|  |  |
| --- | --- |
| 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 –

|  |
| --- |
|  |

# 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 can purchase for a ticket directly at the ticket offices or they can book tickets through the phone. Telephone operators or 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.

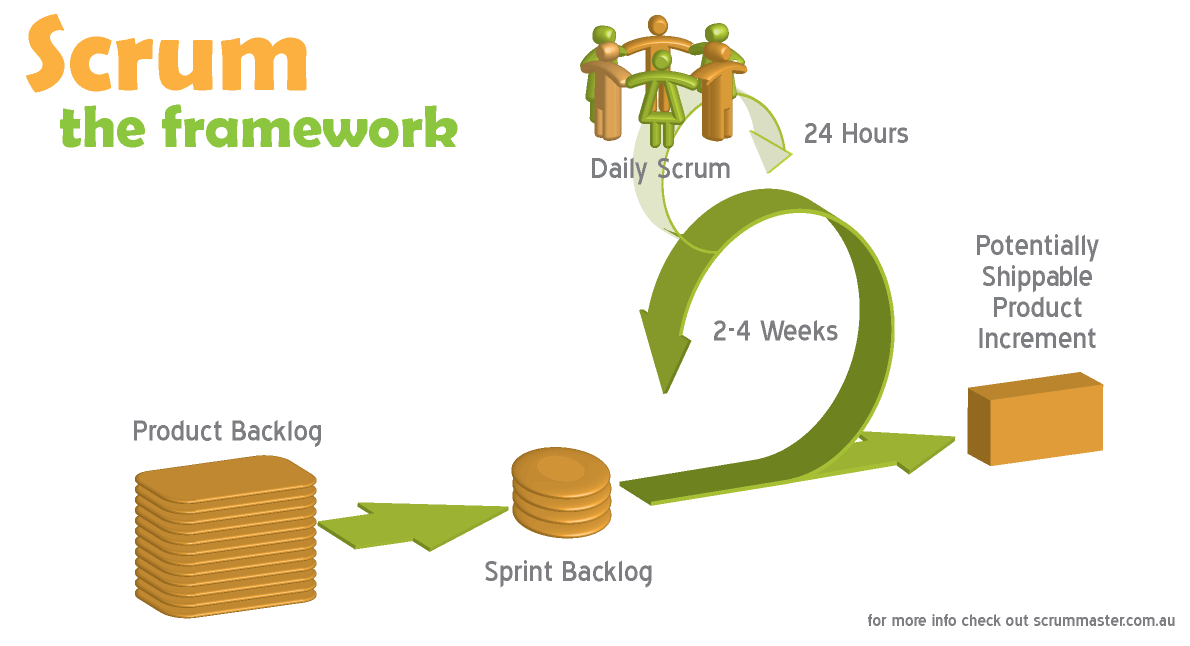


Figure 2‑1 Scrum model

(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 | Build framework | Framework is ready for developing. | 1.4 | Build framework: Struts 2, Spring, Hibernate | 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 |  |
| 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 |  |
| 7 | As a user, I can search for a one-way trip. | Can I search for a one-way trip providing departure, arrival information, number of passenger and preferred bus type? | 3.1 | Create detail design for search function | 3 | Truong |  |
| 3.2 | Code search function | 3 | Truong |  |
| 4.1 | Create test case for search function | 4 | Truong |  |
| 4.2 | Test search function | 4 | Truong |  |

2.3.2. Task Sheet: Assignments and Timetable

2.3.3. All Meeting Minutes

## Coding Convention

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

# Software Requirements Specifications (SRS)

## 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. Edit a route.
12. View list of schedules.
13. Add a new schedule.
14. Delete a schedule.
15. Add new segments.
16. View list of segments.
17. Edit segment.

## System Requirement Specification

### System Features



Figure 3‑1 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 3‑1 Actor description

#### Program flow

1. **Booking journeys**

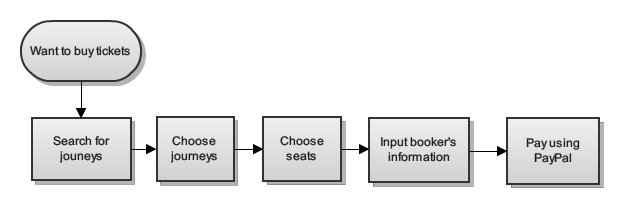


Figure ‑ Flow for booking journeys

1. **Scheduling journeys**

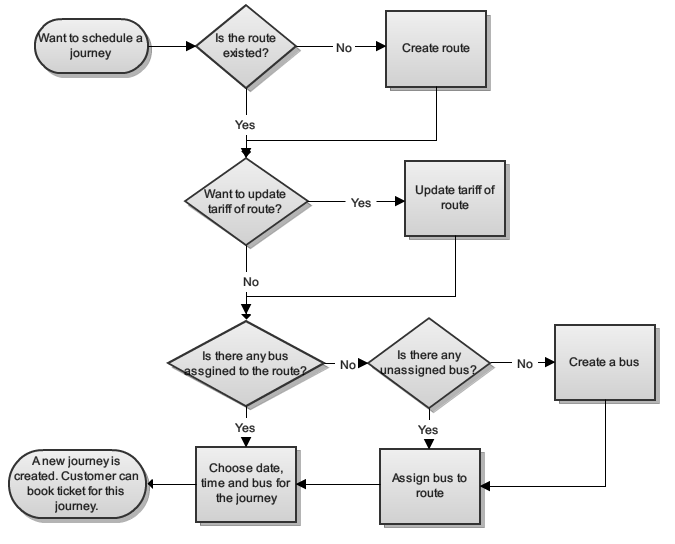
****

Figure 3‑3 Flow for scheduling journey

#### Choose journeys use case



Figure 3‑4 Choose journeys use case

* 1. **Search journeys**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC001 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 3‑2 UC001 Search journeys specification



Figure 3‑5 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 3‑3 Search journey screen description

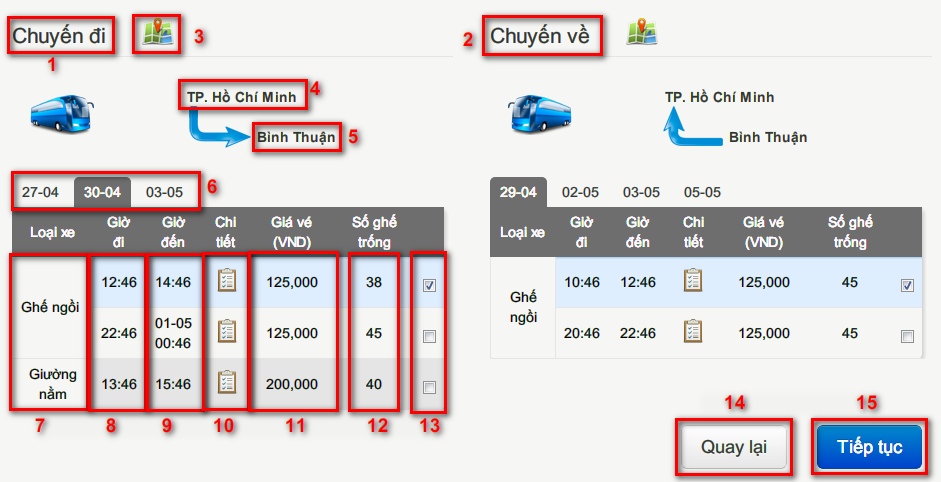


Figure ‑ Search result screen

|  |  |  |
| --- | --- | --- |
| **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 3‑4 Search result screen description

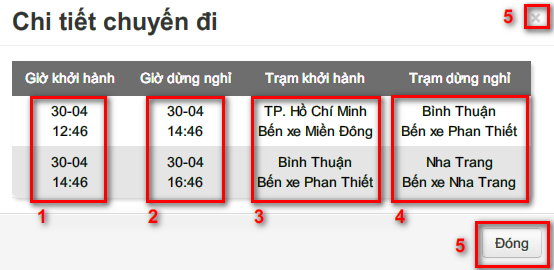


Figure ‑ Journey details screen

|  |  |  |
| --- | --- | --- |
| **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 ‑ Journey details screen description

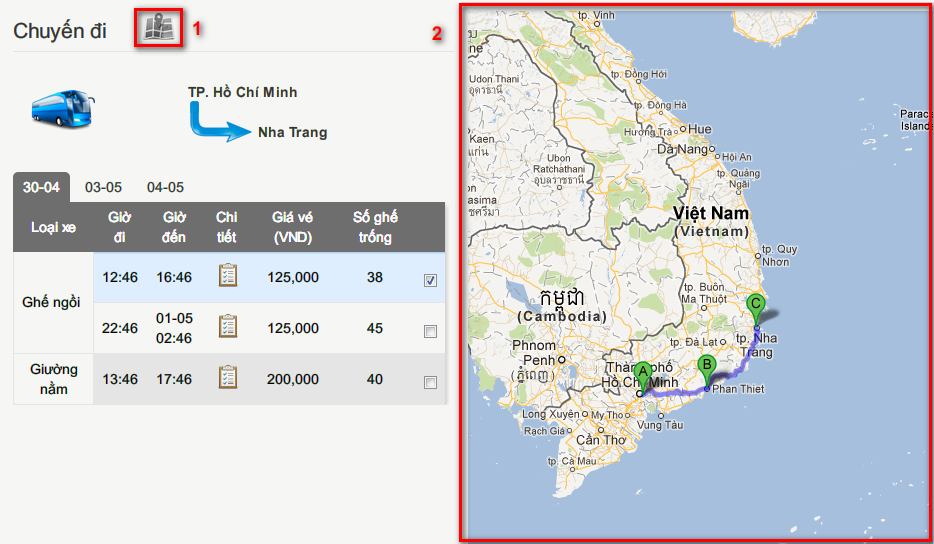


Figure ‑ Map screen

|  |  |  |
| --- | --- | --- |
| **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 ‑ Map screen description

* 1. **Choose journeys**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC001 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:**  <None>  **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 ‑ UC002 Choose journeys use case specification

#### Choose seat use case



Figure ‑ Choose seat use case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC003 | **Use-case Version** | V0.1 |
| **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:** Customer and unauthorized 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 ‑ UC0003 Choose seat use case specification

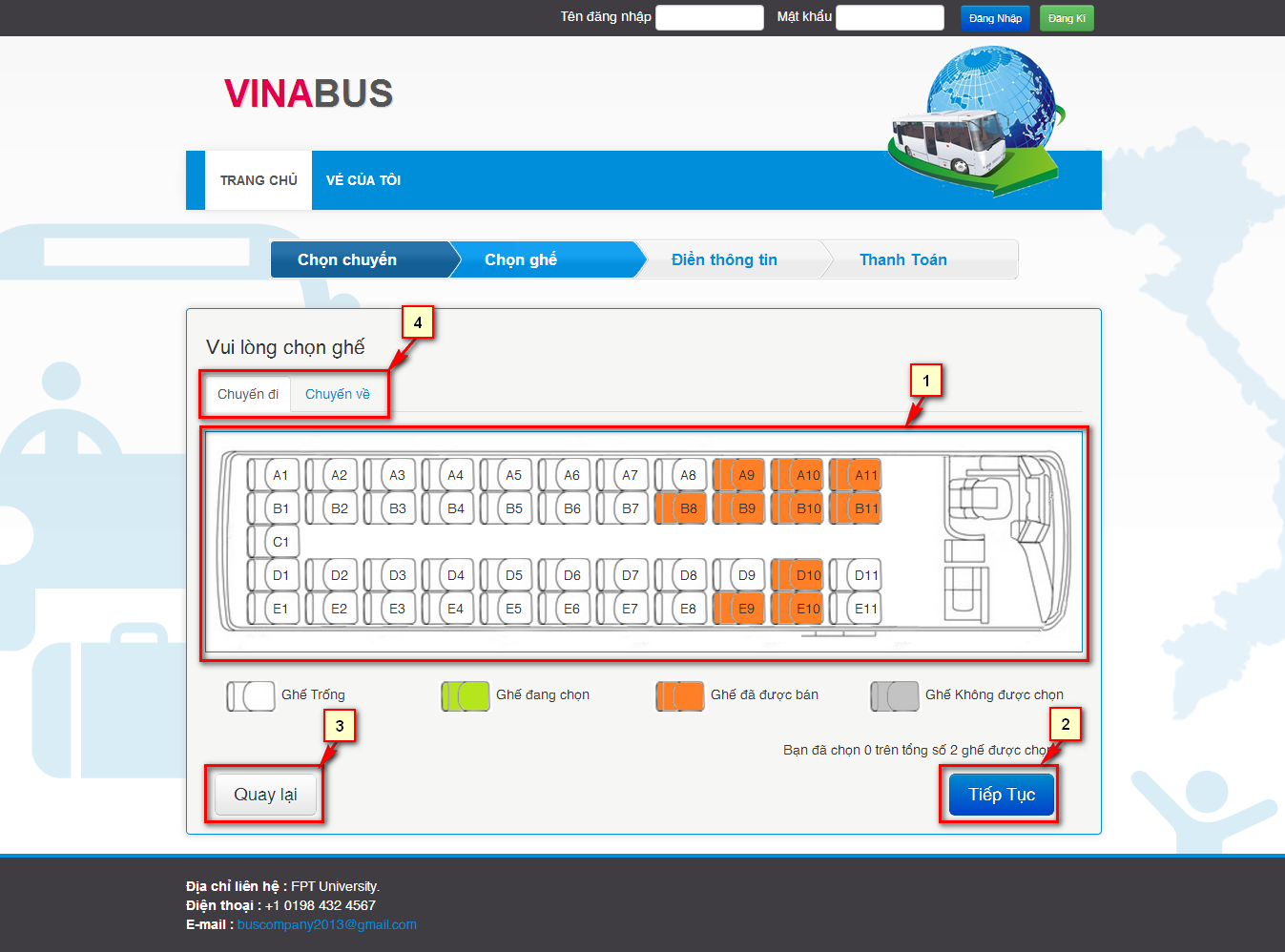


Figure ‑ Choose seat screen

|  |  |  |
| --- | --- | --- |
| **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 ‑ Choose seat screen description

#### Provide booker’s information use case



Figure ‑ Provide booker’s details use case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC003 SPECIFICATION** | | | |
| **Use-case No.** | UC004 | **Use-case Version** | v0.1 |
| **Use-case Name** | Provide booker’s details | | |
| **Author** | Nguyễn Ngọc Sơn, Nguyễn Sơn Trường | | |
| **Date** | 31/1/2013 | **Priority** | High |
| **Actor:** User  **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] * E-mail [“E-mail”: textbox, regular expression] * 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 ‑ UC004 Provide booker’s details user case specification



Figure 4- Customer and Unauthorized User Provide info 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. |

#### Pay use case



Figure ‑ Pay use case

* 1. **Pay with PayPal**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC004 SPECIFICATION** | | | | |
| **Use-case No.** | UC005 | **Use-case Version** | | V0.1 |
| **Use-case Name** | Pay with Paypal | | | |
| **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:**   |  |  | | --- | --- | |  |  | | Actor Actions | System Response | | [Exception 2]  1. Error in requesting payment information or executing payment with PayPal site | 2. Show message “Quá trình thanh toán bằng Paypal đã xảy ra lỗi. Xin quý khách vui lòng thử lại hoặc dùng tài khoản khác.” |   **Relationships:**  Login, Provide booker’s details  **Business Rules:**  Payment process must be completed within 15 minutes after reservation is created. | | | | |

**Table 3‑11: Pay with Paypal Use Case Specification**

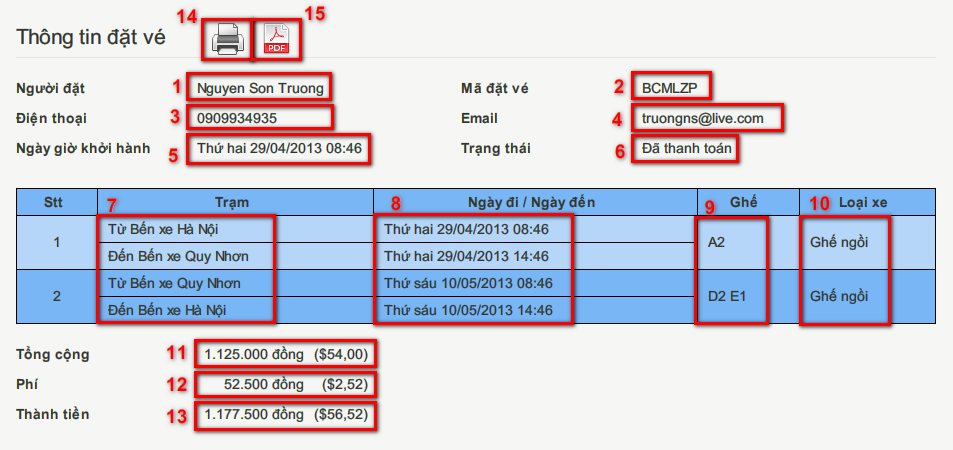


Figure 3‑13- Review reservation details screen

|  |  |  |
| --- | --- | --- |
| **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 3‑12: Review reservation details screen’s Controls**

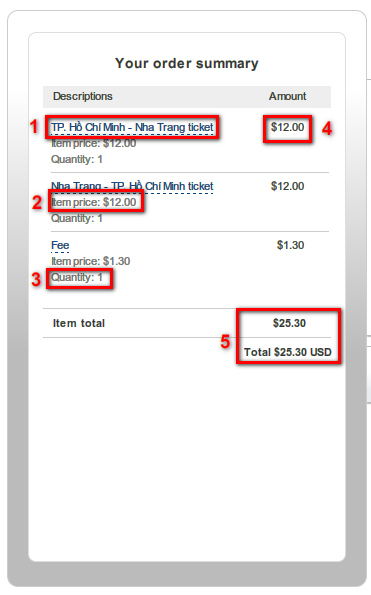


Figure 3‑14- Payment details screen

|  |  |  |
| --- | --- | --- |
| **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 3‑13: Paypal site screen’s Controls**

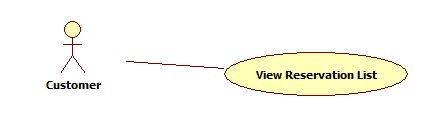
* + - 1.  View Reservation List use case

Figure 3‑15- View Reservation List use case diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC005 SPECIFICATION** | | | | |
| **Use-case No.** | UC005 | **Use-case Version** | | v0.1 |
| **Use-case Name** | View Reservation List | | | |
| **Author** | Nguyen Luong Hai | | | |
| **Date** | 31/01/2013 | **Priority** | High | |
| **Actor:**  User (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:**   |  |  | | --- | --- | | Actor Actions | System Response | | 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 3‑14: View Reservation List Use Case Specification**

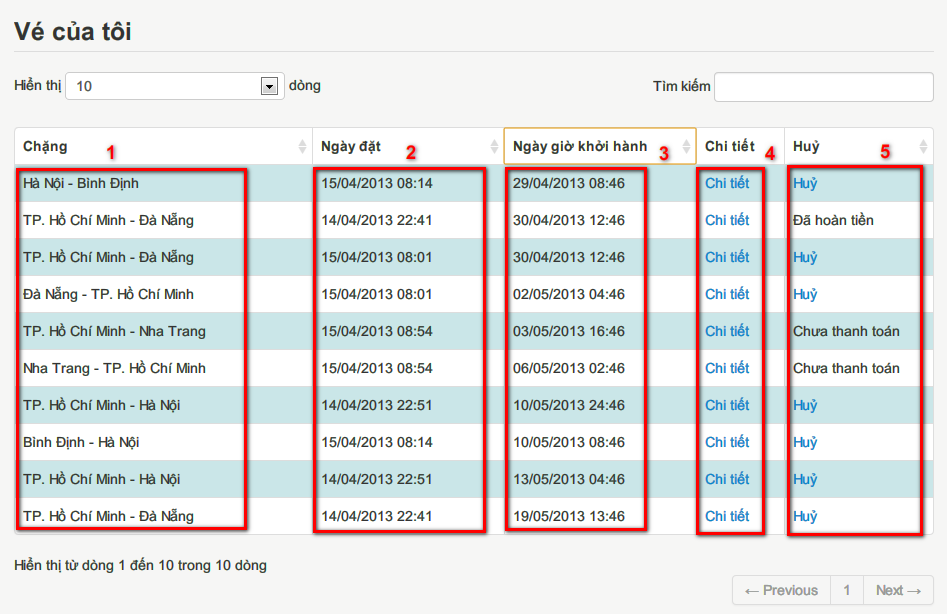


Figure 3‑16- Reservation List screen

|  |  |  |
| --- | --- | --- |
| **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 3‑15: Reservation List screen’s Controls**

* + - 1. View reservation details use case



Figure 3‑17- View reservation details use case diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC006 SPECIFICATION** | | | |
| **Use-case No.** | UC006 | **Use-case Version** | v0.1 |
| **Use-case Name** | View reservation details | | |
| **Author** | Nguyen Luong Hai | | |
| **Date** | 31/01/2013 | **Priority** | High |
| **Actor:**  User  **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 3‑16: View Reservation Details Use Case Specification**

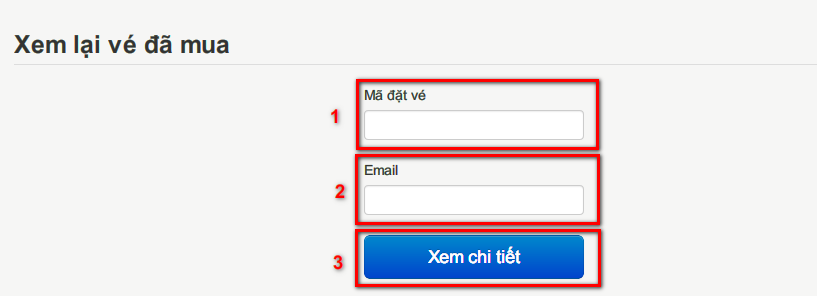


Figure 3‑18- Search Reservation screen

|  |  |  |
| --- | --- | --- |
| **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 3‑17: Search Reservation screen’s controls**

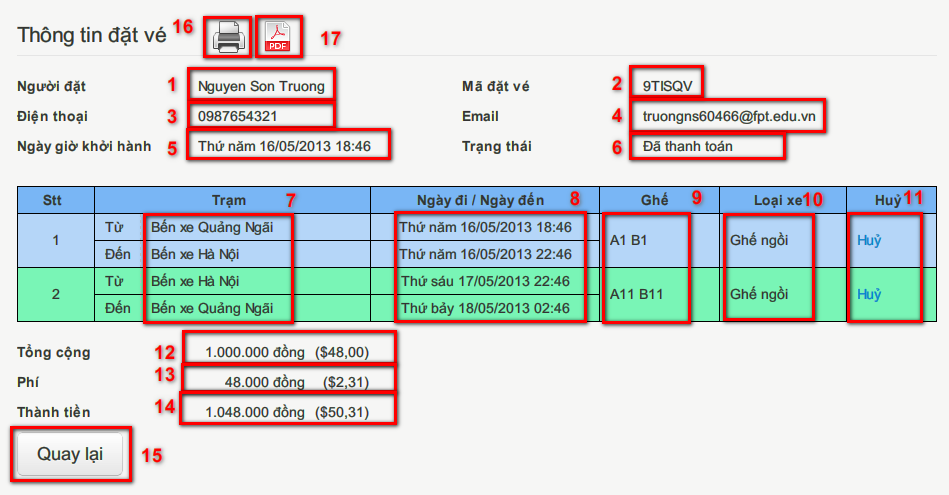


Figure 3‑19- Reservation Details screen

|  |  |  |
| --- | --- | --- |
| **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 3‑18: Reservation Details screen’s Control**

* + - 1. Cancel reservation use case



Figure 3‑20- Cancel Reservation use case diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC007 SPECIFICATION** | | | | |
| **Use-case No.** | UC007 | **Use-case Version** | | v0.1 |
| **Use-case Name** | Cancel Reservation | | | |
| **Author** | Nguyen Luong Hai | | | |
| **Date** | 31/01/2013 | **Priority** | Medium | |
| **Actor:**  User  **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. + times (5 days)  **Post-conditions:**  Success: The reservation is cancelled. The status of the reservation is updated to “Đã hoàn tiền”. 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 3‑19: Cancel Reservation Use Case Specification

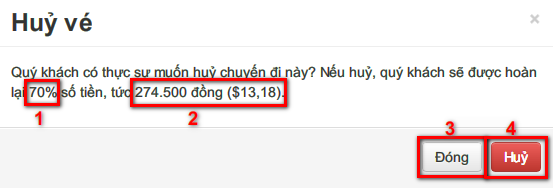


Figure 3‑21- Cancel Reservation Popup

|  |  |  |
| --- | --- | --- |
| **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 3‑20: Cancel Reservation popup description

### 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

* + Can create a reservation, view past reservation details, cancel reservation without logging in.

##### 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

#### Portability

#### Performance

## Entity Relationship Diagram

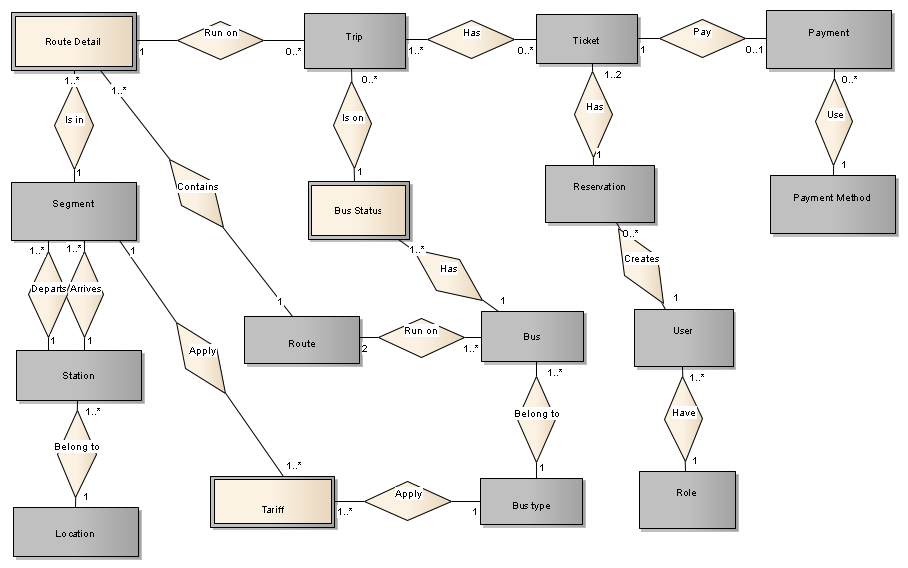


Figure ‑ Entity relationship diagram (with entity only)

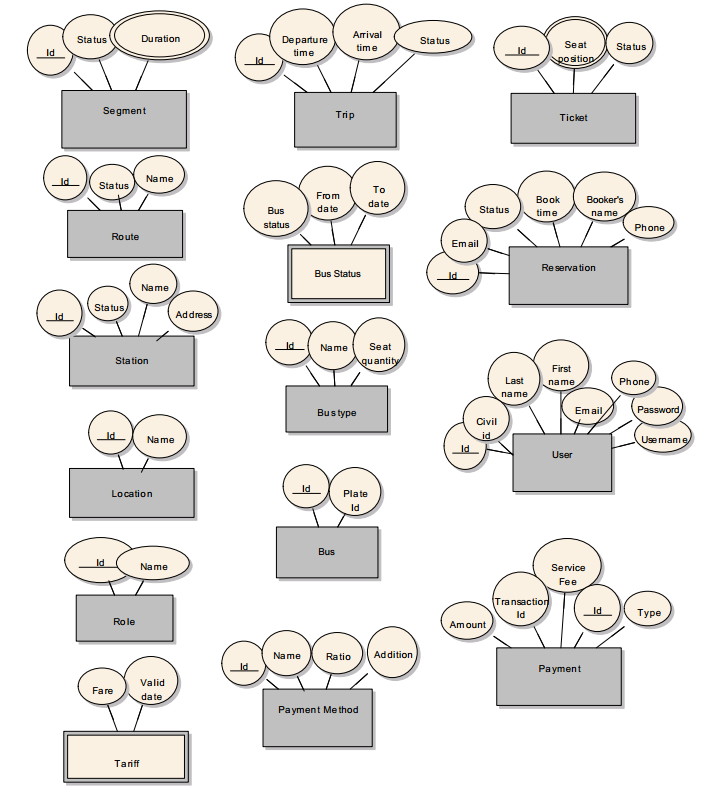


Figure ‑ Attributes of entities

|  |  |  |  |
| --- | --- | --- | --- |
| **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 depature 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 reservation |  | 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 ‑ Entity description

**3.4. Other material (if any)**

# Software Design Description

## Design Overview

## System Architectural Design

### Choice of System Architecture

Presentation Layer

Business Logic Layer

Data Access Layer

Database

Figure ‑ 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.

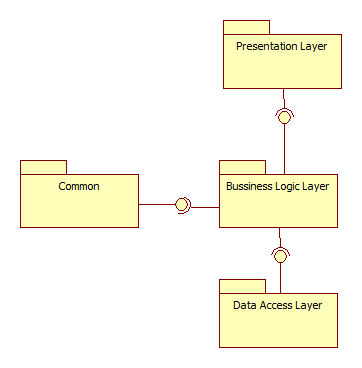


Figure ‑: Component Diagram

### 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 JSP combines with JavaSript, jQuery and CSS.

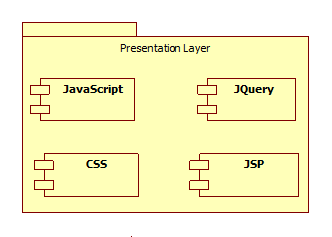


Figure ‑: Presentation Layer's Components

### 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 ‑: 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 ‑: 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 ‑ Common Components

## Detailed Description of Components

### Search trips

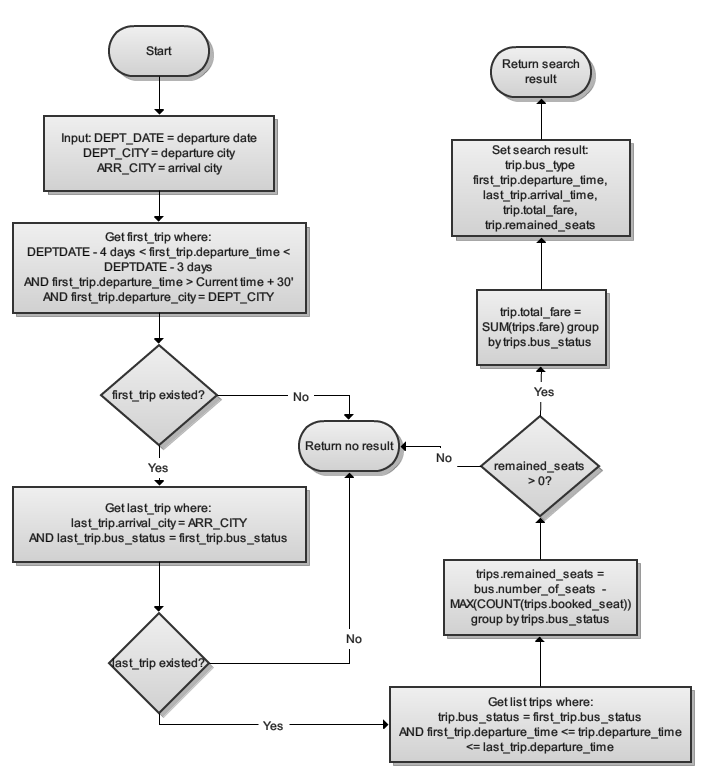
#### Class Diagram



Figure ‑ Search function class diagram

#### Class Diagram Explanation

#### Algorithms of search trip function



## Sequence Diagram

****

**4.6. User Interface Design**

4.6.1 Description of the User Interface

4.12.1.1 Screen Images

4.12.1.2 Objects and Actions

**4.7. Database Design or Data Structures**

*<Provide the detailed database design for the system here. If your team uses a file or in-memory storage facility instead of database, remove this section; use the ‘Data Structures’ section. >*

**4.8. Other material (if any)**

**Report No.5: Software Test Documentation (STD)**

**5.1. Introduction**

1.1 System Overview

1.2 Test Approach

**5.2. Test Plan**

2.1 Features to be tested

2.2 Features not to be tested

2.3 Testing Tools and Environment

**5.3.** **Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** | **Note** |
|  | List roles |  |  |  |  |  |  |
|  | Steps 1. Navigate to Manage Bus screen | | | | | | |
| BM01 | 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 |  |
| BM02 | 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 |  |
| BM03 | 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 |  |
| BM04 | 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!” |  | Passed | 17/4/2013 |  |
| BM05 | 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 |  |
| BM06 | 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 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** | **Note** |
|  | List roles |  |  |  |  |  |  |
|  | Steps 1. Navigate to Manage Segment screen | | | | | | |
| SM01 | 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 |  |
| SM02 | Edit segment duration: Duration is lesser 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 lesser than 1 hour.” |  | Passed | 17/4/2013 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** | **Note** |
|  | List roles |  |  |  |  |  |  |
|  | Steps 1. Navigate to Manage Schedule screen. | | | | | | |
| SCM01 | View details of schedule. | 1. Click button View Details. | 2. Screen “Schedule details” showed the schedule details grid:  Segment Name.  Departure Time.  Arrival Time. |  | Passed | 17/4/2013 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** | **Note** |
|  | List roles |  |  |  |  |  |  |
|  | Steps 1. Navigate to Manage Route screen | | | | | | |
| RM01 | 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 |  |
| RM02 | 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 |  |
| RM03 | 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 |  |
| RM04 | 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!” |  | Passed | 17/4/2013 |  |
| RM05 | 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 |  |
| RM06 | 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 |  |
| RM07 | 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 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** | **Result** | **Test Date** | **Note** |
|  | List roles |  |  |  |  |  |  |
|  | Steps 1. Navigate to Manage Route Details screen | | | | | | |
| RDM01 | 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 |  |
| RDM02 | 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 |  |
| RDM03 | 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 |  |
| RDM04 | 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 |  |
| RDM05 | 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 |  |
| RDM06 | 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 |  |
| RDM07 | 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 |  |
| RDM08 | 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 |  |
| RDM09 | 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 |  |
| RDM10 | 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. |  | Passed | 17/4/2013 |  |
| RDM11 | 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. |  | Passed | 17/4/2013 |  |
| RM12 | 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 |  |
| RM13 | 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. |  | Passed | 17/4/2013 |  |
| RM14 | 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 |  |
| RM15 | 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 |  |
| RM16 | Show the reverse route | 1. Click button “Show the reverse route”. | The reverse route details screen is shown. |  | Passed | 17/4/2013 |  |

**5.5. Checklists**

5.5.1. Checklist of Validation

*<Put the checklist here. Describe how it is used and the resulted checklist>*

5.5.2. Submission Checklist

*<Put the checklist here. Describe how it is used and the resulted checklist>*

**5.5. Other material (if any)** (including appendix A)

APPENDIX A. TEST LOGS

A.n Log for test *n*

A.n.1 Test Results

A.n.2 Incident Report

**Report No.6: Software User’s Manual**

**6.1. Installation Guide**

**6.2. User’s Guide**

**6.3. Other [Optional]**

**Appendix**