start at – label | Start station of segment |
| **4** | End at – label | End station of segment |
| **5** | Duration – label | Estimated travel time of segment in valid date |
| **6** | Price – label | Price of segment by bus type in valid date |
| **7** | Edit – button | Edit travel time of segment |

Table 4‑39 Manage segment screen description

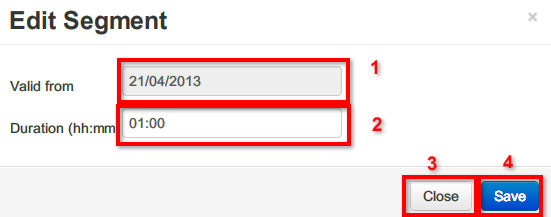


Figure 4‑33 Edit segment screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Valid from – date time picker | The date when the price or duration of the segment is applied. |
| **2** | Duration – text box (hh:mm) | Estimated travel time of segment in valid date |

Table 4‑40 Edit segment screen description

#### Manage schedule use case



Figure 4‑34 Manage schedule use case

1. **Add new schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC018 | **Use-case Version** | 0.1 |
| **Use-case Name** | Add new schedule | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can add schedule for route.  **Goal:** Help user to add a new schedule.  **Triggers:** User want to add a new schedule. (User click “Add new schedule” in Route details screen.)  **Preconditions:**   * User login into system with role Staff. * There is at least one route and one free bus.   **Post Conditions:**  *Success*: Schedule is added.  *Failure*: Schedule is not added. Display message “There're no bus available at this time. Please select other time or click cancel!”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click “Add new schedule” in Route details screen.  3. User choose from date and bus type  5. User choose bus plate number and click button save changes [Alternative 3] | 2. Add new schedule modal is shown.   * From date [date time picker – min: current date + 1 day, max: current date + 3 months] * To date [text box - disabled] * Bus type [drop down list] * Bus plate number [drop down list] * Save changes [button] * Cancel [button]   4. List of bus plate number is loaded based on from date and bus type. To date is calculated and displayed base on from date. [Alternative 1]  6. Schedule is saved. List of schedule is shown. |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. There is no available bus at the selected time.  [Alternative 2]  1. User click button Close. | 2. Display message “There're no bus available at this time. Please select other time or click cancel!”  2. Close current modal and return to route details screen. |   **Exceptions:** N/A  **Relationships:**  Login, Manage schedule, View route details.  **Business Rules:**  Only buses which are located at the start station of the route can be scheduled. | | | |

Table 4‑41 Add new schedule Use case Specification

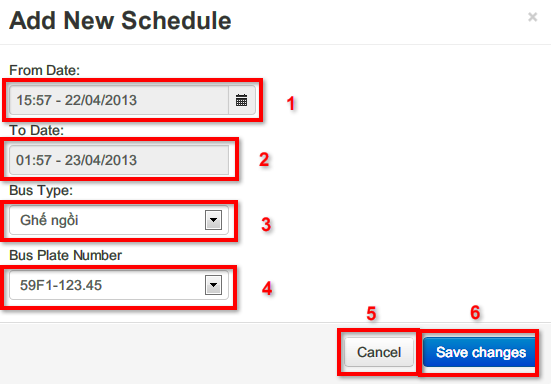


Figure 4‑35 Add new schedule screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | From date – date time picker | The date when the bus starts its journey. |
| **2** | To date – text box (disabled) | The date when the bus ends its journey. |
| **3** | Bus type – drop down list | Type of the bus which is scheduled. |
| **4** | Bus plate number – drop down list | Plate number of the bus that is scheduled. |
| **5** | Cancel – button | Close the modal and back to route details screen |
| **6** | Save changes – button | Save schedule to system. |

Table 4‑42 Add new schedule screen description

1. **View schedule list**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC019 | **Use-case Version** | 0.1 |
| **Use-case Name** | View schedule list | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can view schedule list.  **Goal:** Help user to manage schedule.  **Triggers:** User want to view schedule list. (User click “Lịch xe chạy” in “Quản lý” list or access route details screen.)  **Preconditions:**   * User login into system with role Staff. * There is at least one schedule.   **Post Conditions:**  *Success*: Schedule list is displayed.  *Failure*: Display message “There is no schedule.”  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click “Lịch xe chạy” in “Quản lý” list [Alternative 1]. | 2. Schedule list is displayed. [Alternative 2]   * Route name [label – String] * Bus plate [label – String] * From date [label – Date time] * To date [label – Date time] * Delete [button] |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. User access route details screen.  [Alternative 2]  1. There is no schedule. | 2. Schedule list of the route is displayed [Alternative 2]   * Bus plate [label – String] * Bus type [label – String] * From date [label – Date time] * To date [label – Date time] * Delete [button]   2. Display message “There is no schedule” |   **Exceptions:** N/A  **Relationships:**  Login, Manage schedule, View route details.  **Business Rules:** N/A | | | |

Table 4‑43 UC019 View schedule list Use case Specification

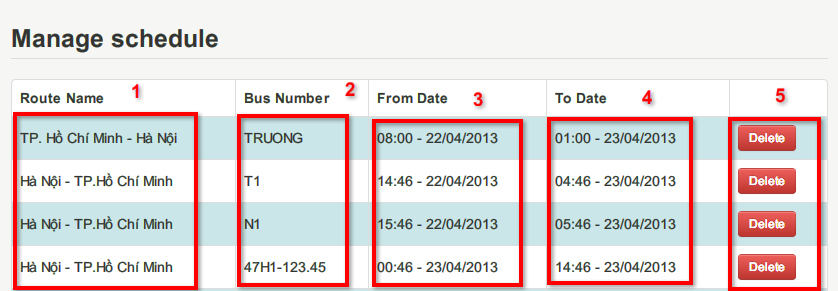


Figure 4‑36 Manage schedule screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Route name – label | Name of the route that the bus is scheduled on. |
| **2** | Bus plate number – label | Plate number of the bus |
| **3** | From date – label | The date when the schedule starts |
| **4** | To date – label | The date when the schedule ends |
| **5** | Delete – button | Delete the schedule |

Table 4‑44 Manage schedule screen description

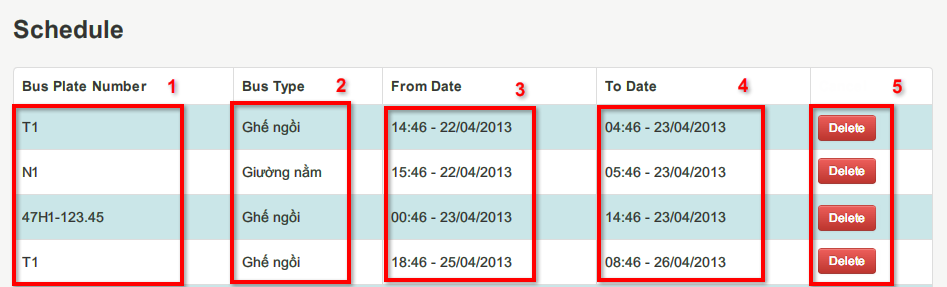


Figure 4‑37 Schedule list in route details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Bus plate number – label | Plate number of the bus |
| **2** | Bus type – label | Type of the bus |
| **3** | From date – label | The date when the schedule starts |
| **4** | To date – label | The date when the schedule ends |
| **5** | Delete – button | Delete the schedule |

Table 4‑45 Schedule list in route details screen description

1. **Cancel schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC020 | **Use-case Version** | 0.1 |
| **Use-case Name** | Cancel schedule | | |
| **Author** | Nguyễn Thị Bích Trâm, Nguyễn Sơn Trường | | |
| **Date** | 30/01/2013 | **Priority** | High |
| **Actor:** Staff  **Summary:** User can cancel a schedule  **Goal:** Help user to cancel a schedule.  **Triggers:** User want to cancel a schedule. (User click “Delete” in Schedule list.)  **Preconditions:**   * User login into system with role Staff. * There is at least one available schedule that has not been started.   **Post Conditions:**  *Success*: Schedule is cancelled.  *Failure*: N/A  **Main Success Scenario:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | 1. User click “Delete” in schedule list.  3. User input reason and click “Yes” [Alternative 2][Alternative 3] | 2. Cancel schedule confirmation modal is shown. [Alternative 1]   * Reason [text box - String] * No [button] * Yes [button]   4. Cancel schedule and save reason. |   **Alternative scenarios:**   |  |  | | --- | --- | | *Actor Actions* | *System Response* | | [Alternative 1]  1. There are reservations on the journey of the cancelling schedule.  [Alternative 2]  1. User do not fill reason and click “Yes”  [Alternative 3]  1. User click “No” | 2. Show number of reservation and amount to be refunded.  2. Display message “Please input reason”  2. Close the modal and return to previous screen. |   **Exceptions:** N/A  **Relationships:**  Login, Manage schedule, View route details.  **Business Rules:**  If there are reservations on the journey of the cancelling schedule, system must refund 100% amount of money that customers paid (by PayPal). | | | |

Table 4‑46 Cancel schedule Use case Specification

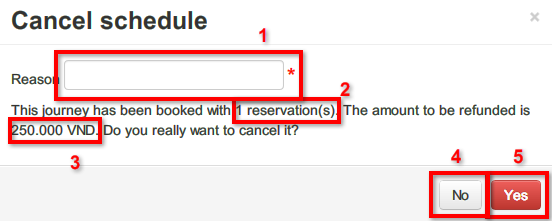


Figure 4‑38 Cancel schedule screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Reason – text box | Reason to cancel schedule |
| **2** | Number of reservation – label | Number of reservation of journey on the current schedule |
| **3** | Refund amount – label | Amount to be refunded |
| **4** | No – button | Confirm not to cancel schedule |
| **5** | Yes – button | Confirm to cancel schedule |

Table 4‑47 Cancel schedule screen description

#### Register and login



Figure 4‑39 Register and login Use case

#### Account management



Figure 4‑40 Account management Use case

### Software System Attributes

#### Usability

##### **Graphic User Interface**

* + All the text, label and image in Guest and Customer page should be in Vietnamese.
  + All the text, label and image in Admin, Staff page should be in English.

##### **Usability for guest and customer**

* + Can create a reservation, view past reservation details, cancel reservation.

##### **Usability for admin and staff**

* + Admin and staff must understand and use English.
  + Website admin and staff should need no more than one week of training to be productive with the system.

##### **Installation**

* + The system must be easy to deploy. Customer can deploy successfully and learn to configure, maintain the system within one day of training.
  + The attached manual guide must be clear. User can read and do itself without developers’ help.

#### Reliability

N/A

#### Availability

N/A

#### Security

* + Privacy: Each role of user has a specific permission to interact with system. System always checks authorization and authenticated before do anything. Transaction history of registered users can be accessed by themselves only.

#### Maintainability

N/A

#### Portability

N/A

#### Performance

N/A

# Software Design Description

## Design Overview

* This document describes the technical and UI design of the Long Travel Bus Reservation system. It includes the architectural design and the detailed design of common functions and business functions. It also includes the design of database model.
* The architectural design describes the overall architecture of the system, and the architecture of each main component and subsystem. It will describe the patterns being used, the role of each component and the role of the system in the working environment.
* The detailed design describes static and dynamic structure for each component and function. It includes class diagrams, class explanations, and sequence diagrams of each use cases. The detailed design uses notations of Java (J2EE 5) as they are the framework and language for developing the system.
* The database design describes the relationship between entities, and details of each entity. It uses notations of MySQL as it is the database server for developing the system.
* Overview:
* Section 5.2: gives an overall description of the system architecture design. It gives what architecture is chosen and its model that is presented in graphic
* Section 5.3: gives component diagrams that describe the connection and integration of the system.
* Section 5.4: gives the detail design description include class diagram, class explanation, sequence diagram... to details the application functions in details
* Section 5.5: describe an ERD with many entity and database design.

## System Architectural Design

### Choice of System Architecture

Presentation Layer

Business Logic Layer

Data Access Layer

Database

Figure 5‑1 3-Layer Architecture

## Component Diagram

* The architecture of Long Travel Bus Reservation system is Three Layers Architecture. Inside each layer, there are some components that handle one or some specific functions.

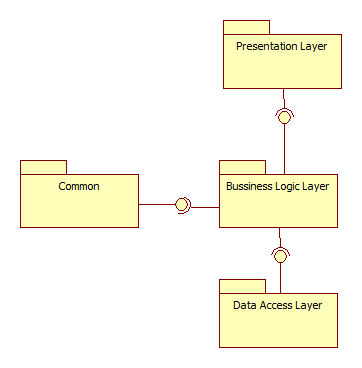


Figure 5‑2: Component Diagram

### Presentation Layer’s Components

* This component has responsibility to deliver the front end to the users and accept their inputs. It will be developed by using HTML combines with JavaSript, jQuery and CSS.



Figure 5‑3: Presentation Layer's Components

### Business Logic Layer’s Components

* This component contains all business logic for the system. It includes business layer object classes, data transfer object classes, struts actions …
* These components have responsibility to process importing data, validation, transferring data to presentation layer or data access layer.

****

Figure 5‑4: Business Layer's Components

### Data Access Layer’s Components

* This component contains all data access objects which is responsible for accessing data in database system component and exchange data information with database for the system. It includes data access classes.



Figure 5‑5: Data Access Layer's Components

### Common Component

* This component contains common logic that can be used throughout the application. It also contains library use for sending email, exporting PDF.



Figure 5‑6 Common Components

## Detailed Description of Components

### Manage bus

<Optional>

### Manage route

<Optional>

### Manage schedule

<Optional>

### Manage segment

<Optional>

### Search for journeys

#### Class Diagram



Figure 5‑7 Search function class diagram

#### Class Diagram Explanation

* + 1. **SearchResultAction**

|  |  |
| --- | --- |
| Class name | SearchResultAction |
| Description | This class handles action when user click button “Đặt vé” at search page. |
| **Attribute**:   |  |  |  |  | | --- | --- | --- | --- | | **Attribute** | **Attribute Type** | **Data Type** | **Description** | | ticketType | Input | String | Ticket type: One way or Round trip. | | departureCity | Input | int | Departure city id | | arrivalCity | Input | Int | Arrival city id | | departureDate | Input | String | Departure date (dd-MM-yyyy format) | | returnDate | Input | String | Return date (dd-MM-yyyy) format | | onwardMap | Output | Map<String, Map<String, List<SearchResultInfo>>> | Search result filtered by bus type and departure date for onward journey | | returnMap | Output | Map<String, Map<String, List<SearchResultInfo>>> | Search result filtered by bus type and departure date for return journey | | deptCity | Output | String | Departure city name | | arrCity | Output | String | Arrival city name | | message | Output | String | Error message | | |
|  | |
| **Methods**:  *public String execute():* Get and organize the trips that match the provided search criteria.   |  |  |  | | --- | --- | --- | | **Parameter/Return** | **Type** | **Description** | | Return | String | SUCCESS: “success”  ERROR: “error” | | |

Table 5‑1 SearchResultAction class explanation

* + 1. **CityDAO**

|  |  |
| --- | --- |
| Class name | CityDAO |
| Description | This class gets information of City in database |
| **Attribute**: N/A | |
| **Methods**:  *public List<CityBean> getDepartCity():* get list of departure city (location) from database   |  |  |  | | --- | --- | --- | | **Parameter/Return** | **Type** | **Description** | | Return | List<CityBean> | List of departure location |   *public List<CityBean> getArriveCity(int departCityId):* get list of arrival city by inputted departure city.   |  |  |  | | --- | --- | --- | | **Parameter/Return** | **Type** | **Description** | | departCityId | Int | Departure city id | | Return | List<CityBean> | List of arrival location | | |

Table 5‑2 CityDAO class explanation

* + 1. **SearchResultLogic**

|  |  |
| --- | --- |
| Class name | SearchResultLogic |
| Description | This class helps search for journeys based on inputted value. |
| **Attribute**: N/A | |
| **Methods**:  *public Map<String, Map<String, List<SearchResultInfo>>> searchTripGroupByDate (int deptCity, int arrvCity, String deptDate, int pssgrNo, String minDate):* search for journeys and filter by bus type and date.   |  |  |  | | --- | --- | --- | | **Parameter/Return** | **Type** | **Description** | | Return | Map<String, Map<String, List<SearchResultInfo>>> | Search result filtered by bus type and departure date | | deptCity | int | Departure city id | | arrCity | int | Arrival city id | | deptDate | String | Departure date inputted by end-user (format: yyyy-MM-dd) | | pssgrNo | int | Number of passenger for the journey | | minDate | String | Minimum date that can be departure date (format: yyyy-MM-dd) | | |

Table 5‑3 SearchResultLogic class explanation

* + 1. **TripDAO**

|  |  |
| --- | --- |
| Class name | TripDAO |
| Description | This class gets information of Trip table in database |
| **Attribute**: N/A | |
| **Methods**:  *public List<SearchResultInfo> searchAvailableTrips(int deptCity, int arrvCity, String deptDate, int pssgrNo, String minDate):* Search for journey by inputted parameters   |  |  |  | | --- | --- | --- | | **Parameter/Return** | **Type** | **Description** | | Return | List<SearchResultInfo> | Return a list of journey (formatted by SearchResultInfo model structure) | | deptCity | int | Departure city id | | arrCity | int | Arrival city id | | deptDate | String | Departure date inputted by end-user (format: yyyy-MM-dd) | | pssgrNo | int | Number of passenger for the journey | | minDate | String | Minimum date that can be departure date (format: yyyy-MM-dd) |   *public List<TripBean> getTripDetails(int busStatus, String deptTime, String arrTime):* get list of trips in a journey.   |  |  |  | | --- | --- | --- | | **Parameter/Return** | **Type** | **Description** | | Return | List<TripBean> | Return a list of trips which is contained in a journey. | | busStatus | int | Id of bus status which performs the journey. | | deptTime | String | Departure time of the journey (format: yyyy/MM/dd HH:mm:ss) | | arrTime | String | Arrival time of the journey  (format: yyyy/MM/dd HH:mm:ss) | | |

Table 5‑4 TripDAO class explanation

* + 1. **SearchTripDetailsJSONAction**

|  |  |
| --- | --- |
| Class name | SearchTripDetailsJSONAction |
| Description | Action activated by AJAX calling that will get list of trips in selected journey. |
| **Attribute**:   |  |  |  |  | | --- | --- | --- | --- | | **Attribute** | **Attribute Type** | **Data Type** | **Description** | | busStatus | Input | int | Id of bus status which performs the journey. | | departTime | Input | String | Departure time of the journey (format: yyyy/MM/dd HH:mm:ss) | | arriveTime | Input | String | Arrival time of the journey (format: yyyy/MM/dd HH:mm:ss) | | tripList | Output | List<TripBean> | List of trips in the journey | | |
|  | |
| **Methods**:  *public String execute():* Get and organize the trips that is contained in selected journey.   |  |  |  | | --- | --- | --- | | **Parameter/Return** | **Type** | **Description** | | Return | String | SUCCESS: “success”  ERROR: “error” | | |

Table 5‑5 SearchTripDetailsJSONAction class explanation

* + 1. **SearchLatitudeJSONAction**

|  |  |
| --- | --- |
| Class name | SearchLatitudeJSONAction |
| Description | Action activated by AJAX calling that will get list of latitudes of each locations which the journey goes by. |
| **Attribute**:   |  |  |  |  | | --- | --- | --- | --- | | **Attribute** | **Attribute Type** | **Data Type** | **Description** | | busStatus | Input | int | Id of bus status which performs the journey. | | departTime | Input | String | Departure time of the journey (format: yyyy/MM/dd HH:mm:ss) | | arriveTime | Input | String | Arrival time of the journey (format: yyyy/MM/dd HH:mm:ss) | | latitudeList | Output | ArrayList | List of latitudes in the journey | | |
|  | |
| **Methods**:  *public String execute():* Get and organize list of latitudes in the journey.   |  |  |  | | --- | --- | --- | | **Parameter/Return** | **Type** | **Description** | | Return | String | SUCCESS: “success”  ERROR: “error” | | |

Table 5‑6 SearchLatitudeJSONAction class explanation

#### Sequence Diagram

****

Figure 5‑8 Sequence diagram of search function

#### Algorithms of search journey function

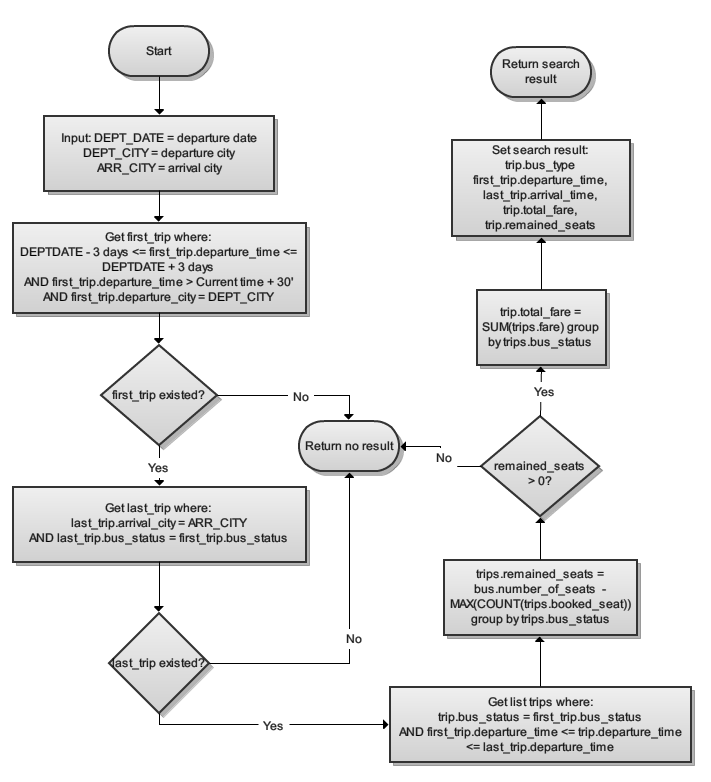


Figure 5‑9 Algorithm of search journeys

### Booking and payment

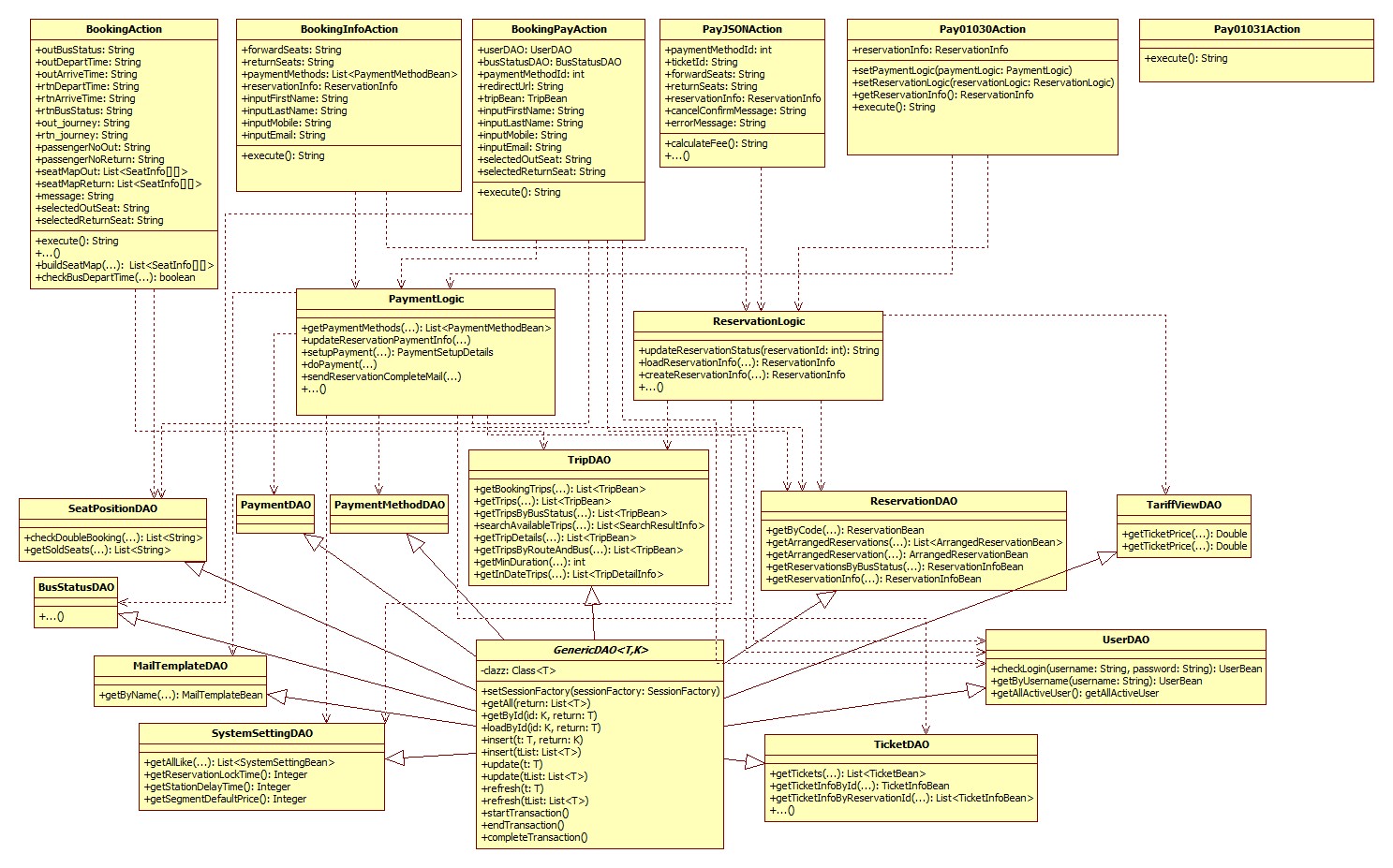


Figure 5‑10 Booking and Payment class diagram

#### Sequence diagram

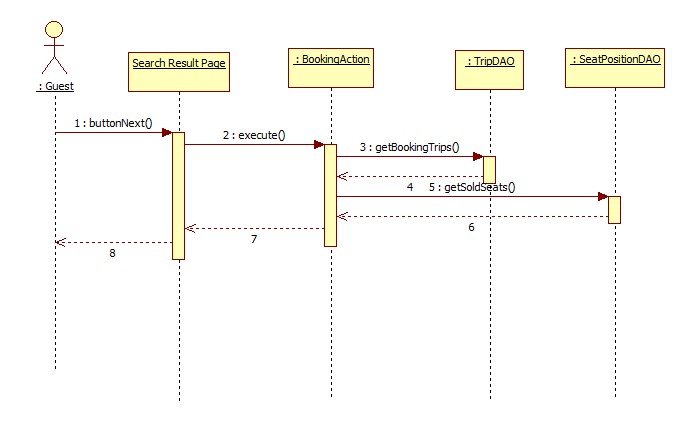


Figure 5‑11 Booking sequence

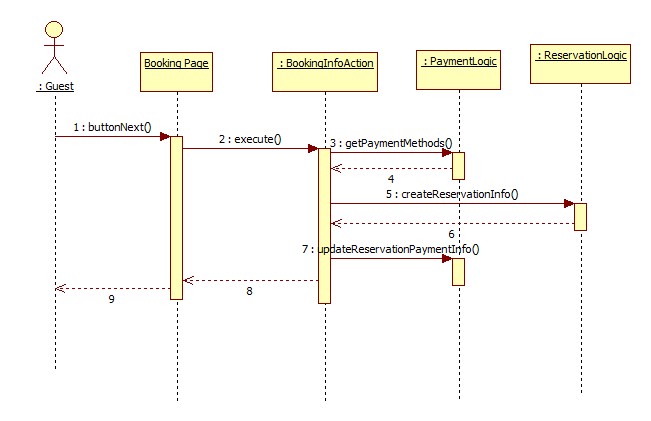


Figure 5‑12 Input customer’s information

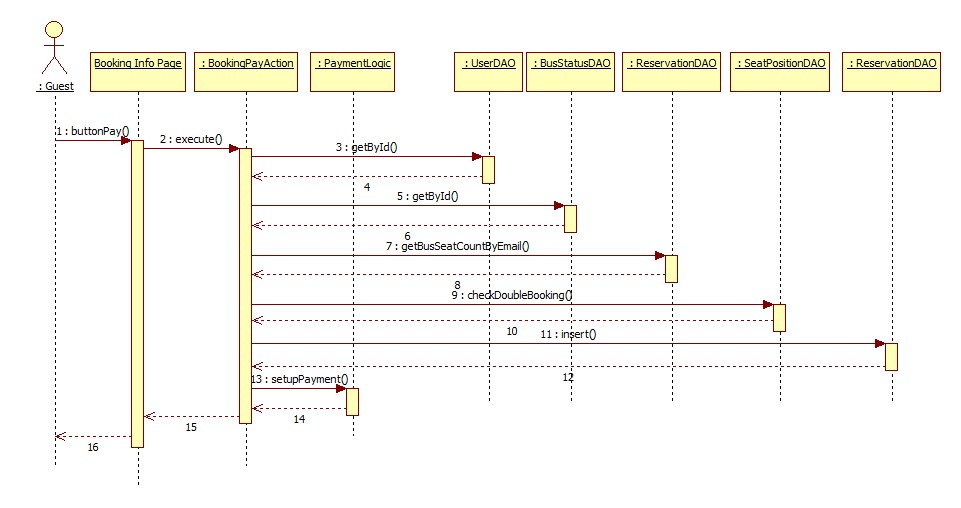


Figure 5‑13 Save reservation

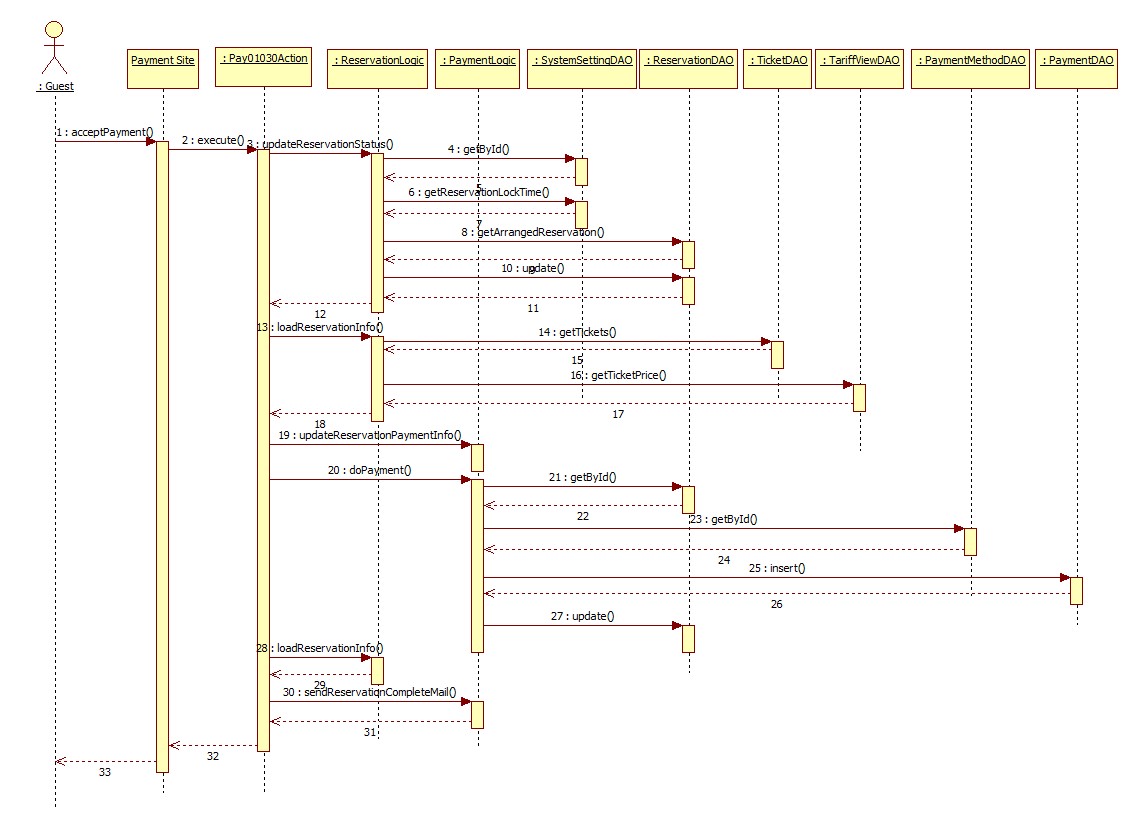


Figure 5‑14 Pay for reservation

## Database Design or Data Structures

### Entity Relationship Diagram



Figure 5‑15 Entity relationship diagram (with entity only)

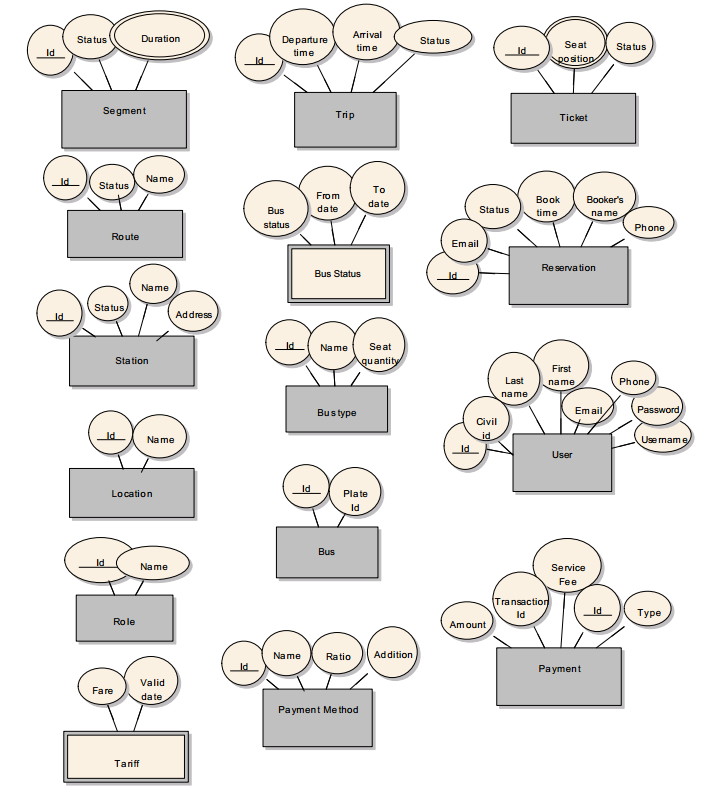


Figure 5‑16 Attributes of entities

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Description** | **Alias** | **Occurrence** |
| Route | A line of travel that buses run on |  | Each route contains one or more route details |
| Each route is run on by one or more buses |
| Route Details | Detailed parts of a route which contain segments that a trip run on |  | Each route detail is only a member of one route |
| Each route detail contains only one segment |
| Each route detail contains one or more trips |
| Segment | A direct track that connects two stations of two cities |  | Each segment has one departure and one arrival station |
| Each segment is in one or more route details |
| Station | A place where buses load or unload passengers |  | Each station belongs to only one city |
| Each station can be a departure or arrival place of many segments |
| Location | A popular location that can be a place of departure or destination Ex: At Lâm Đồng Province, we have two city (or location): Bảo Lộc and Đà Lạt | City | Each city has one or more stations. |
| Bus | Vehicle of the company which is responsible for running trips |  | Each bus has one or more bus statuses. |
| Each bus belong to one bus type. |
| Each bus run on two specific routes: onward route and return route. Ex: Bus 51A-123.45 runs on Hồ Chí Minh City - Hà Nội as onward route and Hà Nội - Hồ Chí Minh City as return route |
| Bus Type | Type of bus. Each type has different facilities and number of seats. |  | Each bus type contains one or more buses. |
| Bus Status | Status of a bus in a period of time indicating that it is busy for running on trips or maintaining at a specific location… |  | When a bus is on a trip, bus status must be recorded. |
| Each bus status indicates that after the bus is busy (for running trip, maintaining…), it is located at one station. |
| Each bus status indicates that the bus is on zero or more trips (there are cases that the bus is busy but not on trip such as maintaining…) |
| Tariff | A schedule of fares apply on a segment for a bus type in a period of time |  | One or more tariff is applied for a segment (different tariff for different period of time) |
| One or more tariff is applied for a bus type (different tariff for different period of time) |
| Trip | A trip going on a route detail which is scheduled before. |  | Each trip runs on only one route details. |
| Each trip is booked on one or more tickets. |
| Each trip is run only by one bus, so it is in only one bus status. |
| Ticket | An indicator for booked seats of a journey (each journey is a list of trips). |  | Each ticket booked for one or more trips (those trips that booked make a journey of a customer) |
| Reservation | Reservation information |  | Each reservation has one or two tickets. In case user reserved for 2 journeys: onward and return, 2 tickets is recorded contained in 1 reservation. |
| Payment | Payment information of a ticket |  | Each payment can using only one payment method |
| Payment Method | Payment methods that customers used to pay for their reservation |  | Each payment method is used in zero or more payment |
| User | People who interact with the system. User can be customer or staff of company |  | Each user can create from zero to many reservation (staff of the company can also create reservations) |
| Role | Role of users |  | Each role contains many users. |

Table 5‑7 Entity description

| **Entity** | **Attribute** | **Description** |
| --- | --- | --- |
| Segment | Id | Id of the segment |
| Status | Status of segment (active or inactive) |
| Duration | Estimated travel time of segment. In each period of time, duration of segment can be different. |
| Route | Id | Id of the route |
| Status | Status of route (active or inactive) |
| Name | Name of the route |
| Station | Id | Id of the station |
| Status | Status of the station (active or inactive) |
| Name | Name of the station |
| Address | Address of the station |
| Location | Id | Id of the location |
| Name | Location name |
| Tariff | Fare | Fare based on segment and bus type. Unit: thousand VND |
| Valid date | From this date on, the fare is applied. Example: There are two records: {Valid date = 30/04/2013, fare = 10} and {Valid date = 19/05/2013, fare = 20} means that from 30/04/2013 to 18/05/2013, fare is 10. From 19/05/2013 to the future, fare is 20. |
| Trip | Id | Id of the trip |
| Departure time | Departure time of the trip |
| Arrival time | Arrival time of the trip |
| Status | Status of the trip (active or inactive) |
| Bus Status | Bus status | Status of the bus (initial, on-trip or return) |
| From date | Date range of the bus status |
| To date |
| Bus | Plate Id | Unique plate number of each bus |
| Bus Type | Id | Id of bus type |
| Name | Name of bus type |
| Seat quantity | Number of seats in each bus type |
| Ticket | Id | Id of ticket |
| Seat position | Seat position that user booked. Each ticket has minimum of 1 seat position and maximum of 5 seat position |
| Status | Status of the ticket (active or inactive) |
| Reservation | Id | Id of reservation |
| Booker's name | Name of the customer who made a reservation |
| Email | Email of booker |
| Phone | Phone number of booker |
| Book time | Date and time that customer booked |
| Status | Status of reservation (paid, unpaid, pending, departed, cancelled, moved, deleted, refunded, refunded2) |
| Payment | Transaction Id | Id of the transaction that system made with online payment service |
| Service fee | Fee of the online payment service such as PayPal |
| Amount | Amount of money must be paid (included service fee) |
| Type | Type of payment (user pay or system refund) |
| Payment Method | Id | Id of payment method |
| Name | Name of payment method |
| Ratio | Percent of money on each orders that user made which will be collected by online payment service. |
| Addition | A fixed amount of money on each transaction that will be collected by online payment service. |

Table 5‑8 Attribute description

|  |  |
| --- | --- |
| **Status** | **Description** |
| Active | Item is active and can be interacted |
| Inactive | Item is removed out of the system and cannot be interacted |
| Initial | The bus has just been assigned to a route and it is located at start station of that route. |
| On-trip | The bus is busy for running on a schedule which journey is reserved by customers. |
| Return | The bus is busy for running on a schedule but there’s no journey for customers to book. |
| Paid | The reservation has been paid and can be cancelled. |
| Unpaid | The reservation has been created, but payment has not been completed. |
| Pending | The reservation can no longer be cancelled. |
| Departed | The journey (or first journey of a reservation for return journey) has departed |
| Cancelled | The reservation has been cancelled due to the journey being cancelled. Awaiting for refund. |
| Deleted | The reservation was not paid within the time limit. |
| Refunded | The reservation was cancelled by user, and refund is completed. |
| Refunded2 | The reservation was cancelled by company, and refund is completed. |

Table 5‑9 Database status description

### Database diagram

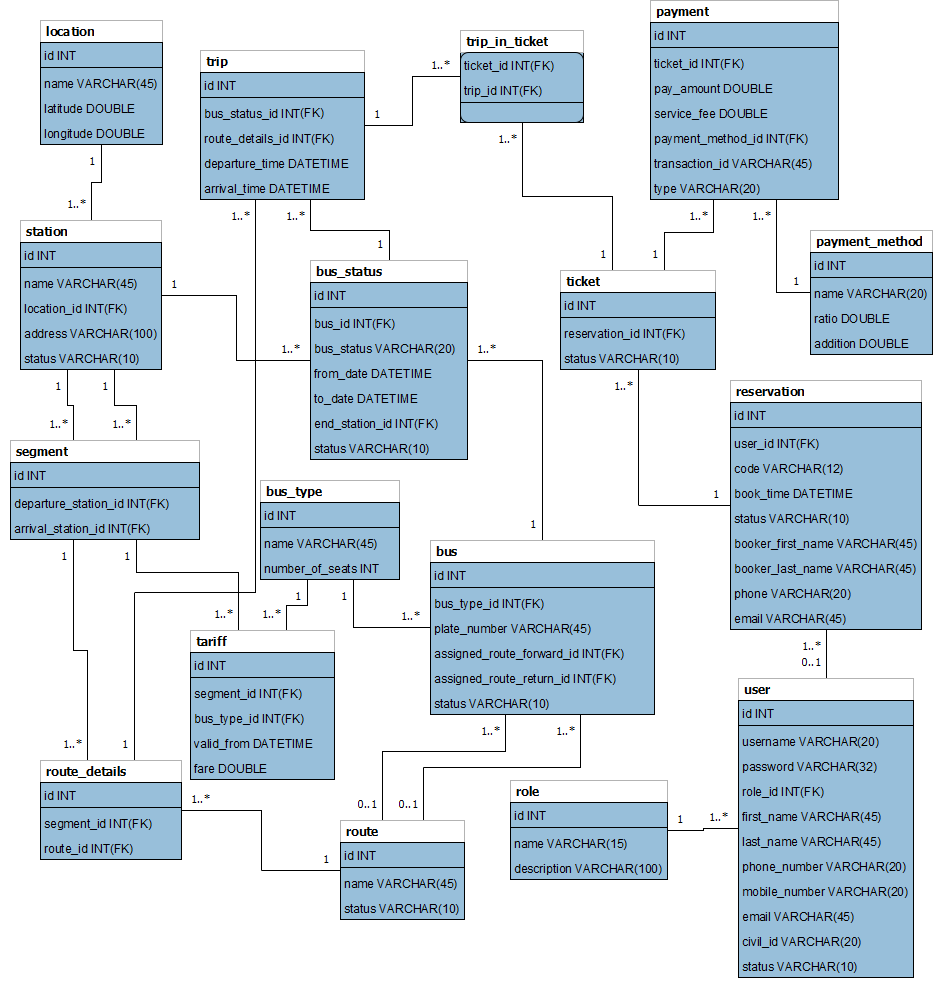


Figure 5‑17 Relational database diagram

# Software Test Documentation

## Introduction

### System Overview

This document describes the approach and methodologies used by the testing group to plan, organize and manage the testing of this system. It describes implementation details of test cases or technical details of how the product features should work.

The Long Travel Bus Reservation system allows users to perform the following actions:

* Create reservation: Search journeys, Choose journeys, Choose seats, Provide booker’s information and Payment.
* View reservation history, reservation details
* Cancel reservation
* Manage bus: Add new bus, View bus list, Remove bus
* Manage segment: View segment list, Edit segment travel time
* Manage route: Add new route, View route list, View route details, Add price for route, Assign bus to route, Add new schedule for route.
* Manage schedule: View schedule list, Cancel schedule
* Manage user: View user list, Add new staff, and Reset password.
* Register as a customer and login to the system

### Test Approach

* The testing for this project will consist of Unit, System and Acceptance test levels. It is hoped that there will be as less problem as possible in project.
* Unit Testing will be done by the developers and will be approved by the implementing team leader. All unit test information will be managed by developers and it must be showed when team required.
* System Testing will be performed by tester with assistance from the individual developers as required.
* Acceptance Testing will be performed by all the member of team. The acceptance test will be done for a period of 2 weeks after completion of the System test process. Programs will enter into Acceptance test after all critical and major defects have been corrected. A program may have one major defect as long as it does not obstruct testing of the program. Prior to final completion of acceptance testing all open critical and major defects must be corrected.

## Test Plan

### Features to be tested

The following is the list of features that will be focused on the test phase of the program:

#### Create reservation

* Test these function with both Guest and Customer role: Search for journey, Choose journey, Choose seat, Provide booker’s details, Pay with PayPal

#### View reservation history

* Customer can view reservation history, reservation details and cancel a reservation
* Guest can view reservation details using reservation code and email than cancel a reservation.

#### Manage bus

* Staff can add new bus, view bus list and delete bus

#### Manage route

* Staff can add new route, view route list, view route details, add price for route, assign bus to route, add new schedule for route.

#### Manage segment

* Staff can view segment list and edit segment travel time

#### Manage schedule

* Staff can view schedule list and cancel schedule.

### Features not to be tested

The following features will not be focused on the test phase. They are just got the unit test of the developers:

#### Register

#### Login

#### Manage user

* Admin can view user list, add new staff and reset password

#### Print and export reservation details to PDF file

## Test Cases

### Manage bus

| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** |
| --- | --- | --- | --- | --- | --- | --- |
|  | List roles |  |  |  |  |  |
|  | Steps 1. Navigate to Manage Bus screen | | | | | |
| MB001 | Add new bus. | 1. Click button Add New Bus.  2. When popup “Add New Bus” showed, input “Bus Plate Number” of new bus, select bus type from dropdown list.  3. Click button Save. | Popup “Add new Bus” is closed.  Manage Bus screen is reload and recent added bus is shown in the first row of bus grid. |  | Passed | 17/4/2013 |
| MB002 | Add new bus: Validate Bus Plate number | 1. Click button Add New Bus.  2. When popup “Add New Bus” showed, click button Save. | Error message shown: “Please enter bus plate number!” |  | Passed | 17/4/2013 |
| MB003 | Add new bus: Validate Bus Plate number | 1. Click button Add New Bus.  2. When popup “Add New Bus” showed, input “Bus Plate Number” of new bus which already existed in system, select bus type from dropdown list.  3. Click button Save.products | Error message shown: “Bus Plate number is existed!” |  | Passed | 17/4/2013 |
| MB004 | Delete bus. | 1. Click button Delete in each row of table.  2. When Delete confirm popup shown, click button Delete. | Delete confirm popup is closed.  Popup Delete Result is shown message: “Bus is deleted successfully!” | MB001 | Passed | 17/4/2013 |
| MB005 | Delete bus: Delete bus that has trip(s) in the future. | 1. Click button Delete in each row of table to delete that bus.  2. When Delete confirm popup shown, click button Delete. | Delete confirm popup is closed.  Popup Delete Result is shown message: “Cannot delete this bus due to this bus has 5 trips in the future!” |  | Passed | 17/4/2013 |
| MB006 | Delete bus: Delete bus that being assigned to a route. | 1. Click button Delete in each row of table.  2. When Delete confirm popup shown, click button Delete. | Delete confirm popup is closed.  Popup Delete Result is shown message: “Cannot delete this bus due to this bus is being assigned to route Hà Nội - TP. Hồ Chí Minh. Please unassign this bus and try again!” |  | Passed | 17/4/2013 |

Table 6‑1 Manage bus test case

### Manage segment

| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** |
| --- | --- | --- | --- | --- | --- | --- |
|  | List roles |  |  |  |  |  |
|  | Steps 1. Navigate to Manage Segment screen | | | | | |
| MS001 | Edit segment duration. | 1. Click button Edit.  2. When popup “Edit segment” showed, select valid from date of new duration, input new duration.  3. Click button Save. | Popup “Edit segment” is closed.  Manage Segment screen is reloaded. |  | Passed | 17/4/2013 |
| MS002 | Edit segment duration: Duration is less than minimum duration. | 1. Click button Edit.  2. When popup “Edit segment” showed, select valid from date of new duration, input new duration with value lesser than 1hour.  3. Click button Save. | Duration text box is highlight by red.  Error message is displayed: “Duration should not less than 1 hour.” |  | Passed | 17/4/2013 |

Table 6‑2 Manage segment test case

### Manage route and schedule

| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** |
| --- | --- | --- | --- | --- | --- | --- |
|  | List roles |  |  |  |  |  |
|  | Steps 1. Navigate to Manage Route screen | | | | | |
| MR001 | Display route in grid. | Navigate to Manage Route screen. | Button “Activate” is display in inactive routes.  Button “Deactivate” is display in active routes. |  | Passed | 17/4/2013 |
| MR002 | Add new route. | 1. Click button Add New Route.  2. When popup “Add New Route” showed, input following information:  - Start At (2 dropdown lists): The first dropdown list is start city of segment; the second is start station of segment.  - End At (2 dropdown lists): The first dropdown list is end city of segment; the second is end station of segment.  - Duration (text box) duration of segment.  3. Click button Add.  4. Click button Save.  5. When popup “Add Route result” shown, click button OK. | Popup “Add new Route” is closed.  Popup “Add Route result” is shown: “New Route saved successfully!”  Manage Route screen is reloaded and 2 new routes is shown in the routes grid. |  | Passed | 17/4/2013 |
| MR003 | Add new route: Add existed segment into new route. | 1. Click button Add New Route.  2. When popup “Add New Route” showed, input segment information which is already existed in system. | Right after input End station of segment, text box Duration is auto filled with the existed segment’s duration and then disabled. |  | Passed | 17/4/2013 |
| MR004 | Add new bus: Add existed route. | 1. Click button Add New Route.  2. When popup “Add New Route” showed, input segments information, the list segments inputted is duplicated with a route in system.  3. Click button Save. | Popup “Add new Route” is closed.  If the existed route is active. Popup “Add Route result” is shown: “Route existed! Please verify again!”  If the existed route is active. Popup “Add Route result” is shown: “Existed route is re-activated successfully!” | MR002 | Passed | 17/4/2013 |
| MR005 | Deactivate Route: deactivate route which doesn’t have any trips. | 1. Click button Deactivate in each row of table. | Manage route screen reload; route and its reverse route are deleted from the route grid. |  | Passed | 17/4/2013 |
| MR006 | Deactivate Route: deactivate route which has at least one trip in the pass and has no trip in the future. | 1. Click button Deactivate in each row of table. | Delete route result shown: “Route Hải Phòng - Thanh Hoá and route Thanh Hoá - Hải Phòng are deleted successfully! All buses belong to these routes are unassigned successfully!”  Manage route screen reload; The route grid shows the route and its reverse route are deactivated. |  | Passed | 17/4/2013 |
| MR007 | Deactivate Route: deactivate route which has at least one trip in the future. | 1. Click button Deactivate in each row of table. | Delete route result shown: “Cannot delete this route due to this route has 12 available trip(s) depend on it”  Manage route screen reload. |  | Passed | 17/4/2013 |
| MR008 | Manage Route details loading screen: Route is active and does not have any bus. | 1. Select a route in screen Manage Route and click button “View details” | Button “Add new Schedule” is disabled.  Button “Edit route” is displayed. |  | Passed | 17/4/2013 |
| MR009 | Manage Route details loading screen: Route is active and has at least 1 bus. | 1. Select a route in screen Manage Route and click button “View details” | Button “Add new Schedule” is enabled.  Button “Edit route” is hide. |  | Passed | 17/4/2013 |
| MR010 | Manage Route details loading screen: Route is inactive. | 1. Select a route in screen Manage Route and click button “View details” | Buttons “Edit route price”, “Assign Bus to route”, “Add new Schedule” are enabled.  Button “Edit route” is hide. |  | Passed | 17/4/2013 |
| MR011 | View route duration and price by Bus type. | 1. Select Bus type in dropdown list bus type. | Column Price in segment grid is reloaded and shows price of route by following bus type. |  | Passed | 17/4/2013 |
| MR012 | View route duration and price by Bus type. | 1. Select Date time in date time picker. | Column Duration and Price in segment grid are reloaded and show duration price of route by following bus type and date time. |  | Passed | 17/4/2013 |
| MR013 | Edit Price of route: Only edit current route. | 1. Click on button Edit route price.  2. Popup “Edit route Price” is shown and column Price is prefilled with price base on Valid date and Bus type.  3. Edit price of each segment.  4. Click button Save. | Popup “Edit route Price” is closed.  Route detail list screen is reloaded.  Choose the bus type and valid date to view the edited price. |  | Passed | 17/4/2013 |
| MR014 | Edit Price of route: Edit price of current route and its reverse route. | 1. Click on button Edit route price.  2. Popup Edit route Price is shown and column Price is prefilled with price base on Valid date and Bus type.  3. Edit price of each segment.  4. Click on check box: “Add price for reverse route too?”  5. Click button Save. | Popup Edit route Price is closed.  Route detail list screen is reloaded.  Choose the bus type and valid date to view the edited price.  Click button “Show the reverse route” and choose the bus type and valid date to view the edited price. |  | Passed | 17/4/2013 |
| MR015 | Edit Price of route: Invalid input value ‘0’ or blank. | 1. Click on button Edit route price.  2. Popup Edit route Price is shown and column Price is prefilled with price base on Valid date and Bus type.  3. Edit one price by value ‘0’.  4. Leave one price blank.  5. Click button Save. | 2 invalid input fields are highlighted by red. |  | Passed | 17/4/2013 |
| MR016 | Assign bus to route: Display bus list. | 1. Click button Assign bus to route. | If the bus has at least 1 trip in the future, button Delete is disabled. |  | Passed | 17/4/2013 |
| MR017 | Assign bus to route. | 1. Click button Assign bus to route.  2. When popup “List buses in route” is shown, select bus type.  4. When the “Select bus” dropdown list is reloaded, select a bus and click button “Add”.  5. New bus is added into bus grid.  6. Click button “Save”. | Bus is assigned into route. | MB001 | Passed | 17/4/2013 |
| MR018 | Assign bus to route: Delete bus from list. | 1. Click button Assign bus to route.  2. When popup “List buses in route” is shown, click on “Delete” button to delete that bus from route. | Bus is removed from list.  Bus Plate number is displayed in bus dropdown list. | MR017 | Passed | 17/4/2013 |
| MR019 | Assign bus to route: Dropdown list bus is empty. | 1. Click button “Assign bus to route”.  2. Click button “Add” bus until the dropdown list is empty. | Message shown: “There is no free bus at the moment!” |  | Passed | 17/4/2013 |
| MR020 | Add new schedule. | 1. Click button “Add new Schedule”.  2. When popup “Add new Schedule” shown:  - Select From date in date time picker  - Select bus type in dropdown list  - Select bus plate number in dropdown list  3. Click button “Save Changes”. | Popup “Add new Schedule” is closed.  Manage Route details screen is reloaded and recent added schedule shown in schedule list. | MR018 | Passed | 17/4/2013 |
| MR021 | Add new schedule: No available bus to create schedule. | 1. Click button “Add new Schedule”.  2. When popup “Add new Schedule” shown, input From date and bus type information that don’t have any bus available for that. | Error message displayed: “There is no bus available at this time. Please select other time or click cancel!” |  | Passed | 17/4/2013 |
| MR022 | Add new schedule: Set auto-return to start station. | 1. Click button “Add new Schedule”.  2. When popup “Add new Schedule” shown:  - Select From date in date time picker  - Select bus type in dropdown list  - Select bus plate number in dropdown list  3. Check the checkbox “Set auto-return to start station”.  4. Select From date to return the bus.  3. Click button “Save Changes”. | Popup “Add new Schedule” is closed.  Manage Route details reload and recent added schedule shown in schedule list.  Add new schedule success for the recent bus which has From date bigger than To date of the return schedule. |  | Passed | 17/4/2013 |
| MR023 | Show the reverse route | 1. Click button “Show the reverse route”. | The reverse route details screen is shown. |  | Passed | 17/4/2013 |

Table 6‑3 Manage route and schedule test case

### Create reservation

| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** |
| --- | --- | --- | --- | --- | --- | --- |
|  | List roles |  |  |  |  |  |
|  | Steps 1. Search a trip at home page.  2. Choose one trip at search-result page. | | | | | |
| BS 001 | One email only book 5 seats in one trip | At booking page, choose 3 seats then checkout.  Then search that trip again and booking 3 seat then checkout | Redirect user to search result page and show message “Bạn chỉ được đặt mua 5 ghế trên mỗi chuyến vui lòng chọn chuyến khác.” |  | Passed | 15/4/2013 |
| BS 002 | Two users choose the same seats (remain seats : 6) | User1: Choose 3 seats.  User2: Choose 3 seats (1 seat is chosen by user1).  User1: complete payment.  User2: complete fulfill booking info | Redirect user2 to booking page show message “Ghế bạn chọn đã có người đặt vui lòng chọn ghế khác.” and let user2 choose another seats |  | Passed | 13/3/2013 |
| BS 003 | Two users choose the same seats (remain seats : 5) | User1: Choose 3 seats.  User2: Choose 3 seats (1 seat is chosen by user1).  User1: complete payment.  User2: complete fulfill booking info | Redirect user2 to booking page show message “Chuyến bạn vừa chọn không đủ số ghế vui lòng đặt ít hơn hoặc chọn chuyến khác.” and let user2 choose another seats |  | Passed | 13/3/2013 |
| BS 004 | Two users choose the same seats (remain seats : 3) | User1: Choose 3 seats.  User2: Choose 3 seats (3 seat is chosen by user1).  User1: complete payment.  User2: complete fulfill booking info | Redirect user2 to search result page show message “Chuyến bạn vừa chọn đã hết ghế vui lòng chọn chuyến khác.” and let user2 choose another trip. |  | Passed | 13/3/2013 |

## Checklists

### Checklist of Validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | | Yes | No | N/A |
| 1 | Does a failure of validation on every field cause a sensible user error symbol? | x |  |  |
| 2 | Is the user required to fix entries which have failed validation tests? |  |  | x |
| 3 | Have any fields got multiple validation rules and if so are all rules being applied? | x |  |  |
| 4 | If the user enters an invalid value and clicks on the insert button is the invalid entry identified and highlighted correctly with an error symbol? | x |  |  |
| 5 | Is validation consistently applied at screen level unless specifically required at field level? | x |  |  |
| 6 | For all numeric fields check whether negative numbers can and should be able to be entered. |  | x |  |
| 7 | For all numeric fields check the minimum and maximum values and also some mid-range values allowable? |  | x |  |
| 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? | x |  |  |
| 9 | Do all mandatory fields require user input? | x |  |  |
| 10 | If any of the database columns do not 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.) | x |  |  |

Table 6‑4: Checklist of Validation

### Submission Checklist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | | Yes | No | N/A |
| TABLES AND FIGURES | | | | |
| 1 | Does every table column, including the stub column, have a heading? | x |  |  |
| 2 | Are all tables referred to in text? | x |  |  |
| 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? |  |  | x |
| 4 | Is lettering in a figure no smaller than 8 points and no larger than 14 points? |  | x |  |
| 5 | Are the figures being submitted in a file format acceptable to the publisher? |  |  | x |
| 6 | Has the figure been prepared at a resolution sufficient to produce a high-quality image? |  |  | x |
| 7 | Are all figures numbered consecutively with Arabic numerals? |  | x |  |
| 8 | Are all figures and tables mentioned in the text and numbered in the order in which they are mentioned? | x |  |  |
| REFERENCES | | | | |
| 1 | Are references cited both in text and in the reference list? |  |  | x |
| 2 | Do the text citations and reference list entries agree both in spelling and in date? |  |  | x |
| 3 | Are journal titles in the reference list spelled out fully? |  | x |  |
| 4 | Are the references (both in the parenthetical text citations and in the reference list) ordered alphabetically by the authors’ surnames? |  | x |  |
| 5 | Are inclusive page numbers for all articles or chapters in books provided in the reference list? |  | x |  |
| 6 | Are references to studies included in your meta-analysis preceded by an asterisk? |  | x |  |
| FORMAT | | | | |
| 1 | Have you checked the journal’s website for instructions to authors regarding specific formatting requirements for submission? |  | x |  |
| 2 | Is the entire manuscript—including quotations, references, author note, content footnotes, and figure captions—double-spaced? Is the manuscript neatly prepared? |  | x |  |
| 3 | Are the margins at least 1 in. (2.54 cm)? |  |  | x |
| 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? |  |  | x |
| 5 | Are all pages numbered in sequence, starting with the title page? | x |  |  |
| PARAGRAPHS AND HEADINGS | | | | |
| 1 | Is each paragraph longer than a single sentence but not longer than one manuscript page? |  |  | x |
| 2 | Do the levels of headings accurately reflect the organization of the paper? | x |  |  |
| 3 | Do all headings of the same level appear in the same format? | x |  |  |

Table 6‑5: Submission Checklist

# Software User’s Manual

## Installation Guide

### Setting up environment at server side

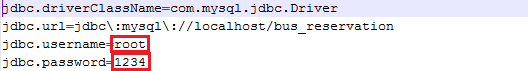
The following software must be installed into the server machine:

* Microsoft Windows 7 or a higher version
* Apache Tom Cat version 6 (can download from <http://tomcat.apache.org/>)
* Java Runtime Environment version 6 (can download from <http://www.microsoft.com/en-us/download/details.aspx?id=21>)
* mysql-5.1.65-win32 or higher (can download at <http://www.mysql.com/downloads/> )

After all necessary software is installed into the server; the deployed device needs to get the path to the Tom Cat server’s folder by setting CATALINA\_HOME environment variable to Tom Cat server’s folder

### Deployment at server sites

* Open MySQL and import bus-reservation.sql to create a new schema named bus\_reservation
* Copy bus-reservation.war file to %CATALINA\_HOME%\webapps
* Start Tomcat server by running %CATALINA\_HOME%\bin\startup.bat
* After the server successfully start up, modify the following files:
  + %CATALINA\_HOME%\webapps\bus-reservation\src\main\resources\ database.properties and change the following rows:



Change the value of jdbc.username to MySQL login username

Change the value of jdbc.password to MySQL login password

### Setting up environment at client side

The client devices need to have one of the following browsers to access the bus-reservation websites:

* Google Chrome version 20 or higher(can download at <https://www.google.com/intl/en/chrome/browser/>)

## User’s Guide

### Guest Guide

#### Register



Figure 7‑1: Register button

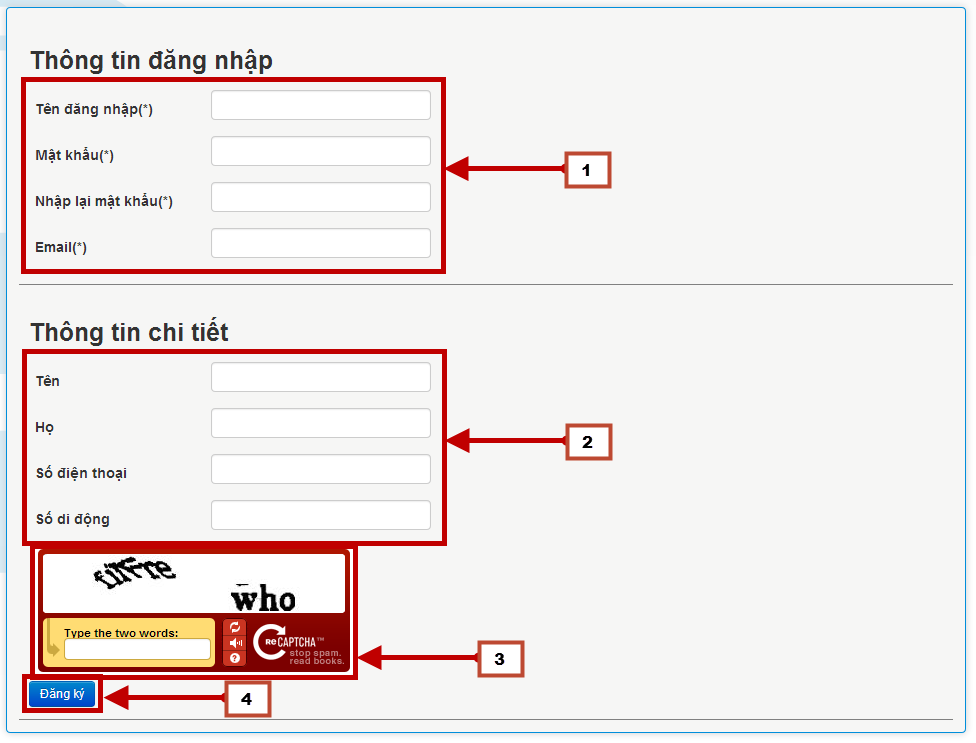


Figure 7‑2 Register page

| **STEP** | **DESCRIPTION** |
| --- | --- |
| **1** | Click button “Đăng kí” |
| **2** | Fill username, password, confirmed password, email [1] (required)  Fill customer’s personal information [2] (optional)  Type the displayed captcha [3] (required) |
| **3** | Click button “Đăng kí” [4] |
| **4** | Check user’s mailbox for activation mail. Follow the instruction inside to complete registration process |

Table 7‑1 Register steps

#### Create reservation:

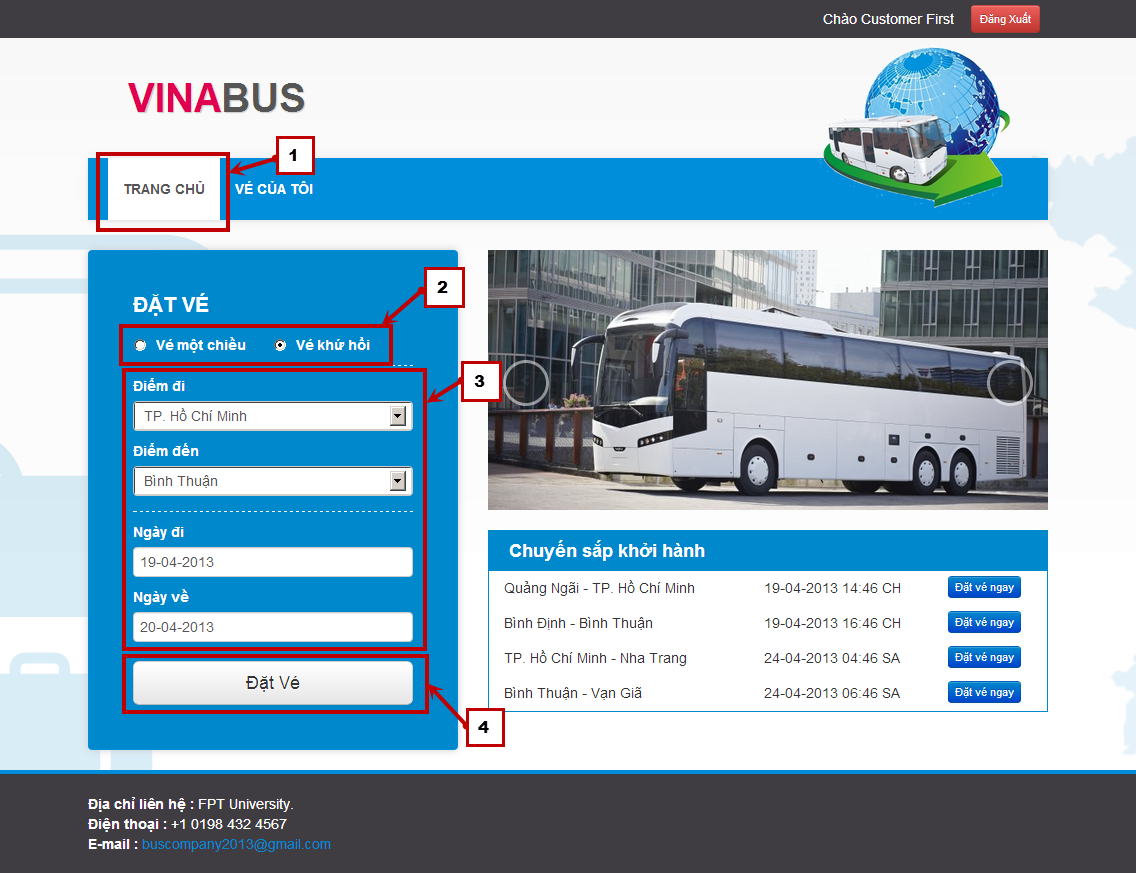


Figure 7‑3: Input search criteria

| **STEP** | **DESCRIPTION** |
| --- | --- |
| **1** | Click tab “Trang chủ” |
| **2** | Choose ticket type, one-way or round-trip |
| **3** | Choose origin, destination, departure date, arrival date\*  \* Only choose arrival date for round trip |
| **4** | Click button “Đặt vé” |

Table 7‑2 Search steps

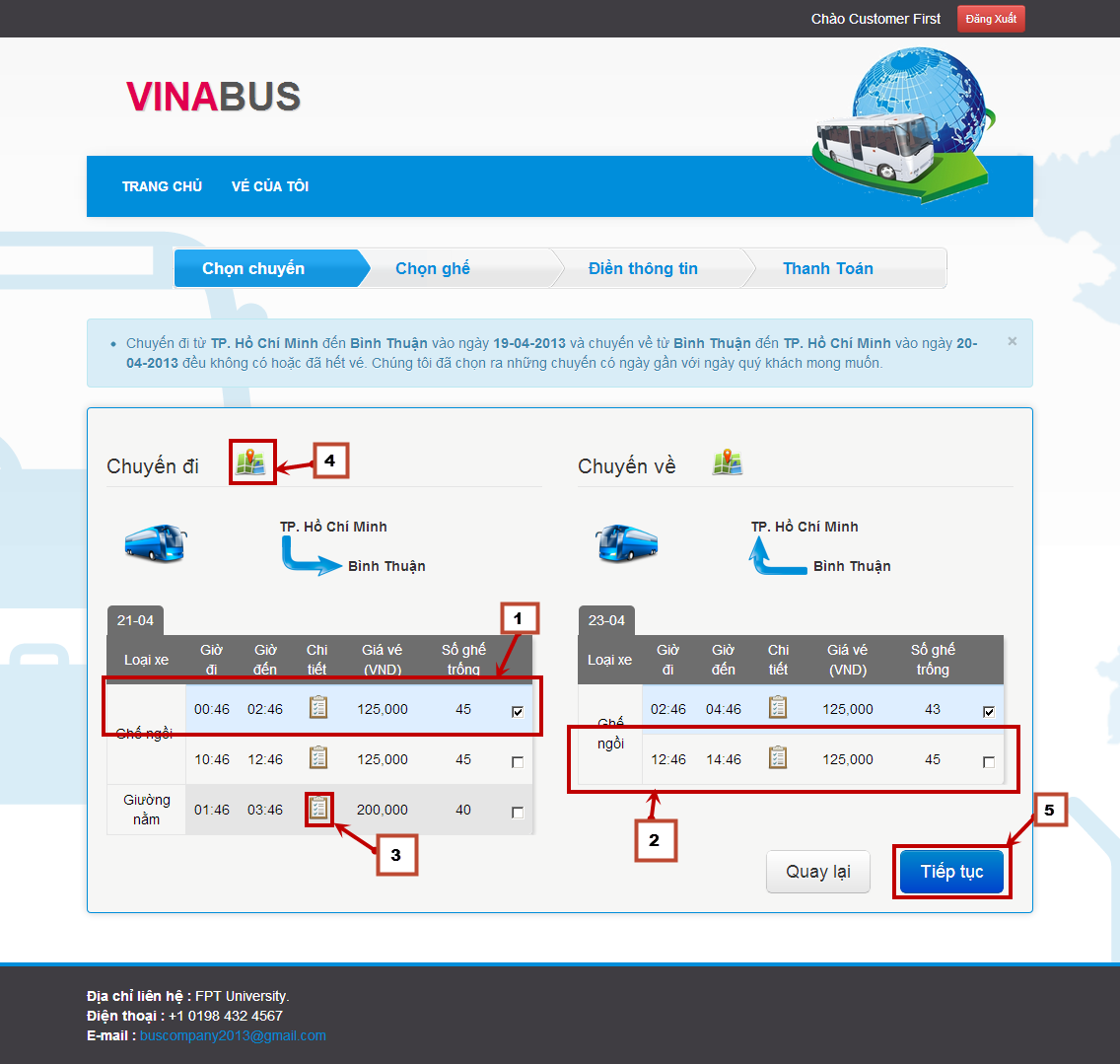


Figure 7‑4: Choose trip

| **STEP** | **DESCRIPTION** |
| --- | --- |
| **5** | Click on a trip to select it [2]  Click on a selected trip to deselect it [1]  Click button view details [3] to view the trip details  Click button view map [4] to view the route |
| **6** | Select at least one trip and click “Tiếp tục” to book for the selected trip(s) |

Table 7‑3 Choose trip steps

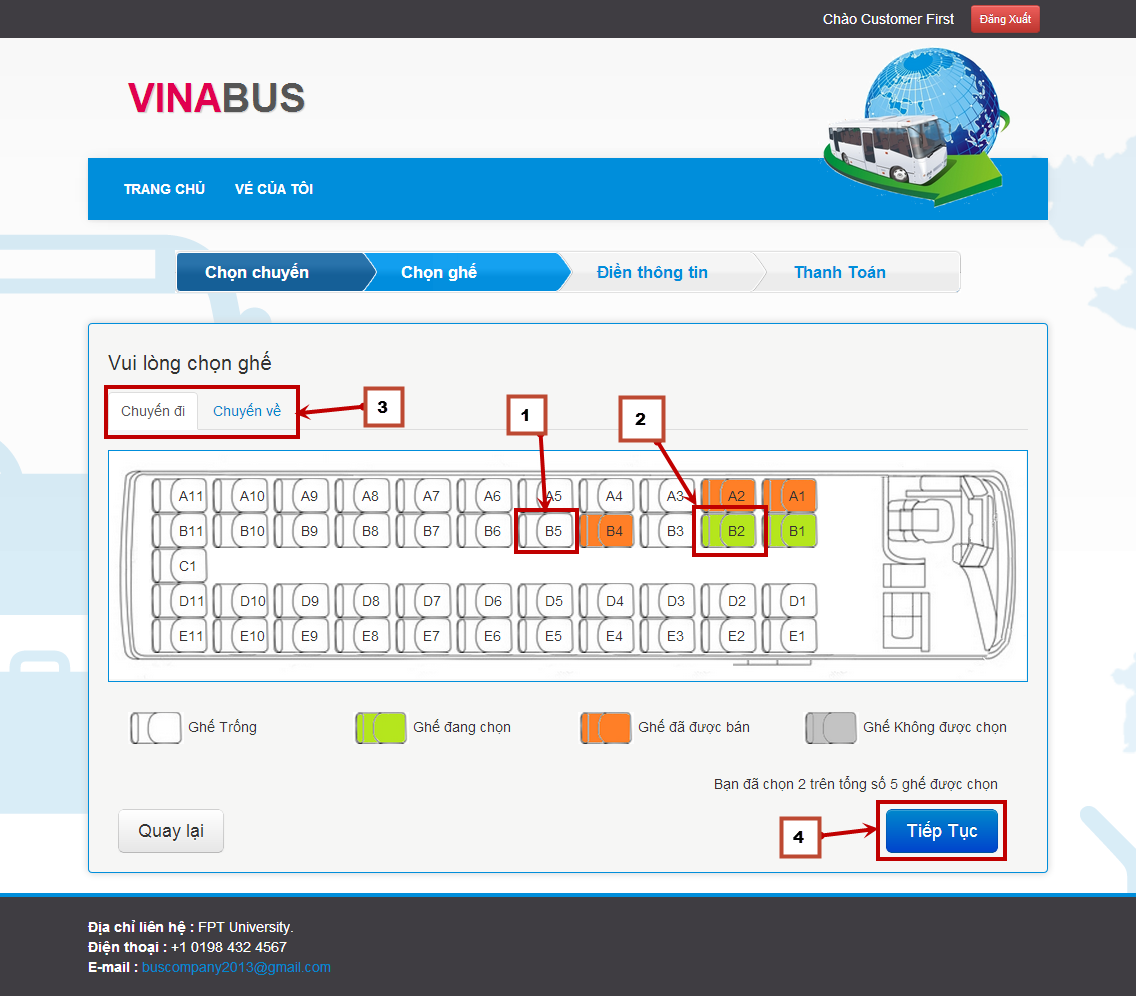


Figure 7‑5: Choose seat

| **STEP** | **DESCRIPTION** |
| --- | --- |
| **7** | Click on a white seat [1] to select it  Click on a selected seat [2] to deselect it  Click on the trip tab [3] to switch trip (for round-trip only) |
| **8** | Select from 1 to 5 seats for each trip and click button “Tiếp tục” |

Table 7‑4 Choose seat steps

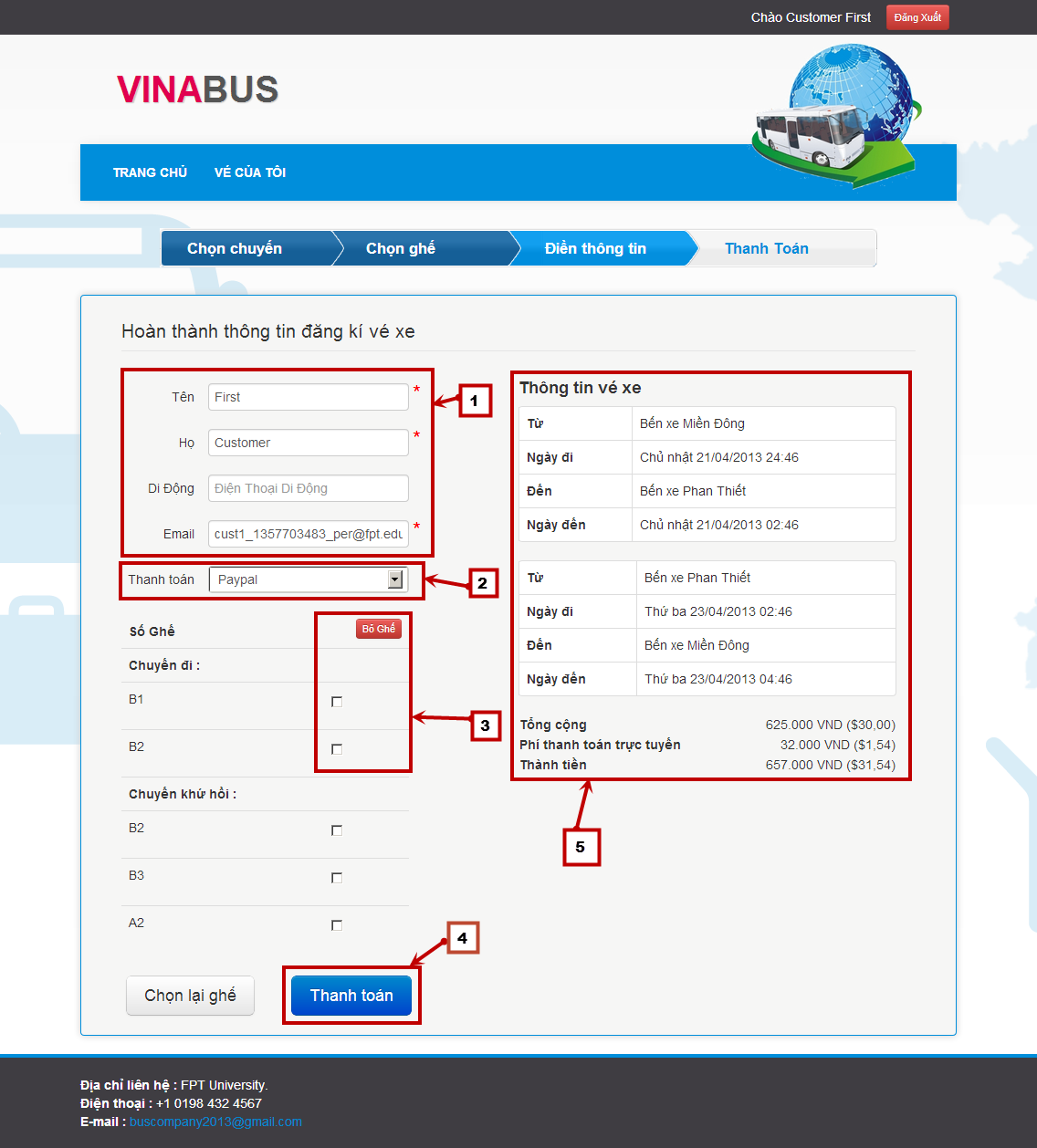


Figure 7‑6: Fill booker’s information

| **STEP** | **DESCRIPTION** |
| --- | --- |
| **9** | Fill booker’s information [1] |
| **10** | Choose a payment method [2] |
| **11** | Select one or many seat and click button “Bỏ ghế”[3] to remove that seat from the reservation |
| **12** | Confirm the reservation information [5] |
| **13** | Click button “Thanh toán” to pay for the reservation |

Table 7‑5 Fill booker’s information steps

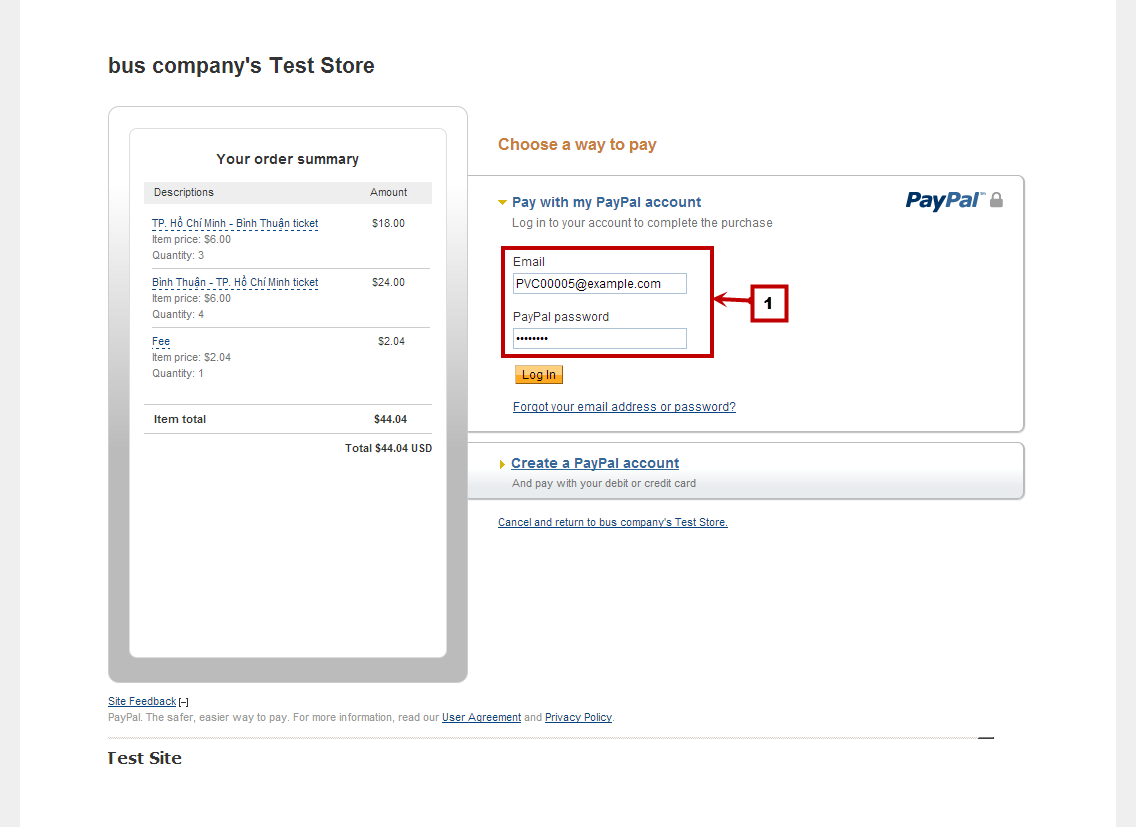


Figure 7‑7: Pay with PayPal

| **STEP** | **DESCRIPTION** |
| --- | --- |
| **11** | Login into PayPal account |
| **12** | Authorize the payment to complete booking process |

Table 7‑6 Pay with PayPal steps

#### View reservation details

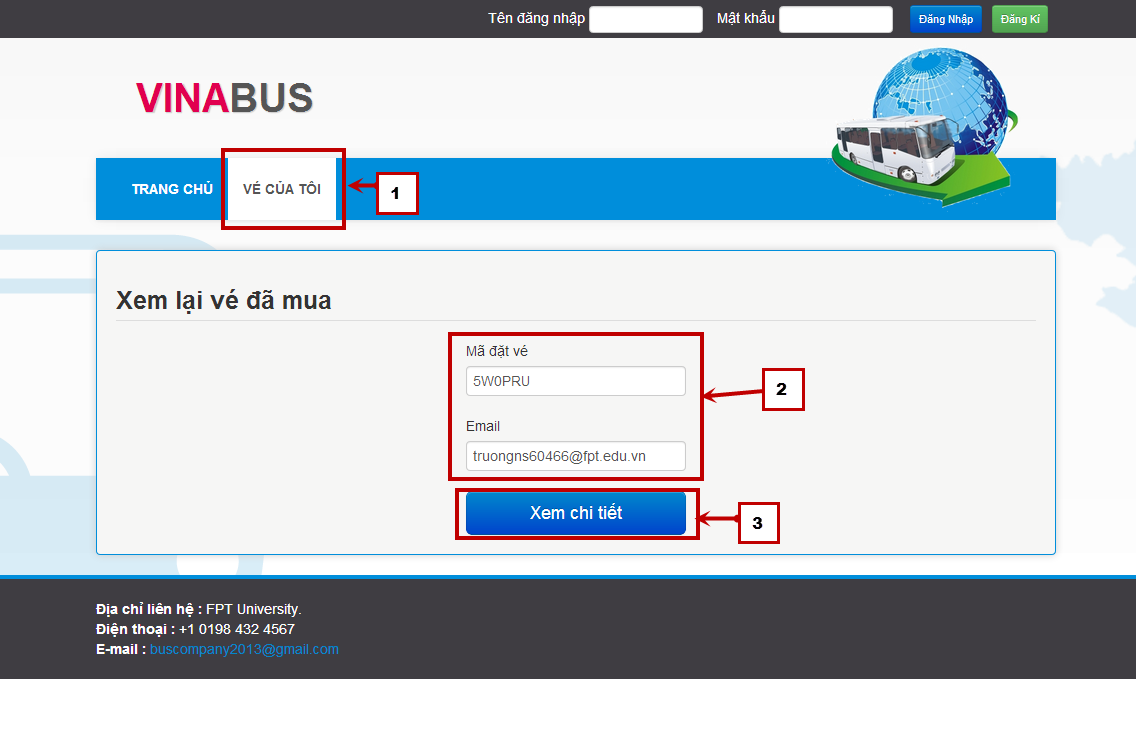


Figure 7‑8: Search reservation

| **STEP** | **DESCRIPTION** |
| --- | --- |
| **1** | Click tab “Vé của tôi” [1] |
| **2** | Enter reservation code and booker’s email [2] |
| **3** | Click button “Xem chi tiết” |

Table 7‑7 Search reservation steps

### Customer Guide

#### View reservation history

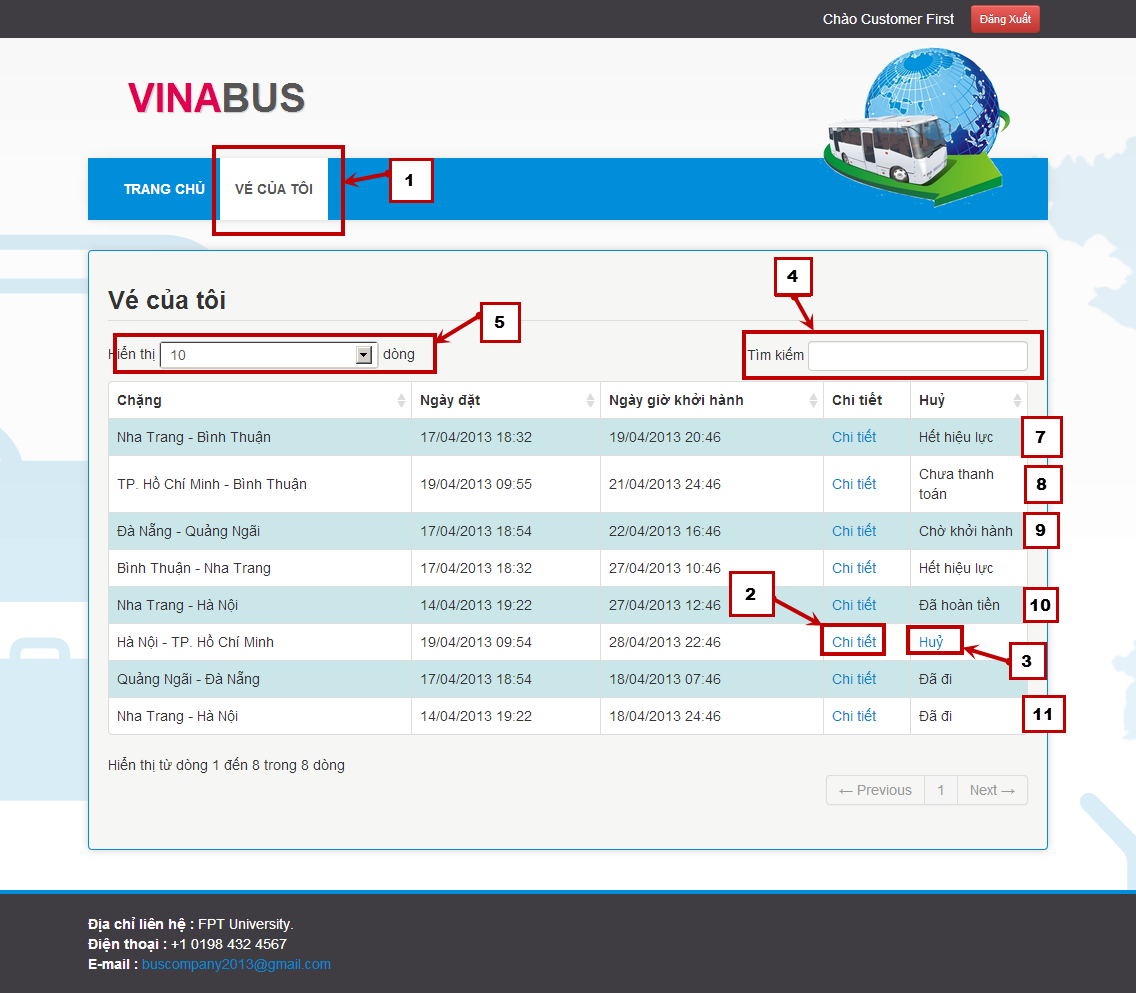


Figure 7‑9: Reservation list

| **STEP** | **DESCRIPTION** |
| --- | --- |
| **1** | Login into customer’s account |
| **2** | Click tab “Vé của tôi” [1] |
| **3** | Click link “Chi tiết”[2] to view reservation details |
| **4** | Click link “Huỷ”[3] to cancel the journey |

Table 7‑8 View reservation steps

### Staff Guide

#### Manage Bus

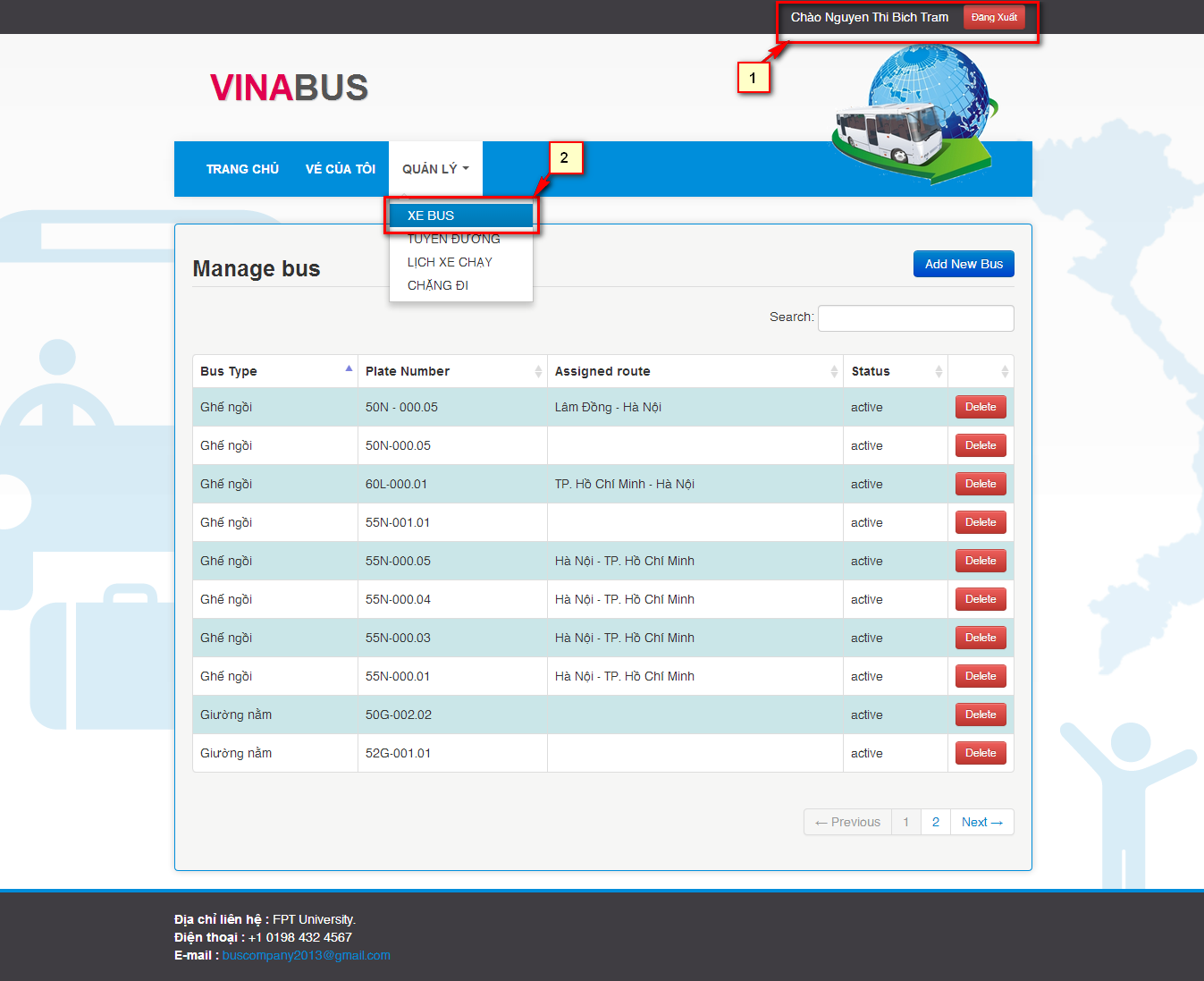


Figure 7‑10 View manage bus screen.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Login with “Staff” role |
| **2** | View list of Bus at “Xe Bus” in drop down menu “Quản lý” |

Table 7‑9: View manage bus screen step

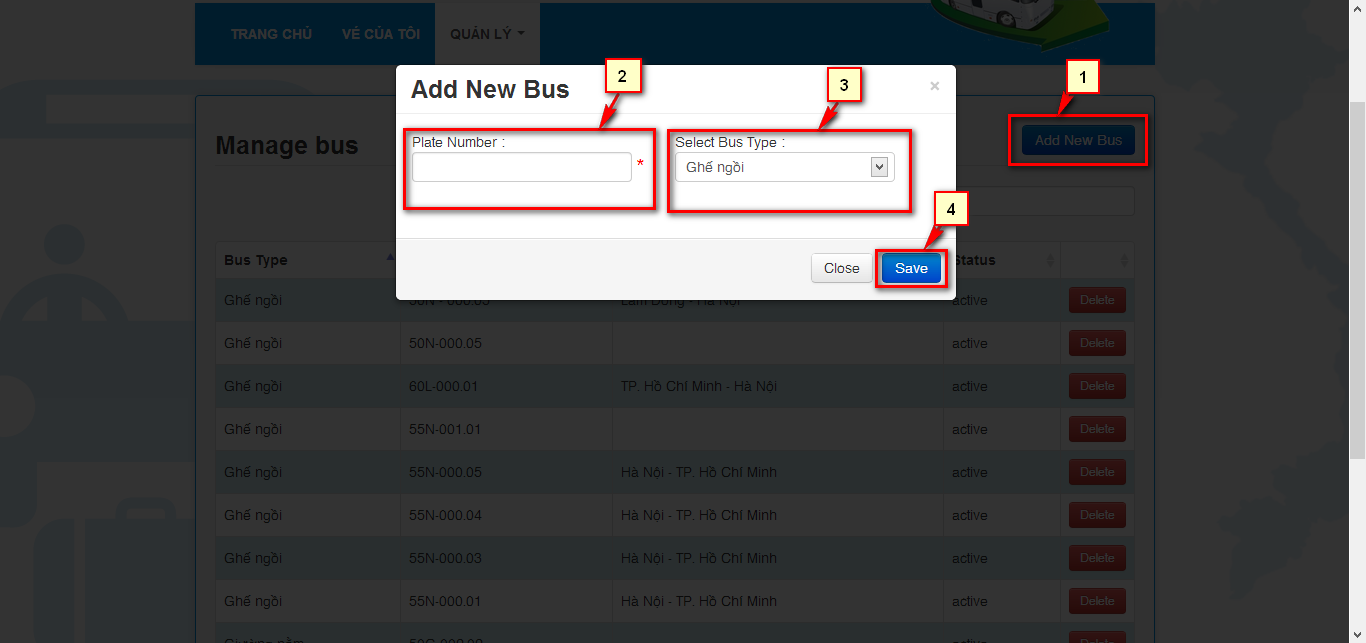


Figure 7‑11 Add bus popup

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on buttion “Add New Bus” |
| **2** | Fill Plate number of bus want to add |
| **3** | Chose bus type in dropdown list |
| **4** | Click on “Save” to save a new bus to database |

Table 7‑10: Add bus step.

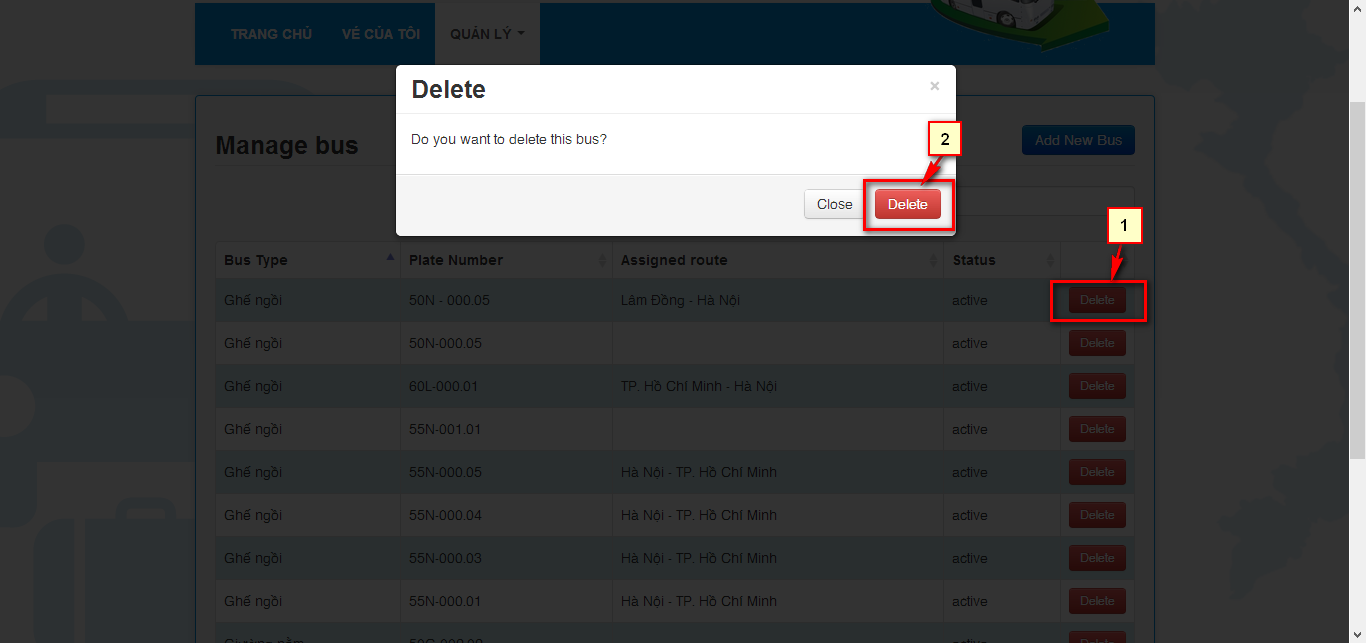


Figure 7‑12 Delete bus popup

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Delete” button add bus want to delete |
| **2** | Click on “Delete” button |

Table 7‑11: Delete bus step.

#### Manage Route:

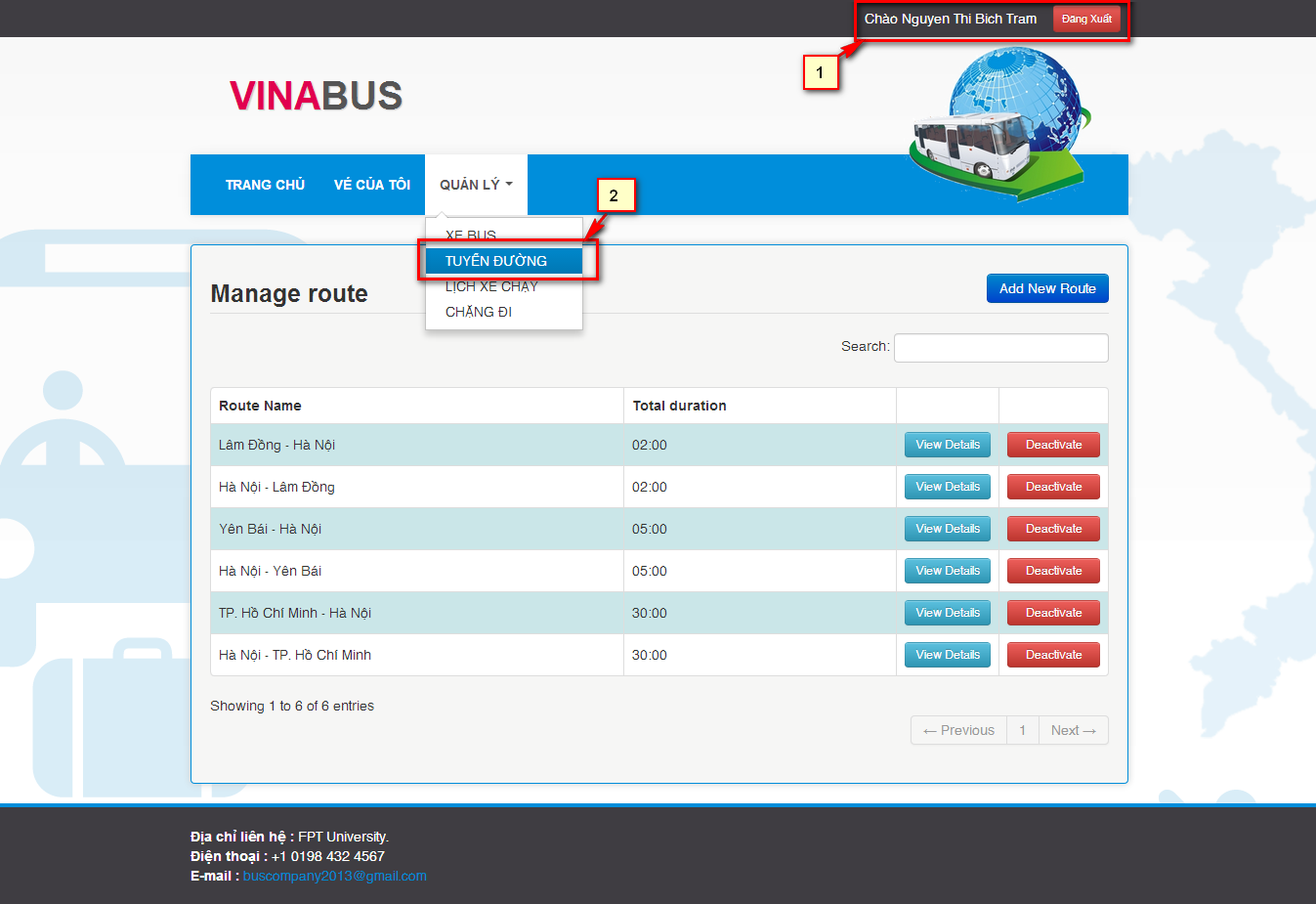


Figure 7‑13 View manage route screen.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Login with “Staff” role |
| **2** | View list of Bus at “Tuyến Đường” in drop down menu “Quản lý” |

Table 7‑12: View manage route screen step.

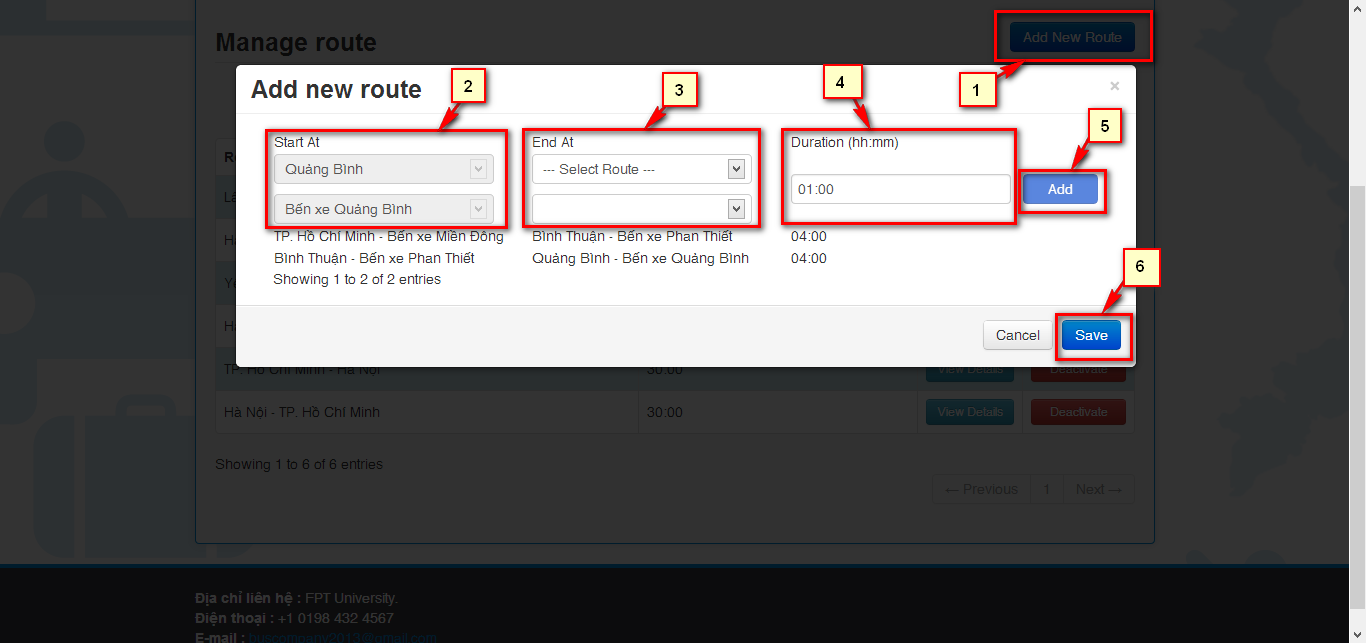


Figure 7‑14 Add route popup.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Add New Route” |
| **2** | Chose place the route start at in “Start At” dropdown list |
| **3** | Chose place the route end at in “End At” dropdown list |
| **4** | Fill the duration of one segment |
| **5** | Click on “Add” to add the segment to the route |
| **6** | Click on “Add” to add route to system |

Table 7‑13: Add route step.

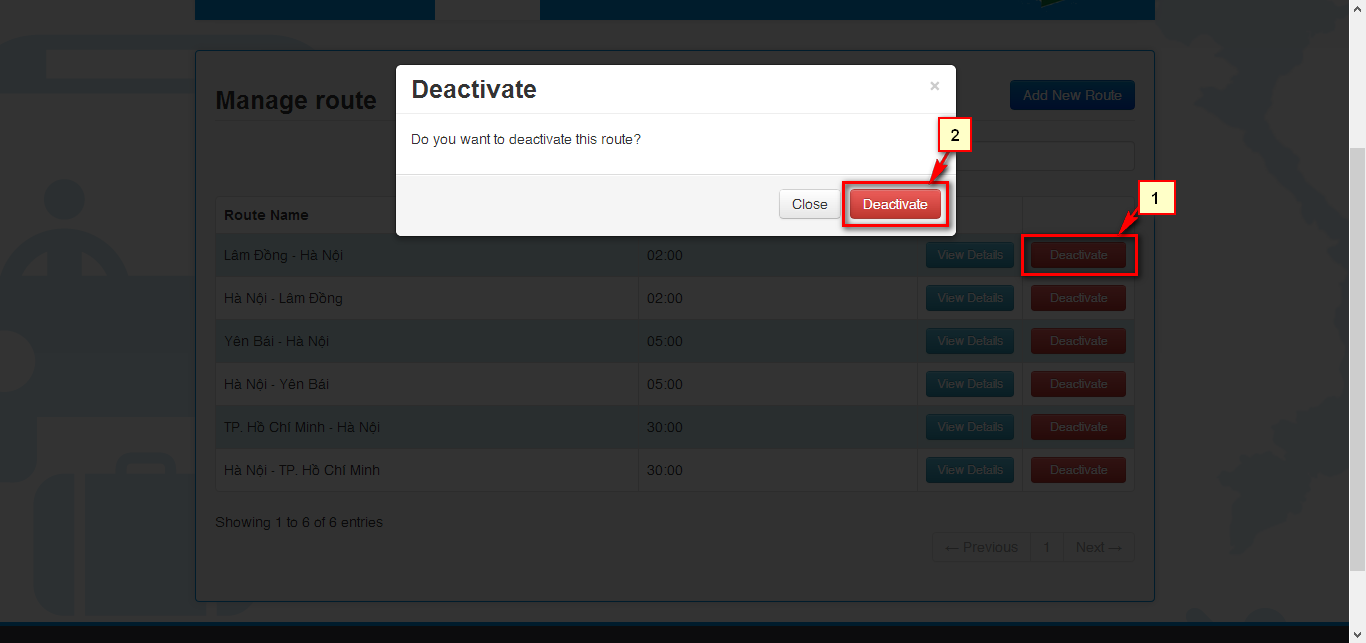


Figure 7‑15 Deactivate route popup.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Deactivate” in route want to deactivate |
| **2** | Click on “Deactivate” to deactivate route |

Table 7‑14: Deactivate route step.

#### Manage Route Details:

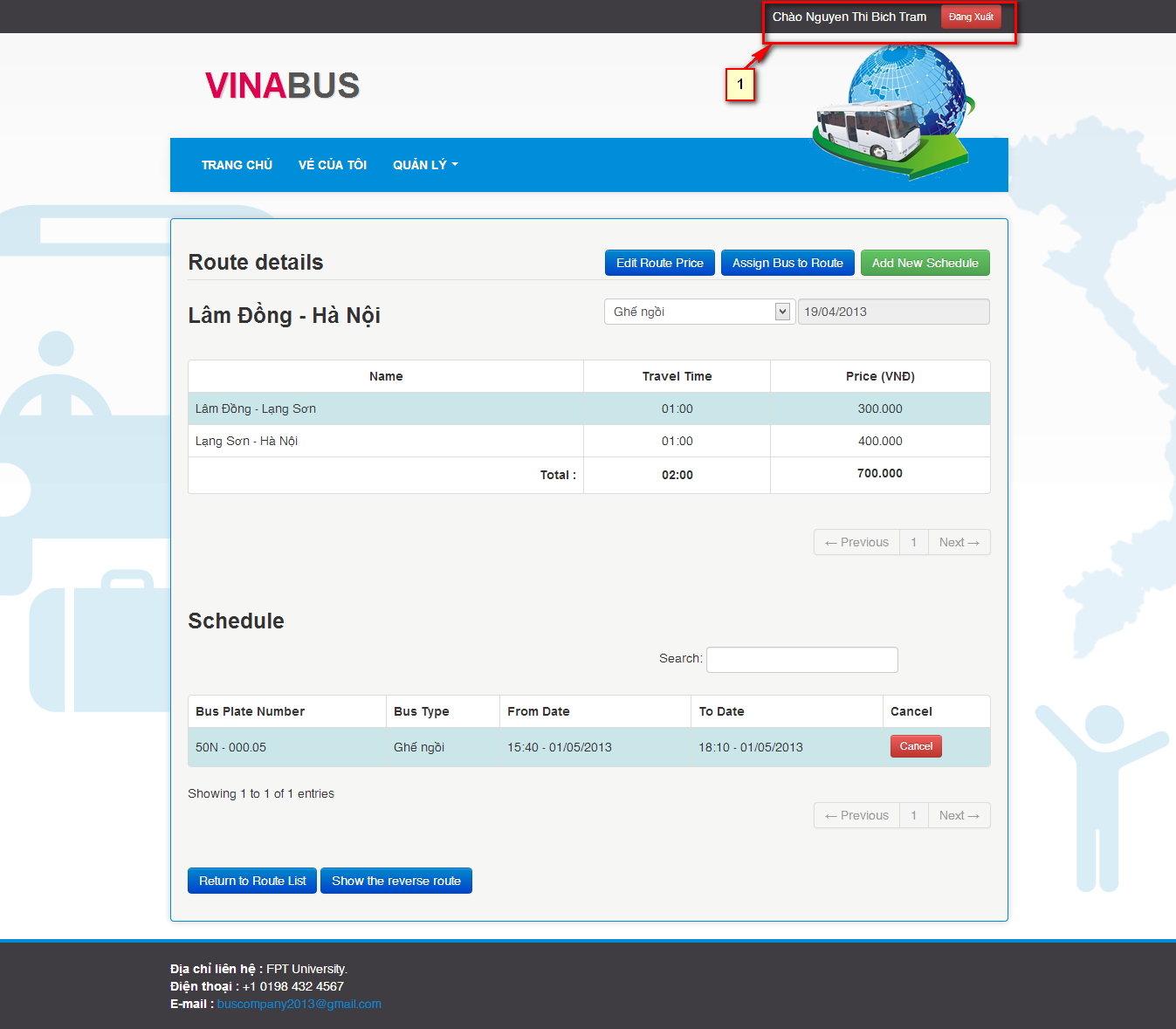


Figure 7‑16 View manage route details screen.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Login with “Staff” role |
| **2** | Click on “View Details” in Manage Route Screen. |

Table 7‑15 View manage route details screen step

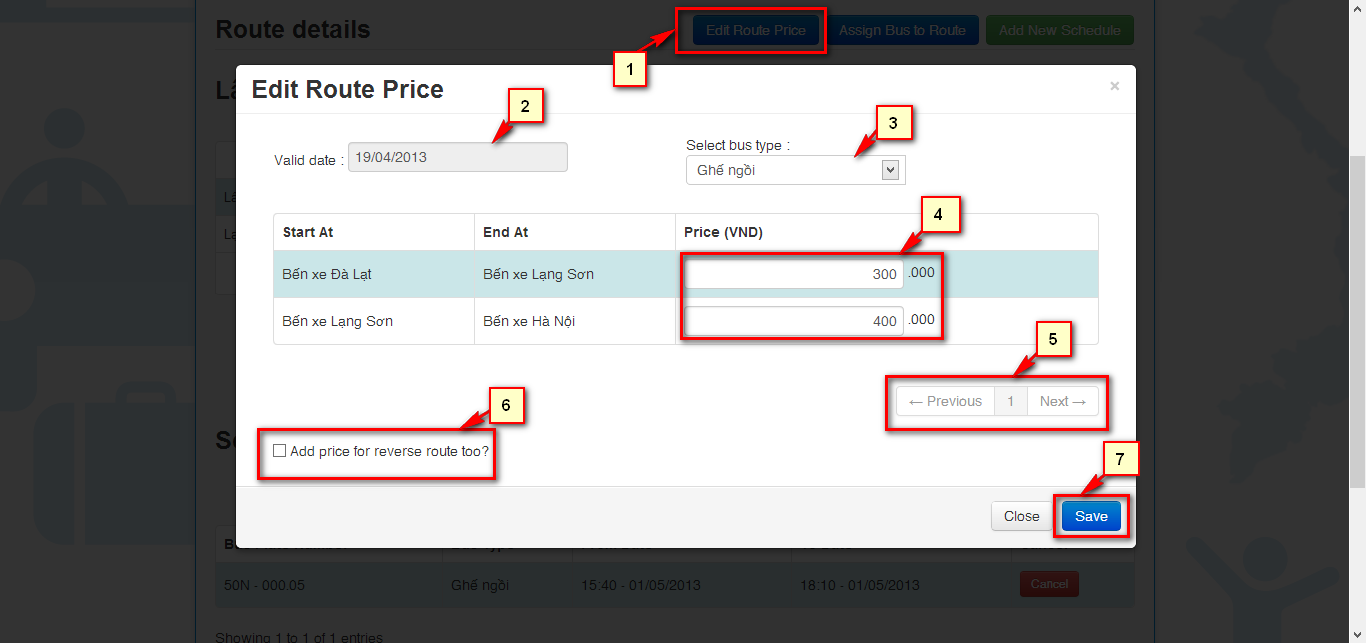


Figure 7‑17 Edit Price popup.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Edit Route Price”. |
| **2** | Chose the “Valid date” when the price of this route will be apply. |
| **3** | Chose “Select bus type” which bus type this price will be apply. |
| **4** | Fill the new price for each segment. |
| **5** | Paging the price list. |
| **6** | Chose “Add price for reverse route too?” to add price for the reverse route. |
| **7** | Click on “Save” to save the new price for this route. |

Table 7‑16: Edit Price step.

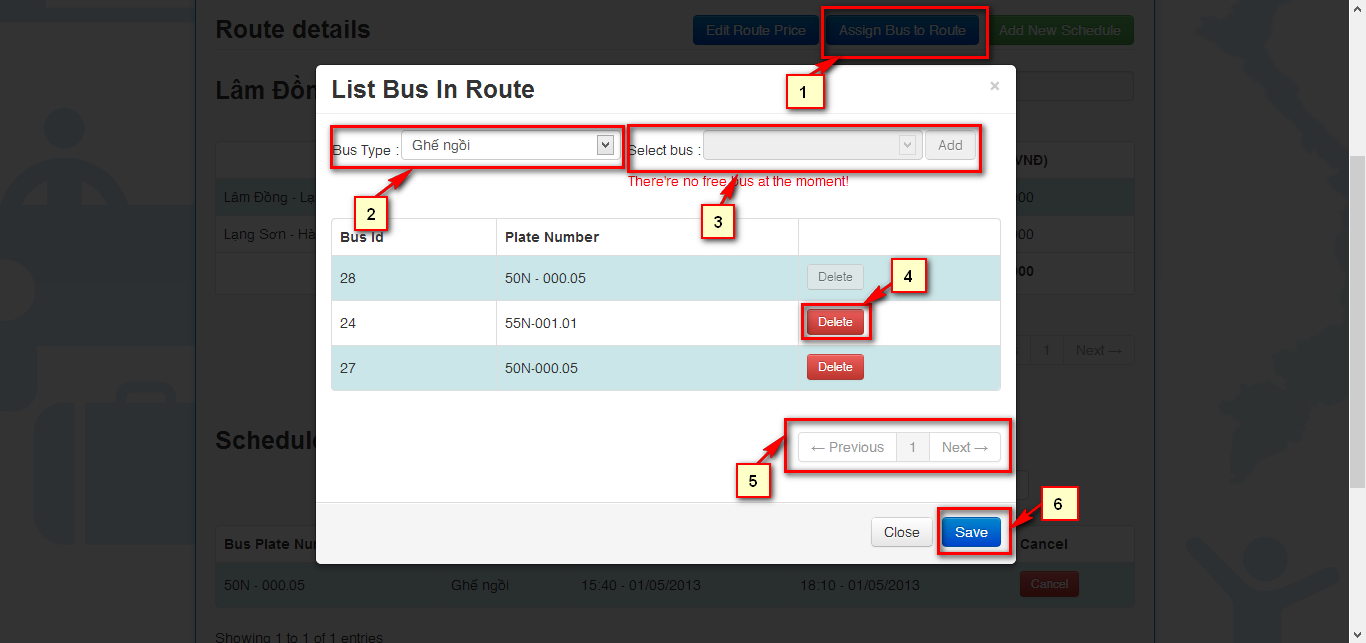


Figure 7‑18 Assign bus to route popup.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Assign Bus To Route”. |
| **2** | Chose “Bus Type” which bus types want to add. |
| **3** | Chose “Select Bus” which buses want to add. |
| **4** | Click on “Delete” in bus want to remove from that route. |
| **5** | Paging the bus list. |
| **6** | Click on “Save” to assign bus to that route. |

Table 7‑17: Assign bus to route step.



Figure 7‑19 Add new schedule popup.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Add New Schedule”. |
| **2** | Chose “From Date” the date start of this schedule. |
| **3** | Chose “To Date” the date end of this schedule. |
| **4** | Chose “Bus type” the bus of the bus run on this schedule. |
| **5** | Chose “Bus Plate Number” the plate number of bus run on this schedule. |
| **6** | Chose “Set auto return to start station” to let bus return to start station. |
| **7** | Chose “From Date” the date this bus will start to return. |
| **8** | Chose “To Date” the date this bus return to start station. |
| **9** | Click on “Save changes” to save the schedule. |

Table 7‑18 Add new schedule step.

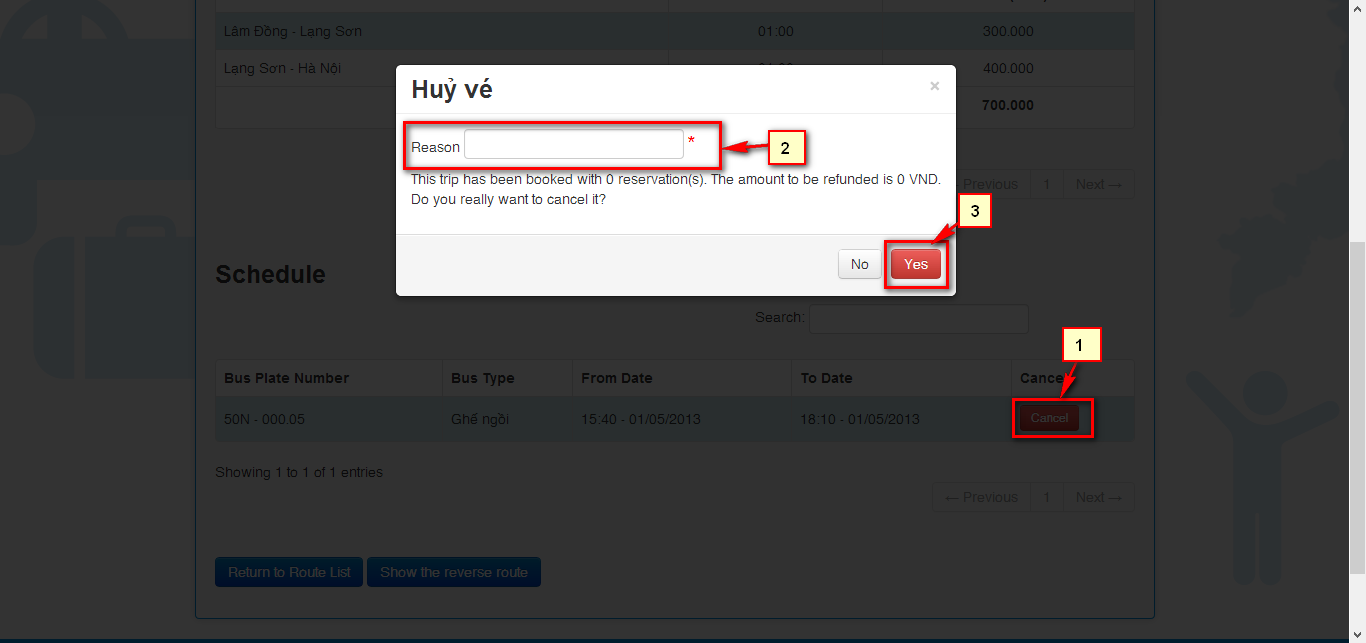


Figure 7‑20 Cancel Schedule popup.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Cancel”. |
| **2** | Fill “Reason” why to cancel the schedule. |
| **3** | Click on “Yes” Cancel the schedule. |

Table 7‑19: Cancel Schedule step.

#### Manage Schedules:

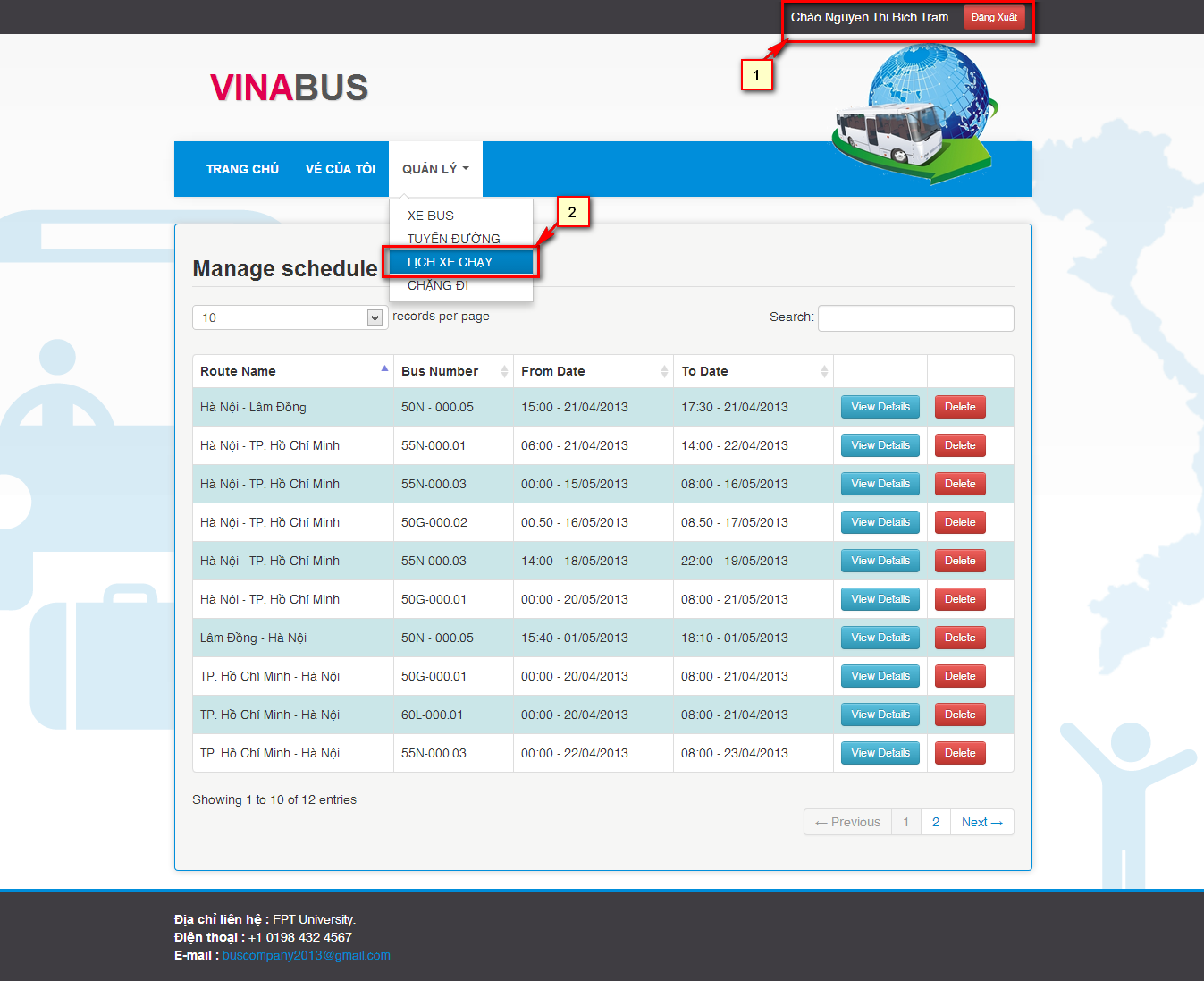


Figure 7‑21 View manage schedule Screen.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Login with “Staff” role |
| **2** | View list of Schedule at “Lịch xe chạy” in drop down menu “Quản lý” |

Table 7‑20 View manage schedule step.

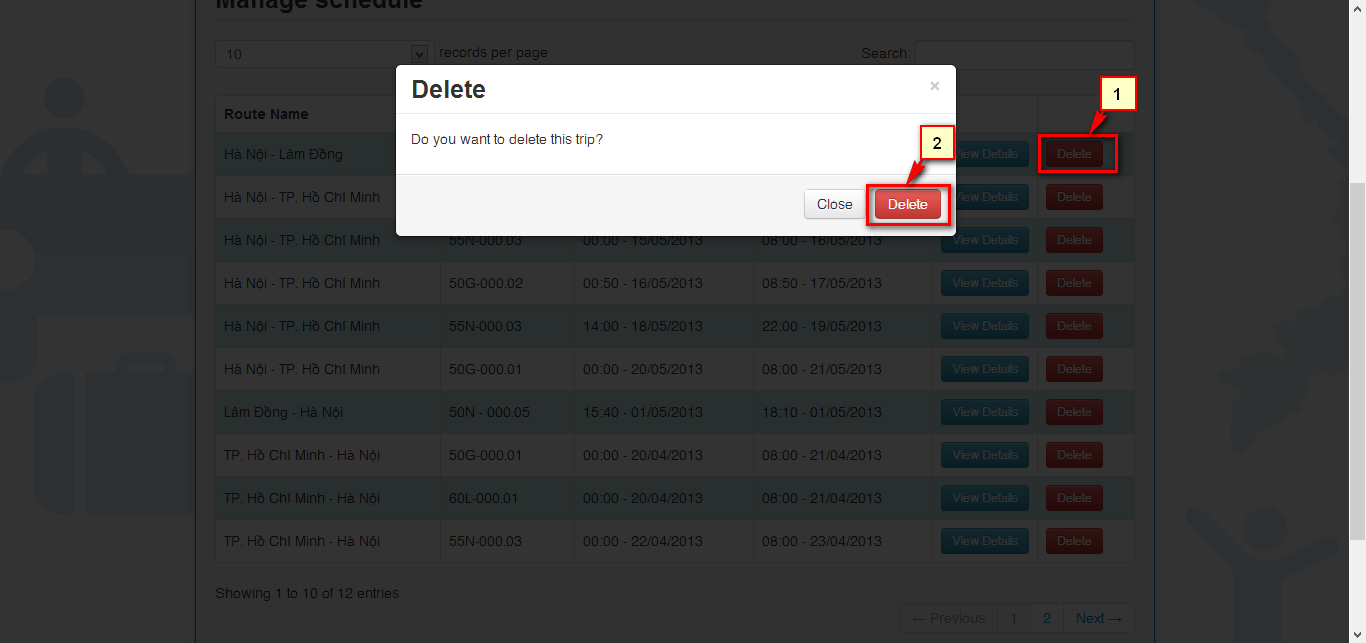


Figure 7‑22 Delete Schedule popup.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Delete” on which schedule want to delete. |
| **2** | Click on “Delete” Delete schedule. |

Table 7‑21: Delete Schedule step.

#### Schedules Details:



Figure 7‑23 View schedule Details Screen.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Login with “Staff” role |
| **2** | Click on “View Details” on Manage Schedule Screen. |

Table 7‑22 View schedule details step.

#### Manage Segment

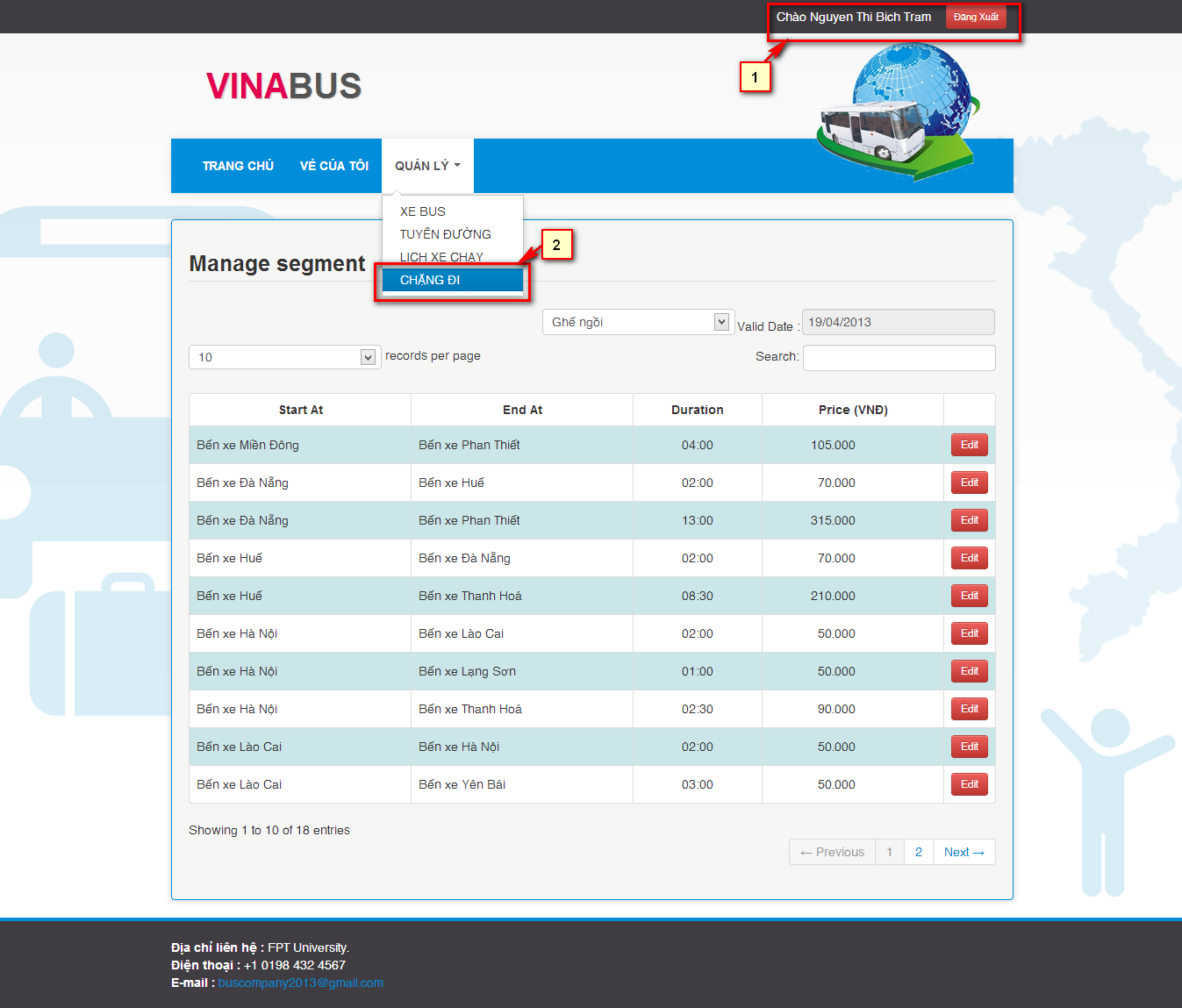


Figure 7‑24 Manage Segment Screen.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Login with “Staff” role |
| **2** | View list of Segment at “Chặng đi” in drop down menu “Quản lý” |

Table 7‑23 View manage segment step.

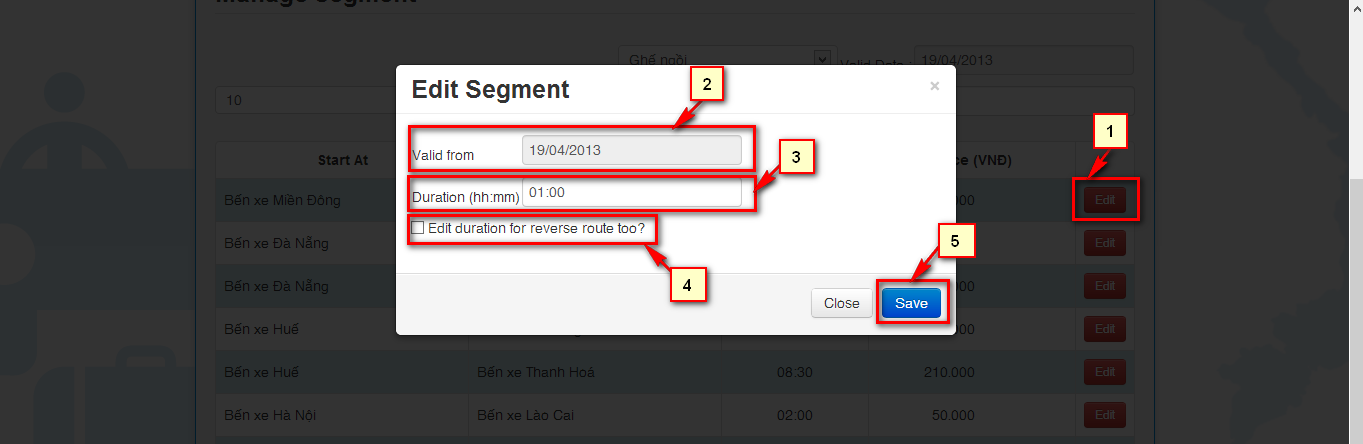


Figure 7‑25 Edit segment popup.

|  |  |
| --- | --- |
| **STEP** | **DESCRIPTION** |
| **1** | Click on “Edit” on what segment want to edit |
| **2** | Chose “Valid from” the date that new duration will be apply. |
| **3** | Fill the new duration. |
| **4** | Choose “Edit duration for reverse segment too?” to edit the duration of reverse segment. |
| **5** | Click on “Save” to save the segment information. |

Table 7‑24 Edit segment step