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

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Long-distance bus reservation |
|  |
| |  |  | | --- | --- | | **Group 4** | | | **Group Members** | Nguyen Son Truong – Team Leader – 60466  Nguyen Luong Hai – Member – 60335  Nguyen Thi Bich Tram – Member – 60…  Nguyen Ngoc Son – Member – 60409 | | **Supervisor** | Kieu Trong Khanh | | **Ext Supervisor** | - | | **Capstone Project code** | Bus Reservation | |
|  |

- Ho Chi Minh City, 01/2013 –

|  |
| --- |
|  |

# 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 into teams by its route, and each team has a fixed route and terminations.
  + Each team has a Team Manager (TM), who has all the status of their bus in their group. The TM will control the schedule of the buses and manage the seat in each bus. They report daily to the Operator Officers (OO) about the buses status and schedule.
  + The status of the buses help the TM controls each bus on their team and builds the schedule base on it. The status is defined by following :
    - Available: the bus is not planned for any departure date.
    - Waiting: the bus is in the station and waiting for departure date.
    - Busy: the bus is on their trip.
    - Maintenance: the bus is on maintaining.
  + Each bus will have a routine of departure date.
  + If the route is always in full or lacking seat, TM will make a request to the OO to have more bus to their team.
  + Otherwise, if the route usually remains excess seats, the TM will request for returning some buses back to OO and extended the time between departure dates.
  + And if excess seats problem occurs in a long time, the TM may ask for removing the routes and all buses in the team will return to OO.

### Ticket managing

* + The entire scheduled journey will be submitted to OO by the TM. Each journey will have a fitted fare defined by the policies of the company which are changed frequently by the OO.
  + Once schedule of the bus and seats division are planned, TM will send them to OO. The OO will base on ticket fare table/checklist to calculating the fare for each ticket, and then they will be sent to ticket offices to be sold.
  + 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 and some bonus, or they can request for the other ticket in the next bus.
  + Customers can purchase for a ticket directly at the ticket offices or they can book tickets through the phone. Telephonist 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 OO to manage all the bus status, route, fare and ticket balancing.
* It hard to control directly bus-to-bus when the bus status are not updated permanently to OO. All the bus status information only can be access through the TM.
* 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, 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 allow each customer to contact one another and change their seat on a journey.
* 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: date-time, number of remaining seats, price
  + Information is ordered by type of journey (full-journey is displayed first, then its constituents.)

### Booking

* Users select journey based on search results
* Users select seats
  + Display seat map for corresponding coach 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
* Cancel and refund

### Scheduling

* Create route and schedule
* Update route/schedule
* Automatically cancel a journey if number of passengers less than 25% of seat numbers
  + Inform and refund for customers if the journey is cancelled

### Vehicles Management

* Add new coach
* Update coach information and status
* Coach category management

### Fare and Promotions

* Fare is based on route, coach type and departure time (for example: price on holidays is higher than usual)
* Promotion for customer loyalty or special events

### User management

* Registration of members (customers) to the site
* View transaction history
* View user info and notifications
* Update user info and status

# Software Project Management Plan (SPMP)

## Project Overview

The name of this project is “Long-distance bus reservation” or “Bus Reservation”. It is a website for bus operators to manage their vehicles, build up schedule and journeys, and provide their customer 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:**

* Personal computer for developing with minimum configuration of 1GB RAM and 20GB of hard disk, Dual Core 2.0 GHz.
* Server computer for testing with minimum configuration of 2GB RAM and 20GB of hard disk, Dual Core 2.0 GHz.

**Software Requirements:**

* Operating system: Windows 7, Windows Server 2008
* IDE: Eclipse Indigo.
* DBMS: MySQL Server 5.5
* Source control: Tortoise SVN.

## Project organization

### ScrumCycle Software Process Model

The model used for developing this project is Scrum model.

Figure 2.2.1. Scrum Model

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

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Full name | Role | Responsibilities |
| 1 | Kieu Trong Khanh | Project Owner | * Specify user requirements * Control the development process * Support in technique and business analysis |
| 2 | Nguyen Son Truong | Scrum Master  Developer  Tester | * Manage process * Design database * Develop and analyze requirement * Prepare documents * Code * Test |
| 3 | Nguyen Luong Hai | Developer  Tester | * Design database * Develop and analyze requirement * Build framework * Support technique * Code * Test |
| 4 | Nguyen Thi Bich Tram | Developer  Tester | * Develop and analyze requirement * Prepare documents * Create test plan and system test cases * Code * Test |
| 5 | Nguyen Ngoc Son | Developer  Tester | * Develop and analyze requirement * Create prototype and design GUI * Code * Test |

### Tools and Techniques

* Front-end: HTML, CSS, jQuery, AJAX
* Server: Apache Tomcat
* Framework: Struts 2, Spring 3, Hibernate 3

### 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

### Naming Convention

• Variable, properties and method name: underscore followed by lowerCamelCase. (Example: int variableField)

• Class name: UpperCamelCase (or PascalCase). (Example: public ClassName)

• Do not use consecutive underscores in name.

• Do not use Hungarian style.

### Lengths

• Public name should not be longer than 32 characters or 7 words.

• Methods should contain no more than 70 lines of code (if it is, it must be divided into methods).

• Methods should contain no more than 5 levels of indentation (if it is, it must be divided into methods).

• A line of code should contain no more than 80 characters.

### Other material

# Software Requirements Specifications (SRS)

## User Requirement Specification

*<Summarize the customer requirements in a compact form>*

## System Requirement Specification (Specific Requirements)

*<Summarize the system requirements in a compact form>*

### External Interface Requirements

3.2.1.1. User Interfaces

3.2.1.2. Hardware Interfaces

3.2.1.3. Software Interfaces

3.2.1.4. Communications Protocol

### System Features



Figure 2‑1 System use case

|  |  |  |
| --- | --- | --- |
| No. | Actor | Description |
| 1 | User | All the people that use the Bus-Reservation System. |
| 2 | Guest | User who does not have an account or has not logged in the system. |
| 3 | Logged-in user | User who has logged in. |
| 4 | Customer | User who is the customer of the transportation company and uses the services provided by the company. |
| 5 | Staff | Employee of the transportation company. |

#### Program flow



Figure 2‑2 Program flow

#### Search for trips use case



Figure 2‑3 Search for trips use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC001 SPECIFICATION** | | | | |
| **Use-case No.** | UC001 | **Use-case Version** | | 0.1 |
| **Use-case Name** | Search for trips | | | |
| **Author** | Nguyen Son Truong | | | |
| **Date** | 2013/01/29 | **Priority** | High | |
| **Actor:**  User  **Summary:**  User can view list of available trips  **Goal:**  Helps user view all the available trips satisfying their needs.  **Triggers:**  User wants to search for trips.  (User provides search conditions and click button “Đặt vé”)  **Pre-conditions:**  The home page is fully loaded. Lists of all departure/arrival cities and bus types are loaded from database.  **Post-conditions:**  Success:   * The search result page is fully loaded. * User can view all of the available trips matching their search conditions.   Failure: System displays error message. Content of error message will be specified in Exception section.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User access the home page  3. User chooses type of trips, departure/arrival cities, number of passengers, departure date, arrival date and clicks on button “Đặt vé” [Alternative 1][Alternative 2]  5. User choose a trip and Click “Tiếp tục” button. | 1. Display home page include search criteria as below:    * Round trip [Khứ hồi: Radio button]    * One way [Một chiều: Radio button]    * Departure city [Điểm đi: Drop down list]    * Arrival city [Điểm đến: Drop down list]    * Number of passenger [Số lượng hành khách: Drop down list, min: 1, max 5]    * Depart date [Ngày đi: Date time picker, min: current date, max: 2 months after current date]    * Return date [Ngày về: Date time picker, min: current date, max: 2 months after current date]]    * Bus type [Loại xe bus: Drop down list]    * Book tickets [Đặt vé: Button]   4. Display search result page with details of trips matching search condition. [Exception 3]   * Departure city [Điểm khởi hành: string, label] * Arrival city [Điểm kết thúc: string, label] * Departure time [Giờ đi: DateTime, label] * Arrival time [Giờ đến: DateTime, label] * Fare [Giá vé: double, label] * Select [Chọn chuyến: radio button] * Continue [Tiếp tục: button]     6. Display seat selection page. [Exception 1][Exception 2] |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Alternative 1]  1. User chooses “Khứ hồi” radio button.  [Alternative 2]  1. User chooses “Một chiều” radio button | 2. Display “Ngày về” control.  2. Hide “Ngày về” control. |   **Exceptions:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Exception 1]  1. Departure and arrival cities are the same  [Exception 2]  1. Arrival date is earlier than departure date  [Exception 3]  1. There are no available trips satisfied the conditions | 2. Show info message: “Trạm khởi hành và trạm kết thúc không phù hợp, xin vui lòng chọn lại.”  2. Show info message: “Ngày đi phải nhỏ hơn ngày về, xin vui lòng chọn lại.”  2. Show info message: “Rất tiếc, hiện tại không có chuyến đi nào phù hợp với yêu cầu của quý khách.” |   **Relationships:**  <None>  **Business Rules:**  User can only reserve for a trip for maximum of 30 days before departure date.  Maximum number of passengers for each reservation is 5. | | | | |

#### Choose seat use case



Figure 3‑4 Choose Seat use case diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC002 SPECIFICATION** | | | |
| **Use-case No.** | UC002 | **Use-case Version** | 0.1 |
| **Use-case Name** | Choose seat | | |
| **Author** | Nguyen Ngoc Son | | |
| **Date** | 31/1/2013 | **Priority** | Normal |
| **Actor:**  User  **Summary:**  User choose seat after search trip  **Goal:**  It helps user choose their desired seats for the whole trip.  **Triggers**  User selected a trip in select trip page and click button “Tiếp tục”.  **Preconditions:**  The page is fully loaded.  **Post Conditions:**  Success:   * Redirect to provide booker’s info page after user click button “Tiếp Tục”   Failure: System displays error message. Content of error message will be specified in Exception section.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. Choose trip and click button “Tiếp tục”   3. Click seats. [Alternative 1]  5. Click “Tiếp Tục” button to submit view. | 2. Display a seats map.   * Green seats : selected seats * Orange seats : not available seats * White seats : available seats * Submit [“Tiếp Tục”: button] * Back [“Quay Lại”: button] * Legend : label   4. Change selected seat color [Alternative 2][Alternative 3][Exception 1][Exception 2]  6. Redirect to page provide booker’s info [Exception 3] |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Alternative 1]  1. User clicks on “Quay Lại” button.  [Alternative 2]  1. User clicks on white seat  [Alternative 3]  1. User clicks on green seat | 2. Back to search result page.  2. Change seat color to green.  2. Change seat color to white. |   **Exceptions:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Exception 1]  1. User clicks on orange seat.  [Exception 2]  1. User chooses more seats than number of passengers.  [Exception 3]  1. User chooses less seats than number of passengers. | 2. Show message: “Ghế này đã được đặt. Xin vui lòng chọn ghế khác.”  2. Show message: “Quý khách đã chọn đủ số lượng ghế mà bạn đã đặt. Để thay đổi, vui lòng bỏ chọn ghế hiện tại và chọn ghế khác”  2. Show message: “Xin quý khách vui lòng chọn đúng số ghế đã đặt”. |   **Relationships:**  Search for trips.  **Business Rules:**  The number of seats must match the number of passengers.  Reserved seats cannot be chosen. | | | |

Table 13: Customer and Unauthorized User Choose Seat Use Case Specification

<to be added>

Figure 2‑5- Customer and Unauthorized User Choose Seat

|  |  |  |
| --- | --- | --- |
| **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 |

Table 17: Customer and Unauthorized User Choose Seat’s Control

3.2.2.3. Provide booker’s details use case



Figure 3‑6 Provide booker’s details use case diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC003 SPECIFICATION** | | | |
| **Use-case No.** | UC003 | **Use-case Version** | v0.1 |
| **Use-case Name** | Provide booker’s details | | |
| **Author** | Nguyen Ngoc Son | | |
| **Date** | 31/1/2013 | **Priority** | Normal |
| **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:**  The page is fully loaded.  **Post Conditions:**  Success:   * Save user’s reservation. * 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: 4, max length: 30] * Last Name [“Họ”: textbox, min length: 4, 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] * Clear [“Xóa” : 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. User clicks on “Quay Lại” button. | 2. Back to choose seat page. |   **Relationships:**  Login, Create reservation  **Business Rules:**  The unauthorized user and customer must provide info to create reservation. | | | |

Table 13: Customer and Unauthorized User Provide Info Use Case Specification

<to be added>

Figure 2‑7- Customer and Unauthorized User Choose Seat

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | First Name - input – text (Họ) | Input for first name |
| **2** | Last Name - input – text (Tên) | Input for last name |
| **3** | Mobile Phone - input – text (Điện thoại di động) | Input for mobile phone number |
| **4** | Email – input – text | Input for email |
| **5** | Submit – button (Hoàn Tất) | Save reservation with status active and redirect to payment page |
| **6** | Back – button (Quay Lại) | Back to page choose seat |

Table 17: Customer and Unauthorized User Choose Seat’s Control

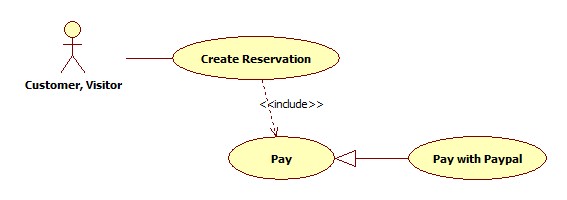
3.2.2.4. Pay with Paypal use case

Figure 2‑8- Pay with Paypal use case diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC004 SPECIFICATION** | | | | |
| **Use-case No.** | UC004 | **Use-case Version** | | v1.0 |
| **Use-case Name** | Pay with Paypal | | | |
| **Author** | Nguyen Luong Hai | | | |
| **Date** | 31/01/2013 | **Priority** | High | |
| **Actor:**  User(Customer, Unauthorized user)  **Summary:**  User can pay for reservation with paypal  **Goal:**  Informs user of the amount that must be paid and guides user through the payment process  **Triggers:**  User want to pay for a reservation  (User clicks “Thanh toán bằng paypal” button)  **Preconditions:**  The page is fully loaded.  If user is customer, login with “Customer” role.  **PostConditions:**  Success:   * Process payment successfully * Generate and display reservation code for user * Complete reservation process   Failure: System displays error message. Content of error message will be specified in Exception section.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User clicks on “Thanh toán bằng paypal”.        3. Complete payment process in paypal site[Alternative 1] [Exception 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. Show message “Quy trình đặt vé hoàn tất”. Display reservation code. |   **Alternative Scenario:**   | Ator Actions | System Response | | --- | --- | | [Alternative 1]  1. User cancel payment process and return to bus reservation site | 2. Show message “Thông tin đặt vé của quý khách đã được huỷ” |   **Exceptions:**   |  |  | | --- | --- | |  |  | | Actor Actions | System Response | | 1. User takes longer than 15 minutes to complete payment process on paypal site. | 2. Show message “Vì đã quá 15 phút, thông tin đặt vé của quý khách đã bị huỷ”. |   **Relationships:**  Login.  **Business Rules:**  Payment process must be completed within 15 minutes | | | | |

Table 6: Pay with Paypal Use Case Specification

<to be added>

Figure 2‑9- Review reservation details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | “Thanh toán bằng Paypal” button | Process payment with Paypal |

Table 7: Review reservation details screen’s Controls

<to be added>

Figure 2‑10- Payment details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Ticket price - Read-only field(Price) | Display the price of ticket |
| **2** | Quantity (Quantity) - Read-only field | Display the quantity of tickets |
| **3** | Fee (Fee) – Read-only field | Display fee for using online booking |
| **4** | Total amount (Total amount) - Read-only field | Display total amount that must be paid |

Table 8: Paypal site screen’s Controls

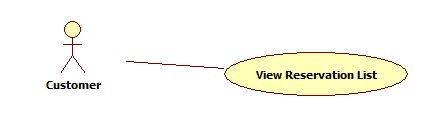
3.2.2.5. View Reservation List use case

Figure 2‑11- View Reservation List use case diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC005 SPECIFICATION** | | | | |
| **Use-case No.** | UC005 | **Use-case Version** | | v1.0 |
| **Use-case Name** | View Reservation List | | | |
| **Author** | Nguyen Luong Hai | | | |
| **Date** | 31/01/2013 | **Priority** | High | |
| **Actor:**  User(Customer)  **Summary:**  User views list of past reservations.  **Goal:**  Show all past reservations made by the user  **Triggers:**  User click “Thông tin đặt vé” link in page header  **Pre-conditions:**  The page is fully loaded.  User logins with “Customer” role  **Post-conditions:**  Success: The list of reservations are listed  Failure: System displays error message. Content of error message will be specified in Exception section.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User click “Thông tin đặt vé” link | 2. Display the list of reservations made by user[Alternative 1]   * Number [Stt: number, label] * Subroute [Chặng: string, label] * Departure date [Ngày giờ khởi hành: string, label] * Details [Chi tiết: string, link] * Cancel [Huỷ: string, link] [Alternative 2] [Alternative 3] |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Alternative 1]  1. User hasn’t made any reservation    [Alternative 2]  1. It’s past departure date  [Alternative 3]  1. Reservation has been cancelled | 2. Display “Quý khách chưa đặt vé lần nào”    2. Display Departed [Đã đi: string, label]    2. Display Cancelled [Đã huỷ: string, link] |   **Exceptions:**   |  |  | | --- | --- | | Actor Actions | System Response | | N/A | |   **Relationships:**  Login  **Business Rules:** | | | | |

Table 9: View Reservation List Use Case Specification

<to be added>

Figure 2‑12- Reservation List screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Number - Read-only field (Stt) | Display index number of reservation |
| **2** | Subroute - Read-only field (Chặng) | Display the starting point and destination of the trip |
| **3** | Departure Date - Read-only field (Ngày giờ khởi hành) | Display the departure date and time of the trip |
| **4** | Details - Link (Chi tiết) | Display reservation details |
| **5** | Cancel - Link (Hủy) | Cancel reservation |

Table 10: Reservation List screen’s Controls

* + - 1. View reservation details use case



Figure 2‑13- View reservation details use case diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC006 SPECIFICATION** | | | |
| **Use-case No.** | UC006 | **Use-case Version** | v1.0 |
| **Use-case Name** | View reservation details | | |
| **Author** | Nguyen Luong Hai | | |
| **Date** | 31/01/2013 | **Priority** | High |
| **Actor:**  Customer, Unauthorized 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)  **Preconditions:**  The page is fully loaded.  **Post Conditions:**  Success:   * Displays reservation details   Failure: System displays error message. Content of error message will be specified in Exception section.  **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]   * Subroute [Chặng: string, label] * Departure date [Ngày giờ khởi hàng: string, label] * Departure station address [Địa chỉ trạm đi: string, label] * Arrival date [Ngày giờ đến: string, label] * Arrival station address [Địa chỉ trạm đến: string, label] * Seat numbers [Số ghế: string, label] * Ticket price [Giá vé: string, label] * Online transaction fee [Phí đặt vé trực tuyến: string, label] * Total amount [Tổng cộng: string, label] * Status [Trạng thái: string, label]   Display button to cancel reservation [Alternative 1] [Alternative 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Actor Actions | System Response | | | [Alternative 1]  1. The departure time has past  [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 does not match any reservation    [Exception 2]  1. Reservation was made by a registered user, but user has not logged in    [Exception 3]  User has logged in, but reservation was made by a different registered user | 2. Display message “Không tìm thấy thông tin đặt vé. Xin vui lòng kiểm lại mã đặt vé.”    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.” |   **Relationships:**  Login, Create Reservation and View Reservation List.  **Business Rules:**  For reservations made by registered users, only the booker can access the reservation | | | |

Table 13: View Reservation Details Use Case Specification

<to be added>

Figure 2‑14- Search Reservation screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Reservation code - Textbox (Mã đặt vé) | Get code of the reservation to be viewed |
| **2** | Display - button (Hiển thị) | Display reservation details |

Table 14: Search Reservation screen’s controls

<to be added>

Figure 2‑15- Reservation Details screen

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Subroute – read-only text (Chặng) | Display the starting point and destination of the trip |
| **2** | Departure Date – read-only text (Ngày giờ khởi hành) | Display the departure date and time of the trip |
| **3** | Departure Station Address – read-only text (Địa chỉ trạm đi) | Display the address of the departure station |
| **4** | Arrival Date – read-only text (Ngày giờ đến) | Display the arrival date and time of the trip |
| **5** | Arrival Station Address – read-only text (Địa chỉ trạm đến) | Display the address of the arrival station |
| **6** | Seat numbers – read-only text (Số ghế) | Display the booked seat numbers |
| **7** | Ticket price – read-only text (Giá vé) | Display the price of each ticket |
| **8** | Online transaction fee – read-only text (Phí đặt vé trực tuyến) | Display the fee for online booking |
| **9** | Total amount – read-only text (Tổng cộng) | Display the total amount of the reservation |
| **10** | Status – read-only text (Trạng thái) | Display the status of the reservation |
| **11** | Cancel – button (Huỷ) | Cancel the reservation |
| **12** | Refund amount – read-only text (Số tiền hoàn lại) | Display the amount refunded when the reservation was cancelled |

Table 15: Reservation Details screen’s Control

3.2.2.7 Cancel reservation use case



Figure 2‑16- Cancel Reservation use case diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-UC007 SPECIFICATION** | | | | |
| **Use-case No.** | UC007 | **Use-case Version** | | v1.0 |
| **Use-case Name** | Cancel Reservation | | | |
| **Author** | Nguyen Luong Hai | | | |
| **Date** | 31/01/2013 | **Priority** | Medium | |
| **Actor:**  Customer, Unauthorized user  **Summary:**  User cancel 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).  **Preconditions:**  The page is fully loaded.  **PostConditions:**  Success: The reservation is cancelled.  Failure: System displays error message. Content of error message will be specified in Exception section.  **Main Success Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User click button “Huỷ” [Exception 1] [Exception 2]      3. Click “Xác nhận” button [Alternative 1] | 2. Display confirm popup with refund details   * Refund amount [Số tiền hoàn lại: string, label]   4. Cancel the reservation, close popup  - Refresh current screen |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Actions | System Response | | [Alternative 1]  1. Click “Quay lại” 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]  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:**  The amount to be refunded is based on the time until the departure date | | | | |

Table 16: Cancel Reservation Use Case Specification

<to be added>

Figure 2‑17- Cancel Reservation Popup

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| **1** | Refund amount – read-only text (Số tiền hoàn lại) | Display the amount to be refunded |
| **2** | Confirm – button (Xác nhận) | Cancel the reservation |
| **3** | Back – button (Quay lại) | Cancel the cancel process |

Table 17: Cancel Reservation popup’s Control

3.2.3. Software System Attributes

3.2.3.1 Reliability

3.2.3.2 Availability

3.2.3.3 Security

3.2.3.4 Maintainability

3.2.3.5 Portability

3.2.3.6 Performance

**3.3. Entity Relationship Diagram or Data Structures**

*<Provide the ERD Diagram for the system here. If your team uses a file or in-memory storage facility instead of a database, replace this section by ‘Data Structures’. Note, use only ERD or Data Structures>*

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

**Report No.4: Software Design Description (SDD)**

**4.1. Design Overview**

**4.2. System Architectural Design**

4.2.1 Choice of System Architecture

4.2.2 Discussion of Alternative Designs

4.2.3 Description of System Interface

**4.3. Component Diagram**

**4.4. Detailed Description of Components**

**4.4.1. CRC Cards (Class-Responsibility-Collaborators)**

4.4.1.*n.* Component-*n*

4.4.1.n.1. Class Diagram

4.4.2.n.2. Class Diagram Explanation

*<Provide a brief explanation of the class diagram above. You do not need to explain “obvious” parts of your class diagram>*

4.4.3.n.3. Algorithms of important methods in each class, specified in pseudo code or by Flow-Chart

**4.5. 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**

3.*n* Case-*n*

3.*n*.1 Purpose

3.*n*.2 Inputs

3.*n*.3 Expected Outputs & Pass/Fail criteria

3.*n*.4 Test Procedure

**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**