

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG WEBSITE BÁN VÉ TÀU ONLINE**

**Giảng viên hướng dẫn:** ThS.ĐẶNG THỊ THU HÀ

**Sinh viên thực hiện:** Phan Tuấn Tài - 16026151

Lâm Hồng Huy - 16011221

**TP Hồ CHí MInh, Năm 2020**

**INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY**

**FACULTY OF INFORMATION TECHNOLOGY**

**PHAN TUAN TAI**

**LAM HONG HUY**

**BUILD WEBSITE TO SELL TRAIN TICKETS ONLINE**

Major: Software Engineering

**Supervisor: Mrs. Dang Thi Thu Ha**

HO CHI MINH CITY, 2020

**Title**: Build website to sell train tickets online.

**Abtract**

**Reason for writing:** Today, with the development of information technology. The trading of cashew is done via internet. But this is still limited in the purchase of train tickets. So this is the main reason why my group chose this topic.

**Problem:** With purchase train tickets traditional, people have to wait a long time to be able to purchase the ticket and this is very waste time. So my group will resolve this problem by allow people booking ticket via online and payment online. This is very convenient for people.

**Methods:** The main object of website is to help people can purchase train ticket in the most convenient and anytime, anywhere. To do this thing then my group will use build an application to sell train ticket to people can purchase ticket.

The project started with the development of a database using SQL Server as a database management system, and the development of the application to manage the database that was coded in Visual Code. Front-end will use EJS as the main view engine and VueJS is library support for render and re-render when anything changes on the UI. Back-End will use NodeJS to get data from database and return result for user because it run very fast. SocketIO for real-time when select seat to make sure this seat will not be selected by another user in 10 minutes. Authentication will use Google Firebase and payment system will use Ngan Luong Gateways.

**Result:** Provided service sell train ticket convenient for user. Reduce time to purchase ticket. Can purchase ticket anytime, anywhere. Simple procedure when print ticket only enter booking code at ticket printer or ask staff support for print ticket.

**Conclusion:** We have learned basics of "Software Development Life Cycle" and new skill such as: VueJS, NodeJS, EJS, Google Firebase, integrate payment system from another system and deploy website, database.

**GIÁO VIÊN HƯỚNG DẪN ĐÁNH GIÁ**

**GIÁO VIÊN PHẢN BIỆN ĐÁNH GIÁ**

# MỤC LỤC

[MỤC LỤC 5](#_Toc13916)

[DANH MỤC CÁC HÌNH VẼ 7](#_Toc24199)

[DANH MỤC CÁC BẢNG BIỂU 9](#_Toc13503)

[LỜI MỞ ĐẦU 10](#_Toc22431)

[CHƯƠNG 1 : GIỚI THIỆU 11](#_Toc2397)

[1.1 Tổng quan 11](#_Toc17465)

[1.2 Mục tiêu cơ bản đề tài 11](#_Toc1491)

[1.3 Phạm vi đề tài 11](#_Toc808)

[1.4 Mô tả yêu cầu chức năng 12](#_Toc15018)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 13](#_Toc25552)

[CHƯƠNG 3 : PHÂN TÍCH 17](#_Toc7885)

[3.1 Mô hình Use case tổng quát 17](#_Toc6711)

[3.2 Danh sách các tác nhân và mô tả 17](#_Toc29641)

[3.3 Danh sách Use case và mô tả 17](#_Toc20689)

[3.4 Đặc tả các yêu cầu chức năng 20](#_Toc21751)

[3.4.1 UC001\_Tìm kiếm lịch trình 20](#_Toc11643)

[3.4.2 UC002\_Mua vé 22](#_Toc4347)

[3.4.3 UC003\_Tra thông tin vé đã đặt 26](#_Toc2947)

[3.4.4 UC004\_Tra tuyến theo lịch trình có sẵn 28](#_Toc10307)

[3.4.5 UC005\_In vé 30](#_Toc31404)

[3.4.6 UC006\_Huỷ vé 33](#_Toc24676)

[3.4.7 UC007\_Lên lịch trình 36](#_Toc21390)

[3.4.8 UC008\_Thống kê vé bán được và doanh thu 39](#_Toc7055)

[3.4.9 UC009\_Quản lý hành khách theo lịch trình 41](#_Toc29636)

[3.4.10 UC010\_Đăng nhập 43](#_Toc23661)

[3.4.11 UC011\_Tra vé theo mã đặt vé 45](#_Toc2247)

[3.4.12 UC012\_Lấy danh sách vé chờ huỷ 49](#_Toc25127)

[CHƯƠNG 4 : THIẾT KẾ VÀ HIỆN THỰC 52](#_Toc18027)

[4.1 ClassDiagram 52](#_Toc25959)

[4.2 Sitemap 52](#_Toc3659)

[4.3 Database Diagram 54](#_Toc1675)

[4.4 Mô tả Website 55](#_Toc30643)

[4.4.1 Một số chức năng và giao diện chính 55](#_Toc15985)

[MỤC : KẾT LUẬN 60](#_Toc25547)

[5.1 Kết quả đạt được 60](#_Toc28275)

[5.2 Hạn chế của đồ án 60](#_Toc26629)

[5.3 Hướng phát triển 60](#_Toc20891)

[TÀI LIỆU THAM KHẢO 61](#_Toc26918)

[PHỤ LỤC 62](#_Toc24593)

# DANH MỤC CÁC HÌNH VẼ

[Hình 3‑3 Activity Diagram UC001 – Tìm kiếm lịch trình 22](#_Toc39960531)

[Hình 3‑4 Sequence Diagram UC001 – Tìm kiếm lịch trình 23](#_Toc39960532)

[Hình 3‑5 Activity Diagram UC002 – Mua vé 25](#_Toc39960531)

[Hình 3‑6 Sequence Diagram UC002 – Mua vé 26](#_Toc39960532)

[Hình 3‑7 Activity Diagram UC003 – Tra thông tin vé đã đặt 28](#_Toc39960531)

[Hình 3‑8 Sequence Diagram UC003 – Tra thông tin vé đã đặt 29](#_Toc39960532)

[Hình 3‑9 Activity Diagram UC004 – Tra tuyến theo lịch trình có sẵn 21](#_Toc39960531)

[Hình 3‑10 Sequence Diagram UC004 – Tra tuyến theo lịch trình có sẵn 22](#_Toc39960532)

[Hình 3‑11 Activity Diagram UC005 – In vé 32](#_Toc39960531)

[Hình 3‑12 Sequence Diagram UC005 – In vé 33](#_Toc39960532)

[Hình 3‑13 Activity Diagram UC006 – Hủy vé 21](#_Toc39960531)

[Hình 3‑14 Sequence Diagram UC006 – Hủy vé 36](#_Toc39960532)

[Hình 3‑15 Activity Diagram UC007 – Lên lịch trình 38](#_Toc39960531)

[Hình 3‑16 Sequence Diagram UC007 – Lên lịch trình 39](#_Toc39960532)

[Hình 3‑17 Activity Diagram UC008 – Thống kê vé bán được và doanh thu 40](#_Toc39960531)

[Hình 3‑18 Sequence Diagram UC008 – Thống kê vé bán được và doanh thu 41](#_Toc39960532)

[Hình 3‑19 Activity Diagram UC009 – Quản lý khách hàng theo lịch trình 42](#_Toc39960531)

[Hình 3‑20 Sequence Diagram UC009 – Quản lý khách hàng theo lịch trình 42](#_Toc39960532)

[Hình 3‑21 Activity Diagram UC010 – Đăng nhập 44](#_Toc39960531)

[Hình 3‑22 Sequence Diagram UC010 – Đăng nhập 45](#_Toc39960532)

[Hình 3‑23 Activity Diagram UC011 – Tìm kiếm vé theo mã đặt vé 47](#_Toc39960531)

[Hình 3‑24 Sequence Diagram UC011 – Tìm kiếm vé theo mã đặt vé 48](#_Toc39960532)

[Hình 3‑25 Activity Diagram UC012 – Lấy danh sách vé chờ hủy 50](#_Toc39960531)

[Hình 3‑26 Sequence Diagram UC012 – Lấy danh sách vé chờ hủy](#_Toc39960532) 51

# DANH MỤC CÁC BẢNG BIỂU

[Bảng Danh sách các usecase và mô tả tóm tắt Usecase](#_Toc39960533) 19

[Bảng 3‑3 Mô tả UC001 – Tìm kiếm lịch trình](#_Toc39960534) 20

[Bảng 3‑4 Mô tả UC002 – Mua vé](#_Toc39960534) 23

[Bảng 3-5 Mô tả UC003 – Tra thông tin vé đã đặt](#_Toc39960534) 26

[Bảng 3‑6 Mô tả UC004 – Tra tuyến theo lịch trình có sẵn](#_Toc39960534) 29

[Bảng 3‑7 Mô tả UC005 – In vé](#_Toc39960534) 31

[Bảng 3‑8 Mô tả UC006 – Hủy vé](#_Toc39960534) 34

[Bảng 3-9 Mô tả UC007 – Lên lịch trình](#_Toc39960534) 37

[Bảng 3-10 Mô tả UC008 – Thống kê vé bán được và doanh thu](#_Toc39960534) 40

[Bảng 3-11 Mô tả UC009 – Quản lý khách hàng theo lịch trình](#_Toc39960534) 42

[Bảng 3‑12 Mô tả UC010 – Đăng nhập](#_Toc39960534) 43

[Bảng 3‑13 Mô tả UC011 – Tìm kiếm vé theo mã đặt vé](#_Toc39960534) 46

[Bảng 3‑14 Mô tả UC012 – Lấy danh sách vé chờ hủy](#_Toc39960534) 49

# LỜI MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài

* Tính cấp thiết của đề tài nghiên cứu là giúp cho việc số hoá dịch vụ bán vé tàu trong thời đại 4.0 hiện nay. Tạo nên sự thuận tiện cho người mua cũng như người bán.
* Mục đích, đối tượng và phạm vi nghiên cứu được nhắm vào hành khách đi tàu hoả, bên bán vé tàu.
* Phương pháp nghiên cứu dựa vào các bài báo nói về thực trạng khi mua vé tại ga vào các dịp lễ, tết để đưa ra giải pháp phù hợp với thực trạng đó.

1. Ý nghĩa khoa học và thực tiễn của đề tài

* Về ý nghĩa khoa học của đề tài là giải quyết những vấn đề nan giải trong khâu mua vé tàu vào những dịp cao điểm trong năm như là đợi mua vé quá lâu, hết vé trong quá trình đợi mua vé, bị động trong khâu mua vé. Trong thời đại 4.0 thì việc số hoá toàn bộ dịch vụ là điều hết sức cần thiết nên đề tài cũng giúp giải quyết vấn đề này trong dịch vụ mua bán vé tàu thay vì bán vé theo cách thủ công bây giờ là bán vé thông qua internet và thanh toán thông qua internet.
* Về ý nghĩa thực tiễn của đề tài sẽ giúp cho người mua vé cũng như bên bán vé thoải mái hơn lúc chưa có website. Về phía người mua vé họ sẽ được chủ động hơn trước đây thay vì xếp hàng chờ mua vé thì họ chỉ cần laptop, desktop hoặc điện thoại có kết nối internet, thẻ ATM và email để thanh toán và nhận mã đặt vé sau khi đặt vé thành công và họ sẽ không lo hết vé như lúc trước nữa. Còn đối với bên bán vé thì họ chỉ cần nhập mã đặt vé và email của người đặt vé là họ có thể in vé một cách nhanh chóng thay vì lúc trước nhập N trường dữ liệu giờ họ chị cần nhập 2 trường dữ liệu và nhấn nút là đã có thể in vé.

# : GIỚI THIỆU

## Tổng quan

* Trước giờ việc đi lại bằng tàu hỏa ở các quốc gia là vô cùng phổ biến và lúc đó việc đặt vé điều thông qua ga tàu. Và càng ngày việc đặt vé thông qua cách thức đó bị quá tải và thời gian chờ ngày càng tăng. Nên một số quốc gia trên thế giới đã số hóa việc này cho phép đặt vé thông qua internet. Và lúc này ngành CNTT lại nổi lên vì việc số hóa đó.
* Website bán vé tàu online nhầm đáp ứng một mục đích duy nhất là tạo sự tự do và nhanh chóng cho người hành khách muốn mua vé. Họ chỉ cần tối thiểu một số thứ như là email, thẻ ATM và một số giấy tờ để việc đặt vé diễn ra nhanh chóng và thuận tiện nhất.
* Website bán vé tàu online cũng giúp cho bên bán vé đỡ phần nào về thủ tục nhập xuất rất nhiều, thay vì lúc trước họ phải điền form có rất nhiều trường. Giờ họ chỉ cần nhập 2 thứ là mã đặt vé và email người đặt vé và kiểm tra giấy tờ trước khi họ in vé và việc này diễn ra nhanh hơn việc nhập form trước đó.

## Mục tiêu cơ bản đề tài

* Giúp cho việc đặt vé tàu trở nên đơn giản và ít tốn thời gian hơn trước
* Giúp cho bên bán vé tàu có thể đơn giản hóa thủ tục bán vé hơn trước đây và việc quản lý doanh thu trong tháng hoặc năm được nhanh chóng hơn so với trước đây.

## Phạm vi đề tài

* Website bán vé tàu gồm các chức năng:
* Tìm kiếm lịch trình
* Mua vé (đặt vé)
* Tra thông tin đặt vé
* Tra giá vé
* In vé
* Hủy vé
* Lên lịch trình
* Thống kê doanh thu
* Thống kê vé bán được trong tháng hoặc năm
* Kỹ thuật thực hiện:
* Front-end:
* EJS (View engine)
* VueJS (Viết script nhúng vào EJS)
* Back-end:
* NodeJS (Express)
* Google Firebase (Authentication)
* Truy xuất cơ sở dữ liệu:
* Sequelize (Tạo model giống entity framework)
* Tedious (Tạo kết nối tới database)
* Cơ sở dữ liệu:
* Sql Server
* Google Firebase (Database lưu trữ thông tin nhân viên)

## Mô tả yêu cầu chức năng

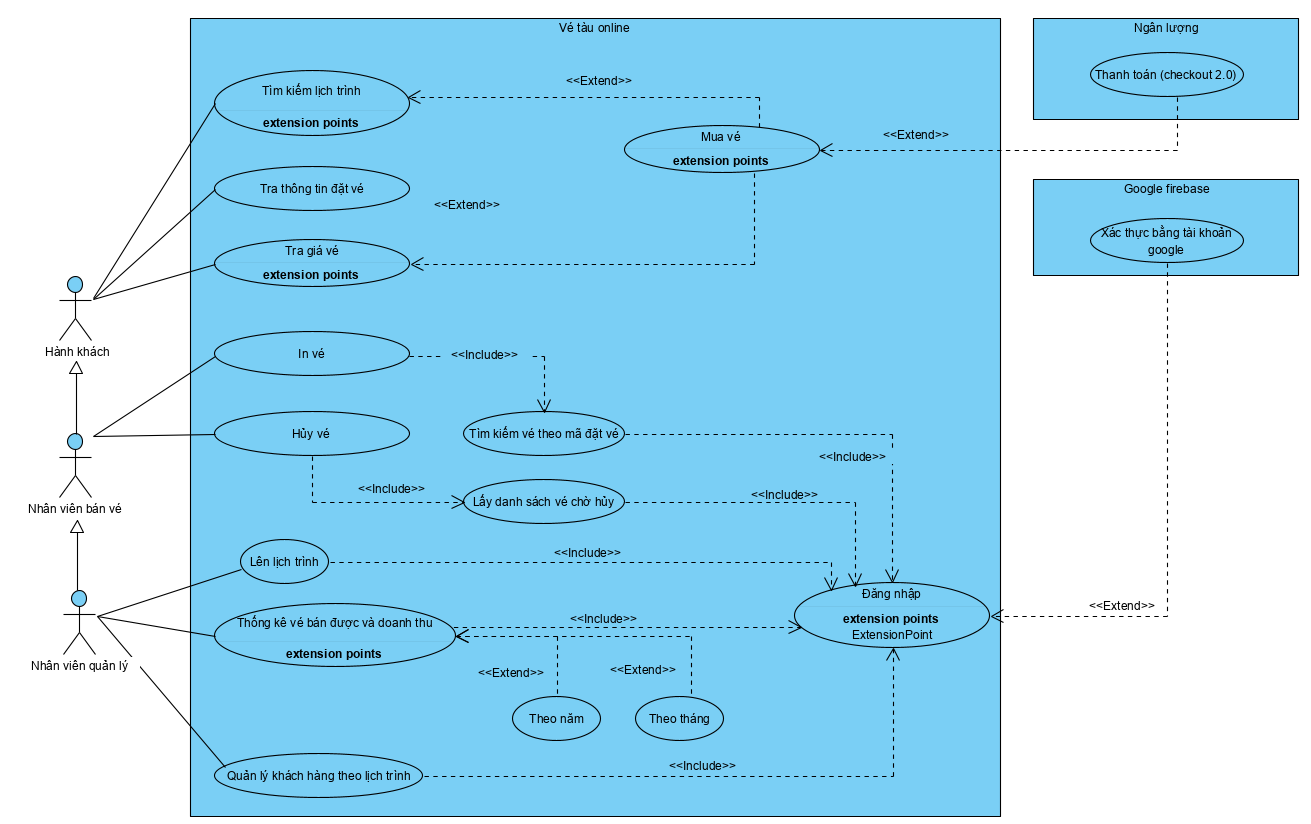
* Các tác nhân: Hành khách, Nhân viên bán vé, Nhân viên quản lý.
* Các chức năng ứng với mỗi tác nhân cụ thể
* Tìm kiếm lịch trình (Hành khách)
* Mua vé (đặt vé) (Hành khách)
* Tra thông tin đặt vé (Hành khách)
* Tra tuyến theo lịch trình có sẵn (Hành khách)
* In vé (Nhân viên bán vé)
* Huỷ vé (Nhân viên bán vé)
* Lên lịch trình (Nhân viên quản lý)
* Thống kê doanh thu và vé bán được (Nhân viên quản lý)
* Đăng nhập (Nhân viên bán vé, nhân viên quản lý)
* Lấy danh sách vé chờ huỷ (Hệ thống)
* Quản lý hành khách theo lịch trình (Nhân viên quản lý)
* Tra vé theo mã đặt vé (Nhân viên bán vé)

# : CƠ SỞ LÝ THUYẾT

* ***Front-End***
* EJS (Embedded JavaScript)
* Là một ngôn ngữ cho phép người lập trình có thể nhúng mã JavaScript vào HTML. Và nó được thực hiện bằng cách đặt mã JavaScript bên trong cặp dấu <%%> nếu muốn khai báo biến, <%=%> nếu muốn hiển thị biến lên trên HTML và <%-%> nếu muốn nhúng một file EJS khác vào.
* Những tính năng của EJS là:
  + - * + Nó sử dụng ngôn ngữ JavaScript thuần và điều này sẽ làm cho người lập trình biết về JavaScript dễ tiếp cận hơn.
        + Thời gian phát triển nhanh vì cú pháp nó là JavaScript nên nó sẽ không bắt người lập trình học quá nhiều về nó.
        + Cú pháp đơn giản.
        + Thời gian thực hiện nhanh chóng vì EJS sẽ lưu lại các hàm JS trung gian (cache) nên sẽ ít tốn thời gian để tải lại các hàm đó.
        + Dễ dàng Debugging khi lỗi chổ nào thì EJS sẽ báo lỗi ngay chỗ đó (ngay line bị lỗi)
* VueJS
* Là một framework linh động. Khác với các framework nguyên khối, Vue được thiết kế từ đầu theo hướng phát triển ứng dụng theo từng bước. Nghĩa là khi phát triển lớp giao diện (view layer), người dùng chỉ cần dùng thư viện lõi (core library) của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn. Nếu kết hợp với những kĩ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ, Vue cũng đáp ứng được dễ dàng nhu cầu xây dựng những ứng dụng một trang (SPA - Single-Page Applications) với độ phức tạp cao hơn nhiều.
* Các đặc tính của VueJS
  + - * + Dễ dàng tiếp cận chỉ cần người phát triển biết về HTML, CSS và JS là có thể bất đầu viết VueJS. Vì VueJS chủ yếu là code JavaScript.
        + Đa năng VueJS có một hệ sinh thái rất đa dạng với các thành phần từ căn bản đến nâng cao và từ một thư viện đơn giản tới một thư viện nâng cao.
        + Xử lý mạnh mẽ VueJS xử lý cực nhanh với virtual DOM và dễ dàng tối ưu.
* ***Back-End***
* NodeJS
* Node.js là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Bình thường thì bạn cũng có thể tải bộ V8 và nhúng nó vào bất cứ thứ gì; Node.js làm điều đó đối với các web server. JavaScript suy cho cùng cũng chỉ là một ngôn ngữ - vậy thì không có lý do gì để nói nó không thể sử dụng trên môi trường server tốt như là trong trình duyệt của người dùng được
* Trong một môi trường server điển hình LAMP (Linux-Apache-MySQL-PHP), bạn có một web server là Apache hoặc NGINX nằm dưới, cùng với PHP chạy trên nó. Mỗi một kết nối tới server sẽ sinh ra một thread mới, và điều này khiến ứng dụng nhanh chóng trở nên chậm chạp hoặc quá tải - cách duy nhất để hỗ trợ nhiều người dùng hơn là bằng cách bổ sung thêm nhiều máy chủ. Đơn giản là nó không có khả năng mở rộng tốt. Nhưng với Node.js thì điều này không phải là vấn đề. Không có một máy chủ Apache lắng nghe các kết nối tới và trả về mã trạng thái HTTP - bạn sẽ phải tự quản lý kiến trúc lõi của máy chủ đó. May mắn thay, có một số module giúp thực hiện điều này được dễ dàng hơn, nhưng công việc này vẫn gây cho bạn một chút khó khăn khi mới bắt đầu. Tuy nhiên, kết quả thu được là một ứng dụng web có tốc độ thực thi cao.
* Tốc độ của một ứng dụng NodeJS gần như đặt tới real-time vì vậy một website chịu tải lớn như là website vé tàu vào dịp lễ thì việc sử dụng NodeJS là vô cùng hợp lí và cần xử lý real-time trong việc chọn ghế.
* Các đặc tính của NodeJS:
  + - * + Không đồng bộ: Tất cả các API của Nodejs đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của Nodejs Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
        + Chạy rất nhanh: Nodejs được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
        + Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
        + Không đệm: Nodejs không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
* Google Firebase (Authentication)
* Authentication là một phần không thể thiếu trong việc làm ứng dụng có web server làm hệ thống back-end. Trong thời buổi hiện tại, ngoài việc tạo trang sign up để người dùng đăng kí tài khoản trực tiếp trên server của bạn, bạn còn nên hỗ trợ thêm việc sign up / sign in qua các nhà cung cấp authentication khác, chẳng hạn như Google, Facebook hoặc Twitter. Việc tự xây dựng hệ thống đăng nhập không phải là khó, nhưng vấn đề bảo trì và bảo mật sẽ gây không ít khó khăn cho các bạn, đặc biệt là các bạn single developer hoặc các tổ chức có quy mô nhỏ. Với sự lớn mạnh của mình, Firebase team at Google cung cấp cho chúng ta một giải pháp (solution) rất dễ dàng và an toàn, đó chính là Firebase Authentication.
* Không chỉ hỗ trợ User Authentication để liên kết với các dịch vụ khác của Firebase, Firebase Authentication vẫn cho phép bạn sử dụng cho hệ thống cơ sở dữ liệu riêng của bạn. Do đó, bạn không bắt buộc phải dùng tất cả các sản phẩm khác của Firebase như Analytics, Test Lab hay Notification thì vẫn có thể “dùng nhờ” Firebase Authentication được. Và tốt hơn hết, bạn nên thiết lập Authentication trước khi bắt đầu với các phần khác, chẳng hạn Realtime Database, để đảm bảo tính bảo mật cho cơ sở dữ liệu.
  + - Thanh toán online thì nhóm em chọn sẽ là cổng thanh toán ngân lượng vì họ cung cấp key API hoàn toàn nhanh chóng và không có ràng buộc về khoảng phải là doanh nghiệp mới được cấp key so với các cổng thanh toán khác bên họ chỉ cần số thẻ, CMND để xác thực tài khoản là sẽ cấp key API ngay và tài liệu hướng dẫn tích hợp của họ cũng rất dễ hiểu so với các bên khác.
* ***Database Access***
* Vì là nhóm em chọn SQL Server làm nơi lưu dữ liệu nên việc truy xuất vào database sẽ thông qua Tedious.
* Các đối tượng để mapping với các bảng dưới database sẽ thông qua Sequelize.
* ***Database***
* SQL Server
* SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) sử dụng câu lệnh SQL (Transact-SQL) để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS. SQL Server được phát triển và tiếp thị bởi Microsoft.
* SQL Server hoạt động độc quyền trên môi trường Windows trong hơn 20 năm. Năm 2016, Microsoft đã cung cấp phiên bản trên Linux. SQL Server 2017 ra mắt vào tháng 10 năm 2016 chạy trên cả Windows và Linux, SQL Server 2019 sẽ ra mắt trong năm 2019. SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server….

# : PHÂN TÍCH

## Mô hình Use case tổng quát



## Danh sách các tác nhân và mô tả

* Hành khách: Là tác nhân chính của hệ thống, được thực hiện các chức năng như là mua vé, tìm kiếm lịch trình, tra thông tin đặt vé và tra giá vé.
* Nhân viên bán vé: Là nhân viên ngồi tại quầy bán vé và có thể sử dụng các chức năng như là in vé và hủy vé.
* Nhân viên quản lý: Là nhân viên không trực tiếp ngồi tại quầy bán vé và có thể dùng các chức năng như là lên lịch trình, thống kê doanh thu, thông kê vé bán được theo năm, theo tháng.

## Danh sách Use case và mô tả

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Tên Use case | Mô tả ngắn gọn Use case | Chức năng | Ghi chú |
| UC001 | Tìm kiếm lịch trình | Tìm kiếm lịch trình theo các trường như là ga đi, ga đến, ngày đi, ngày về, số lượng hành khách, khứ hồi hay một chiều. | Tìm kiếm lịch trình | Hành khách không cần đăng nhập vào hệ thống để thực hiện chức năng. |
| UC002 | Mua vé | Đặt vé tàu dựa trên kết quả tìm kiếm được tù chức năng tìm kiếm lịch trình | Mua vé | Hành khách muốn thực hiện chức năng này thì phải thực hiện chức năng tìm kiếm lịch trình sau đó chọn lịch trình cần đi và nhấn nút đặt vé. Hành khách không cần đăng nhập vào hệ thống để thực hiện chức năng |
| UC003 | Tra thông tin vé đã đặt | Hành khách dùng chức năng này để kiểm tra thông tin của vé tàu sau khi thanh toán thành công. | Tra thông tin đặt vé | Sau khi thanh toán thành công thì hành khách sẽ nhận được mã đặt vé trong email và vào trang chủ chọn “Manage Booking” và điền thông tin vào form và nhân nút tìm kiếm để tra thông tin. Hành khách không cần đăng nhập vào hệ thống để thực hiện chức năng |
| UC004 | Tra tuyến theo lịch trình có sẵn | Hiển thị toàn bộ lịch trình có ngày khởi hành lớn hơn hoặc bằng ngày hiện tại. | Tra tuyến theo lịch trình có sẵn | Hành khách vào trang chủ chọn “List Schedule”. Hành khách không cần đăng nhập vào hệ thống để thực hiện chức năng |
| UC005 | In vé | Nhân viên nhập mã đặt vé để lấy toàn bộ vé đã đặt để tiến hành in vé. | In vé | Nhân viên cần đăng nhập vào hệ thống để thực hiện chức năng này. |
| UC006 | Hủy vé | Nhân viên bấm vào nút Cancel Ticket để chọn vé để hủy. |  |  |
| UC007 | Lên lịch trình | Nhân viên quản lý lên lịch trình cho tàu các tàu đang rảnh. | Lên lịch trình | Nhân viên quản lý cần đăng nhập vào hệ thống với quyền là quản lý để thực hiện chức năng này. |
| UC008 | Thống kê vé bán được và doanh thu | Thống kê vé bán được và doanh thu | Thống kê vé bán được và doanh thu | Nhân viên quản lý cần đăng nhập vào hệ thống với quyền là quản lý để thực hiện chức năng này. |
| UC009 | Quản lý hành khách theo lịch trình | Quản lý thông tin của các hành khách liệt kê theo lịch trình | Quản lý hành khách theo lịch trình | Nhân viên quản lý cần đăng nhập vào hệ thống với quyền là quản lý để thực hiện chức năng này. |
| UC010 | Đăng nhập | Nhân viên của bên bán vé dùng để đăng nhập và hệ thống. | Đăng nhập | Truy cập vào trang admin sẽ hiển thị form đăng nhập |
| UC011 | Tìm kiếm vé theo mã đặt vé | Nhân viên của bên bán vé dùng để tìm kiếm vé của hành khách | Tra vé theo mã đặt vé | Nhân viên cần đăng nhập vào hệ thống để thực hiện chức năng này. |
| UC012 | Lấy danh sách vé chờ huỷ | Hệ thống tự động thực hiện khi có yêu cầu lấy danh sách chờ huỷ | Lấy danh sách vé chờ huỷ | Cần được bấm vào nút “Cancel ticket” ở menu chính |

Bảng 3‑1 Danh sách các Usecase và mô tả tóm tắt Usecase

## Đặc tả các yêu cầu chức năng

### UC001\_Tìm kiếm lịch trình

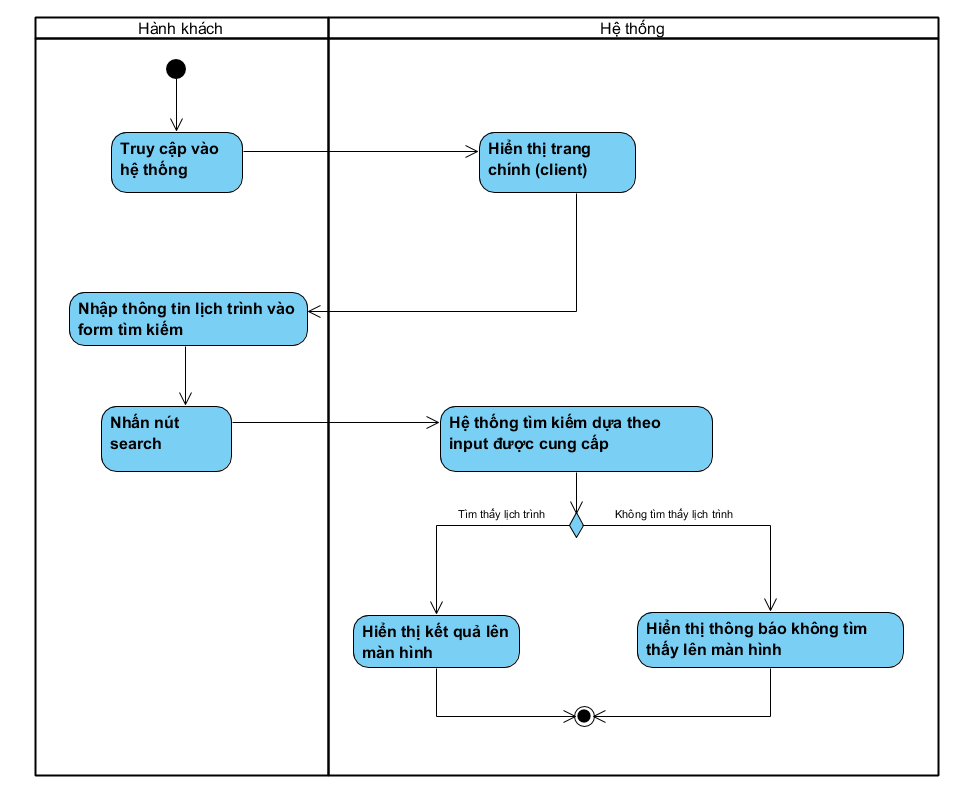
#### Mô tả use case UC001

|  |  |  |
| --- | --- | --- |
| **Use case:** Tìm kiếm lịch trình | | |
| Mục đích: | Tìm kiếm lịch trình. | |
| Mô tả: | Giúp hành khách có thể tìm kiếm được lịch trình để có thể đặt vé trên hệ thống. | |
| Tác nhân: | Hành khách. | |
| Điều kiện trước: | Truy cập vào hệ thống. | |
| Điều kiện sau: | Hiển thị ra danh sách các lịch trình phù hợp với những input được đưa vào. | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
| 1. Truy cập vào hệ thống. | |  |
|  | | 1. Hiển thị trang chính (client). |
| 1. Nhập thông tin lịch trình vào form và nhấn nút tìm kiếm. | |  |
|  | | 1. Hệ thống tìm kiếm dựa theo các input được cung cấp |
|  | | 1. Lấy kết quả tìm được và render lên giao diện. |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 5.1 Nếu không tìm thấy kết quả phù hợp với input được cung cấp sẽ trả về trang web với thông báo là không tìm thấy. | | |

Bảng 3‑3 Mô tả UC001 – Tìm kiếm lịch trình

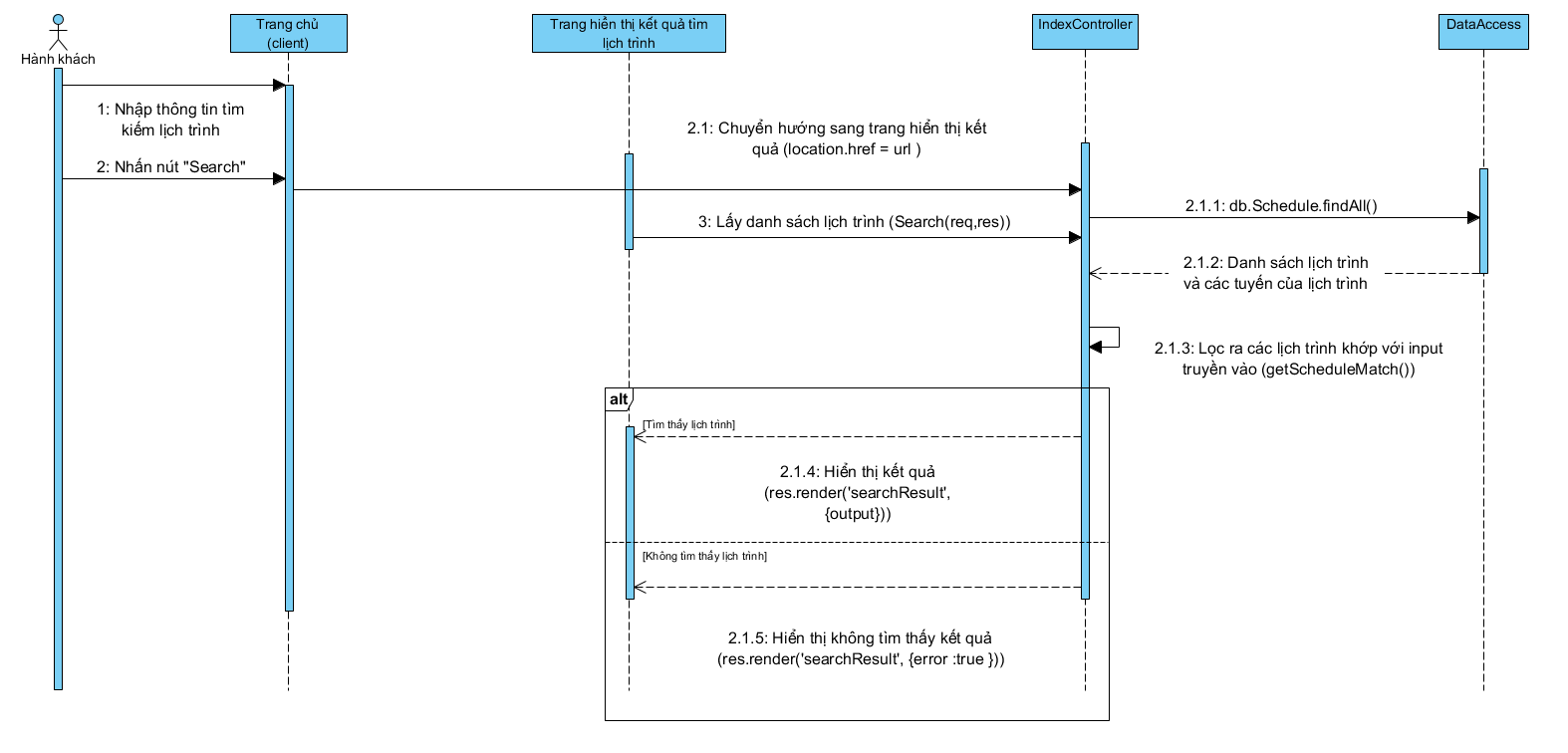
#### Biểu đồ

##### Activity Diagram



Hình 3‑3 Activity Diagram UC001 – Tìm kiếm lịch trình

##### Sequence Diagram



Hình 3‑4 Sequence Diagram UC001 – Tìm kiếm lịch trình

### UC002\_Mua vé

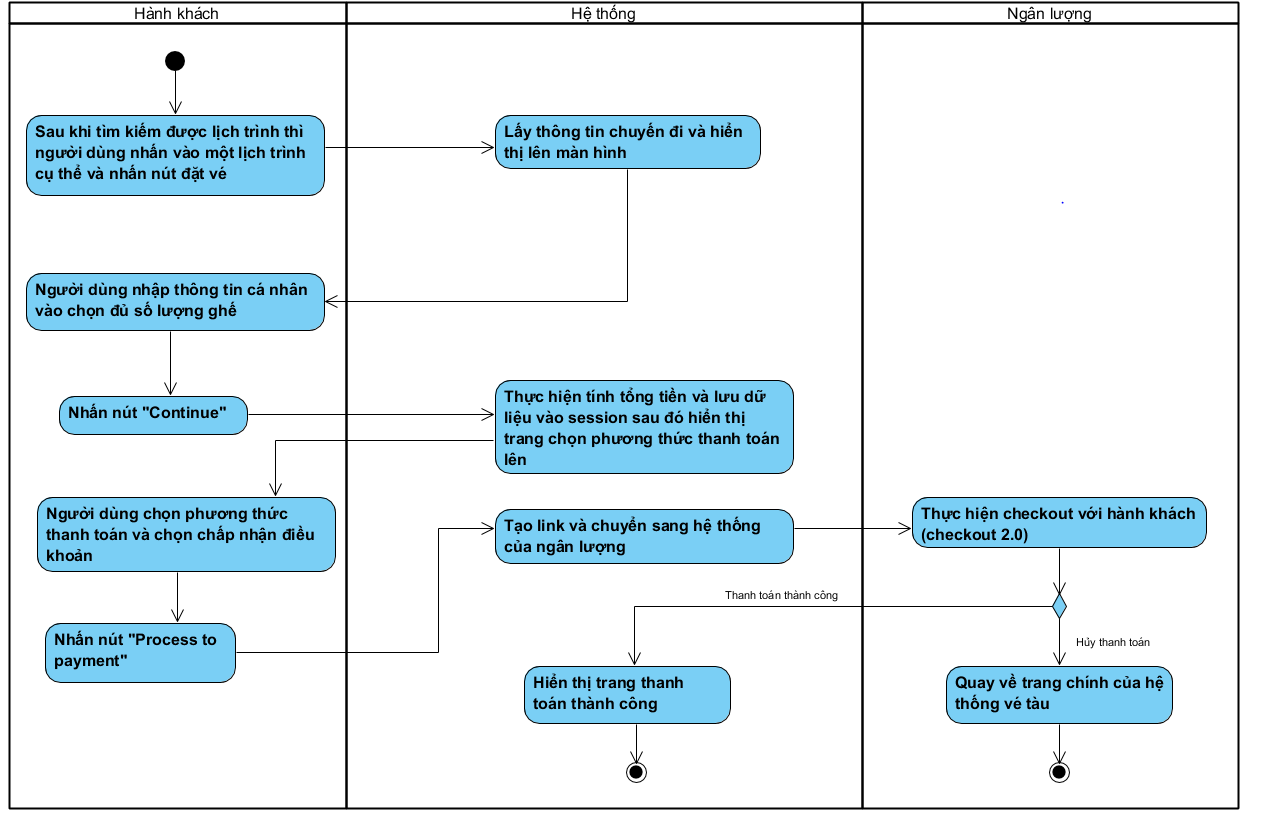
#### Mô tả use case UC002

|  |  |  |
| --- | --- | --- |
| **Use case:** Mua vé | | |
| Mục đích: | Đặt vé tàu từ hệ thống. | |
| Mô tả: | Giúp hành khách có thể mua được vé tàu thông qua internet. | |
| Tác nhân: | Hành khách. | |
| Điều kiện trước: | Truy cập vào hệ thống và tìm kiếm lịch trình hoặc chọn chức năng tra giá vé. | |
| Điều kiện sau: | Thêm thành công dữ liệu vào database, hiển thị thông báo mua vé thành công và có thể sử dụng được chức năng tra thông tin đặt vé. | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
| 1. Hành khách chọn lịch trình cần đi và chọn loại ghế và nhấn nút đặt vé. | |  |
|  | | 1. Hệ thống lấy thông tin chuyến đi và hiển thị lên cho người dùng thực hiện bước tiếp theo. |
| 1. Hành khách nhập thông tin cá nhân và chọn ghế và nhấn nút thanh toán | |  |
|  | | 1. Hệ thống thực hiện tính toán tổng tiền và tạo các đối tượng cần thiết và lưu vào session của hệ thống. Sau đó hiển thị trang phương thức thanh toán lên để người dùng chọn phương thức thanh toán cũng như là xem tổng tiền. |
| 1. Hành khách chọn phương thức thanh toán và chấp nhận các điều khoản và nhân nút thanh toán (hiện giờ chỉ có phương thức thanh toán thông qua ngân lượng) | |  |
|  | | 1. Hệ thống tạo link và chuyển hành khách sang cổng thanh toán Ngân Lượng. |
| 1. Người dùng thực hiện quá trình thanh toán với ngân lượng. | |  |
|  | | 1. Ngân lượng chuyển hành khách về hệ thống vé tàu online, lấy dữ liệu từ session ra và thêm vào database và hiển thị thông tin đặt vé về cho hành khách. Và gửi mail thông tin đặt vé về cho khách hàng. |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 8.1 Nếu có lỗi trong quá trình thêm dữ liệu thì hệ thống vẫn gửi mail về cho khách hàng và hành khách phải gửi mail về để cho bên bán vé hổ trợ hoàn tiền hoặc tiếp tục mua vé. | | |

Bảng 3-4 Mô tả UC002-Mua vé

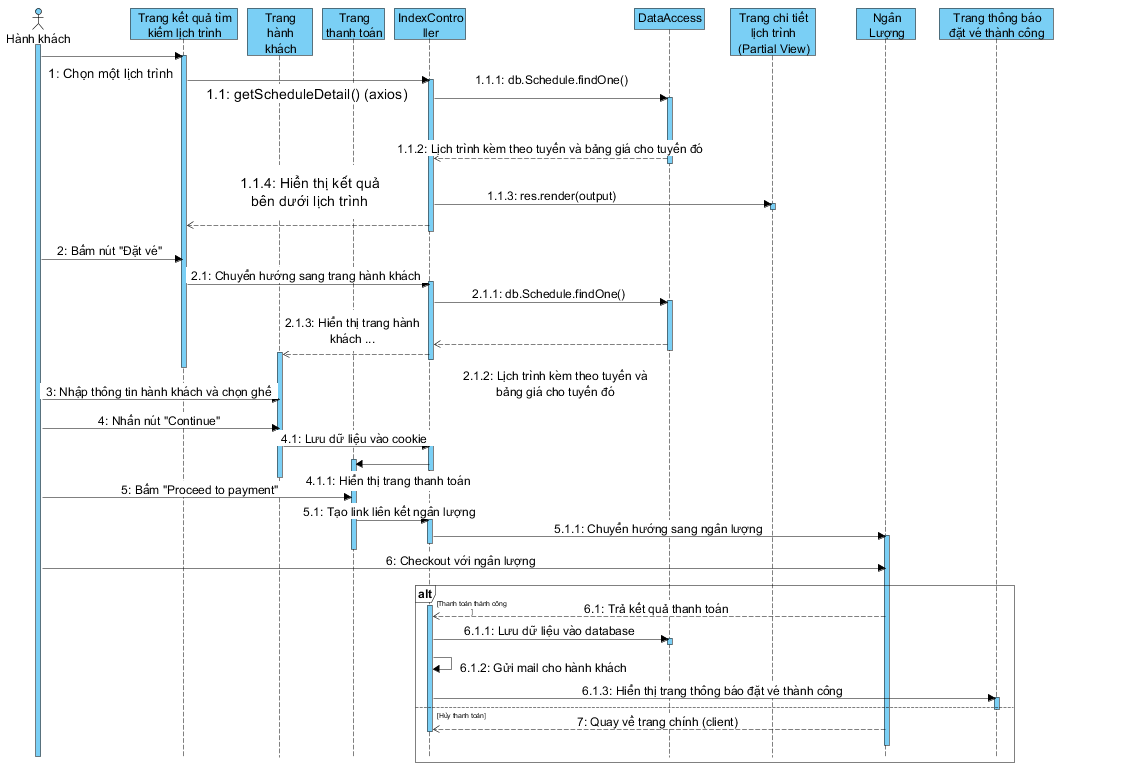
#### Biểu đồ

##### Activity Diagram



Hình 3‑5 Activity Diagram UC002 – Mua vé

##### Sequence Diagram



Hình 3‑6 Sequence Diagram UC002 – Mua vé

### UC003\_Tra thông tin vé đã đặt

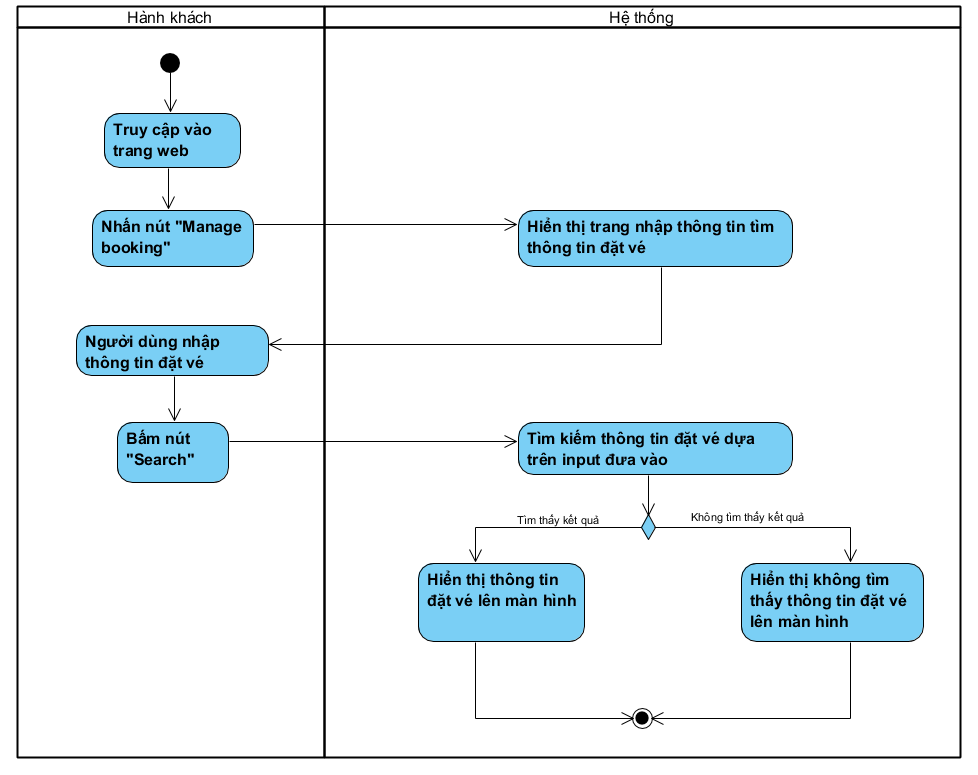
#### Mô tả use case UC003

|  |  |  |
| --- | --- | --- |
| **Use case:** Tra thông tin vé đã đặt | | |
| Mục đích: | Giúp hành khách tra thông tin đặt vé của mình có thành công hay không. | |
| Mô tả: | Tra thông tin đặt vé sau khi đặt vé để chắc rằng họ đã đặt vé thành công | |
| Tác nhân: | Hành khách. | |
| Điều kiện trước: | Truy cập vào hệ thống và nhấn nút “Manage Booking” | |
| Điều kiện sau: | Hiển thị ra thông tin đặt vé và hành khách. | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
| 1. Người dùng nhập email và mã đặt vé và nhấn “Search” | |  |
|  | | 1. Hệ thống lấy thông tin người đại diện đặt vé và các hành khách liên quan và hiển thị lên. |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 2.1 Nếu không tìm thấy thông tin hệ thống sẽ hiển thị không tìm thấy thông tin đặt vé. | | |

Bảng 3‑5 Mô tả UC003 – Tra thông tin đặt vé

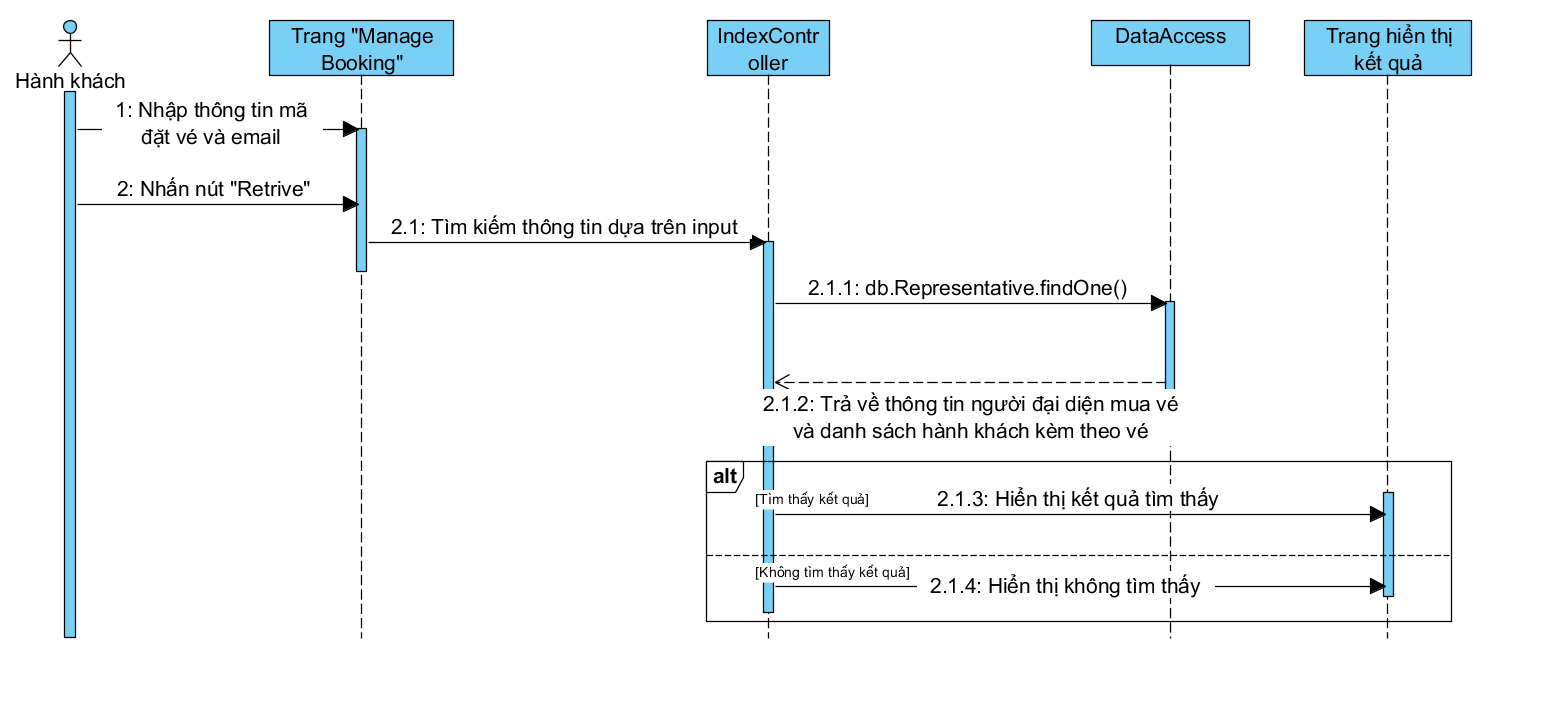
#### Biểu đồ

##### Activity Diagram



Hình 3‑7 Activity Diagram UC003 – Tra thông tin đặt vé

##### Sequence Diagram



Hình 3‑8 Sequence Diagram UC003 – Tra thông tin đặt vé

### UC004\_Tra tuyến theo lịch trình có sẵn

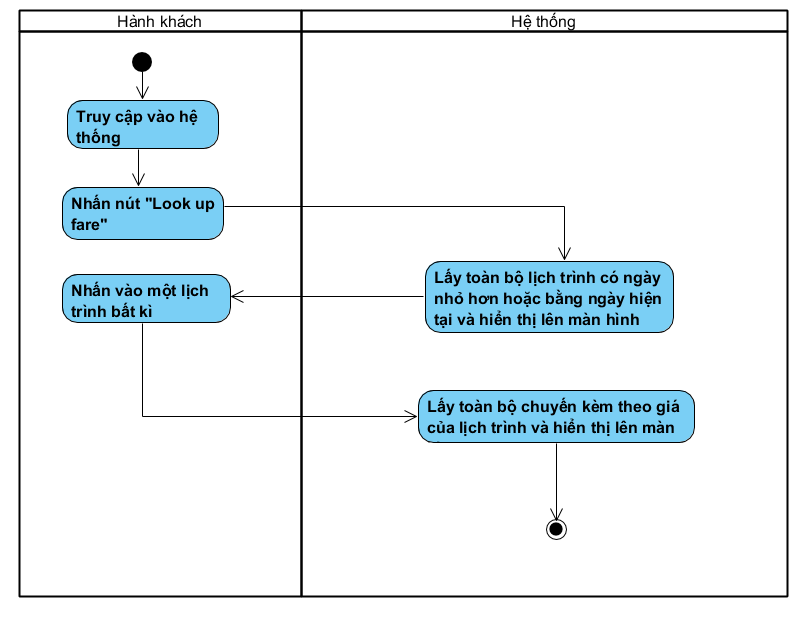
#### Mô tả use case UC004

|  |  |  |
| --- | --- | --- |
| **Use case:** Tra tuyến theo lịch trình có sẵn | | |
| Mục đích: | Tra giá vé của từng chuyến trên một lịch trình. | |
| Mô tả: | Hành khách dùng chức năng này để tra giá vé và có thể chọn đặt vé ngay trên đó. | |
| Tác nhân: | Hành khách. | |
| Điều kiện trước: | Truy cập vào hệ thống | |
| Điều kiện sau: | Hiển thị danh sách lịch trình có ngày lớn hơn hoặc bằng ngày hiện tại và có thể lấy được toàn bộ chuyến kèm theo giá của lịch trình đó. | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
| 1. Người dùng nhấn vào nút “List Schedule” | |  |
|  | | 1. Lấy danh sách lịch trình có ngày lớn hơn hoặc bằng ngày hiện tại |
| 1. Nhấn vào một lịch trình bất kì. | |  |
|  | | 1. Lấy toàn bộ chuyến kèm theo giá của lịch trình đó. |
| Luồng sự kiện phụ (Alternative Flows) : | | |
|  | | |

Bảng 3-6 Mô tả UC004 – Tra tuyến theo lịch trình có sẵn

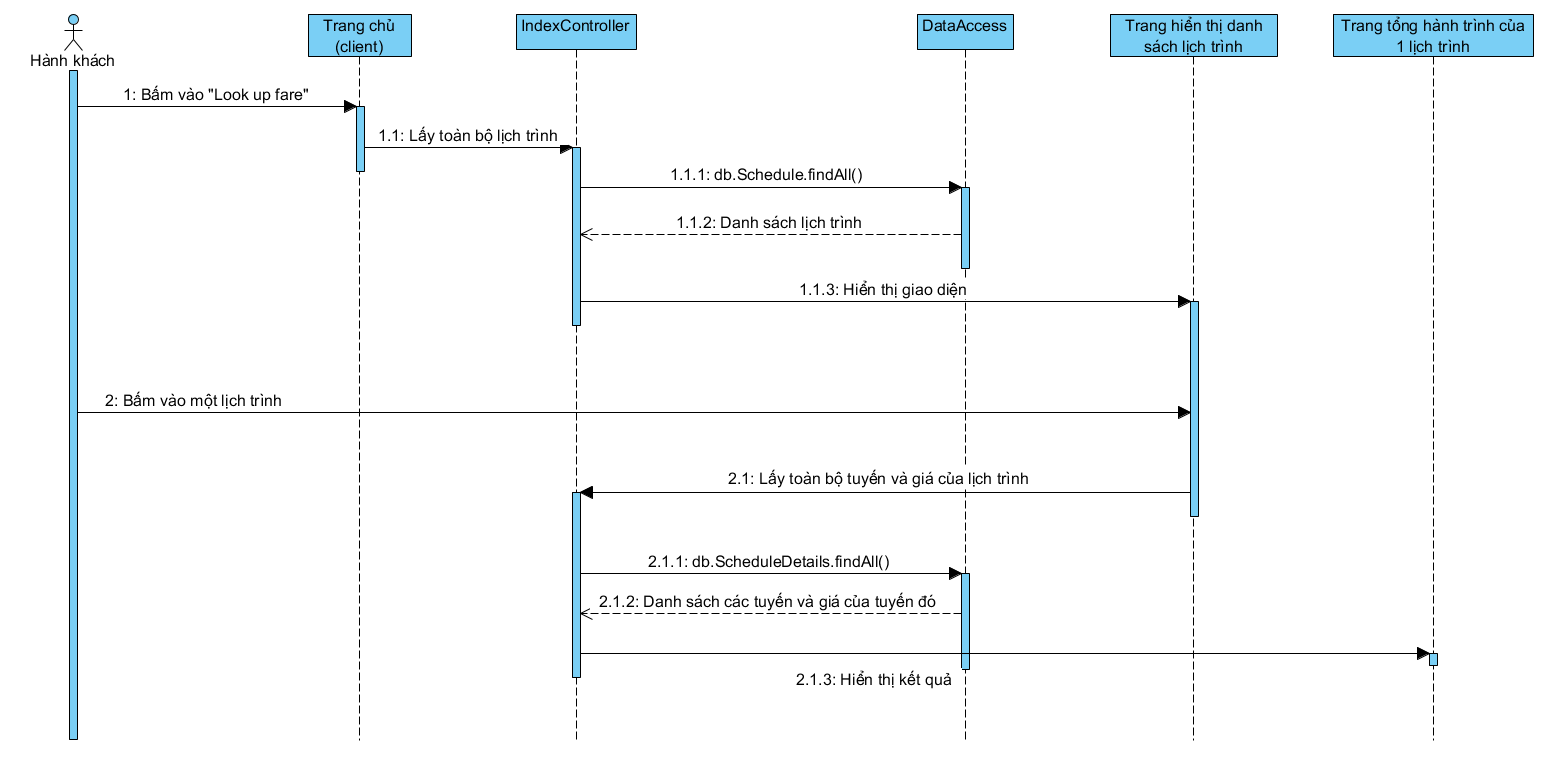
#### Biểu đồ

##### Activity Diagram



Hình 3‑9 Activity Diagram UC004 – Tra tuyến theo lịch trình có sẵn

##### Sequence Diagram



Hình 3‑10 Sequence Diagram UC004 – Tra tuyến theo lịch trình có sẵn

### UC005\_In vé

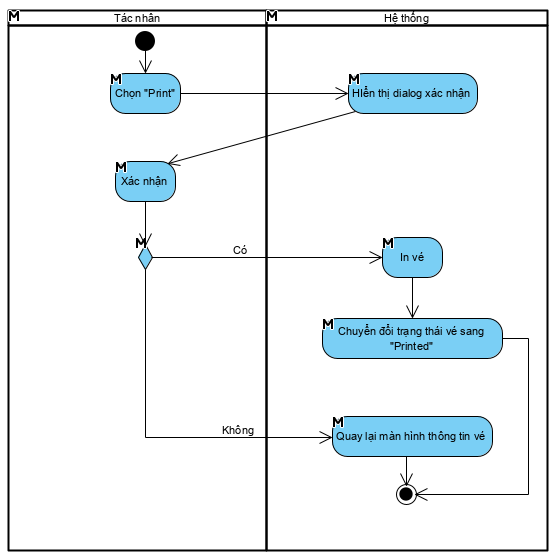
#### Mô tả use case UC005

|  |  |  |
| --- | --- | --- |
| **Use case:** In vé | | |
| Mục đích: | In vé cho khách | |
| Mô tả: | Giúp nhân viên có thể in vé đã được thanh toán cho khách hàng | |
| Tác nhân: | Nhân viên bán vé | |
| Điều kiện trước: | Đã có thông tin vé hợp lệ | |
| Điều kiện sau: | In vé thành công cho khách hàng, chuyển đổi trạng thái vé sang đã in thành công | |
| Luồng sự kiện chính (Basic flows): | | |
| **Actor** | | **System** |
| 1. Chọn “Print” | |  |
|  | | 1. Hiển thị dialog xác nhận |
| 1. Chọn “Yes” | |  |
|  | | 1. In vé |
|  | | 1. Chuyển đổi trạng thái vé sang “Printed” |
| Luồng sự kiện phụ (Alternative flows): | | |
| 3.1 Chọn “No” | | |
| 4.1 Quay lại màn hình thông tin vé | | |

Bảng 3-7 Mô tả UC005 – In vé

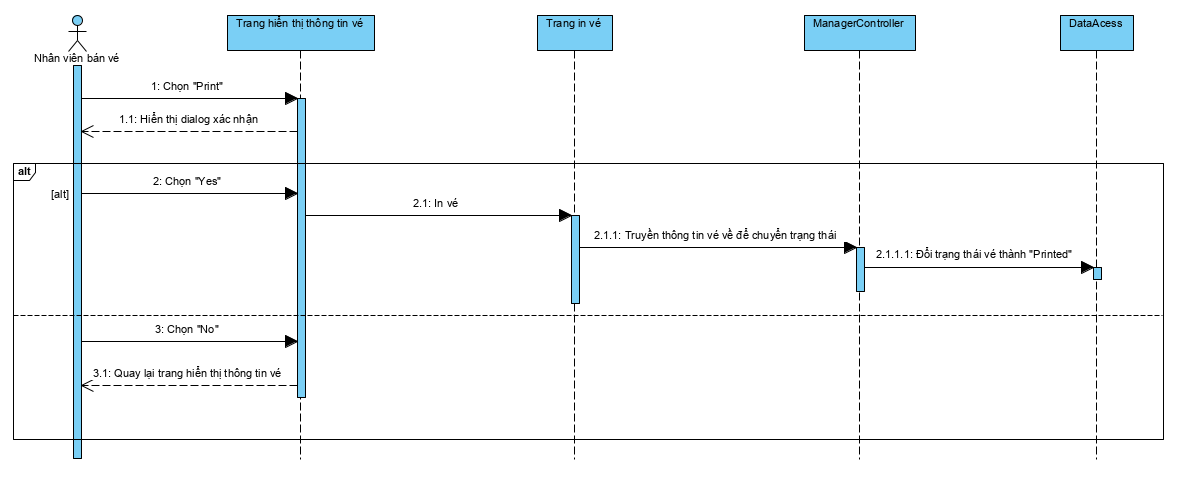
#### Biểu đồ

##### Activity Diagram



Hình 3‑11 Activity Diagram UC005 – In vé

##### Sequence Diagram



Hình 3‑12 Sequence Diagram UC005 – In vé

### UC006\_Huỷ vé

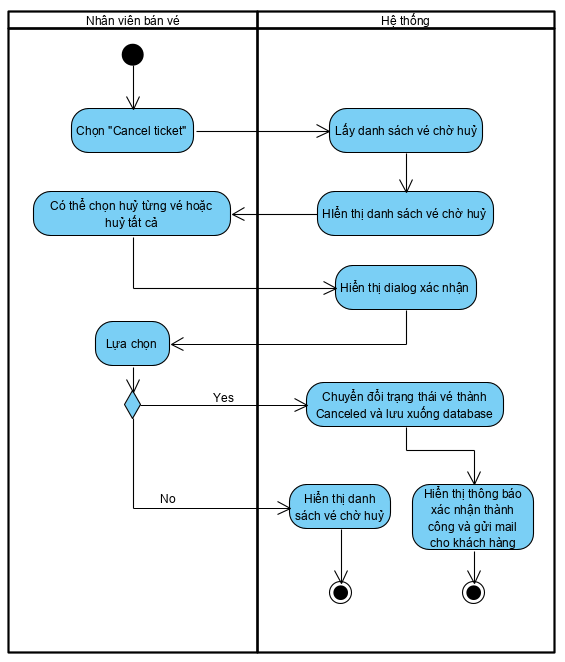
#### Mô tả use case UC006

|  |  |  |
| --- | --- | --- |
| **Use case:** Huỷ vé | | |
| Mục đích: | Huỷ vé cho khách | |
| Mô tả: | Giúp nhân viên có thể huỷ vé đã được thanh toán cho khách hàng | |
| Tác nhân: | Nhân viên bán vé | |
| Điều kiện trước: | Đã có thông tin vé hợp lệ | |
| Điều kiện sau: | Huỷ vé thành công cho khách hàng, chuyển đổi trạng thái vé sang huỷ | |
| Luồng sự kiện chính (Basic flows): | | |
| **Actor** | | **System** |
| 1. Chọn “Cancel ticket” | |  |
|  | | 1. Lấy danh sách vé chờ huỷ (UC012) |
|  | | 1. Hiển thị danh sách vé chờ huỷ |
| 1. Có thể chọn huỷ từng vé hoặc huỷ tất cả vé | |  |
|  | | 1. Hiển thị dialog xác nhận |
| 1. Chọn “Yes” | |  |
|  | | 1. Chuyển đổi trạng thái vé sang Canceled, lưu thông tin xuống database |
|  | | 1. Hiển thị thông báo thành công và gửi mail cho khách hàng |
| Luồng sự kiện phụ (Alternative flows): | | |
| 6.1 Chọn “No”, hệ thống sẽ quay lại màn hình hiển thị danh sách vé chờ huỷ | | |

Bảng 3-8 Mô tả UC006 – Hủy vé

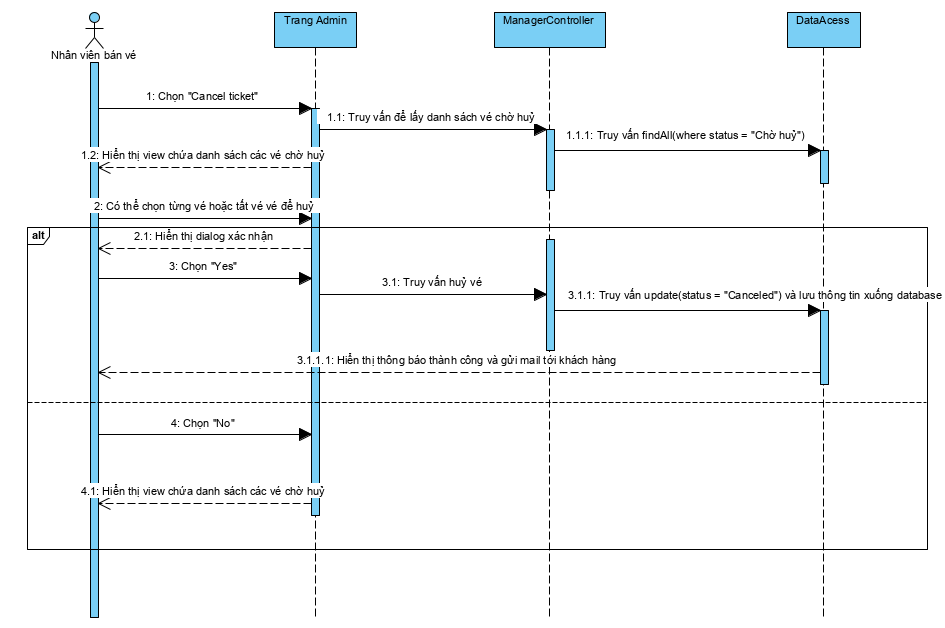
#### Biểu đồ

##### Activity Diagram



Hình 3‑13 Activity Diagram UC006 – Hủy vé

##### Sequence Diagram



Hình 3‑14 Sequence Diagram UC006 – Hủy vé

### UC007\_Lên lịch trình

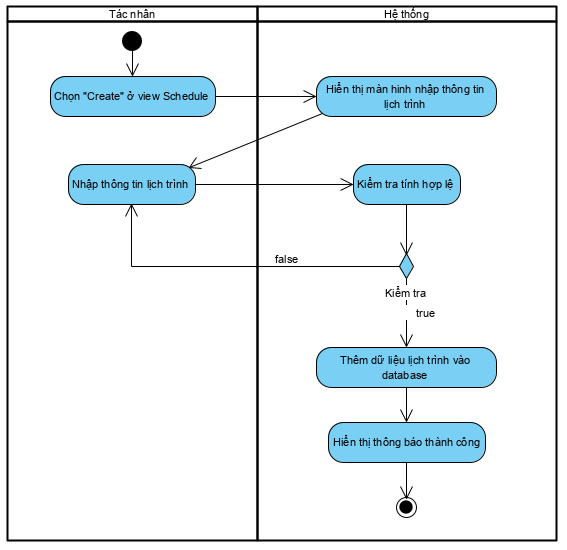
#### Mô tả use case UC007

|  |  |  |
| --- | --- | --- |
| **Use case:** Lên lịch trình | | |
| Mục đích: | Lên lịch trình cho các chuyến tàu | |
| Mô tả: | Giúp quản lý có thể lên lịch trình cho các chuyến tàu phục vụ nhu cầu mua vé của hành khách | |
| Tác nhân: | Nhân viên quản lý | |
| Điều kiện trước: | Đã đăng nhập vào hệ thống với quyền admin | |
| Điều kiện sau: | Thêm lịch trình thành công vào hệ thống (bao gồm cả những tuyến con bên trong lịch trình đó) | |
| Luồng sự kiện chính (Basic flows): | | |
| **Actor** | | **System** |
| 1. Chọn “Create” ở view Schedule | |  |
|  | | 1. Hiển thị màn hình để nhập thông tin lên lịch trình |
| 1. Nhập thông tin lên lịch trình (bao gồm tàu, nơi đi, nơi đến, thời gian,…) | |  |
|  | | 1. Kiểm tra tính hợp lệ của thông tin đã nhập |
|  | | 1. Thêm dữ liệu lịch trình vào database |
|  | | 1. Hiển thị thông báo thành công |
| Luồng sự kiện phụ (Alternative flows): | | |
| 4.1 Thông tin không hợp lệ, quay lại bước 3 | | |

Bảng 3-9 Mô tả UC007 – Lên lịch trình

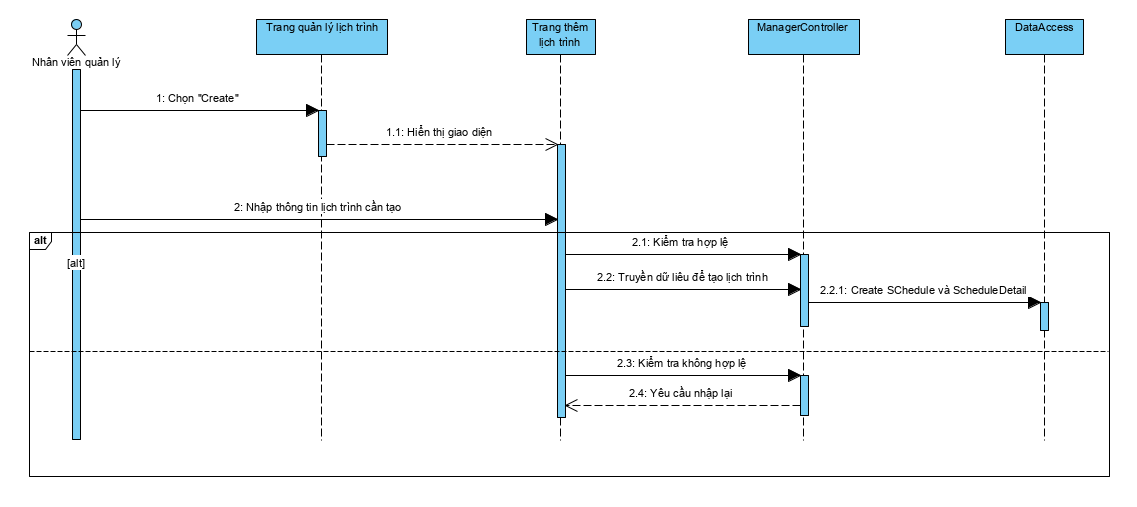
#### Biểu đồ

##### Activity Diagram



Hình 3‑15 Activity Diagram UC007 – Lên lịch trình

##### Sequence Diagram



Hình 3‑16 Sequence Diagram UC007 – Lên lịch trình

### UC008\_Thống kê vé bán được và doanh thu

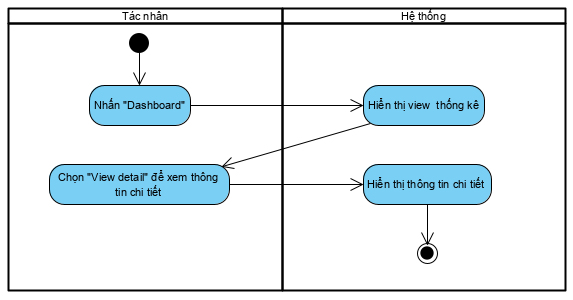
#### Mô tả use case UC008

|  |  |  |
| --- | --- | --- |
| **Use case:** Thống kê vé bán được và doanh thu | | |
| Mục đích: | Thống kê vé bán được và doanh thu của tổ chức | |
| Mô tả: | Giúp quản lý có thể kiểm soát thống kê vé bán được và doanh thu | |
| Tác nhân: | Nhân viên quản lý | |
| Điều kiện trước: | Đã đăng nhập vào hệ thống với quyền admin | |
| Điều kiện sau: | Hiển thị thành công các thông tin thống kê liên quan | |
| Luồng sự kiện chính (Basic flows): | | |
| **Actor** | | **System** |
| 1. Nhấn “Dashboard” | |  |
|  | | 1. Hiển thị view chứa thông tin thống kê |
| 1. Có thể kiểm xem thông tin chi tiết bằng cách nhấn “View detail” | |  |
|  | | 1. Hiển thị thông tin chi tiết vé bán được và doanh thu |
| Luồng sự kiện phụ (Alternative flows): | | |
| N/A | | |

Bảng 3-10 Mô tả UC008 – Thống kê vé bán được và doanh thu

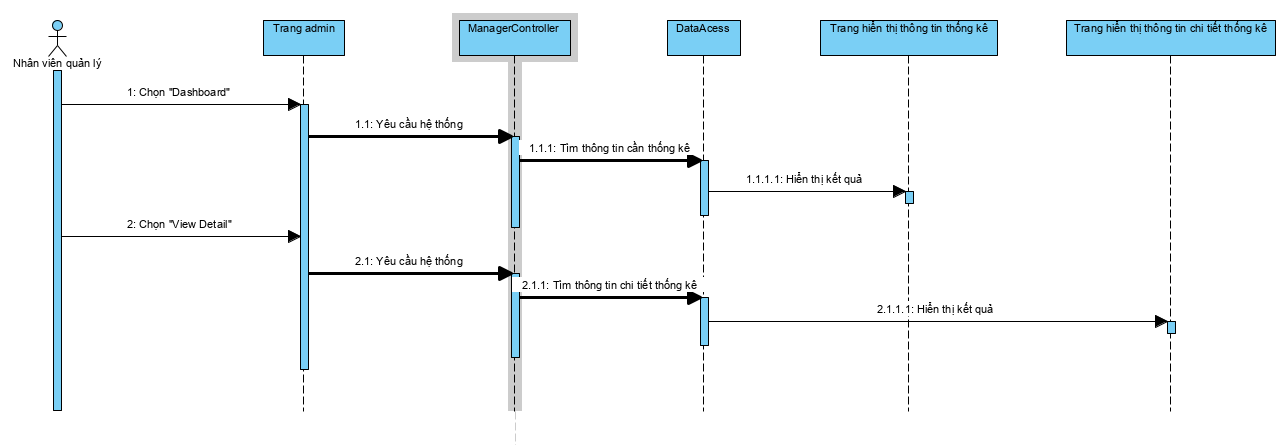
#### Biểu đồ

##### Activity Diagram



Hình 3‑17 Activity Diagram UC008 – Thống kê vé bán được và doanh thu

##### Sequence Diagram



Hình 3‑18 Sequence Diagram UC008 – Thống kê vé bán được và doanh thu

### UC009\_Quản lý hành khách theo lịch trình

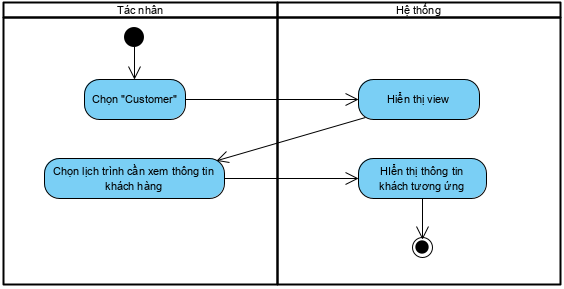
#### Mô tả use case UC009

|  |  |  |
| --- | --- | --- |
| **Use case:** Quản lý hành khách theo lịch trình | | |
| Mục đích: | Quản lý thông tin của hành khách theo lịch trình | |
| Mô tả: | Giúp quản lý có thể kiểm soát thông tin của hành khách theo lịch trình | |
| Tác nhân: | Nhân viên quản lý | |
| Điều kiện trước: | Đã đăng nhập vào hệ thống với quyền admin | |
| Điều kiện sau: | Hiển thị thành công các thông tin thống kê liên quan | |
| Luồng sự kiện chính (Basic flows): | | |
| **Actor** | | **System** |
| 1. Chọn “Customer” | |  |
|  | | 1. Hiển thị view |
| 1. Chọn lịch trình cần xem thông tin hành khách | |  |
|  | | 1. Hiển thị thông tin tương ứng |
| Luồng sự kiện phụ (Alternative flows): | | |
| N/A | | |

Bảng 3-11 Mô tả UC009 – Quản lý khách hàng theo lịch trình

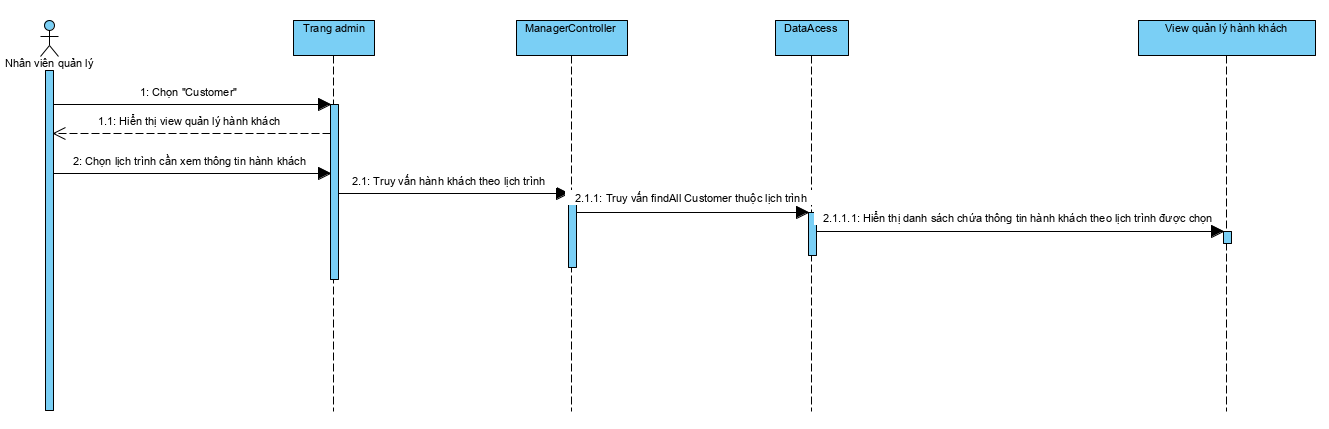
#### Biểu đồ

##### Activity Diagram



Hình 3‑19 Activity Diagram UC009 – Quản lý hành khách theo lịch trình

##### Sequence Diagram



Hình 3‑20 Senquence Diagram UC009 – Quản lý hành khách theo lịch trình

### UC010\_Đăng nhập

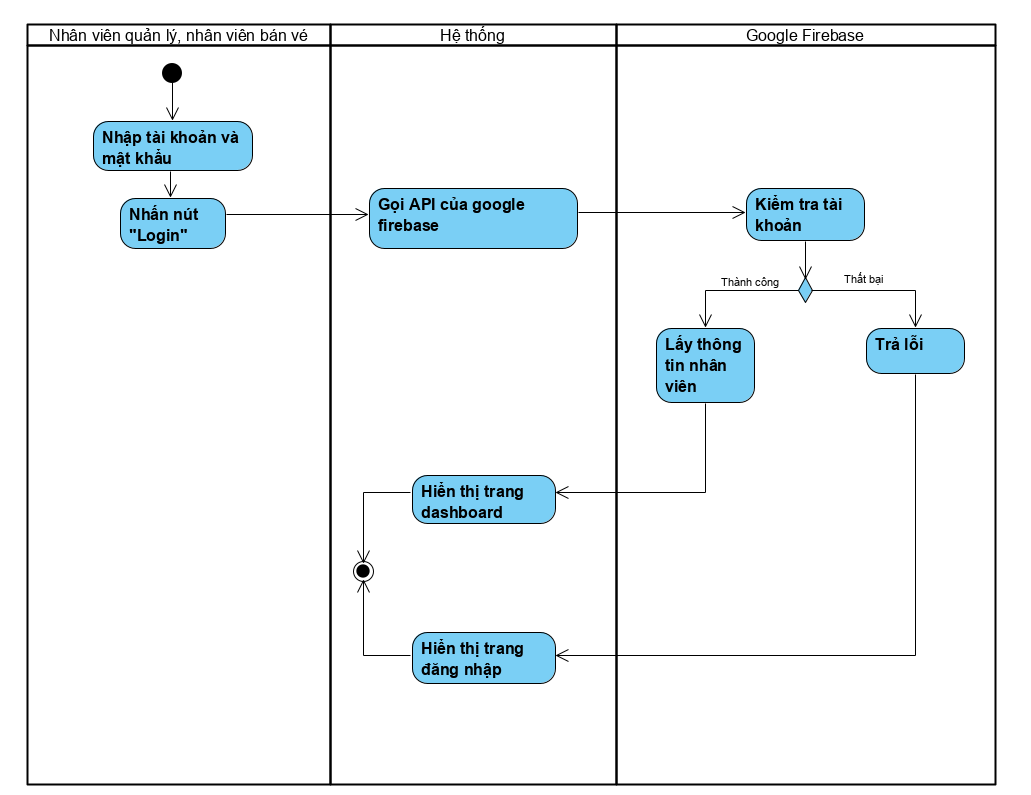
#### Mô tả use case UC010

|  |  |  |
| --- | --- | --- |
| **Use case:** Đăng nhập | | |
| Mục đích: | Cho nhân viên đăng nhập vào hệ thống | |
| Mô tả: | Giúp cho nhân viên của bên bán có thể đăng nhập vào hệ thống để thực hiện chức năng. | |
| Tác nhân: | Nhân viên quản lý, nhân viên bán vé | |
| Điều kiện trước: | Chưa đăng nhập vào hệ thống | |
| Điều kiện sau: | Chuyển về trang dashboard của hệ thống quản lý | |
| Luồng sự kiện chính (Basic flows): | | |
| **Actor** | | **System** |
| 1. Nhập tài khoản và mật khẩu | |  |
| 2. Nhấn nút đăng nhập | |  |
|  | | 3.Kết nối đến google firebase để xác thực và lấy thông tin |
|  | | 4. Hiển thị trang dashboard |
| Luồng sự kiện phụ (Alternative flows): | | |
| 4.1. Nếu kết quả trả về là một đoạn báo lỗi thì sẽ hiển thị lại trang đăng nhập kèm theo lỗi. | | |

Bảng 3-12 Mô tả UC010 – Đăng nhập

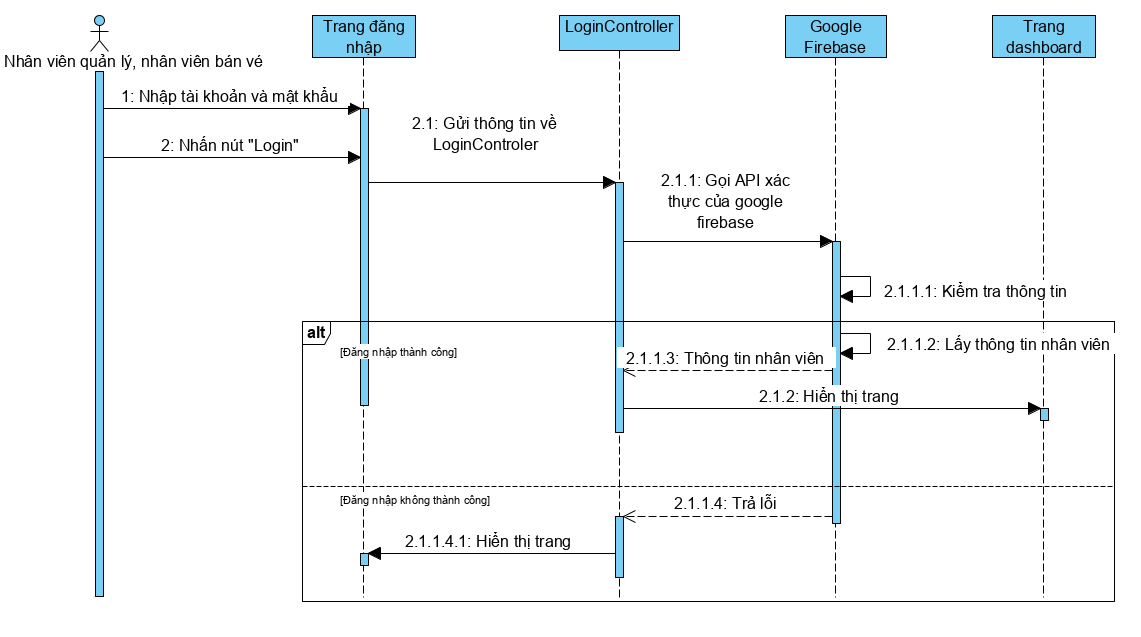
#### Biểu đồ

##### Activity Diagram



Hình 3‑21 Activity Diagram UC010 – Đăng nhập

##### Sequence Diagram



Hình 3‑22 Sequence Diagram UC010 – Đăng nhập

### UC011\_Tra vé theo mã đặt vé

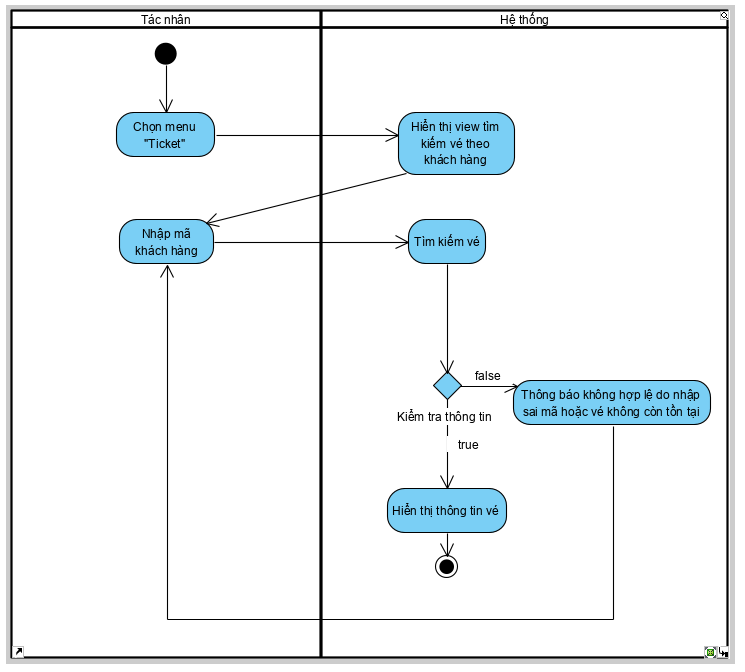
#### Mô tả use case UC011

|  |  |  |
| --- | --- | --- |
| **Use case:** Tìm kiếm vé theo mã đặt vé | | |
| Mục đích: | Tìm kiếm vé dựa theo mã khách hàng cung cấp để in hoặc huỷ vé | |
| Mô tả: | Giúp nhân viên có thể tìm kiếm vé dựa theo mã khách hàng cung cấp để in hoặc huỷ vé | |
| Tác nhân: | Nhân viên bán vé | |
| Điều kiện trước: | Đã đăng nhập thành công vào hệ thống và được cấp quyền là nhân viên bán vé | |
| Điều kiện sau: | Hiển thị thông tin vé lên màn hình thành công | |
| Luồng sự kiện chính (Basic flows): | | |
| **Actor** | | **System** |
| 1. Chọn menu “Ticket” | |  |
|  | | 1. Hiển thị view tìm kiếm vé theo khách hàng |
| 1. Nhập mã khách hàng | |  |
|  | | 1. Tìm kiếm vé |
|  | | 1. Hiển thị thông tin vé |
| Luồng sự kiện phụ (Alternative flows): | | |
| 5.1 Hệ thống hiển thị thông tin không hợp lệ nếu mã khách hàng cung cấp sai | | |
| 5.2 Hệ thống hiển thị thông tin vé không còn tồn tại nếu vé đã được in trước đó | | |

Bảng 3-13 Mô tả UC011 – Tra vé theo mã đặt vé

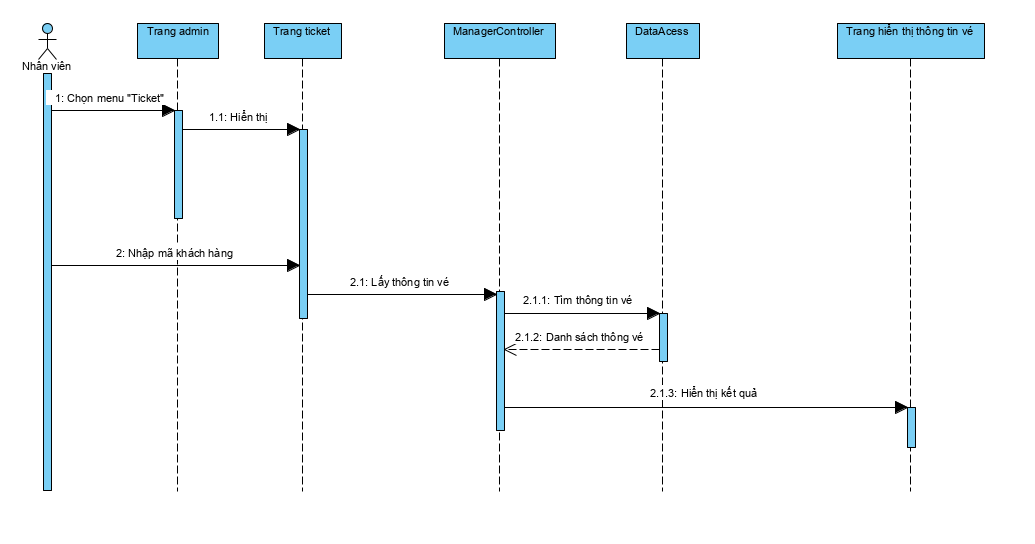
#### Biểu đồ

##### Activity Diagram



Hình 3‑23 Activity Diagram UC011 – Tra vé theo mã đặt vé

##### Sequence Diagram



Hình 3‑24 Sequence Diagram UC011 – Tra vé theo mã đặt vé

### UC012\_Lấy danh sách vé chờ huỷ

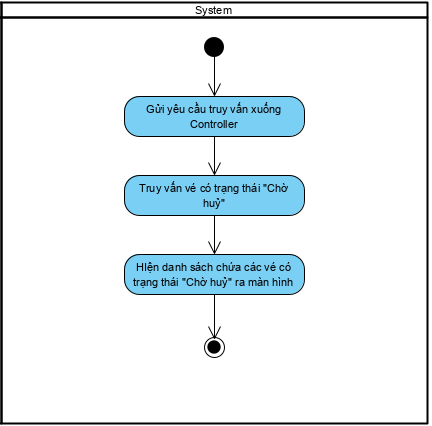
#### Mô tả use case UC012

|  |  |
| --- | --- |
| **Use case:** Lấy danh sách vé chờ huỷ | |
| Mục đích: | Lấy danh sách các vé được chờ huỷ từ yêu cầu của khách hàng |
| Mô tả: | Giúp lấy danh sách cá vé chờ huỷ từ database |
| Tác nhân: | Hệ thống khi được gọi khi bấm nút “Cancel ticket” |
| Điều kiện trước: | Đã bấm nút “Cancel ticket” ở menu chính yêu cầu lấy danh sách vé chờ huỷ |
| Điều kiện sau: | Load danh sách vé chờ huỷ thành công |
| Luồng sự kiện chính (Basic flows): | |
| **System** | |
| 1. Gửi yêu cầu xuống Controller | |
| 1. Truy vấn các vé có trạng thái “Chờ huỷ” | |
| 1. Load danh sách chứa các vé có trạng thái “Chờ huỷ” lên màn hình | |
| Luồng sự kiện phụ (Alternative flows): | |
| N/A | |

Bảng 3-14 Mô tả UC012 – Lấy danh sách vé chờ hủy

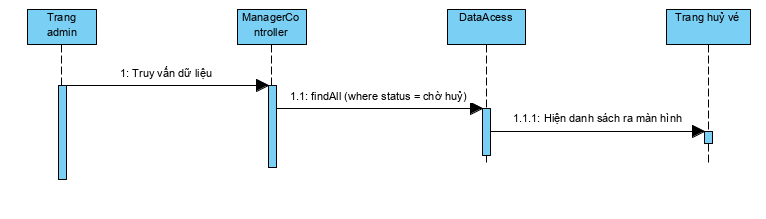
#### Biểu đồ

##### Activity Diagram



Hình 3‑25 Activity Diagram UC012 – Lấy danh sách vé chờ hủy

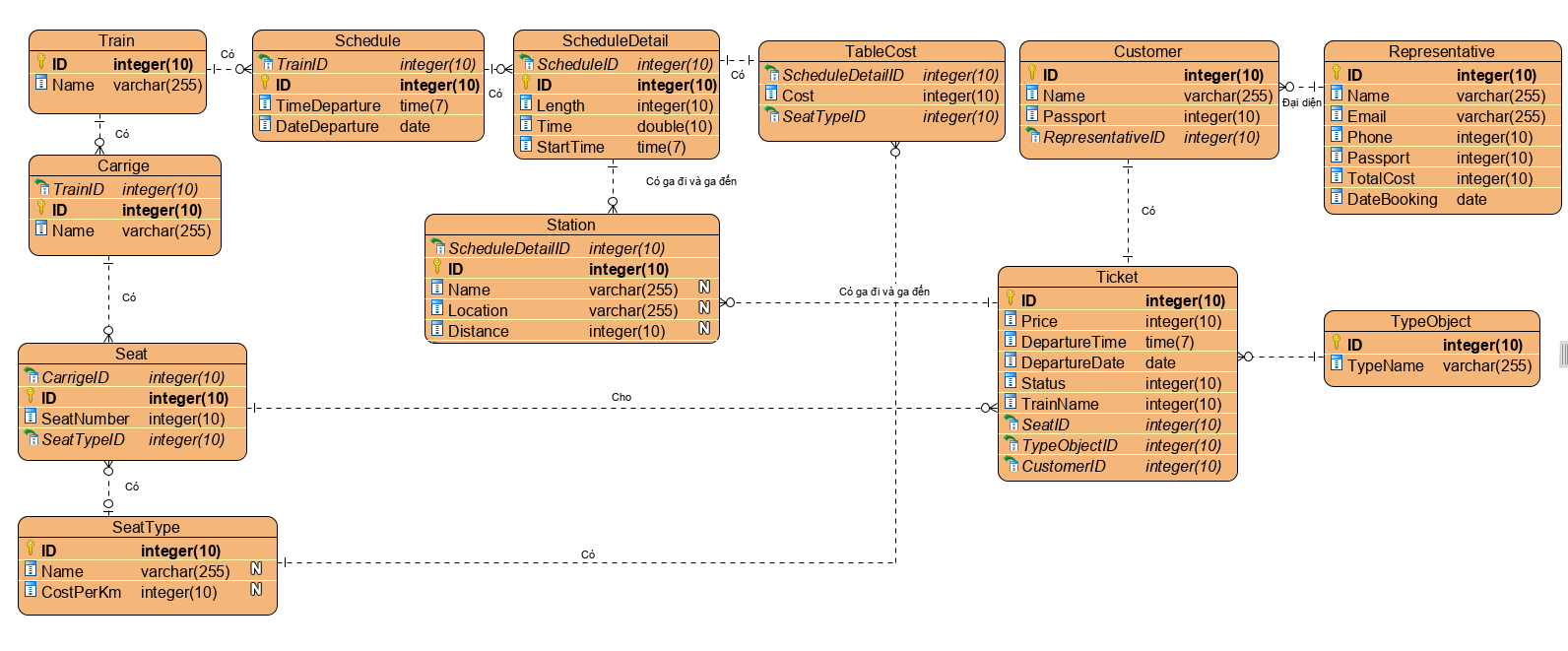
##### Sequence Diagram



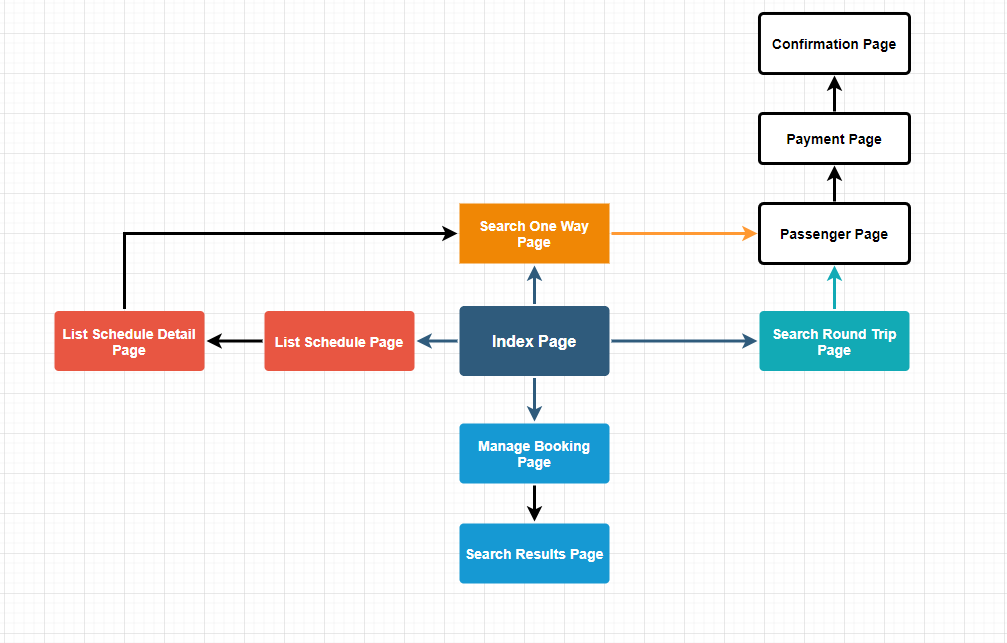
Hình 3‑26 Sequence Diagram UC012 – Lấy danh sách vé chờ hủy

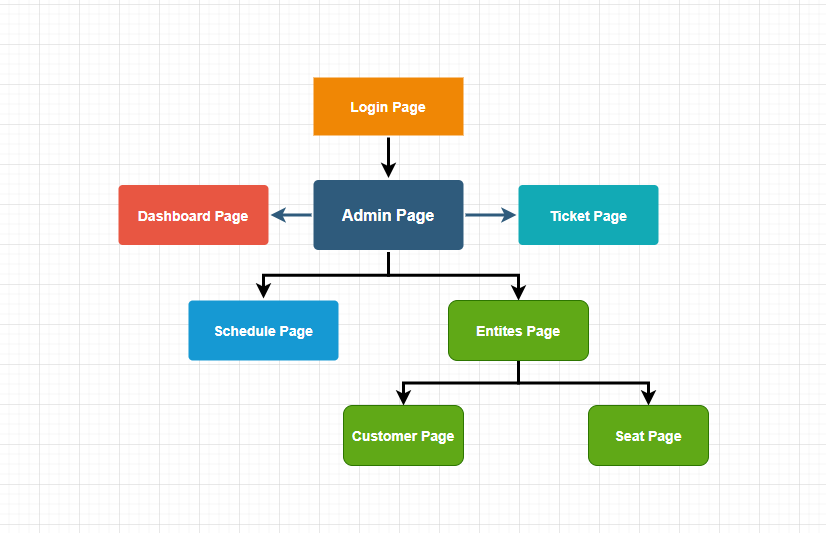
# : THIẾT KẾ VÀ HIỆN THỰC

## ClassDiagram

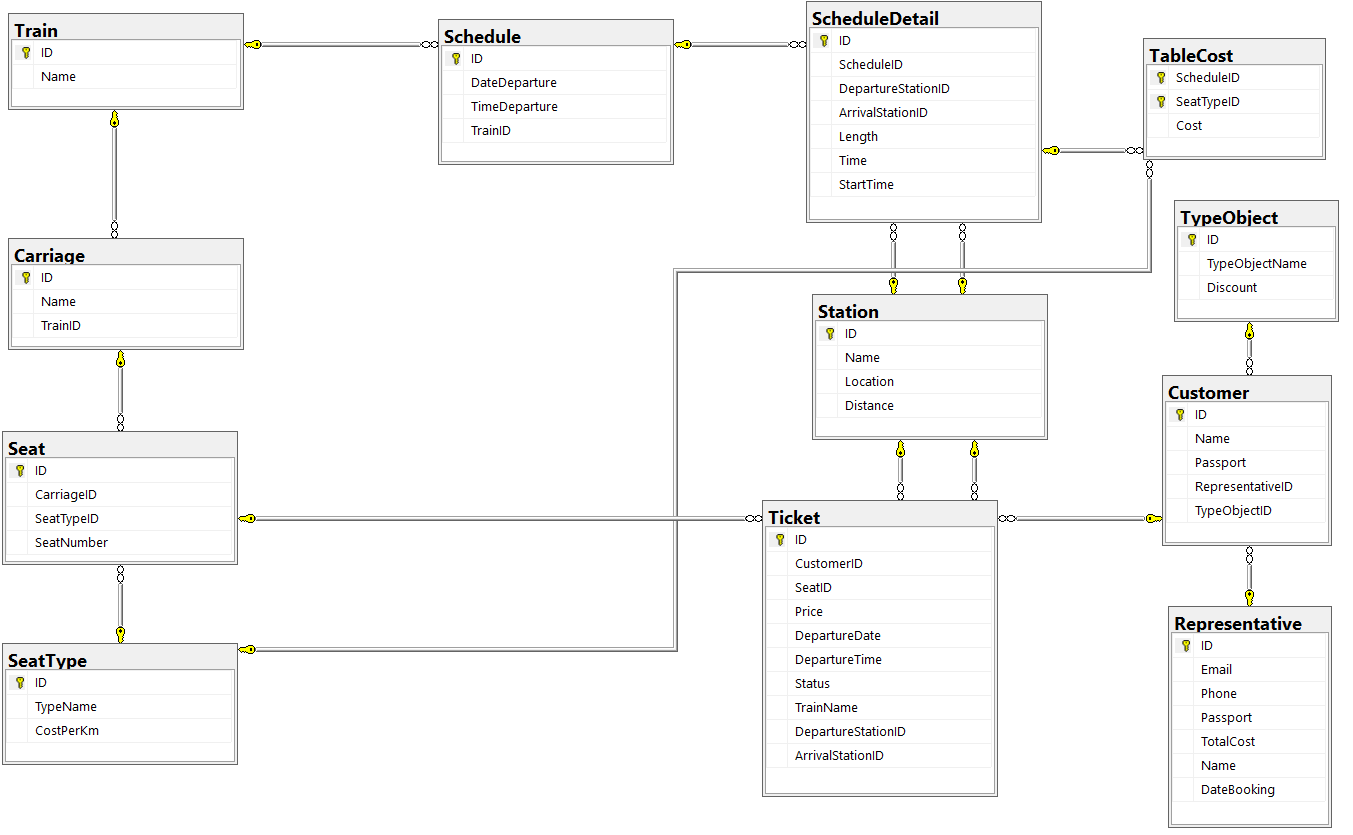


## Sitemap

Site Map Trang Hành Khách

Site Map Trang Quản Trị

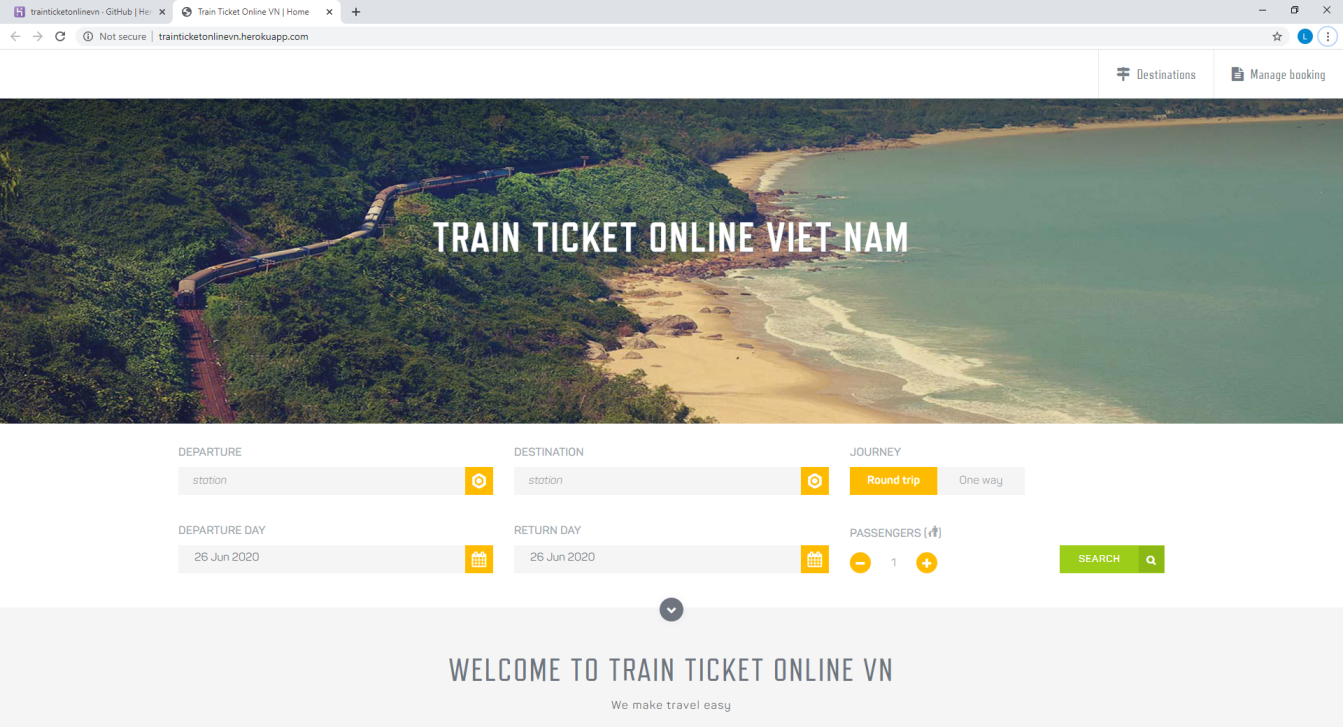
## Database Diagram



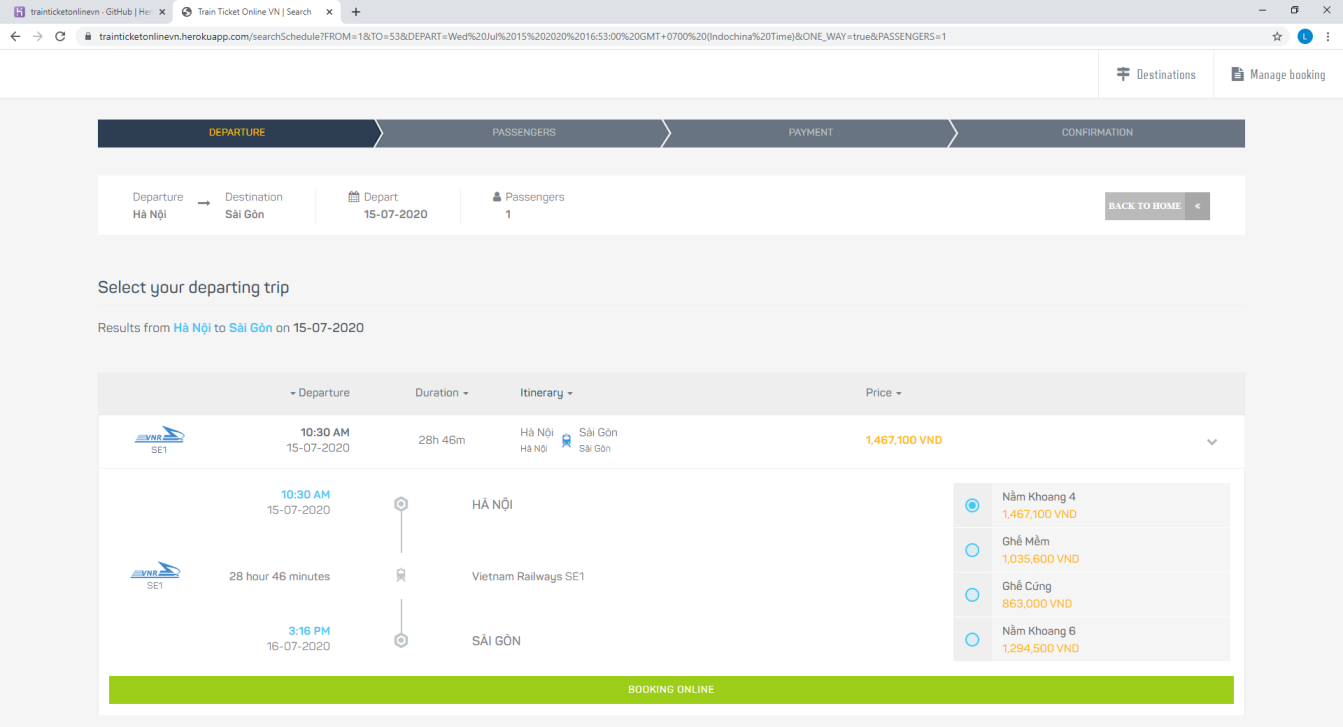
## Mô tả Website

### Một số chức năng và giao diện chính

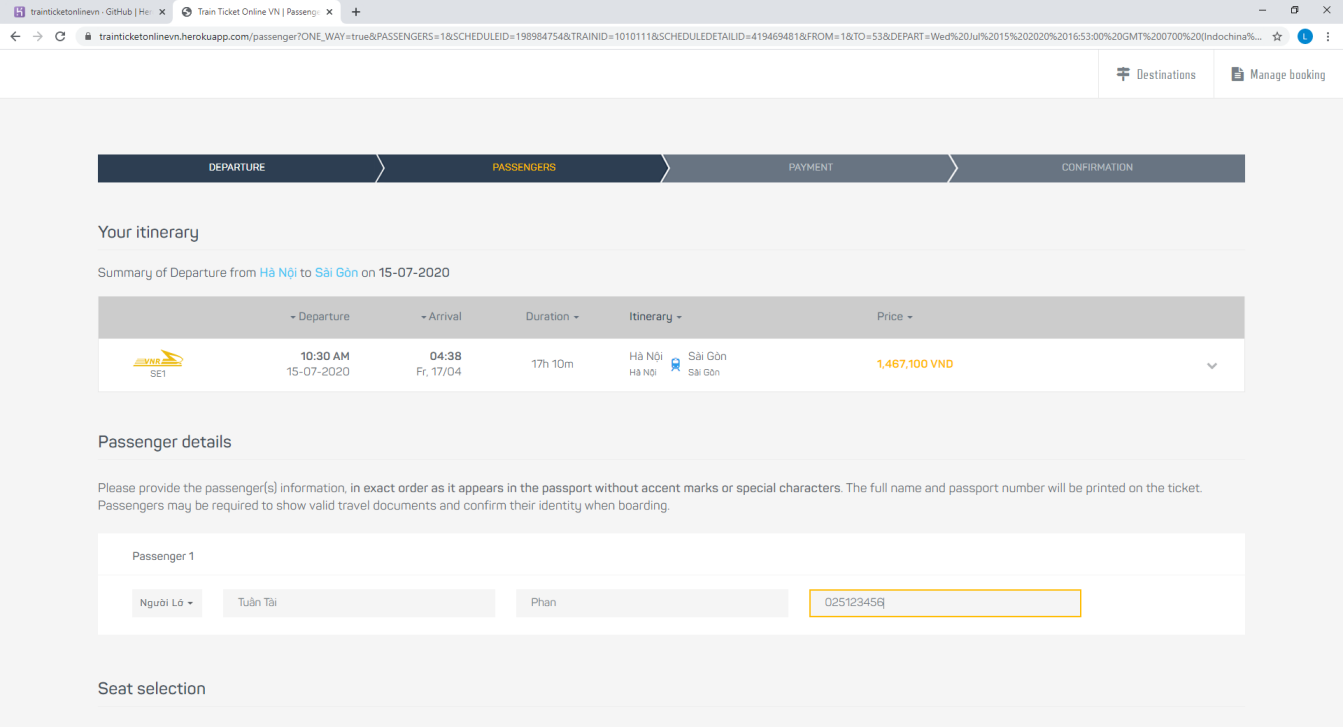
Trang chủ khách hàng

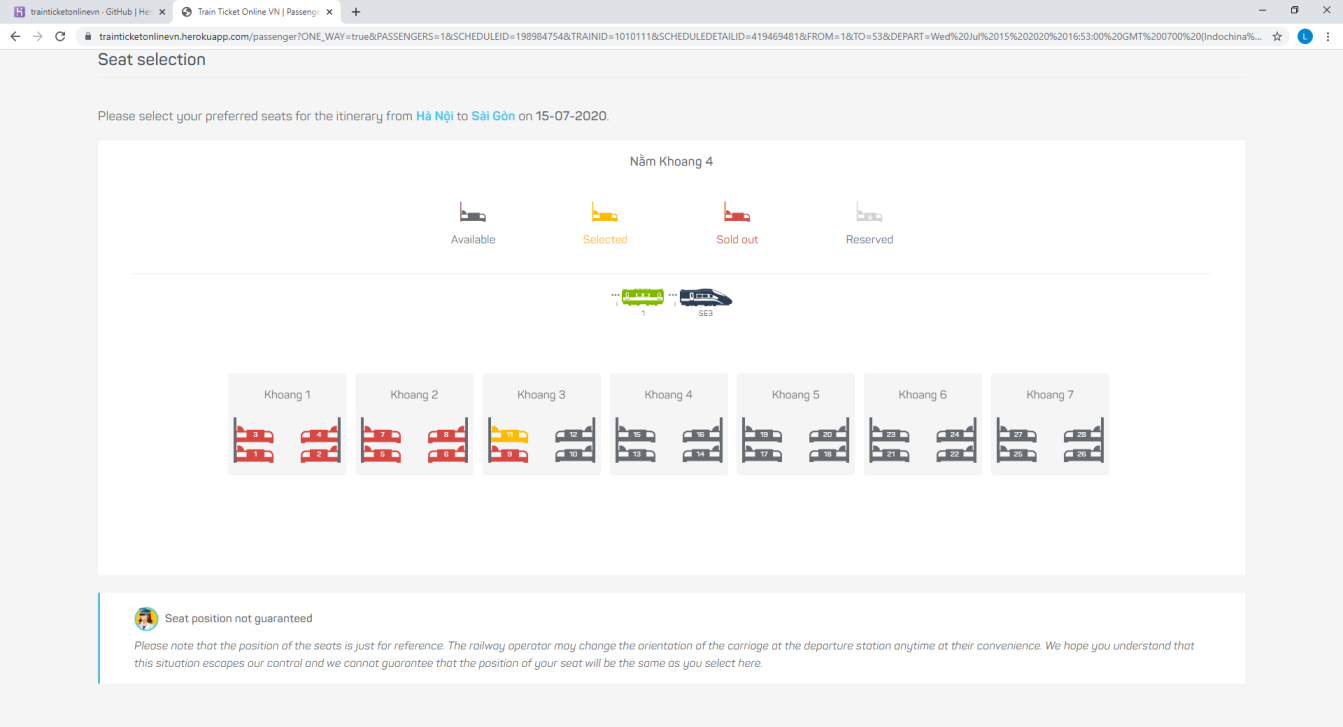


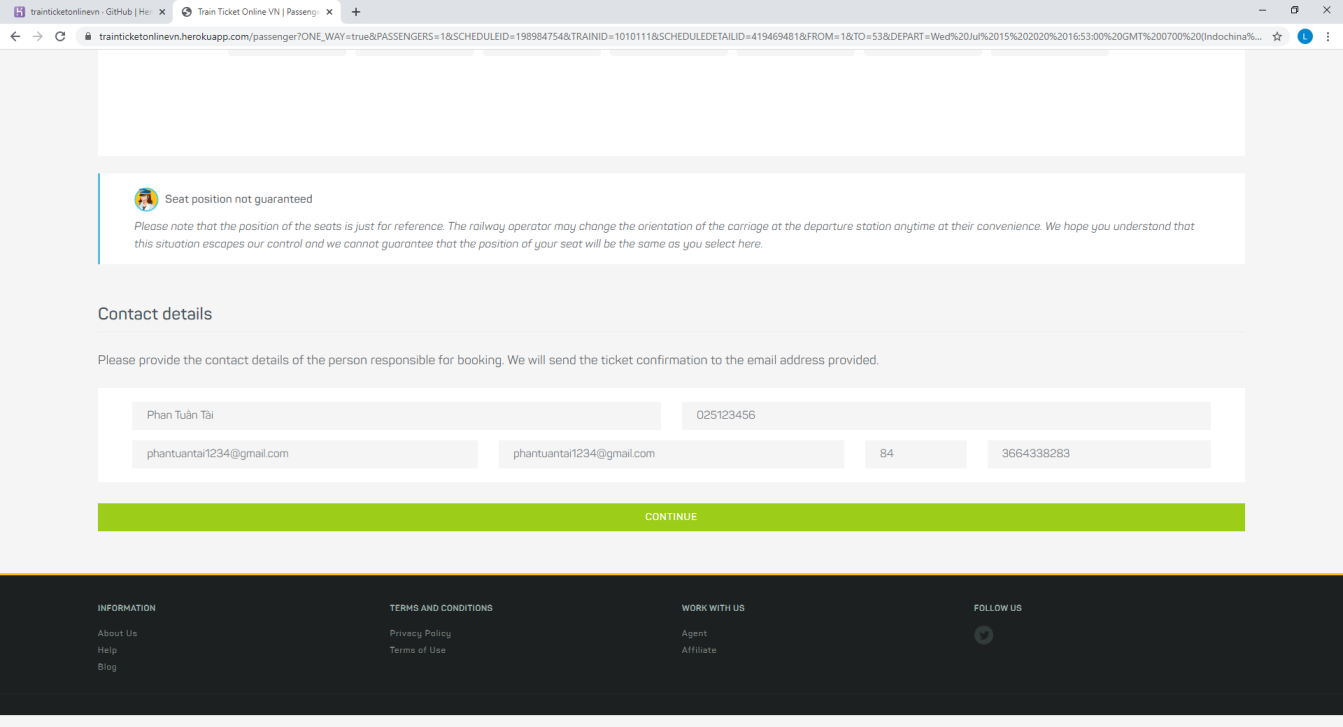
Trang kết quả tìm kiếm lịch trình



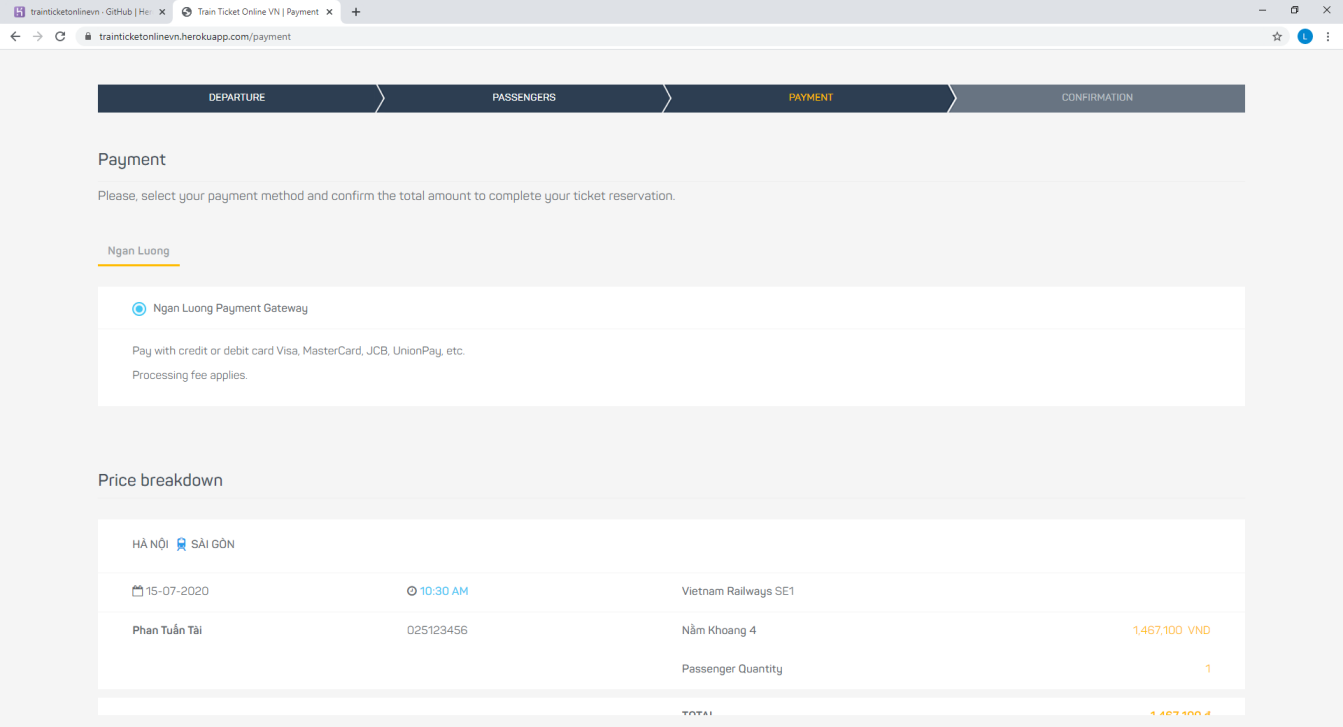
Trang chọn ghế và điền thông tin khách hàng

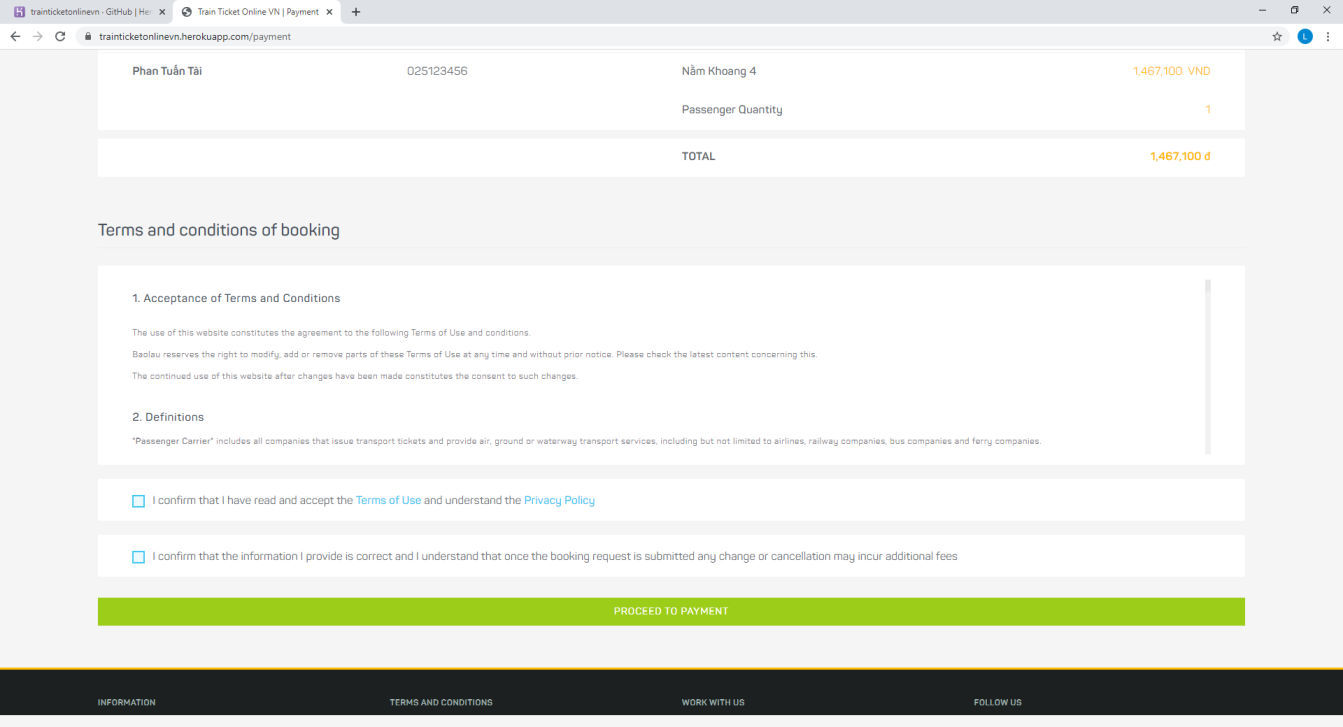




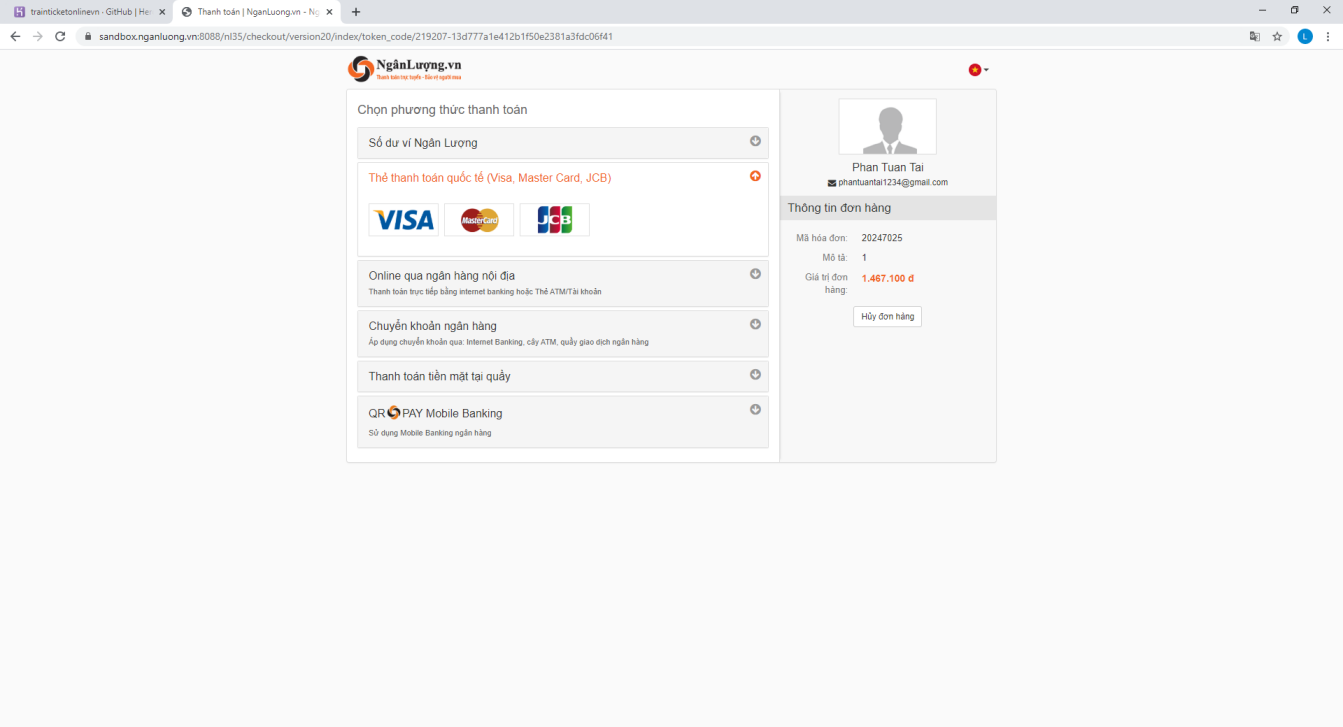


Trang xem lại thông tin và chọn phương thức thanh toán

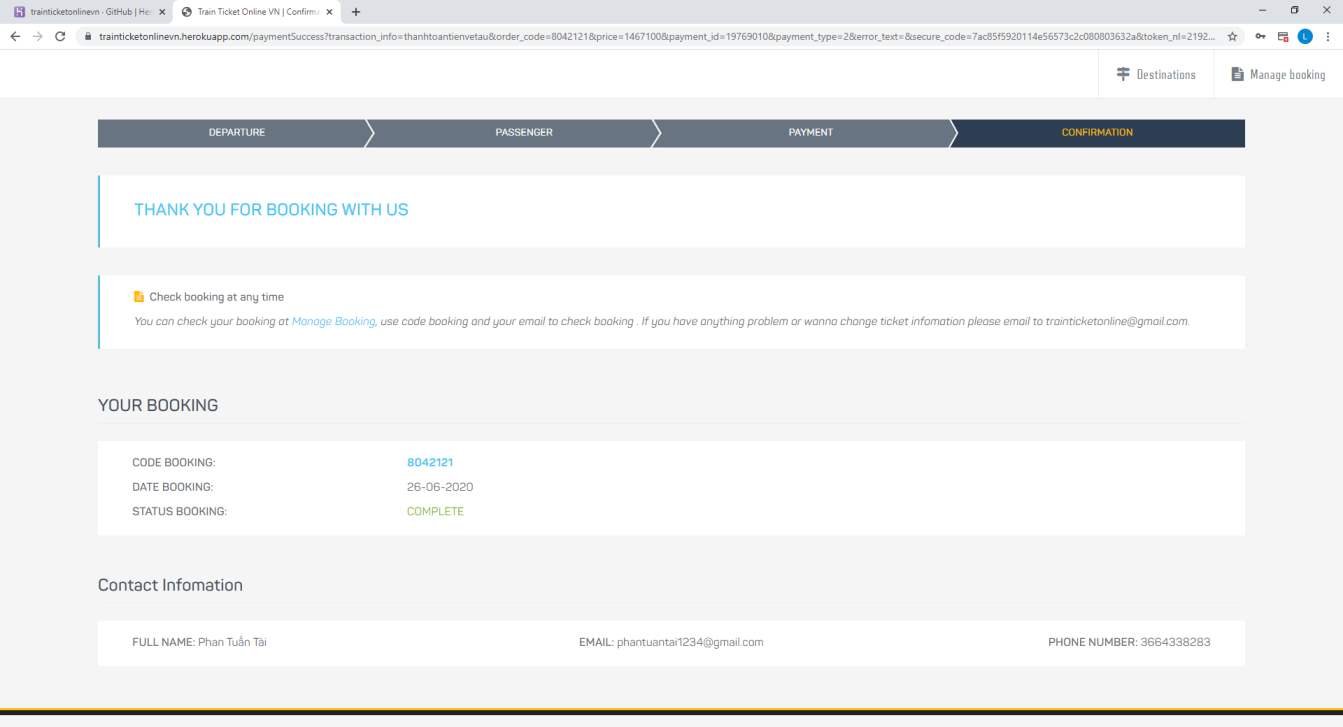




Trang thanh toán của ngân lượng



Trang xác nhận sau khi thanh toán thành công



# MỤC: KẾT LUẬN

Báo cáo đã phân tích những chức năng cơ bản mà hệ thống cần thực hiện. Từ bước phân tích thiết kế đến tạo lập hệ thống ứng dụng. Nhìn chung hệ thống đã đáp ứng được các chức năng cơ bản của một ứng dụng bán vé tàu online.

## Kết quả đạt được

* Hệ thống cho phép người dùng thực hiện việc mua vé tàu online một cách nhanh chóng và dễ dàng.
* Hệ thống giúp cho bên bán vé dễ kiểm soát vé bán được, doanh thu theo tháng và năm, số lượng vé bán được trên từng lịch trình.

## Hạn chế của đồ án

* Chưa đáp ứng tốt phần hổ trợ cho các thiết bị di động.
* Vẫn còn một số lỗi nhỏ ở phần hiển thị thông tin.

## Hướng phát triển

* Làm thêm ứng dụng di động để hổ trợ tốt nhất cho các thiết bị di động.
* Mở thêm các dịch vụ bán vé như là vé máy bay, vé xe khách.
* Chuyển website từ MVC (Model – View - Controller) sang SPA (Single Page Application)
* .

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Việt

1. Tài liệu hướng dẫn tích hợp thanh toán của ngân lượng <https://drive.google.com/file/d/1kPUmbY7aaydVL1ALMzx9Xv02xF_UEvw/view?usp=sharing>

Các tài liệu Tiếng Anh

1. Tài liệu về EJS: <https://ejs.co/#docs>
2. Tài liệu về VueJS: <https://vuejs.org/v2/guide/>
3. Tài liệu về Google Firebase (Authentication): <https://firebase.google.com/docs/auth?authuser=0>
4. Tài liệu về Heroku dùng để deploy website: <https://devcenter.heroku.com/articles/getting-started-with-nodejs>
5. Tài liệu về Sequelize: <https://sequelize.org/master/>

# PHỤ LỤC