

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

**xây dựng website bán vé tàu online**

Giảng viên hướng dẫn: ThS. Đặng Thị Thu Hà

Sinh viên thực hiện: Lê Hoàng – 16021961

Cao Quốc Đông – 16016211

**TP Hồ CHí MInh ,Tháng ….. Năm 20……..**

**GIÁO VIÊN HƯỚNG DẪN ĐÁNH GIÁ**

**GIÁO VIÊN PHẢN BIỆN ĐÁNH GIÁ**

# MỤC LỤC

[MỤC LỤC 3](#_Toc39960503)

[DANH MỤC CÁC HÌNH VẼ 4](#_Toc39960504)

[DANH MỤC CÁC BẢNG BIỂU 6](#_Toc39960505)

[LỜI MỞ ĐẦU 7](#_Toc39960506)

[CHƯƠNG 1 : GIỚI THIỆU 8](#_Toc39960507)

[1.1 Tổng quan 8](#_Toc39960508)

[1.2 Mục tiêu cơ bản đề tài 8](#_Toc39960509)

[1.3 Phạm vi đề tài 8](#_Toc39960510)

[1.4 Mô tả yêu cầu chức năng 8](#_Toc39960511)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 9](#_Toc39960512)

[CHƯƠNG 3 : PHÂN TÍCH 10](#_Toc39960513)

[3.1 Mô hình Use case tổng quát 10](#_Toc39960514)

[3.2 Danh sách các tác nhân và mô tả 10](#_Toc39960515)

[3.3 Danh sách Use case và mô tả 10](#_Toc39960516)

[3.4 Đặc tả các yêu cầu chức năng 11](#_Toc39960517)

[3.4.1 UC001\_Đăng nhập 11](#_Toc39960518)

[CHƯƠNG 4 : THIẾT KẾ VÀ HIỆN THỰC 14](#_Toc39960519)

[4.1 ClassDiagram 14](#_Toc39960520)

[4.2 Sitemap 14](#_Toc39960521)

[4.3 Database Diagram 14](#_Toc39960522)

[4.4 Mô tả Website 14](#_Toc39960523)

[4.4.1 Một số chức năng và giao diện chính 14](#_Toc39960524)

[CHƯƠNG 5 : KẾT LUẬN 15](#_Toc39960525)

[5.1 Kết quả đạt được 15](#_Toc39960526)

[5.2 Hạn chế của đồ án 15](#_Toc39960527)

[5.3 Hướng phát triển 15](#_Toc39960528)

[TÀI LIỆU THAM KHẢO 16](#_Toc39960529)

[PHỤ LỤC 17](#_Toc39960530)

# DANH MỤC CÁC HÌNH VẼ

[Hình 3‑2 Activity Diagram UC001 – Đăng nhập 10](#_Toc39960531)

[Hình 3‑3 Sequence Diagram UC001 – Đăng nhập 11](#_Toc39960532)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑2 Danh sách các usecase và mô tả tóm tắt Usecase 9](#_Toc39960533)

[Bảng 3‑3 Mô tả UC001 – Đăng nhập 10](#_Toc39960534)

# LỜI MỞ ĐẦU

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

* Cách mạng Công nghiệp 4.0 đang diễn ra tại nhiều nước phát triển như là Việt Nam ta dẫn đến việc số hoá các dịch vụ là điều cần thiết bởi sự tiện lợi mà nó mang lại cho cả người mua và người bán.
* Lĩnh vực được nghiên cứu thông qua tình hình thực tiễn tại các nhà ga mỗi khi Tết đến xuân về hay các dịp lễ lớn trong năm, nhằm hướng đến việc cung cấp giải pháp phù hợp với thực trạng đấy.
* Mục đích, đối tượng & phạm vi của đề tài bao gồm phía hành khách và nhân viên nhà ga.

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

* Về mặt ý nghĩa khoa học, đề tài này giúp giải quyết những vấn đề khi mua vé truyền thống như: chen lấn, thời gian chờ đợi lâu, đợi đến lượt mình thì hết vé, rủi ro về mặt giao thông trên đường (kẹt xe, tai nạn…). Với dịch vụ bán vé tàu trực tuyến, các vấn đề trên có thể được giải quyết một cách hiệu quả cũng như là giúp bên bán và bên mua thực hiện toàn bộ quá trình giao dịch thông qua website.
* Về mặt ý nghĩa thực tiễn, chính nhờ việc toàn bộ quá trình đều được thực hiện thông qua website mà nó giúp cho người mua và người bán tiện lợi hơn so với hình thức bán vé truyền thống. Người mua giờ đây có thể bỏ qua những bất lợi kể trên chỉ với một chiếc máy tính để bàn hay laptop, thậm chí là điện thoại miễn là có kết nối internet, thì cũng đã có thể mua vé tàu, thậm chí nếu có thẻ ngân hàng thì họ có thể thanh toán ngay mà không cần phải thanh toán trực tiếp. Đối với bên bán vé thì chỉ cần một ít thông tin cá nhân mà khách hàng đã cung cấp từ trước là đã có thể in vé cho khách đã đặt trực tuyến đến thanh toán tại nhà ga.

# : GIỚI THIỆU

## Tổng quan

* Ngành công nghệ thông tin là một ngành khoa học đang trên ngày càng phát triển mạnh và ứng dụng rộng rãi trên nhiều lĩnh vực trong thời kỳ công nghiệp 4.0 hiện nay.
* Việc đi lại bằng tàu hoả ở nước ta vẫn luôn phổ biến từ trước đến nay và nhu cầu của nó ngày càng tăng khiến cho hình thức mua vé truyền thống khó có thể thoả mãn khách hàng. Và để đáp ứng nhu cầu đó của khách hng, website bán vé tàu online ra đời là nhu cầu tất yếu.
* Website bán vé tàu online được tạo ra nhằm đáp ứng một mục đích duy nhất là tạo sự tự do và nhanh chóng cho hành khách muốn mua vé. Họ chỉ cần có một số thứ như là email, thẻ ATM và các giấy tờ tuỳ thân để việc đặt vé diễn ra nhanh chóng và thuận tiện nhất.

## Mục tiêu cơ bản đề tài

* Tăng sự tiện lợi khi đặt mua vé tàu, giảm thiểu việc tiêu tốn thời gian.
* Giúp bên bán vé dễ dàng quản lý được doanh thu cũng như là đơn giản hoá thủ tục bán vé cho hành khách.

## Phạm vi đề tài

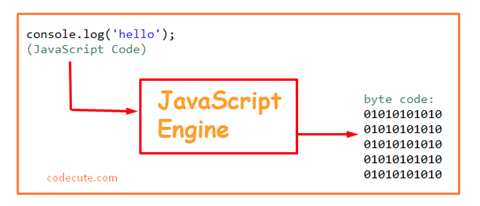
* Website bán vé tàu online gồm các chức năng:
* Tìm kiếm thông tin lịch trình.
* Đặt vé/mua vé.
* Tra cứu thông tin đặt vé.
* Tra cứu giá vé.
* In vé.
* Huỷ vé.
* Thống kê số vé bán được.
* Thống kê doanh thu.
* Lên lịch trình.
* Kỹ thuật thực hiện :
* NodeJS.
* EJS.
* VueJS.
* Tedious.
* Sequelize.
* Sql Server.
* Google Firebase.

## 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).
  + Đặt vé (Hành khách).
  + Tra thông tin vé đã đặt (Hành khách).
  + Tra tuyến theo lịch trình đã có (Hành khách).
  + In vé (Nhân viên bán vé).
  + Huỷ vé (Nhân viên bán vé).
  + Thống kê doanh thu và lượng vé đã bán (Nhân viên quản lý).
  + Lên lịch trình (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 hành trình (Nhân viên quản lý).
  + Tra vé theo mã đặt vé (Nhân viên bán vé).

# : CƠ SỞ LÝ THUYẾT

* **NodeJS:**
* **Chrome V8**, hay đơn giản là **V8**, là một **JavaScript engine (JavaScript engine là một chương trình hoặc trình thông dịch thực thi mã Javascript)** mã nguồn mở phát triển bởi The Chromium Project. Đây là engine được thiết kế đặc biệt và tối ưu hóa hiệu năng cho các dự án Javascript lớn, như Node.js. Node.js là một mã nguồn được xây dựng dựa trên nền tảng **Javascript V8 Engine**, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp.



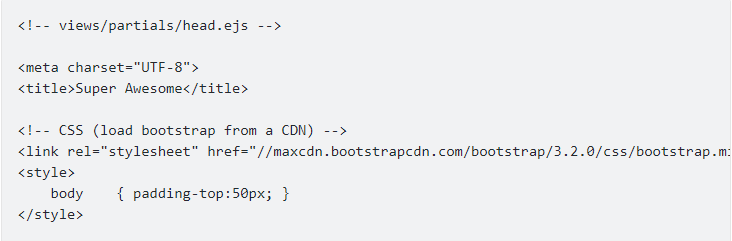
* + NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.
  + Tốc độ xử lý các giao tiếp từ client tới server gần như đạt đến mức real-time nên việc lựa chọn sử dụng NodeJS để xây dựng website bán vé tàu với lượt truy cập cao là vô cùng hợp lý, đặc biệt là trong quá trình hành khách chọn ghế.
  + Vì sao nên dùng **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**: NodeJ 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**: Hãy so sánh với kỹ thuật web-serving truyền thống: mỗi kết nối (request) sinh ra một thread (luồng xử lý), chiếm một lượng bộ nhớ RAM hệ thống và dần dần chiếm hết dung lượng RAM khả dĩ. Node.js không như vậy. Nó chỉ chạy trên 1 thread duy nhất, sử dụng lời gọi non-blocking I/O, cho phép nó xử lý được cả ngàn kết nối cùng lúc.

A screen shot of a smart phone

Description automatically generated

* + - **Không đệm**: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
    - **Có giấy phép**: NodeJS đã được cấp giấy phép bởi MIT License.
* **EJS:**
  + **EJS** là viết tắt của **“Embeded Javascript templating”** –một thư viện được sử dụng để phân tích các tập tin ejs và tạo ra trang html trả về cho trình duyệt (client), sử dụng nó bằng cách đặt mã JavaScript bên trong <%%> khi khai báo biến, <%=%> khi muốn hiển thị bến lên html và <%-%> khi muốn nhúng một file EJS khác vào.

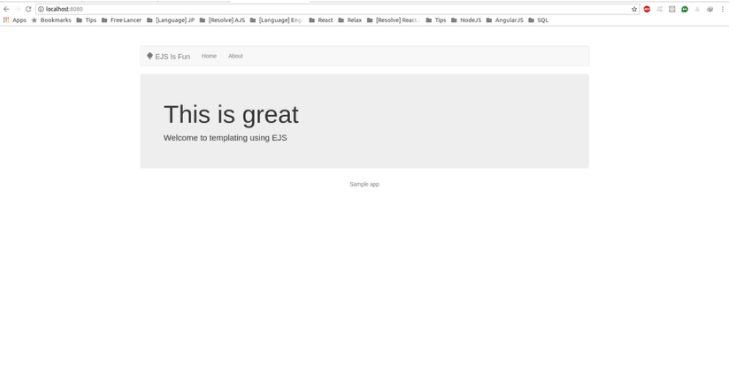
**Ví dụ:** Tạo thư mục views/partials và sau đó thêm các file như hình bên dưới



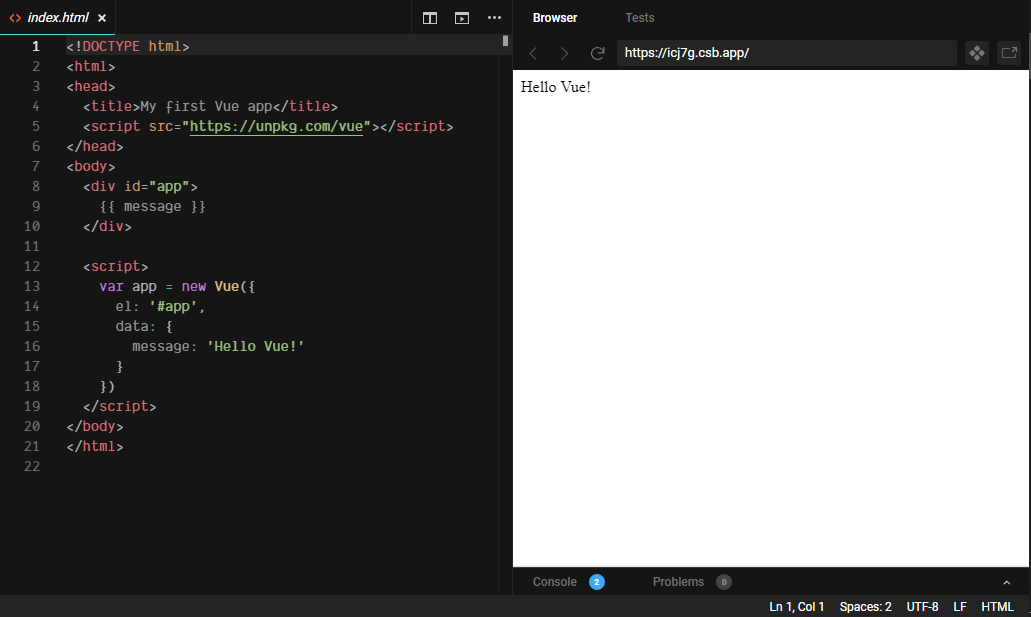




Sau khi thêm các tệp trên, refresh lại localhost:8080 và ta có kết quả sau:



* + Vì sao nên dùng **EJS**:
    - Sử dụng ngôn ngữ thuần JavaScript nên nó giúp cho những lập trình viên có kiến thức về JavaScript dễ tiếp cận hơn.
    - Cú pháp đơn giản là một thế mạnh của EJS, điều này khiến cho người mới dễ tiếp cận hơn, rút ngắn thời gian phát triển ứng dụng.
    - Thời gian thực thi ngắn nhờ vào việc lưu trữ các hàm JavaScript trung gian.
    - EJS có một cơ chế xử lý lỗi thực sự thông minh được tích hợp ngay trong nó. Nó chỉ ra cho bạn số dòng mà trên đó đã xảy ra lỗi để bạn không phải xem toàn bộ tệp mẫu làm lãng phí thời gian của bạn trong việc tìm kiếm lỗi
* **VueJS:**
  + **VueJS** là một **framework linh động** (nguyên bản tiếng Anh: progressive – tiệm tiến) dùng để xây dựng giao diện người dùng (user interfaces). Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước. 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. Cùng lúc đó, nếu kết hợp với những kĩ thuật hiện đại như [**SFC (single file components)**](https://vi.vuejs.org/v2/guide/single-file-components.html) và [**các thư viện hỗ trợ**](https://github.com/vuejs/awesome-vue#components--libraries), 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.
  + **Vì sao nên dùng VueJS**:
    - **Gần gũi**: đối với lập trình viên đã có kiến thức về HTML, CSS, JavaScript sẽ dễ dàng tiếp cận với VueJS.

****

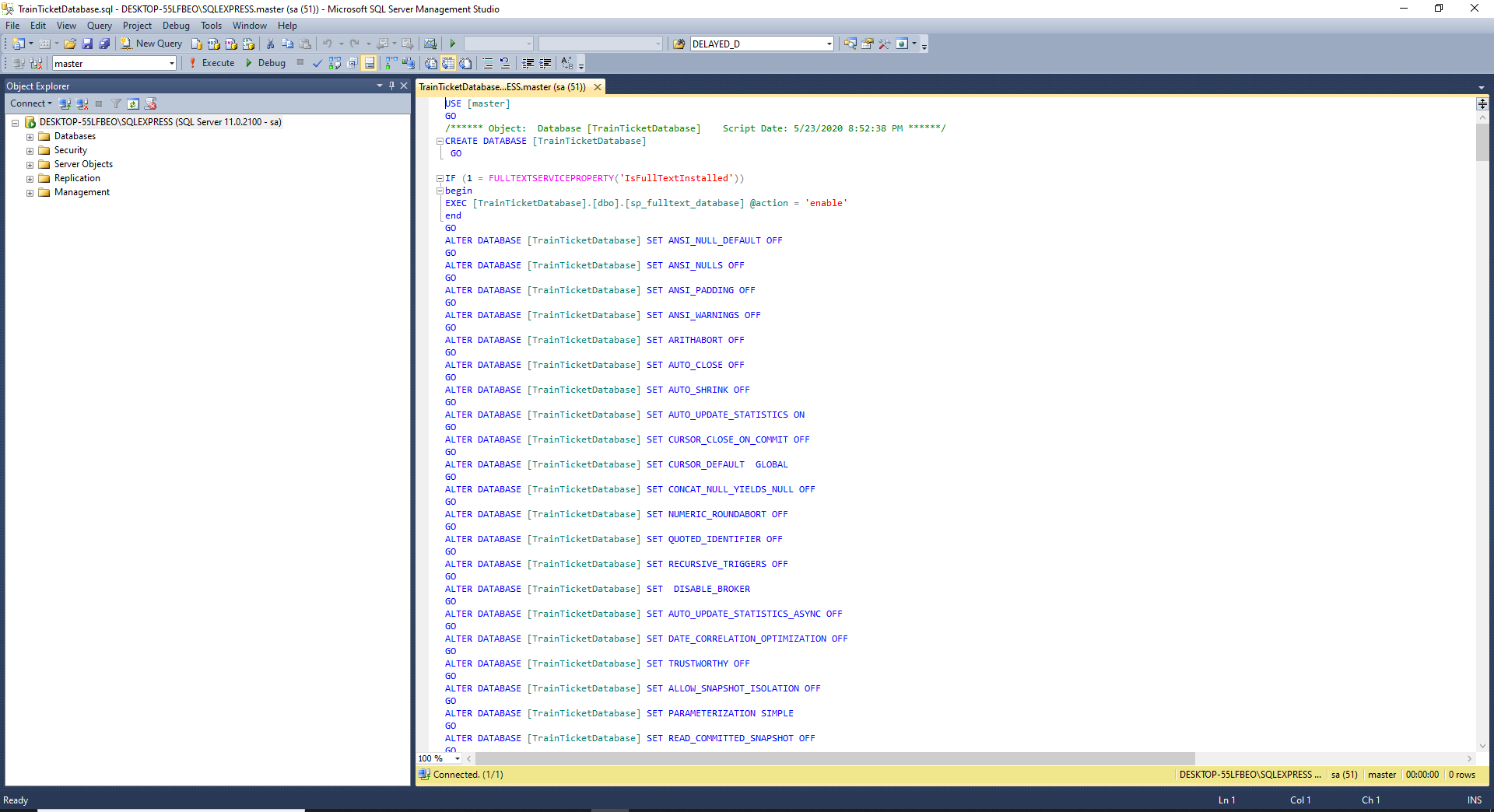
* + - **Đa năng**: VueJS sở hữu một hệ sinh thái linh hoạt từ cơ bản đến nâng cao, từ một thư viện đơn giản đến một framework với đầy đủ các tính năng.
    - **Mạnh mẽ**: dung lượng yêu cầu cực thấp, xử lý cực nhanh với Virtual DOM cũng như là dễ dàng tối ưu.
    - **Nhỏ gọn:** Vue.jscó kích thước nhỏ gọn. Kích thước của framework này là 18-21KB và người dùng không mất thời gian tải xuống và sử dụng nó. Điều này không có nghĩa là nó có tốc độ thấp vì kích thước nhỏ. Thay vào đó, nó đánh bại tất cả các framework cồng kềnh như React.
    - **Dễ phát triển:** Trong trường hợp có bất kỳ sự cố nào, người dùng có thể dễ dàng theo dõi các khối có lỗi. Tất cả điều này là do cấu trúc đơn giản của nó.
* **Sequelize & Tedious:**
  + **Sequelize** là một ORM (Object Relational Mapping) NodeJS dựa trên promise. Nó có thể được sử dụng với PostgreSQL, MySQL, MariaDB, SQLite và MSSQL. Mục đích chính của Sequelize nhằm ánh xạ các đối tượng với các bảng của database.

****

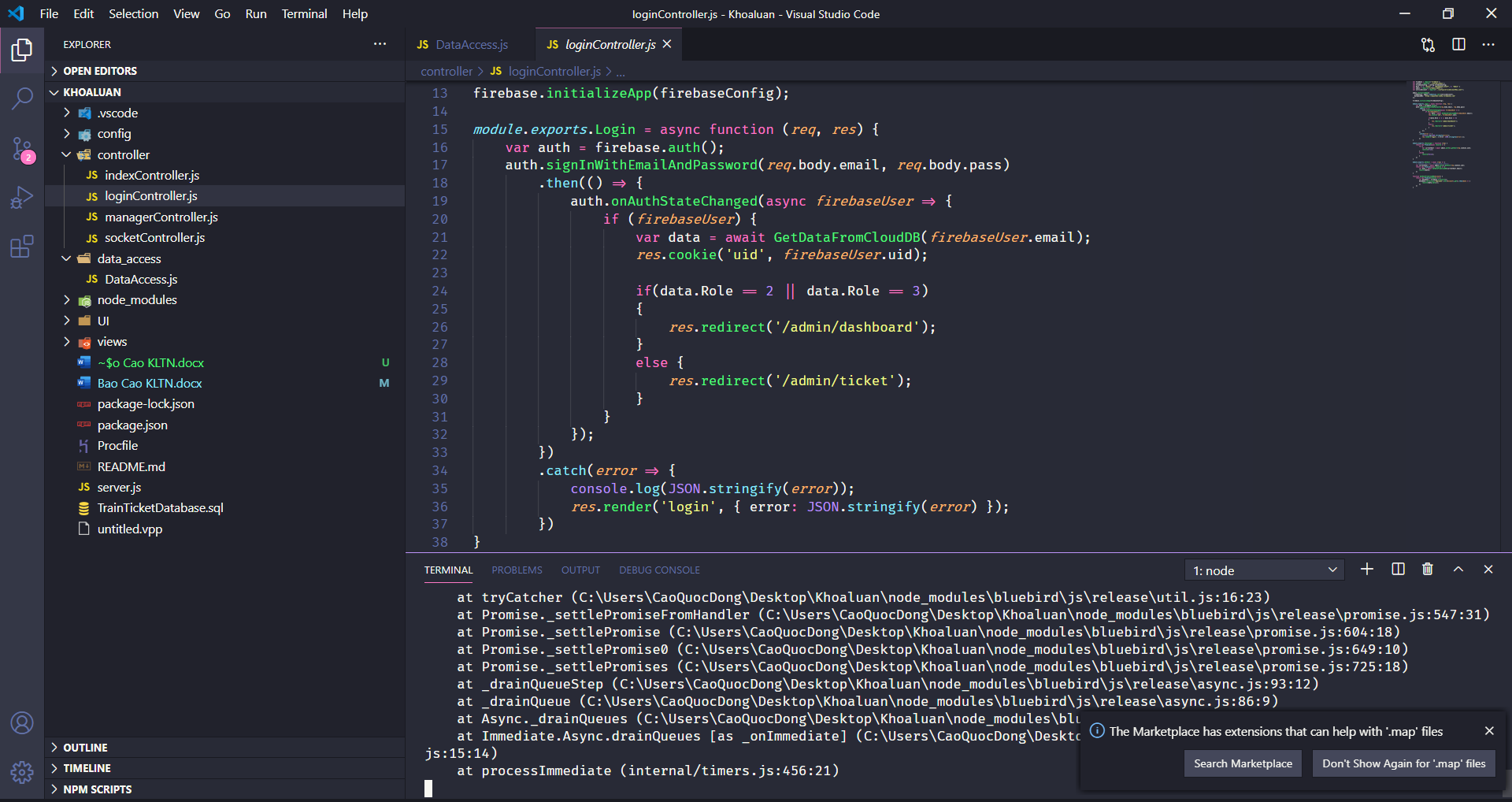
* + **Tedious** là một triển khai thuần tuý JavaScript của giao thức TDS (Tabular Data Stream), được sử dụng để tương tác với MSSQL, đồng thời cung cấp khả năng xác thực và mã hoá.



* **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** 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 tốt với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server….



* + Vì sao nên chọn **SQL Server:**
    - Các truy vấn SQL đơn giản có thể được sử dụng để lấy một lượng lớn dữ liệu từ cơ sở dữ liệu rất nhanh chóng và hiệu quả.
    - SQL rất dễ học và hầu hết mọi DBMS đều hỗ trợ SQL.
    - Việc quản lý cơ sở dữ liệu bằng SQL sẽ dễ dàng hơn vì không cần phải viết nhiều mã.
* **Google Firebase:**



* + **Firebase** là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp lập trình viên đơn giản hóa các thao tác với cơ sở dữ liệu, cụ thể là những giao diện lập trình ứng dụng API đơn giản. Đặc biệt Firebase còn là dịch vụ đa năng và bảo mật cực tốt, hỗ trợ cả 2 nền tảng Android và iOS.
  + Hoạt động nổi bật của **Firebase** là xây dựng các bước xác thực người dùng bằng Email, Facebook, Twitter, GitHub, Google. Đồng thời cũng xác thực nặc danh cho các ứng dụng. Hoạt động xác thực có thể giúp thông tin cá nhân của người sử dụng được an toàn và đảm bảo không bị đánh cắp tài khoản.
  + Vì sao nên chọn **Firebase**:
    - Xây dựng ứng dụng nhanh chóng mà không tốn thời gian, nhân lực để quản lý hệ thống và cơ sơ sở hạ tầng phía sau: Firebase cung cấp cho bạn chức năng như phân tích, cơ sở dữ liệu, báo cáo hoạt động và báo cáo các sự cố lỗi để bạn có thể dễ dàng phát triển, định hướng ứng dụng của mình vào người sử dụng nhằm đem lại các trải nghiệm tốt nhất cho họ.
    - Quản lý cấu hình và trải nghiệm các ứng dụng của Firebase tập trung trong một giao diện website đơn giản, các ứng dụng này hoạt động độc lập nhưng liên kết dữ liệu phân tích chặt chẽ.
    - Uy tín chất lượng đảm bảo từ Google: Firebase được Google hỗ trợ và cung cấp trên nền tảng phần cứng với quy mô rộng khắp thế giới, được các tập đoàn lớn và các ưng dụng với triệu lượt sử dụng từ người dùng

# : PHÂN TÍCH

## Mô hình Use case tổng quát

Diagram

Description automatically generated

Hình 3-1 Mô hình Use Case Tổng Quát

## Danh sách các tác nhân và mô tả

* **Hành khách**: Là đối tượng sử dụng website chính, có thể thực hiện các chức năng bao gồm: mua vé, tìm kiếm lịch trình, tra thông tin vé đã đặt và tra cứu giá vé.
* **Nhân viên bán vé**: Là các nhân viên ngồi tại quầy bán vé ở nhà ga, có thể thực hiện các chức năng bao gồm: in vé và huỷ vé.
* **Nhân viên quản lý**: Là người quản lý ở nhà ga, có thể thực hiện các chức năng bao gồm: lên lịch trình, thống kê doanh thu và lượng vé bán được theo năm, theo tháng và quản lý hành khách theo hành trình.

## 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 | Hành khách nhập các thông tin vào các trường có sẵn để tìm kiếm lịch trình cần đặt vé. | Tìm kiếm lịch trình | Hành khách không cần đăng nhập vào hệ thống để thực hiện chức năng này. |
| UC002 | Đặt vé | Đặt vé tàu dựa trên thông tin đã nhập từ hành khách | Đặt vé | Khách hàng sau khi tìm kiếm và chọn lịch trình phù hợp mới có thể đặt vé. Chức năng này không cần đăng nhập. |
| UC003 | Tra thông tin vé đã đặt | Sau khi đã đặt vé thì hành khách có thể dùng chức năng này để xem lại thông tin trên vé của mình. | Tra thông tin đặt vé | Hành khác sử dụng mã đặt vé được gửi về email để sử dụng chức năng này, không cần đăng nhập. |
| UC004 | Tra tuyến theo lịch trình đã có | Hiển thị tất cả lịch trình có ngày lớn hơn hoặc bằng ngày hiện tại. | Tra tuyến theo lịch trình đã có | Hành khách không cần đăng nhập để thực hiện chức năng này. |
| UC005 | In vé | Nhân viên bán vé dùng mã đặt vé để in vé cho khách hàng | In vé | Nhân viên bán vé cần đăng nhập để thực hiện chức năng này. |
| UC006 | Huỷ vé | Nhân viên bán vé thực hiện chức năng để huỷ vé cho hành khách | Huỷ vé | Nhân viên bán vé cần đăng nhập để thực hiện chức năng này. |
| UC007 | Thống kê doanh thu và lượng vé đã bán | Thống kê doanh thu và lượng vé bán được theo năm/tháng. | Thống kê doanh thu và lượng vé đã bán | Nhân viên quản lý cần đăng nhập tài khoản admin để thực hiện chức năng này. |
| UC008 | Lên lịch trình | Nhân viên quản lý lên lịch trình cho các tàu đang trống lịch. | Lên lịch trình | Nhân viên quản lý cần đăng nhập tài khoản admin để thực hiện chức năng này. |
| UC009 | Đăng nhập | Nhân viên bán vé và nhân viên quản lý thực hiện đăng nhập vào hệ thống. | Đăng nhập | Truy cập vào trang web phía quản lý để đăng nhập vào hệ thống. |
| UC010 | Lấy danh sách vé chờ huỷ | Hệ thống thực hiện chức năng tự động khi có yêu cầu lấy danh sách vé chờ huỷ. | Lấy danh sách vé chờ huỷ | Khi nhân viên sử dụng chức năng “huỷ vé” sẽ kích hoạt hệ thống thực hiện chức năng này. |
| UC011 | Quản lý hành khách theo hành trình | Quản lý thông tin của hành khách theo hành trìnnh | Quản lý hành khách theo hành trình | Nhân viên quản lý cần đăng nhập tài khoản admin để thực hiện chức năng này. |
| UC012 | Tra vé theo mã đặt vé | Nhân viên bán vé dùng mã đặt vé để để tìm kiếm vé cho hành khách | Tra vé theo mã đặt vé | Nhân viên cần đăng nhập để thực hiện chức năng này. |

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 tìm kiếm lịch trình để đặt vé | |
| Tác nhân: | Hành khách | |
| Điều kiện trước: | Truy cập vào website | |
| Điều kiện sau: | Hiển thị được danh sách các lịch trình phù hợp với thông tin đầu vào. | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
|  | |  |
| 1. Truy cập vào hệ thống | |  |
|  | | 1. Hiển thị trang chủ. |
| 1. Nhập thông tin lịch trình rồi nhấn tìm kiếm | |  |
|  | | 1. Hệ thống tìm kiếm lịch trình dựa trên input |
|  | | 1. Hiển thị kết quả tìm được lên màn hình. |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 5.1 Nếu không có kết quả phù hợp hệ thống tự chuyển qua trang thông báo không tìm thấy kết quả. | | |

Bảng 3‑2 Mô tả UC001 – Tìm kiếm lịch trình

#### Biểu đồ

##### Activity Diagram

Diagram, schematic

Description automatically generated

Hình 3‑2 Activity Diagram UC001 – Tìm kiếm lịch trình

##### Sequence Diagram

### UC002\_Đặt vé

#### Mô tả use case UC002

|  |  |  |
| --- | --- | --- |
| **Use case:** Đặt vé | | |
| Mục đích: | Đặt vé | |
| Mô tả: | Giúp hành khách đặt vé trực tuyến | |
| Tác nhân: | Hành khách | |
| Điều kiện trước: | Truy cập vào website và tìm kiếm lịch trình hoặc tra giá vé | |
| Điều kiện sau: | Lưu trữ được dữ liệu vào database, hiển thị thông báo mua vé thành công và có thể sử dụng chức năng tra thông tin vé đã đặt. | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
|  | |  |
| 1. Hành khách chọn lịch trình, chọn loại ghế và nhấn đặt vé. | |  |
|  | | 1. Hệ thống hiển thị thông tin của chuyến đi. |
| 1. Hành khách nhập thông tin cá nhân, chọn ghế và nhấn thanh toán. | |  |
|  | | 1. Hệ thống tính tổng tiền, lưu trữ các objects cần thiết vào session hiện tại sau đó hiển thị trang thanh toán. |
| 1. Hành khách chọn phương thức thanh toán sau đó nhấn nút thanh toán. | |  |
|  | | 1. Hệ thống điều hướng khách hàng sang trang thanh toán của Ngân Lượng. |
| 1. Hành khách tiến hành thanh toán qua Ngân Lượng. | |  |
|  | | 1. Ngân Lượng điều hướng hành khách về lại trang bán vé, dữ liệu từ session được lưu vào database. Hiển thị thông tin đặt vé và gửi mail thông tin đặt vé cho hành khách. |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 8.1 Nếu thanh toán thành công nhưng thông tin vé của hành khách không lưu được thì hệ thống vẫn sẽ gửi mail về cho họ để sử dụng cho việc hoàn tiền hoặc tiếp tục mua vé. | | |

Bảng 3‑3 Mô tả UC002 - Đặt vé

#### Biểu đồ

##### Activity Diagram

Diagram

Description automatically generated

Hình 3‑4 Activity Diagram UC002 – Đặt vé

##### Sequence Diagram

### UC003\_Tra thông tin vé đã đặt

#### Mô tả use case UC003

|  |  |  |
| --- | --- | --- |
| **Use case:** Tra thông tin vé đã đặt | | |
| Mục đích: | Tra thông tin vé đã đặt | |
| Mô tả: | Giúp hành khách tìm kiếm lịch trình để đặt vé | |
| Tác nhân: | Hành khách | |
| Điều kiện trước: | Truy cập vào website | |
| Điều kiện sau: | Hiển thị được danh sách các lịch trình phù hợp với thông tin đầu vào. | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
|  | |  |
| 1. Nhấn “Kiểm tra vé” | |  |
|  | | 1. Hiện trang nhập thông tin |
| 1. Người dùng nhập mã đặt vé và email rồi nhấn tìm kiếm. | |  |
|  | | 1. Hệ thống lấy thông tin người đặt vé và các hành khách hiển thị lên màn hình. |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 4.1 Hiển thị thông báo nếu không tìm thấy thông tin đặt vé. | | |

Bảng 3‑6 Mô tả UC003 – Tra thông tin vé đã đặt

#### Biểu đồ

##### Activity Diagram

Diagram

Description automatically generated

Hình 3‑6 Activity Diagram UC003 – Tra thông tin vé đã đặt

##### Sequence Diagram

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

#### Mô tả use case UC004

|  |  |  |
| --- | --- | --- |
| **Use case:** Tra tuyến theo lịch trình đã có | | |
| Mục đích: | Tra tuyến theo lịch trình đã có. | |
| Mô tả: | Giúp hành khách tra giá vé và đặt vé. | |
| Tác nhân: | Hành khách. | |
| Điều kiện trước: | Truy cập vào website. | |
| 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 “Danh sách lịch trình” | |  |
|  | | 1. Hiển thị danh sách lịch trình có ngày lớn hơn hoặc bằng ngày hiện tại |
| 1. Nhấn vào lịch trình tuỳ chọn | |  |
|  | |  |
|  | | 1. Hiển thị toàn bộ chuyến và giá của lịch trình đó |
| Luồng sự kiện phụ (Alternative Flows) : | | |
|  | | |

Bảng 3‑5 Mô tả UC004 – Tra tuyến theo lịch trình đã có

#### Biểu đồ

##### Activity Diagram

Diagram, schematic

Description automatically generated

Hình 3‑8 Activity Diagram UC004 – Tra tuyến theo lịch trình đã có

##### Sequence Diagram

### UC005\_In vé

#### Mô tả use case UC005

|  |  |  |
| --- | --- | --- |
| **Use case:** In vé | | |
| Mục đích: | In vé cho hành khách | |
| Mô tả: | Giúp nhân viên bán vé in vé cho hành khách đã thanh toán | |
| Tác nhân: | Nhân viên bán vé | |
| Điều kiện trước: | Thông tin vé hợp lệ | |
| Điều kiện sau: | In được vé cho hành khách, chuyển đổi trạng thái của vé sang đã in | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
|  | |  |
| 1. Chọn “In vé” | |  |
|  | | 1. Hiện thông báo xác nhận |
| 1. Chọn “Yes” | |  |
|  | | 1. In vé |
|  | | 1. Chuyển trạng thại vé sang “Đã in” |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| * 1. Chọn “No”   2. Hệ thống quay lại trang thông tin vé | | |

Bảng 3‑6 Mô tả UC005 – In vé

#### Biểu đồ

##### Activity Diagram

Diagram

Description automatically generated

Hình 3‑10 Activity Diagram UC005 – In vé

##### Sequence Diagram

### UC006\_Huỷ vé

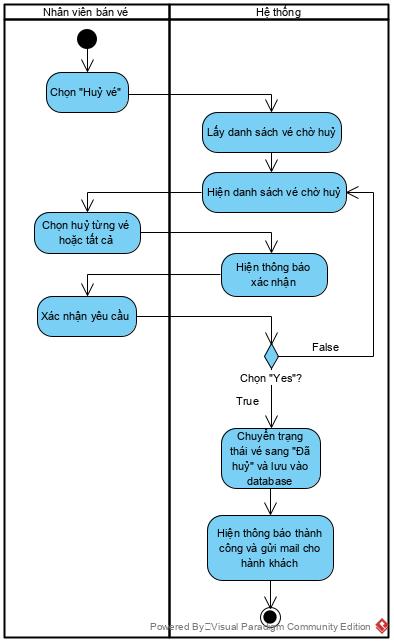
#### Mô tả use case UC006

|  |  |  |
| --- | --- | --- |
| **Use case:** Huỷ vé | | |
| Mục đích: | Huỷ vé cho hành khách | |
| Mô tả: | Giúp nhân viên bán vé huỷ vé cho khách đã thanh toán | |
| Tác nhân: | Nhân viên bán vé | |
| Điều kiện trước: | Thông tin vé hợp lệ | |
| Điều kiện sau: | Huỷ vé thành công, chuyển đổi trạng thái vé sang huỷ | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
|  | |  |
| 1. Chọn “Huỷ vé” | |  |
|  | | 1. Lấy danh sách vé chờ huỷ |
|  | | 1. Hiện danh sách vé chờ huỷ |
| 1. Chọn huỷ từng vé hoặc tất cả vé | |  |
|  | | 1. Hiện thông báo xác nhận |
| 1. Chọn “Yes” | |  |
|  | | 1. Chuyển đổi trạng thái vé sang “Đã huỷ”, lưu thông tin vào database |
|  | | 1. Hiện thông báo thành công và gửi mail về khách hàng |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 6.1 Chọn “No”  6.2 Hệ thống quay lại trang hiển thị danh sách vé chờ huỷ | | |

Bảng 3‑7 Mô tả UC006 – Huỷ vé

#### Biểu đồ

##### Activity Diagram



Hình 3‑12 Activity Diagram UC006 – Huỷ vé

##### Sequence Diagram

### UC007\_Thống kê doanh thu và lượng vé đã bán

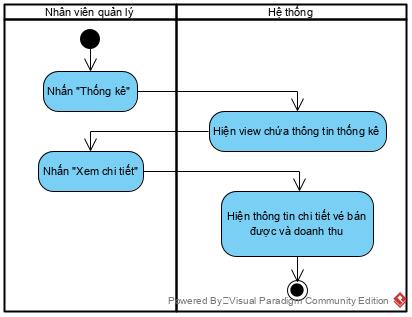
#### Mô tả use case UC007

|  |  |  |
| --- | --- | --- |
| **Use case:** Thống kê doanh thu và lượng vé đã bán | | |
| Mục đích: | Thống kê doanh thu và lượng vé nhà ga bán được | |
| Mô tả: | Giúp quản lý thống kê và kiểm soát danh thu cũng như là lượng vé bán ra | |
| Tác nhân: | Nhân viên quản lý | |
| Điều kiện trước: | Đã đăng nhập vào hệ thống bằng tài khoản admin | |
| Điều kiện sau: | Hiển thị được các thông tin cần thống kê | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
|  | |  |
| 1. Nhấn “Thống kê” | |  |
|  | | 1. Hiện view chứa thông tin thống kê |
| 1. Nhấn “Xem chi tiết” | |  |
|  | |  |
|  | | 1. Hiện thông tin chi tiết vé bán được và doanh thu |
| Luồng sự kiện phụ (Alternative Flows) : | | |
|  | | |

Bảng 3‑8 Mô tả UC007 – Thống kê doanh thu và lượng vé đã bán

#### Biểu đồ

##### Activity Diagram



Hình 3‑14 Activity Diagram UC007 – Thống kê doanh thu và lượng vé đã bán

##### Sequence Diagram

### UC008\_Lên lịch trình

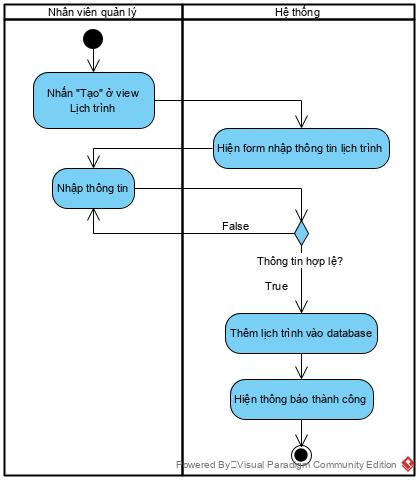
#### Mô tả use case UC008

|  |  |  |
| --- | --- | --- |
| **Use case:** Lên lịch trình | | |
| Mục đích: | Lên lịch trình cho các chuyến tàu | |
| Mô tả: | Giúp quản lý lên lịch trình cho các chuyến tàu nhằm bán vé cho hành khách | |
| Tác nhân: | Nhân viên quản lý | |
| Điều kiện trước: | Đã đăng nhập vào hệ thống bằng tài khoản admin | |
| Điều kiện sau: | Thêm lịch trình chính và các lịch trình phụ vào hệ thống thành công | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
| 1. Nhấn “Tạo” ở view Lịch trình | |  |
|  | | 1. Hiện form nhập thông tin lịch trình |
| 1. Nhập thông tin lịch trình | |  |
|  | | 1. Kiểm tra hợp lệ |
|  | | 1. Thêm dữ liệu lịch trình vào database |
|  | | 1. Hiện thông báo thành công |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 4.1 Thông tin không hợp lệ thì quay lại bước 3 | | |

Bảng 3‑9 Mô tả UC008 – Lên lịch trình

#### Biểu đồ

##### Activity Diagram



Hình 3‑16 Activity Diagram UC008 – Lên lịch trình

##### Sequence Diagram

### UC009\_Đăng nhập

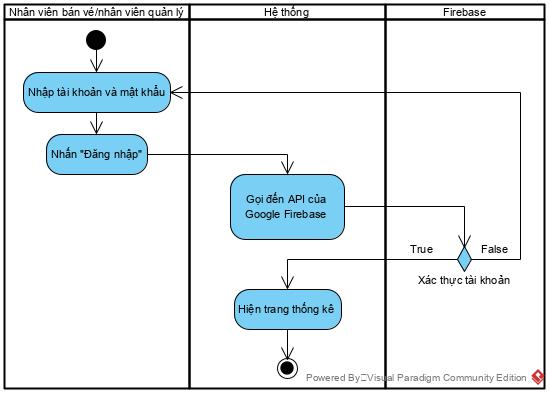
#### Mô tả use case UC009

|  |  |  |
| --- | --- | --- |
| **Use case:** Đăng nhập | | |
| Mục đích: | Đăng nhập vào hệ thống | |
| Mô tả: | Giúp nhân viên bán vé và nhân viên quản lý đăng nhập vào hệ thống để thực hiện các chức năng của mình | |
| Tác nhân: | Nhân viên bán vé, nhân viên quản lý | |
| Điều kiện trước: | Hệ thống chưa được đăng nhập | |
| Điều kiện sau: | Chuyển về trang thống kê của hệ thống quản lý | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
| 1. Nhập tài khoản và mật khẩu | |  |
| 1. Nhấn đăng nhập | |  |
|  | | 1. Kết nối đến Firebase để xác thực và lấy thông tin |
|  | | 1. Hiện trang thống kê |
|  | |  |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 3.1 Xác thực không thành công sẽ hiện thông báo lỗi, quay lại bước 1 | | |

Bảng 3‑10 Mô tả UC009 – Đăng nhập

#### Biểu đồ

##### Activity Diagram



Hình 3‑18 Activity Diagram UC009 – Đăng nhập

##### Sequence Diagram

### UC010\_Lấy danh sách vé chờ huỷ

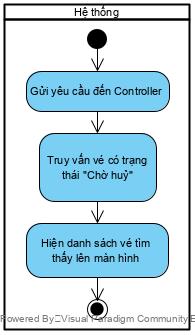
#### Mô tả use case UC010

|  |  |  |
| --- | --- | --- |
| **Use case:** Lấy danh sách vé chờ huỷ | | |
| Mục đích: | Lấy danh sách vé chờ để huỷ nhằm phục vụ yêu cầu của khách hàng | |
| Mô tả: | Giúp lấy danh sách vé chờ huỷ từ database | |
| Tác nhân: | Hệ thống | |
| Điều kiện trước: | Khi người dùng nhấn vào “Huỷ vé” | |
| Điều kiện sau: | Hiển thị được danh sách các vé chờ huỷ thành công | |
| Luồng sự kiện chính (Basic flows): | | |
| **System** | | |
|  | |  |
| 1. Gửi yêu cầu đến Controller | | |
| 1. Truy vấn vé có trạng thái “Chờ huỷ” | | |
| 1. Hiện danh sách các vé có trạng thái “Chờ huỷ” lên màn hình | | |
|  | |  |
| Luồng sự kiện phụ (Alternative Flows) : | | |
|  | | |

Bảng 3‑11 Mô tả UC010 – Lấy danh sách vé chờ huỷ

#### Biểu đồ

##### Activity Diagram



Hình 3‑20 Activity Diagram UC010 – Lấy danh sách vé chờ huỷ

##### Sequence Diagram

### UC011\_Quản lý hành khách theo hành trình

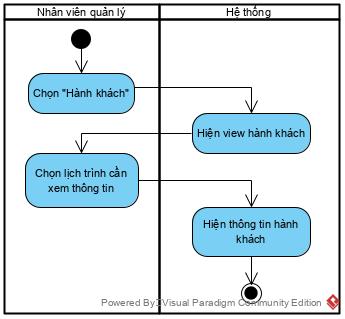
#### Mô tả use case UC011

|  |  |  |
| --- | --- | --- |
| **Use case:** Quản lý hành khách theo hành trình | | |
| Mục đích: | Quản lý thông tin hành khách theo hành trình | |
| Mô tả: | Giúp quản lý kiểm soát thông tin của hành khách theo hành trình | |
| Tác nhân: | Nhân viên quản lý | |
| Điều kiện trước: | Đã đăng nhập vào hệ thống bằng tài khoản admin | |
| Điều kiện sau: | Hiển thị thành công các thông tin của hành khách | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
|  | |  |
| 1. Chọn “Hành khách” | |  |
|  | | 1. Hiện view hành khách |
| 1. Chọn lịch trình cần xem thông tin | |  |
|  | | 1. Hiện thông tin hành khách |
| Luồng sự kiện phụ (Alternative Flows) : | | |
|  | | |

Bảng 3‑12 Mô tả UC011 – Quản lý hành khách theo hành trình

#### Biểu đồ

##### Activity Diagram



Hình 3‑22 Activity Diagram UC011 – Quản lý hành khách theo hành trình

##### Sequence Diagram

### UC012\_Tra vé theo mã đặt vé

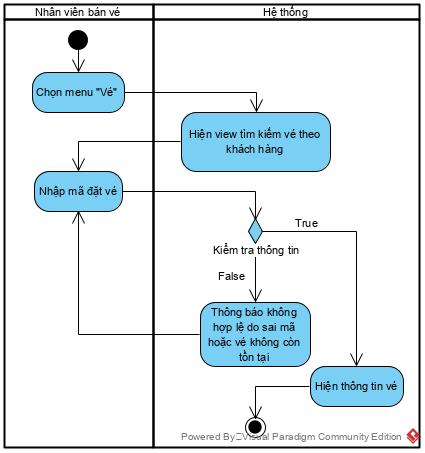
#### Mô tả use case UC012

|  |  |  |
| --- | --- | --- |
| **Use case:** Tra vé theo mã đặt vé | | |
| Mục đích: | Tìm kiếm vé theo mã đặt vé để in hoặc huỷ vé | |
| Mô tả: | Giúp nhân viên bán vé có thể tìm kiếm vé dựa trên mã đặt vé của hành khách để in hoặc huỷ vé | |
| Tác nhân: | Nhân viên bán vé | |
| Điều kiện trước: | Đã đăng nhập vào hệ thống | |
| Điều kiện sau: | Hiện thông tin vé lên màn hình thành công | |
| Luồng sự kiện chính (Basic flows): | | |
| Actor | | **System** |
| 1. Chọn menu “Vé” | |  |
|  | | 1. Hiện view tìm kiếm vé theo khách hàng |
| 1. Nhập mã đặt vé | |  |
|  | | 1. Tìm kiếm vé |
|  | | 1. Hiện thông tin vé |
| Luồng sự kiện phụ (Alternative Flows) : | | |
| 4.1 Nếu mã đặt vé không hợp lệ, hiện thông báo không hợp lệ | | |
| 4.2 Nếu vé đã được in từ trước, hiện thông báo không hợp lệ | | |

Bảng 3‑13 Mô tả UC012 – Tra vé theo mã đặt vé

#### Biểu đồ

##### Activity Diagram



Hình 3‑24 Activity Diagram UC012 – Tra vé theo mã đặt vé

##### Sequence Diagram

# : THIẾT KẾ VÀ HIỆN THỰC

## ClassDiagram

## Sitemap

\

## Database Diagram

## Mô tả Website

### Một số chức năng và giao diện chính

# : 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 quản lý công việc online.

## Kết quả đạt được

* .

## Hạn chế của đồ án

* .

## Hướng phát triển

* .

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Việt

1. AAAAAA
2. BBBBBBB

Các tài liệu Tiếng Anh

1. CCCCCCC

Các tài liệu từ Internet

1. FFFFFFFFFFF

# PHỤ LỤC