

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA: CÔNG NGHỆ THÔNG TIN II



ĐỒ ÁN MÔN HỌC
NHẬP MÔN CÔNG NGHỆ PHẦN MỀM
ĐỀ TÀI: MINI APP HỖ TRỢ BÁN VÉ SỰ KIỆN

Nhóm thực hiện: Nhóm 6

GVHD: ThS. Châu Văn Vân

Danh sách thành viên:

- | | |
|------------------------|------------|
| 1. Trần Nguyễn Bá Huy | N22DCPT037 |
| 2. Nguyễn Đăng Khoa | N22DCPT045 |
| 3. Vũ Hoàng Quang Dũng | N22DCPT011 |

TP. Hồ Chí Minh, tháng 5, năm 2025

MỤC LỤC

I. BÀI TOÁN	4
1.1. Giới thiệu đề tài	4
1.2. Mục tiêu đề tài	4
1.3. Phạm vi thực hiện	5
1.4. Lựa chọn mô hình phát triển phần mềm	5
1.5. Một số trang chính của mini app	6
II. XÁC ĐỊNH YÊU CẦU	7
2.1. Khảo sát	7
2.2. Phân tích yêu cầu chức năng	7
2.3. Phân tích yêu cầu phi chức năng	8
III. CÔNG NGHỆ SỬ DỤNG	9
3.1. Hệ quản trị cơ sở dữ liệu MySQL:	9
3.2. React	9
3.3. TailwindCSS	11
3.4. Zalo MiniApp + ZaUI Components	11
3.5. NodeJs + Express	12
IV. PHÂN TÍCH HỆ THỐNG	14
4.1. Mô hình hóa biểu đồ chức năng (Use case)	14
4.2. Biểu đồ thực thể (Class Diagram)	26
4.3. Biểu đồ tuần tự (Sequence Diagram)	27
4.3.1 Biểu đồ tuần tự của Use case Đăng ký	27
4.3.2. Biểu đồ tuần tự của Use case Đăng nhập	28
4.3.3. Biểu đồ tuần tự của Use case quản lý thông tin cá nhân	29
4.3.4. Biểu đồ tuần tự của Use case quản lý sự kiện	30
4.3.5. Biểu đồ tuần tự của Use case mua vé, hủy vé	31
4.3.6. Biểu đồ tuần tự của Use case tiếp nhận và xử lý đơn hàng	32
V. THIẾT KẾ HỆ THỐNG	33
5.1. Sơ đồ ERD:	33
5.2. Bảng cơ sở dữ liệu:	34

5.3. Mô hình quan hệ giữa các bảng:	36
5.4. Cài đặt cấu hình cơ sở dữ liệu:	36
VI. DEMO CÁC CHỨC NĂNG	37
6.1. Giao diện đăng nhập và đăng ký tài khoản mới:	37
6.2. Giao diện trang chủ khách hàng:	39
6.3. Giao diện Vé của tôi (Ví đựng vé):	42
6.4. Giao diện Chi tiết sự kiện:	46
6.5. Giao diện mua vé:	47
6.6. Giao diện Thông tin cá nhân:	49
6.7. Giao diện Quản lý sự kiện (dành cho admin):	50
VII. KIỂM THỬ PHẦN MỀM	53
7.1. Phần đăng ký và đăng nhập	53
7.2. Phần quản lý sự kiện	54
7.3. Phần quy trình mua hàng của người dùng	54
7.4. Phần quy trình xem vé và tùy chỉnh của người dùng	55
VIII. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	57
TÀI LIỆU THAM KHẢO	59
BẢNG PHÂN CÔNG VÀ ĐÁNH GIÁ	60

I. BÀI TOÁN

1.1. Giới thiệu đề tài

Trong những năm gần đây, sự trỗi dậy mạnh mẽ của các nền tảng mạng xã hội và ứng dụng mini (Mini App) như Zalo, WeChat đã mở ra một kỷ nguyên mới cho việc phát triển các giải pháp công nghệ tiện lợi và tức thời. Đặc biệt tại Việt Nam, Zalo, với cộng đồng hơn 100 triệu người dùng, đã nổi lên như một "siêu ứng dụng" và một kênh phân phối đầy tiềm năng, cho phép các nhà tổ chức sự kiện tiếp cận đối tượng mục tiêu một cách trực tiếp và hiệu quả chưa từng có. Thêm vào đó, sự thay đổi trong thói quen tiêu dùng, đặc biệt được thúc đẩy bởi đại dịch COVID-19, đã làm tăng vọt nhu cầu mua vé trực tuyến, đòi hỏi các giải pháp không chỉ đơn giản và tiện lợi mà còn tích hợp liền mạch các phương thức thanh toán điện tử.

Mini App trên Zalo đáp ứng hoàn hảo những yêu cầu này. Được xây dựng bằng sự kết hợp mạnh mẽ giữa React + ZaUI cho front end và Node.js/Express cho backend, các ứng dụng mini không chỉ giảm thiểu đáng kể chi phí và thời gian phát triển so với các ứng dụng truyền thống mà còn tận dụng tối đa hệ sinh thái sẵn có của Zalo.

Nắm bắt xu hướng này và nhận thấy tiềm năng to lớn của thị trường, nhóm chúng em đã tiến hành thiết kế và phát triển một Mini App bán vé sự kiện hiệu quả, với mục tiêu đáp ứng nhu cầu ngày càng tăng của cả người mua vé và nhà tổ chức sự kiện, đồng thời tận dụng tối đa sức mạnh của nền tảng Zalo và các công nghệ web hiện đại.

1.2. Mục tiêu đề tài

Mini App hướng đến việc giải quyết các khó khăn thường gặp như quy trình mua vé phức tạp, thông tin sự kiện không tập trung, hay hạn chế trong khả năng tương tác giữa nhà tổ chức và người tham dự. Bằng cách tích hợp các tính năng như thanh toán qua Zalo Pay, quét QR code check-in, và gửi thông báo tự động, Mini App không chỉ giúp người dùng tiết kiệm thời gian mà còn hỗ trợ nhà tổ chức quản lý sự kiện hiệu quả hơn, đồng thời tận dụng tối đa hệ sinh thái sẵn có của Zalo tại Việt Nam.

1.3. Phạm vi thực hiện

Xây dựng một hệ thống Mini App trực tuyến, có khả năng triển khai trên trình duyệt web thông thường và tích hợp vào nền tảng Zalo MiniApp thông qua WebView.

Mục tiêu chính của hệ thống là cung cấp một giải pháp toàn diện cho việc bán vé các loại hình sự kiện trực tuyến, đáp ứng nhu cầu mua vé của đa dạng người dùng.

Hệ thống sẽ được xây dựng với hai cấp độ người dùng chính, được phân quyền rõ ràng để đảm bảo tính bảo mật và quản lý hiệu quả

1.4. Lựa chọn mô hình phát triển phần mềm

Để triển khai dự án xây dựng Mini App bán vé sự kiện này, nhóm chúng em đã quyết định áp dụng **Mô hình phát triển phần mềm Thác nước (Waterfall Model)**. Quyết định này được đưa ra dựa trên việc phân tích các đặc điểm và điều kiện cụ thể của dự án đồ án môn học, cũng như những ưu điểm mà mô hình Thác nước có thể mang lại trong một số trường hợp nhất định:

- **Yêu cầu rõ ràng và ổn định:** Các chức năng cho Mini App (người dùng) và Admin đã được xác định chi tiết ngay từ đầu, dự kiến ít thay đổi, tạo điều kiện lập kế hoạch tuần tự.
- **Quy trình tuần tự, dễ quản lý:** Mô hình phân chia dự án thành các giai đoạn rõ ràng (phân tích, thiết kế, lập trình, kiểm thử), giúp nhóm dễ dàng theo dõi tiến độ và quản lý chặt chẽ từng bước. Điều này đặc biệt hữu ích cho một dự án học thuật đòi hỏi sự rõ ràng trong quy trình.
- **Tài liệu hóa chi tiết:** Mỗi giai đoạn đều yêu cầu tài liệu hóa kỹ lưỡng, quan trọng cho việc trình bày đồ án và đánh giá của giảng viên, đảm bảo sự hiểu biết nhất quán trong nhóm.
- **Phù hợp với quy mô và phạm vi rõ ràng:** Với hạn chế về thời gian và nguồn lực của đồ án, việc tập trung hoàn thành các chức năng đã định nghĩa theo luồng tuần tự giúp kiểm soát tốt phạm vi dự án, đảm bảo khả năng hoàn thành đúng thời hạn.

Nhóm chúng em áp dụng mô hình này với các pha tuần tự như sau:

1. **Thu thập yêu cầu:** Xác định rõ chức năng cho Mini App (người dùng) và Admin (quản lý sự kiện).
2. **Phân tích và thiết kế:** Thiết kế kiến trúc tổng thể, cơ sở dữ liệu và giao diện người dùng (UI/UX) chi tiết cho Zalo Mini App.
3. **Cài đặt:** Chuyển đổi các thiết kế thành mã nguồn thực tế cho Mini App.
4. **Kiểm thử:** Đảm bảo các chức năng hoạt động đúng như thiết kế thông qua kiểm thử module và tích hợp.
5. **Triển khai:** Giai đoạn này không bao gồm trong đồ án.
6. **Bảo trì:** Giai đoạn này không bao gồm trong đồ án."

1.5. Một số trang chính của mini app

- Trang chủ: giới thiệu các sự kiện nổi bật, các sự kiện mới liên tục được cập nhật, Thanh tìm kiếm và bộ lọc (ngày, địa điểm, thể loại).
- Trang chi tiết sự kiện: mô tả chi tiết về sự kiện (VD: thời gian, địa điểm, giá vé)
- Trang vé: hiển thị những vé đã mua, cho phép chỉnh sửa hoặc xóa vé
- Trang mua vé và thanh toán: nhập thông tin cá nhân (nếu chưa nhập từ zalo login), chọn phương thức thanh toán, xác nhận thanh toán và nhận QR vé
- Trang profile: hiển thị thông tin cá nhân của người dùng (như tên, SĐT, Email,...)

II. XÁC ĐỊNH YÊU CẦU

2.1. Khảo sát

Nhằm đảm bảo Mini App được xây dựng phù hợp với thực tiễn, chúng em đã tiến hành một loạt các khảo sát nhỏ với những cá nhân và tổ chức thường xuyên tổ chức các sự kiện vừa và nhỏ, đồng thời thu thập ý kiến từ những người có thói quen mua vé sự kiện trực tuyến, có tham khảo các tính năng và quy trình của các nền tảng bán vé phổ biến như Ticketbox.

Nhiều người tổ chức sự kiện vẫn phải đối mặt với các thách thức trong việc thiết lập và theo dõi các loại vé khác nhau, quản lý số lượng vé còn lại, cũng như xử lý các giao dịch thanh toán và xác nhận đặt vé một cách trơn tru. Hơn nữa, việc tiếp cận và tương tác với người mua vé tiềm năng, đặc biệt là trên các nền tảng di động, vẫn chưa được khai thác hiệu quả. Từ phía người mua, yêu cầu về một ứng dụng mua vé nhanh chóng, dễ dàng thao tác, hỗ trợ nhiều hình thức thanh toán điện tử và cung cấp vé điện tử tiện lợi là rất rõ ràng.

Kết quả khảo sát cho thấy sự cần thiết của một Mini App đơn giản, trực quan, với chi phí phải chăng, nhưng vẫn đáp ứng được các nhu cầu cơ bản của cả người bán và người mua vé sự kiện. Do đó, việc phát triển một Mini App bán vé sự kiện trực tuyến, tập trung vào các chức năng quản lý sự kiện, mua/ bán vé online, tích hợp thanh toán trực tuyến và mang lại trải nghiệm mượt mà cho người dùng là một hướng đi có tiềm năng, hứa hẹn mang lại giải pháp hiệu quả cho thị trường bán vé sự kiện trực tuyến.

2.2. Phân tích yêu cầu chức năng

Dựa trên các nghiệp vụ thu thập được, hệ thống Mini App bán vé sự kiện cần hỗ trợ các chức năng chính sau:

- Đăng ký, đăng nhập hệ thống: Phân biệt người dùng (khách hàng mua vé) và Admin (người quản lý sự kiện).
- Xem thông tin sự kiện: Hiển thị danh sách sự kiện, thông tin chi tiết sự kiện (tên, thời gian, địa điểm, mô tả, hình ảnh, giá vé, sơ đồ chỗ ngồi).

- Tìm kiếm sự kiện: Cho phép người dùng tìm kiếm sự kiện theo nhiều tiêu chí (tên, địa điểm, thời gian, thể loại).
- Quản lý sự kiện (chỉ dành cho tài khoản Admin): Thêm, sửa, xóa thông tin sự kiện; quản lý loại vé, số lượng vé, giá vé; quản lý sơ đồ chỗ ngồi.
- Mua vé: Cho phép người dùng chọn loại vé, số lượng vé, thêm vào giỏ hàng (nếu có), thực hiện thanh toán trực tuyến.
- Quản lý đơn hàng: Theo dõi trạng thái đơn hàng (mới, đã thanh toán, đã hủy), xem lịch sử mua vé.
- Quản lý thông tin cá nhân: Cho phép người dùng cập nhật thông tin cá nhân, xem lịch sử mua vé.
- Xem mã vé điện tử (QR code): Hiển thị mã QR code cho vé đã mua.

2.3. Phân tích yêu cầu phi chức năng

Ngoài các chức năng nghiệp vụ, hệ thống Mini App bán vé sự kiện cần đáp ứng các yêu cầu phi chức năng sau:

- Giao diện dễ sử dụng, trực quan: Phù hợp với cả người mua vé và người quản lý sự kiện.
- Ứng dụng chạy trên trình duyệt web và Zalo MiniApp WebView: Đảm bảo khả năng tiếp cận đa dạng người dùng.
- Phân quyền chặt chẽ: Đảm bảo chỉ Admin mới có quyền quản lý sự kiện.
- Vé điện tử (QR code): Cung cấp vé điện tử dưới dạng mã QR code để người dùng dễ dàng sử dụng khi đến sự kiện.
- Dữ liệu được lưu trữ trên cơ sở dữ liệu MySQL: Đảm bảo tính toàn vẹn, bảo mật và dễ dàng bảo trì dữ liệu.
- Hỗ trợ thanh toán trực tuyến: Tích hợp các phương thức thanh toán phổ biến và an toàn.

III. CÔNG NGHỆ SỬ DỤNG

3.1. Hệ quản trị cơ sở dữ liệu MySQL:



MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc hỗ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

3.2. React



React là một thư viện JavaScript mã nguồn mở được sử dụng để xây dựng giao diện người dùng (UI) hoặc các thành phần UI có tính tương tác

cao. Nó được phát triển và duy trì bởi Facebook (nay là Meta) và một cộng đồng lớn các nhà phát triển.

React có cú pháp đơn giản và dễ hiểu, đặc biệt là đối với những người đã quen thuộc với JavaScript. JSX, một phần mở rộng của JavaScript, cho phép lập trình viên viết mã HTML trong JavaScript, làm cho quá trình phát triển trở nên trực quan và dễ dàng hơn. Điều này giúp các lập trình viên mới dễ dàng tiếp cận và bắt đầu sử dụng React nhanh chóng.

Dưới đây là những thư viện phổ biến tương thích với React mà nhóm chúng em đã sử dụng trong đồ án này:

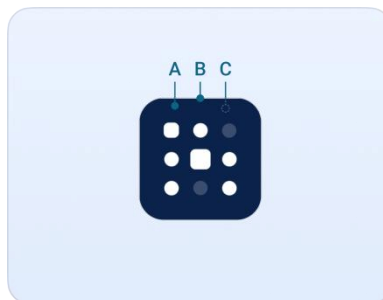
- **react-router-dom:** là một thư viện định tuyến (routing) chuẩn và phổ biến nhất cho các ứng dụng React. Nó cho phép bạn xây dựng các ứng dụng đơn trang (Single Page Applications - SPA) với nhiều "trang" hoặc chế độ xem khác nhau mà không cần tải lại toàn bộ trang web khi người dùng điều hướng. react-router-dom cung cấp các thành phần và hook như BrowserRouter, Routes, Route, Link, useNavigate để quản lý URL, hiển thị các component React tương ứng với URL, và điều hướng giữa các trang một cách mượt mà, mang lại trải nghiệm giống như một ứng dụng gốc cho người dùng.
- **axios** là một thư viện HTTP client dựa trên Promise, được sử dụng rộng rãi trong các ứng dụng JavaScript (cả trên trình duyệt và Node.js) để thực hiện các yêu cầu HTTP (như GET, POST, PUT, DELETE) đến các API backend. Nó giúp đơn giản hóa việc gửi và nhận dữ liệu từ server, cung cấp các tính năng mạnh mẽ như tự động chuyển đổi JSON, khả năng xử lý lỗi toàn diện, hủy yêu cầu, và tự động biến đổi dữ liệu yêu cầu/phản hồi. axios được ưa chuộng vì cú pháp rõ ràng, dễ sử dụng và khả năng tương thích tốt với React, giúp các nhà phát triển dễ dàng tương tác với dữ liệu từ xa.

3.3. TailwindCSS



Tailwind CSS là một framework CSS tiện ích, công cụ này cung cấp các lớp tiện ích sẵn có. Điều này giúp người dùng dễ dàng tạo giao diện người dùng mà không cần viết CSS tùy chỉnh. Khác với các framework như Bootstrap, nơi sử dụng các thành phần đã được định nghĩa trước, Tailwind CSS cho phép bạn tạo kiểu linh hoạt và tùy chỉnh hoàn toàn theo ý muốn.

3.4. Zalo MiniApp + ZaUI Components



Zalo Mini App là những “chương trình nhỏ” xuất hiện trên nền tảng Zalo. Nó bao gồm nhiều lĩnh vực khác nhau như: Giáo dục, sức khỏe, tài chính, mua sắm, giải trí,... Zalo Mini App có vai trò giúp người dùng dễ dàng tương tác với doanh nghiệp một cách nhanh chóng.

ZaUI Components bao gồm các bộ mini app UI Component, các tập file thư viện thiết kế và bộ hướng dẫn đi kèm được thiết kế để giúp bạn rút ngắn thời gian phát triển Mini App và tạo được giao diện thân thiện với người dùng.

Bộ thư viện cung cấp các thành phần thiết kế cơ bản, cần thiết cho một ứng dụng Mini App. Các component, tài liệu thiết kế của ZaUI

3.5. NodeJs + Express



NodeJs:

Node.js là một công cụ đơn giản nhưng hiệu quả, giúp bạn sử dụng JavaScript - một ngôn ngữ quen thuộc ở phía client - để xử lý các tác vụ phía server. Điều này không chỉ hợp nhất ngôn ngữ lập trình trên cả hai phía, mà còn mở ra cánh cửa cho những ai muốn chuyển từ lập trình giao diện (frontend) sang full-stack.

Express:

Express là một framework dành cho nodejs. Nó cung cấp cho chúng ta rất nhiều tính năng mạnh mẽ trên nền tảng web cũng như trên các ứng dụng di động. Express hỗ trợ các phương thức HTTP và middleware tạo ra một API vô cùng mạnh mẽ và dễ sử dụng. Có thể tổng hợp một số chức năng chính của express như sau:

- Thiết lập các lớp trung gian để trả về các HTTP request
- Định nghĩa router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL
- Cho phép trả về các trang HTML dựa vào các tham số.

Dưới đây là những thư viện hỗ trợ nodejs + express mà nhóm chúng em đã sử dụng trong đồ án này:

- **mysql2:** là một trình điều khiển (driver) MySQL cho Node.js, được sử dụng để kết nối và tương tác với cơ sở dữ liệu MySQL. Nó cung cấp các phương thức để thực hiện các truy vấn SQL, nhận kết quả và quản lý kết nối database một cách hiệu quả. mysql2 nổi bật với khả năng hỗ trợ các tính năng hiện đại như Promise API (ngoài

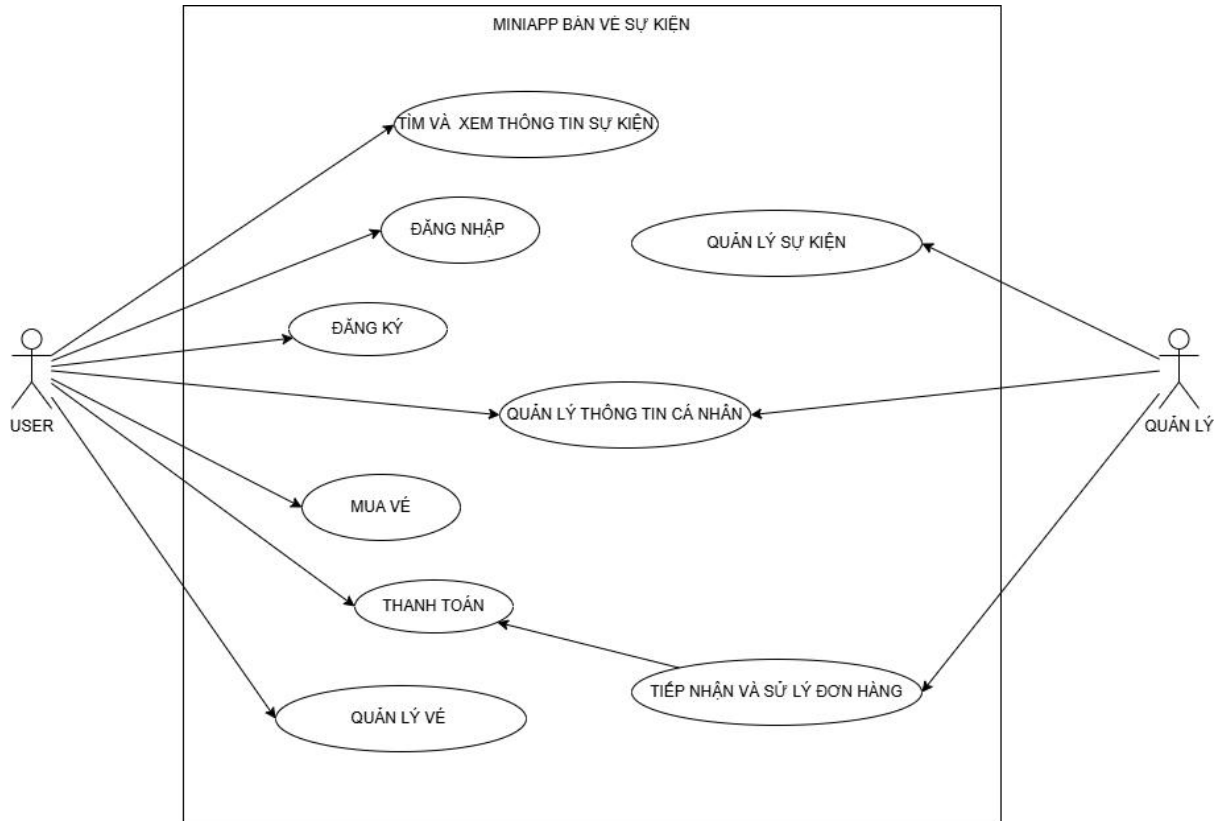
callback) và Prepared Statements, giúp tăng cường bảo mật và hiệu suất.

- **cors:** (Cross-Origin Resource Sharing) là một middleware của Express.js, được sử dụng để xử lý các vấn đề liên quan đến chính sách Same-Origin Policy trên trình duyệt web. Nó cho phép server của bạn chấp nhận các yêu cầu HTTP từ các tên miền khác (khác với tên miền của server).
- **uuid:** là một thư viện Node.js được sử dụng để tạo ra các Mã định danh duy nhất toàn cầu (Universally Unique Identifiers - UUIDs) theo các tiêu chuẩn khác nhau (như RFC 4122). Tác dụng chính là để tạo ra các ID duy nhất một cách ngẫu nhiên, đảm bảo tính không trùng lặp và tiện lợi trong việc quản lý dữ liệu mà không cần phụ thuộc vào cơ chế tự tăng của database.
- **cookie-parser:** là một middleware của Express.js, được sử dụng để phân tích cú pháp (parse) các cookie được gửi từ HTTP request. Nó giúp dễ dàng truy cập và thao tác với các cookie trong ứng dụng Express của bạn thông qua đối tượng req.cookies, điều này rất hữu ích cho việc quản lý phiên đăng nhập, cá nhân hóa người dùng, hoặc các chức năng khác liên quan đến cookie.

IV. PHÂN TÍCH HỆ THỐNG

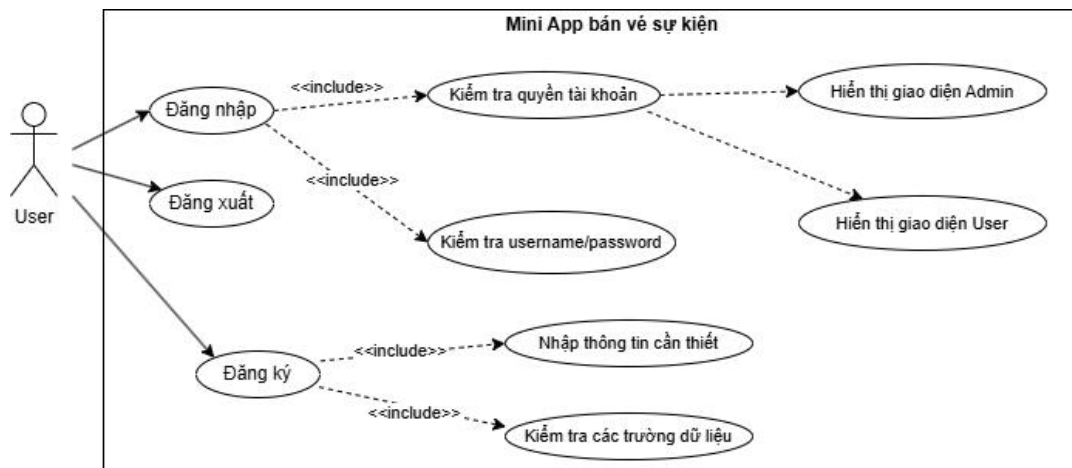
4.1. Mô hình hóa biểu đồ chức năng (Use case)

4.1.1 Sơ đồ Use case tổng quát



Hình 1: Use case tổng quát

4.1.2. Sơ đồ nhóm Use case đăng nhập, đăng ký, đăng xuất



Hình 2: Nhóm Use case đăng nhập, đăng xuất, đăng ký.

Use case	Nội dung
Tên use case	Đăng nhập
Mô tả	Cho phép người dùng (khách hàng, quản lý) đăng nhập hệ thống
Actor	User (Khách hàng), admin(quản lý)
Điều kiện	Khi user/admin chọn chức năng đăng nhập hệ thống
Tiền điều kiện	Có tài khoản
Hậu điều kiện	Đăng nhập thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. User/admin chọn chức năng đăng nhập. 2. Form đăng nhập hiển thị. 3. Nhập tên, mật khẩu. 4. Hệ thống kiểm tra username/password. 6. Nếu đúng: <ul style="list-style-type: none"> - Kiểm tra quyền. - Hiển thị giao diện tương ứng (User/Admin). 6. Nếu sai: Chuyển luồng nhánh A1.
Luồng sự kiện phụ(A1)	<ol style="list-style-type: none"> 1. Thông báo lỗi. 2. Cho phép nhập lại hoặc đăng ký. 3. Nếu nhập lại: Quay lại bước 2. 4. Nếu không: UC kết thúc.

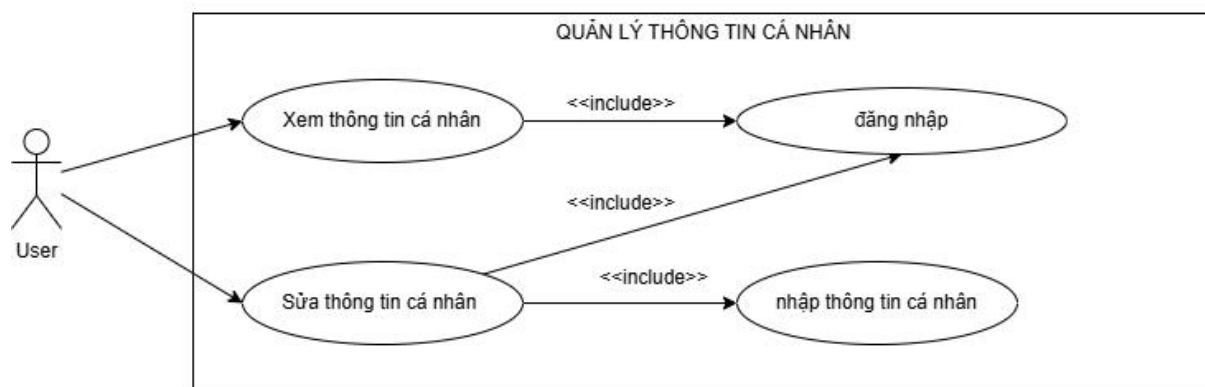
Use case	Nội dung
Tên use case	Đăng xuất
Mô tả	Cho phép user/admin kết thúc phiên làm việc và thoát khỏi hệ thống.
Actor	User(khách hàng), admin(quản lý) (đã đăng nhập)

Điều kiện	User/admin đang ở trạng thái đã đăng nhập và chọn chức năng đăng xuất.
Tiền điều kiện	Đang ở trạng thái đăng nhập
Hậu điều kiện	Hệ thống chuyển về trang chủ hoặc màn hình đăng nhập.
Luồng sự kiện	<ol style="list-style-type: none"> 1. User/admin chọn nút "Đăng xuất". 2. Hệ thống hủy phiên đăng nhập hiện tại. 3. Chuyển về trang chủ/màn hình đăng nhập.

Use case	Nội dung
Tên use case	Đăng ký
Mô tả	Cho phép user (khách hàng) tạo tài khoản mới trên hệ thống
Actor	User (Khách hàng)
Điều kiện	User chọn chức năng "Đăng ký" từ màn hình đăng nhập
Tiền điều kiện	User chưa có tài khoản trong hệ thống
Hậu điều kiện	Tài khoản mới được tạo, thông tin lưu vào CSDL
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. User chọn "Đăng ký". 2. Hệ thống hiển thị form đăng ký (email, mật khẩu, tên, số điện thoại, ...). 3. User nhập thông tin. 4. Kiểm tra xem thông có hợp lệ không. 6. Kiểm tra xem email/tên đã tồn tại chưa 6. Lưu thông tin vào cơ sở dữ liệu. 7. Thông báo đăng ký thành công.
Luồng sự kiện phụ(A1)	<ol style="list-style-type: none"> 1. Nếu email/username đã tồn tại: <ul style="list-style-type: none"> - Thông báo lỗi. - Yêu cầu nhập lại hoặc chuyển sang "Đăng nhập". 2. Nếu dữ liệu không hợp lệ:

	<ul style="list-style-type: none"> - Highlight trường lỗi. - Yêu cầu nhập lại.
--	--

4.1.3. Sơ đồ quản lý thông tin cá nhân

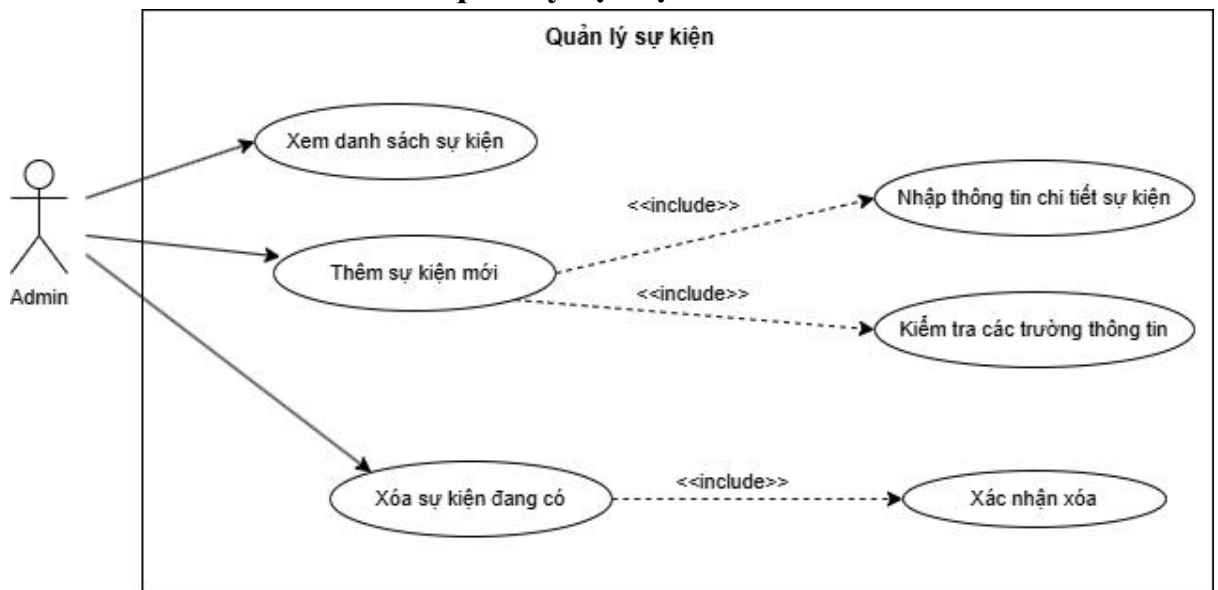


Hình 3: Nhóm Use case quản lý thông tin cá nhân.

Use case	Nội dung
Tên use case	Quản lý thông tin cá nhân
Mô tả	Cho phép người dùng(khách hàng) quản lý thông tin cá nhân của mình
Actor	User (Khách hàng)
Điều kiện	User chọn vào mục cá nhân
Tiền điều kiện	User đã đăng nhập vào hệ thống
Hậu điều kiện	Hiển thị, cập nhật(nếu có thay đổi) thông tin cá nhân
Luồng sự kiện chính	1. Xem thông tin cá nhân: <ul style="list-style-type: none"> - 1.1. User chọn chức năng “Cá nhân”. - 1.2. Hệ thống hiển thị thông tin (tên, email, SĐT, địa chỉ,...). 2. Sửa thông tin cá nhân: <ul style="list-style-type: none"> - 2.1. Hệ thống hiển thị form chỉnh sửa. - 2.2. User nhập thông tin mới (Nhập thông tin cá

	nhân). - 2.3. Hệ thống kiểm tra tính hợp lệ. - 2.4. Người dùng nhấn “Hoàn thành” - 2.6. Lưu thông tin vào CSDL.
Luồng sự kiện phụ	Lỗi dữ liệu không hợp lệ: - Thông báo lỗi cụ thể. - Yêu cầu nhập lại.

4.1.4. Sơ đồ nhóm Use case quản lý sự kiện

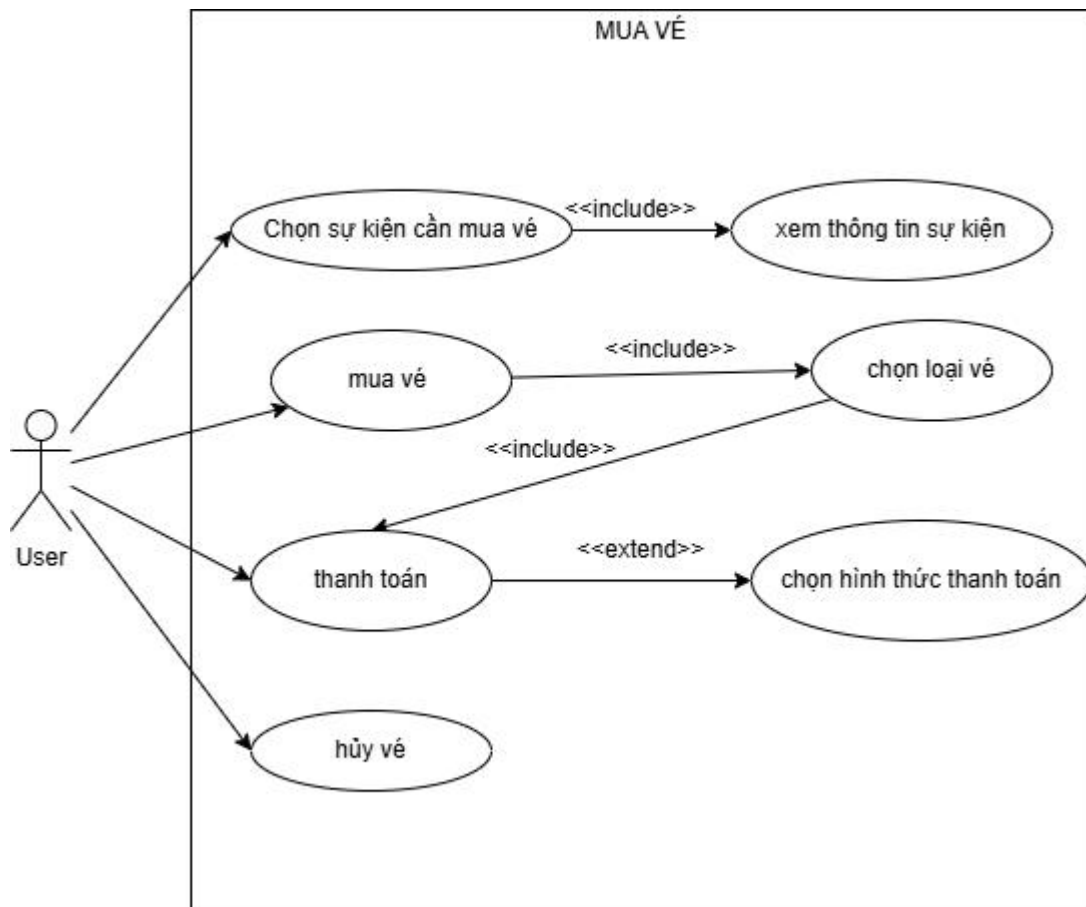


Hình 4: Nhóm Use case quản lý sự kiện.

Use case	Nội dung
Tên use case	Quản lý sự kiện
Mô tả	Cho phép Admin(quản lý) thêm/xóa và xem danh sách các sự kiện trong hệ thống.
Actor	Admin
Điều kiện	Admin đã đăng nhập và có quyền quản lý sự kiện.
Tiền điều kiện	
Hậu điều kiện	Sự kiện mới được thêm vào danh sách (nếu thao tác thành công).
Luồng sự kiện chính	1. Xem danh sách sự kiện: <ul style="list-style-type: none"> - 1.1. Admin chọn mục "Quản lý sự kiện". - 1.2. Hệ thống hiển thị danh sách sự kiện (tên, ngày, trạng thái,...). 2. Thêm sự kiện mới: <ul style="list-style-type: none"> - 2.1. Admin chọn "Thêm sự kiện". - 2.2. Hệ thống hiển thị form nhập thông tin. - 2.3. Admin điền thông tin (tên, mô tả, thời gian, địa điểm,...). - 2.4. Hệ thống kiểm tra thông tin

	<ul style="list-style-type: none"> - 2.6. Lưu sự kiện vào CSDL. <p>3. Xóa sự kiện đang có</p> <ul style="list-style-type: none"> - 3.1. Admin chọn nút “ Xóa ” - 3.2. Hệ thống hiển thị thông báo “Xác nhận xóa sự kiện”. - 3.2. Khi nhấn “Xóa” sẽ hiển thị, “Xác nhận xóa thành công”. - 3.3. Hệ thống xóa thông tin trong database
Luồng sự kiện phụ	<p>Lỗi thiếu thông tin bắt buộc:</p> <ul style="list-style-type: none"> - Thông báo lỗi cụ thể. - Yêu cầu nhập lại.

4.1.5. Sơ đồ nhóm Use case mua vé



Hình 5: Nhóm Use case mua hàng.

Use case	Nội dung
Tên use case	Chọn sự kiện cần mua vé
Mô tả	Cho phép user chọn sự kiện để mua vé
Actor	User (Khách hàng)
Điều kiện	User đã đăng nhập và sự kiện còn vé
Tiền điều kiện	Sự kiện phải hiển thị trên hệ thống.
Hậu điều kiện	Sự kiện cần mua vé được chọn xong.
Luồng sự kiện chính	1. User chọn sự kiện từ danh sách sự kiện 2. Hệ thống hiển thị các sự kiện có trong danh sách 3. User xem thông tin về sự kiện cần mua vé 4. Hệ thống hiển thị số lượng vé → Chuyển sang UC

	"Chọn loại vé". Nếu không thì thực hiện luồng sự kiện phụ 6. UC kết thúc
Luồng sự kiện phụ	Sự kiện hết vé: - Thông báo "Đã hết vé". - Quay lại danh sách.

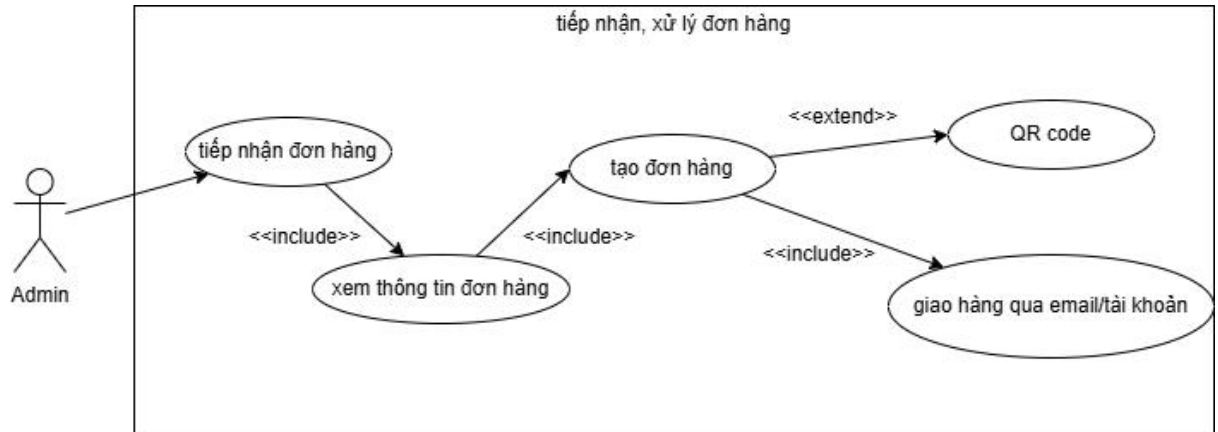
Use case	Nội dung
Tên use case	Chọn loại vé
Mô tả	Cho phép khách hàng chọn loại vé (VIP/Thường) và số lượng.
Actor	User (Khách hàng)
Điều kiện	User đã chọn sự kiện từ UC "Chọn sự kiện".
Tiền điều kiện	Sự kiện còn vé loại đã chọn.
Hậu điều kiện	Loại vé và số lượng được xác nhận
Luồng sự kiện chính	1. Hệ thống hiển thị các loại vé (giá, mô tả). 2. User chọn loại vé và số lượng. 3. Hệ thống kiểm tra số lượng còn lại. Nếu không thì thực hiện luồng sự kiện phụ 4. UC kết thúc
Luồng sự kiện phụ	Hết loại vé đã chọn: - Thông báo loại vé đã hết - Yêu khách hàng chọn loại vé khác

Use case	Nội dung
Tên use case	Thanh toán
Mô tả	Xử lý quy trình thanh toán sau khi Khách hàng chọn vé.
Actor	User (Khách hàng)

Điều kiện	User đã chọn loại vé từ UC "Chọn loại vé".
Tiền điều kiện	User có đủ số dư/tài khoản thanh toán.
Hậu điều kiện	Vé được kích hoạt, số lượng vé trong hệ thống giảm.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị tổng tiền. 2. User chọn hình thức thanh toán (Zalopay, ngân hàng,...). 3. User tiến hành thanh toán. 4. UC kết thúc
Luồng sự kiện phụ	Thanh toán thất bại: - Thông báo lỗi. - Yêu cầu thử lại hoặc chọn phương thức khác.

Use case	Nội dung
Tên use case	Hủy vé đã thanh toán
Mô tả	Cho phép user hủy vé đã thanh toán
Actor	User (Khách hàng)
Điều kiện	User đã mua vé.
Tiền điều kiện	Vé thuộc sự kiện chưa diễn ra
Hậu điều kiện	Vé bị vô hiệu hóa, tiền được hoàn (nếu có), số lượng vé sự kiện tăng lại.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. User mở mục "Vé của tôi". 2. Chọn vé muốn hủy. 3. Hệ thống kiểm tra điều kiện hủy. 4. Xác nhận hủy (có hiển thị phí hủy nếu có). 6. Hệ thống cập nhật trạng thái vé và hoàn tiền (nếu áp dụng).
Luồng sự kiện phụ	Quá thời hạn hủy: - Thông báo "Không thể hủy vé". - Hiển thị chính sách hủy.

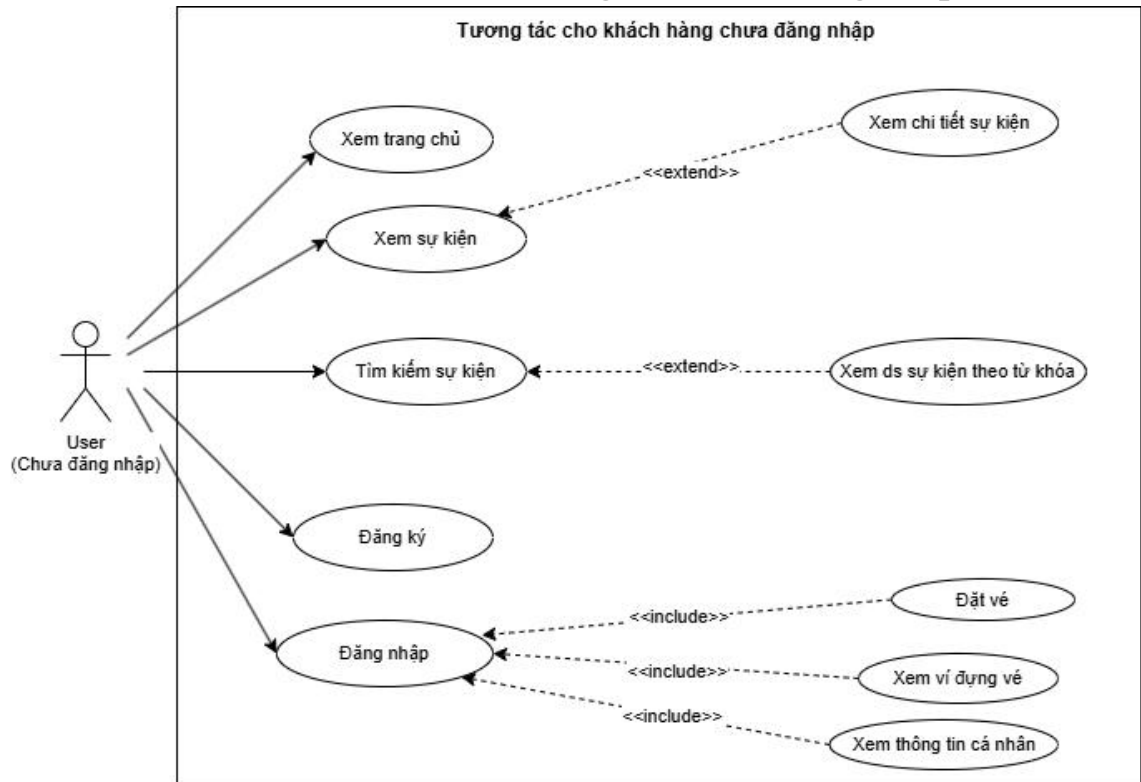
4.1.6. Sơ đồ xử lý vé đã mua



Hình 6: Nhóm Use case tiếp nhận, xử lý đơn hàng.

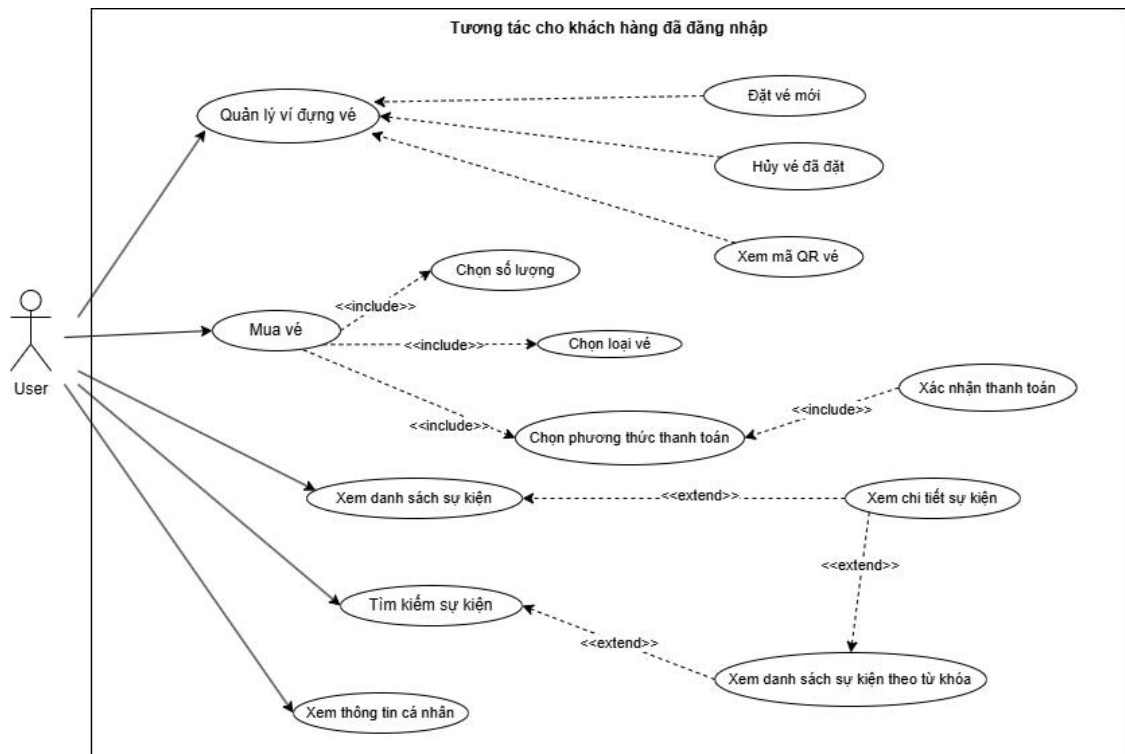
Use case	Nội dung
Tên use case	Tiếp nhận và xử lý đơn hàng
Mô tả	Hệ thống tự động/admin tiếp nhận và xử lý đơn hàng vé từ khách hàng
Actor	User (Khách hàng), admin(quản lý)
Điều kiện	User đã hoàn tất chọn vé và nhấn "Đặt hàng".
Tiền điều kiện	Đơn hàng hợp lệ, số lượng vé còn đủ, thông tin giao dịch đầy đủ.
Hậu điều kiện	Đơn hàng được ghi nhận, trạng thái cập nhật
Luồng sự kiện chính	1. User gửi yêu cầu đặt hàng. 2. Admin tiếp nhận đơn hàng. 3. Tạo đơn hàng với QR code lưu thông tin sự kiện 4. Gửi vé qua email/tài khoản. 6. UC kết thúc.

4.1.7. Sơ đồ nhóm Use case cho những user chưa đăng nhập



Hình 7: Nhóm Use Case cho những user chưa đăng nhập.

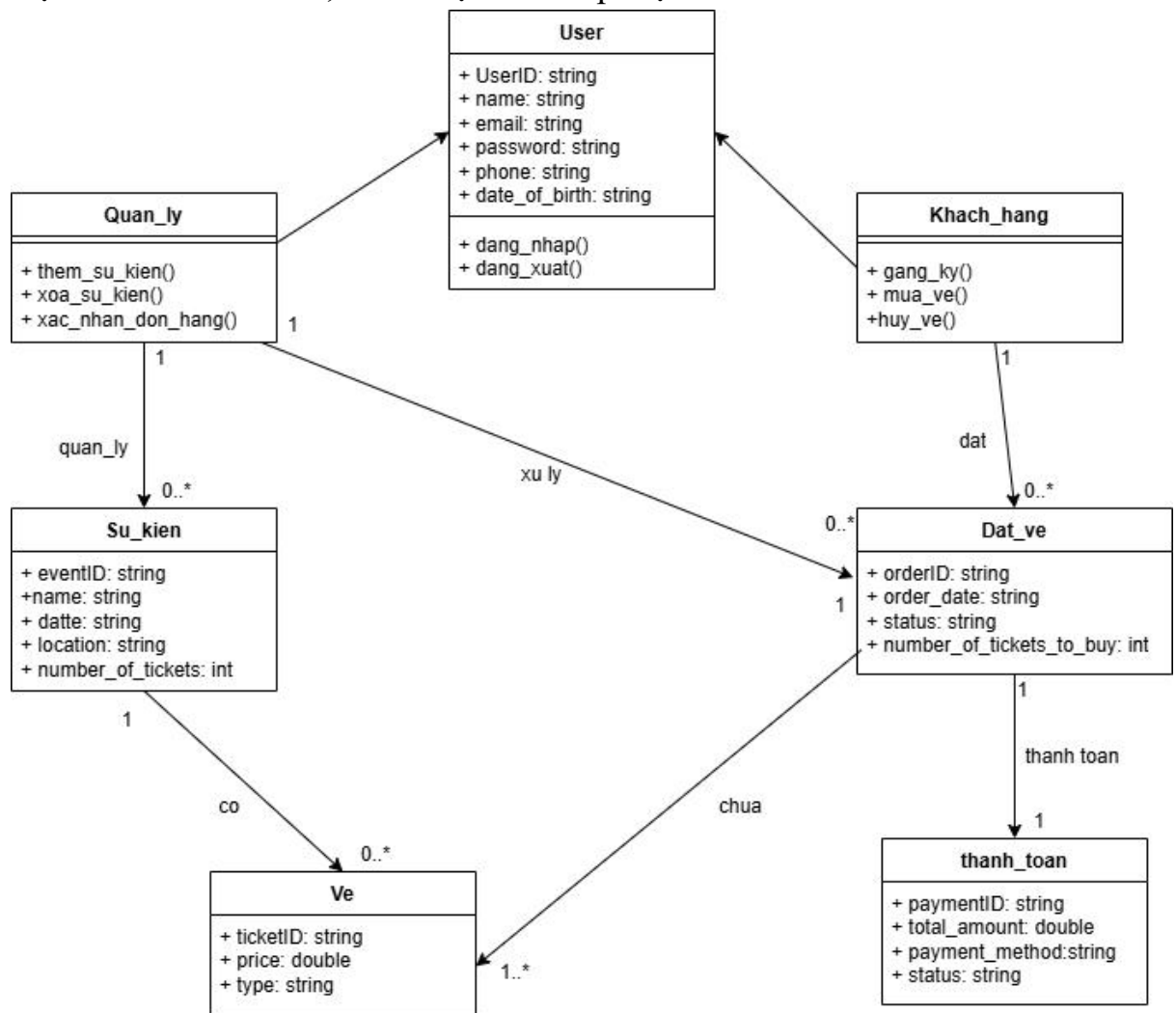
4.1.8 Sơ đồ nhóm Use case cho những user đã đăng nhập



Hình 8: Nhóm Use Case cho những user đã đăng nhập.

4.2. Biểu đồ thực thể (Class Diagram)

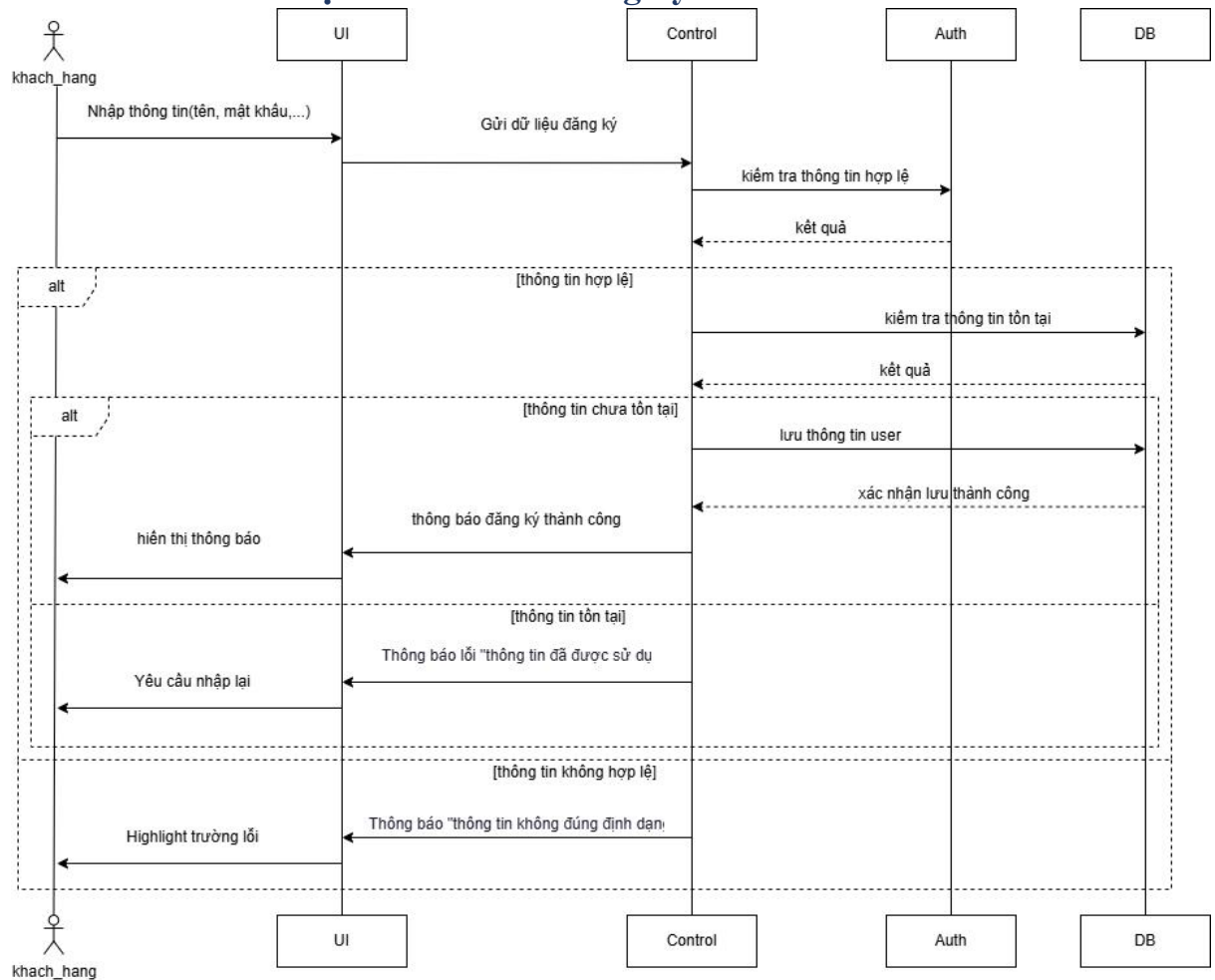
Dựa vào các Use case, ta xác định các lớp thực thể sau:



Hình 9: Sơ đồ các thực thể.

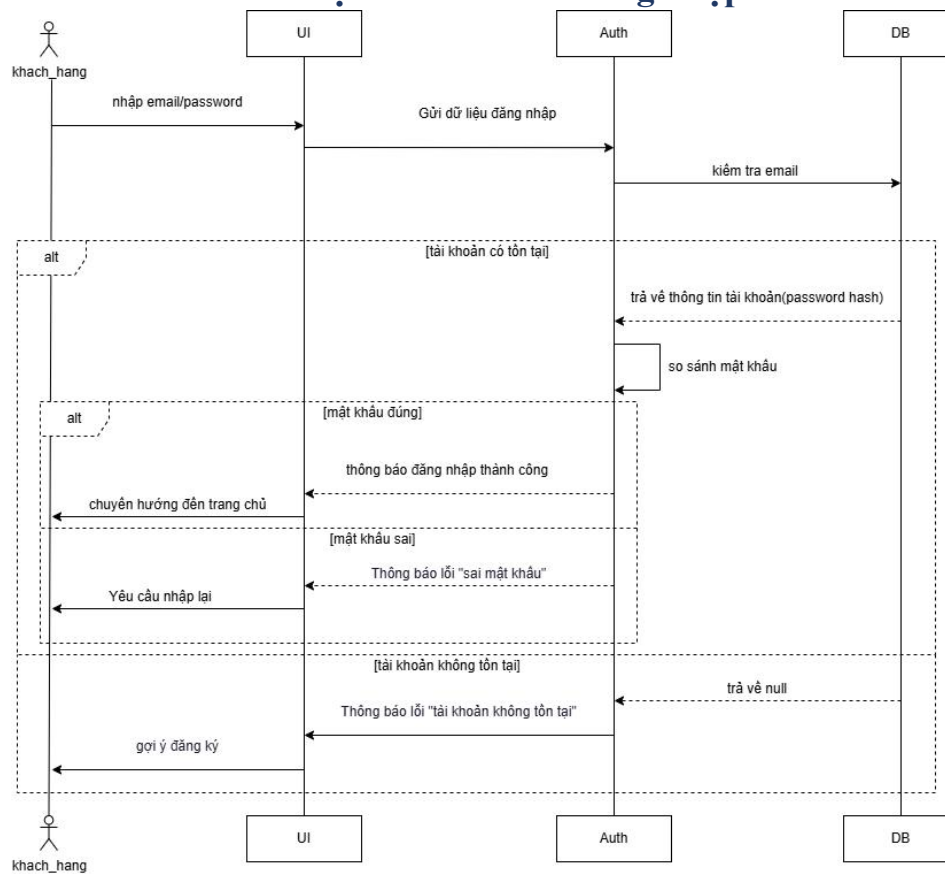
4.3. Biểu đồ tuần tự (Sequence Diagram)

4.3.1 Biểu đồ tuần tự của Use case Đăng ký



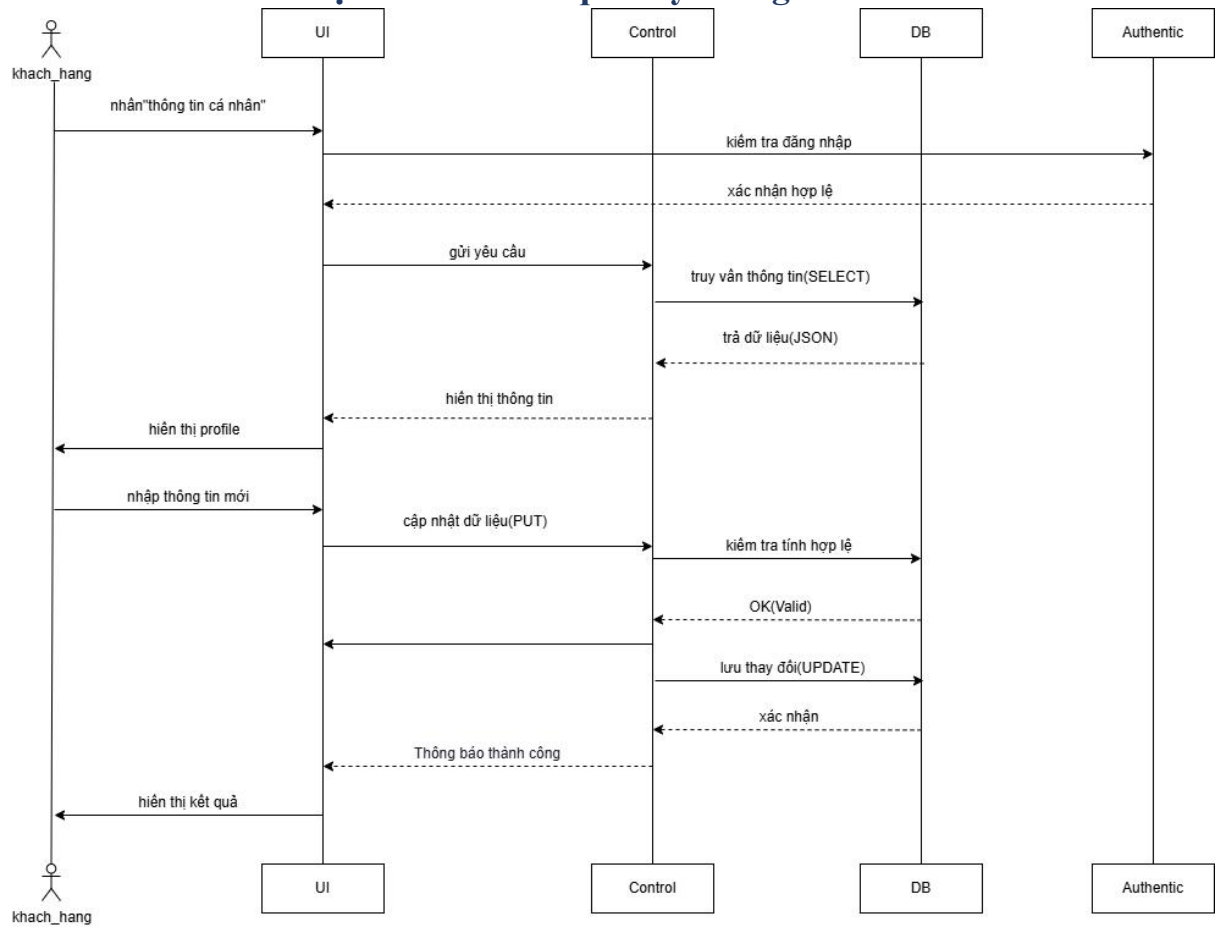
Hình 10: Biểu đồ tuần tự của Use case Đăng ký

4.3.2. Biểu đồ tuần tự của Use case Đăng nhập



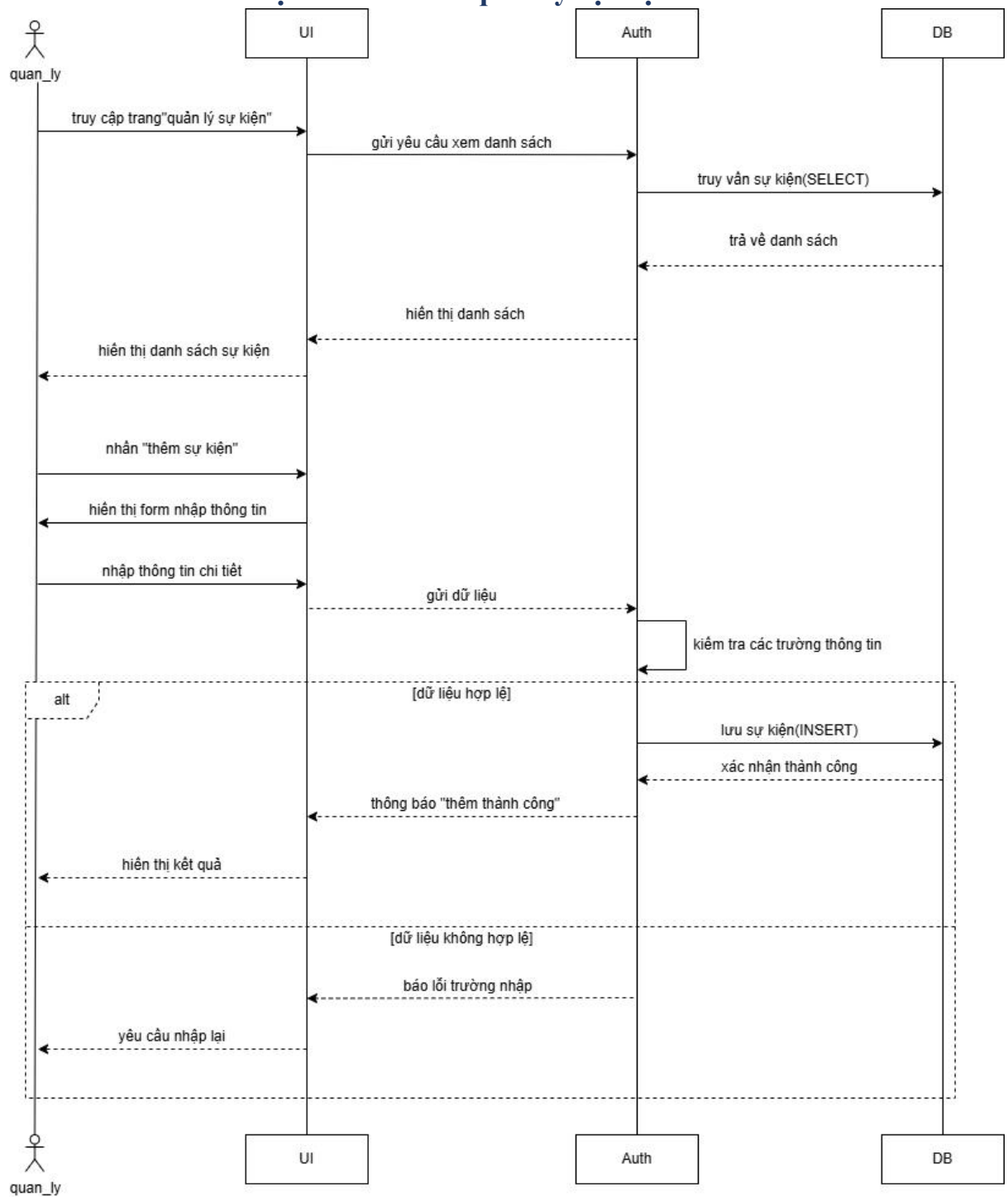
Hình 11: Biểu đồ tuần tự của Use case Đăng nhập.

4.3.3. Biểu đồ tuần tự của Use case quản lý thông tin cá nhân



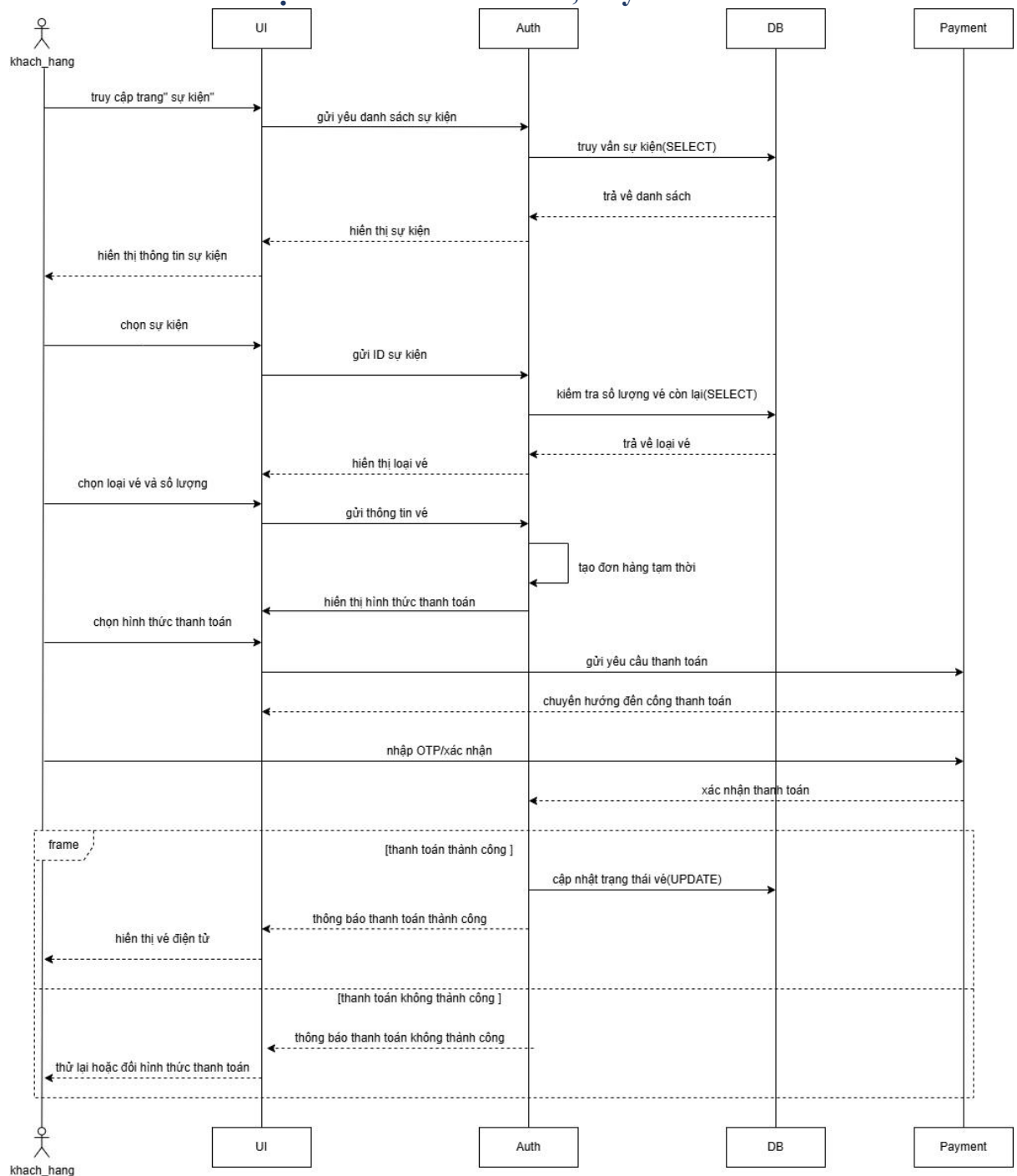
Hình 12: Biểu đồ tuần tự của Use case quản lý thông tin cá nhân.

4.3.4. Biểu đồ tuần tự của Use case quản lý sự kiện

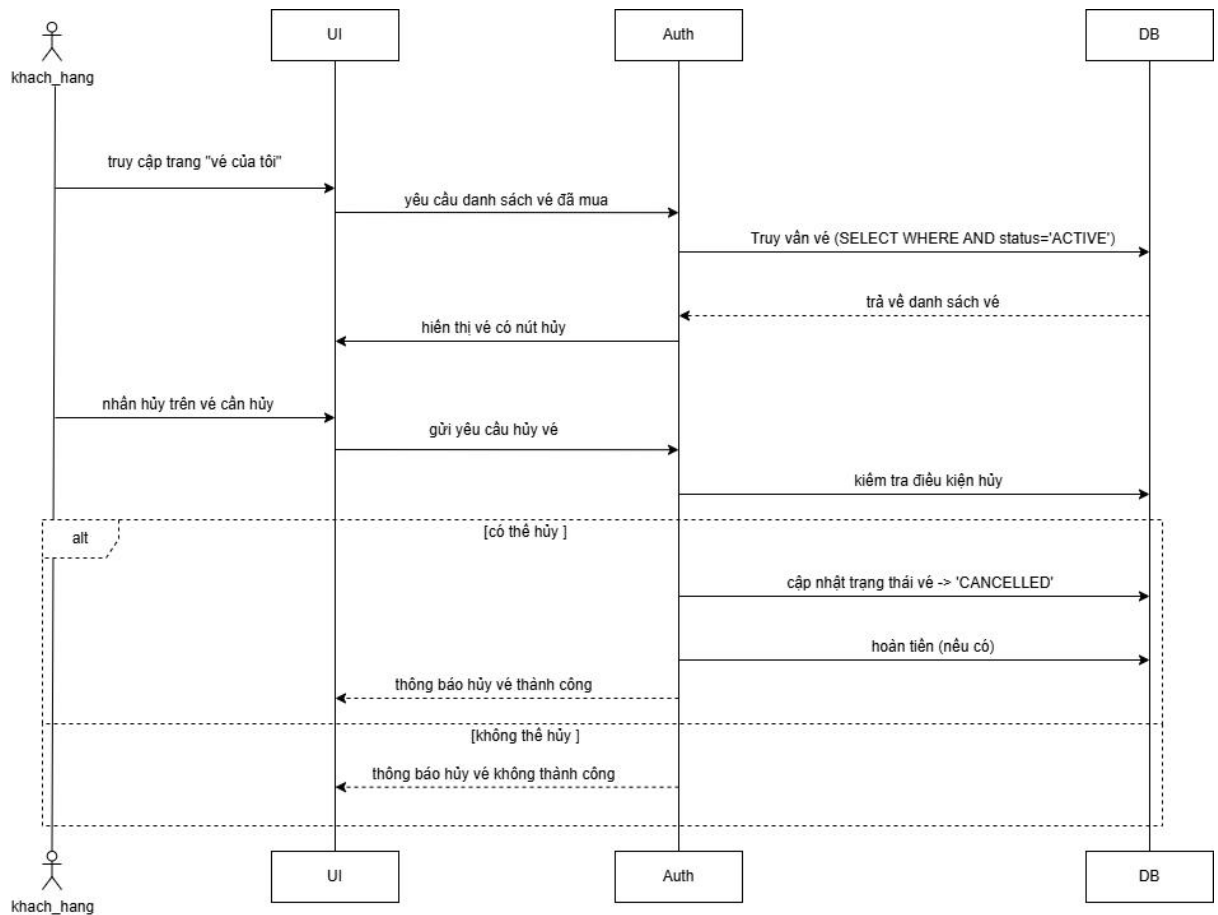


Hình 13: Biểu đồ tuần tự của Use case quản lý sự kiện.

4.3.5. Biểu đồ tuần tự của Use case mua vé, hủy vé

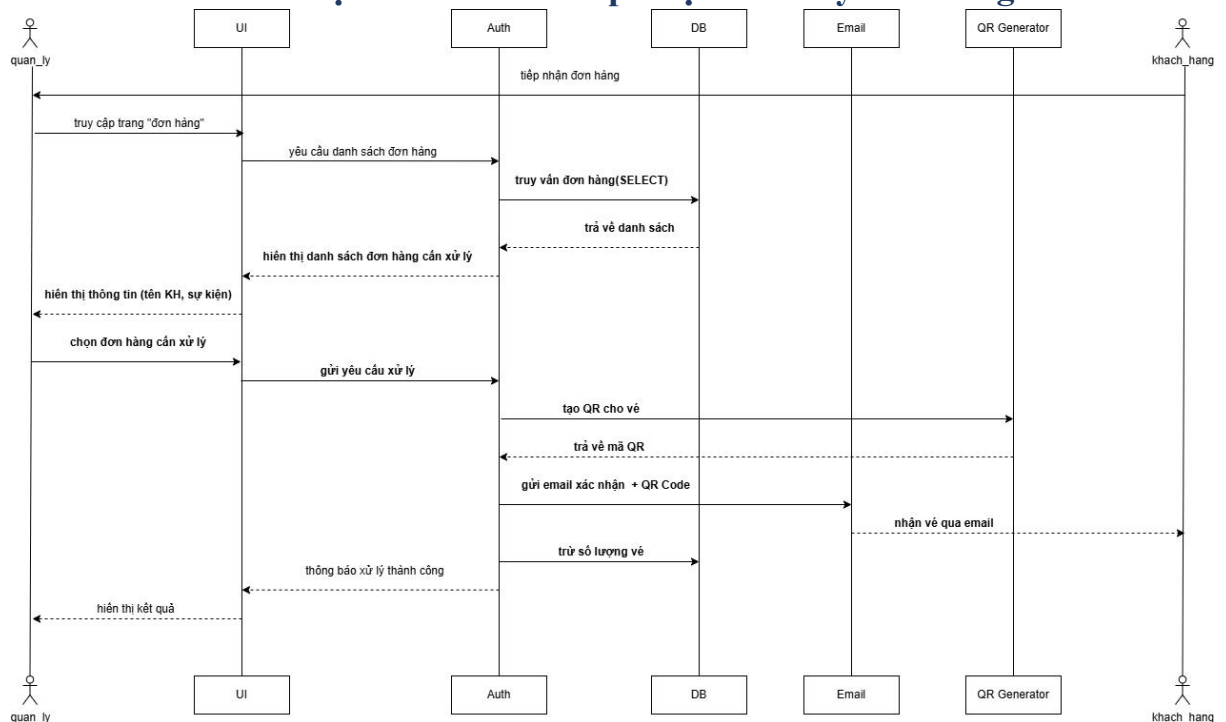


Hình 14: Biểu đồ tuần tự mua vé.



Hình 15: Biểu đồ tuần tự hủy vé.

4.3.6. Biểu đồ tuần tự của Use case tiếp nhận và xử lý đơn hàng



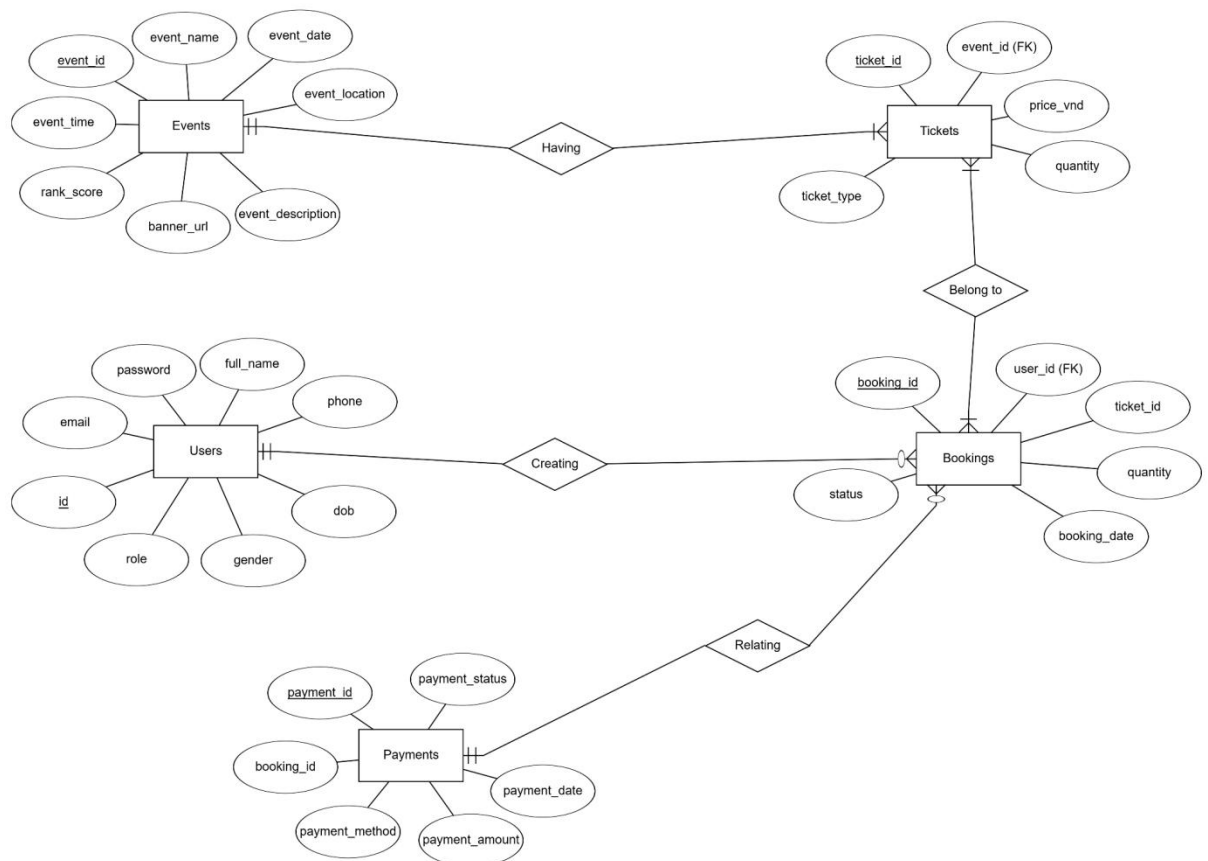
Hình 15: Biểu đồ tuần tự của Use case tiếp nhận và xử lý đơn hàng.

V. THIẾT KẾ HỆ THỐNG

5.1. Sơ đồ ERD:

Xác định các thực thể như sau:

- EVENTS (event_id, event_name, event_date, event_time, event_location, event_description, banner_url, rank_score)
- TICKETS (ticket_id, event_id, ticket_type, price_vnd, quantity)
- USERS (id, email, password, full_name, phone, dob, gender, role)
- BOOKINGS (booking_id, user_id, ticket_id, quantity, booking_date, status)
- PAYMENTS (payment_id, booking_id, payment_method, payment_amount, payment_date, payment_status)



Hình 5.1: Sơ đồ ERD của Mini App Za Ticketing

5.2. Bảng cơ sở dữ liệu:

5.2.1. Bảng events:

Tên biến	Kiểu dữ liệu	Null	Ràng buộc	Mặc định
event_id	VARCHAR(255)	Không	PK	
event_name	VARCHAR(255)	Không		
event_date	DATE	Không		
event_time	TIME	Không		
event_location	VARCHAR(255)	Không		
event_description	TEXT	Không		
banner_url	TEXT	Không		
rank_score	INT	Không		

5.2.2. Bảng tickets:

Tên biến	Kiểu dữ liệu	Null	Ràng buộc	Mặc định
tickets_id	VARCHAR(255)	Không	PK	
event_id	VARCHAR(255)	Không	FK	
ticket_type	ENUM()	Không		
price_vnd	INT	Không		
quantity	INT	Không		

5.2.3. Bảng users:

Tên biến	Kiểu dữ liệu	Null	Ràng buộc	Mặc định
id	INT	Không	PK	
email	VARCHAR(255)	Không		

password	VARCHAR(255)	Không		
full_name	VARCHAR(255)	Không		
phone	VARCHAR(255)	Không		
dob	DATE	Không		
gender	ENUM()	Không		
role	ENUM()			user

5.2.4. Bảng bookings:

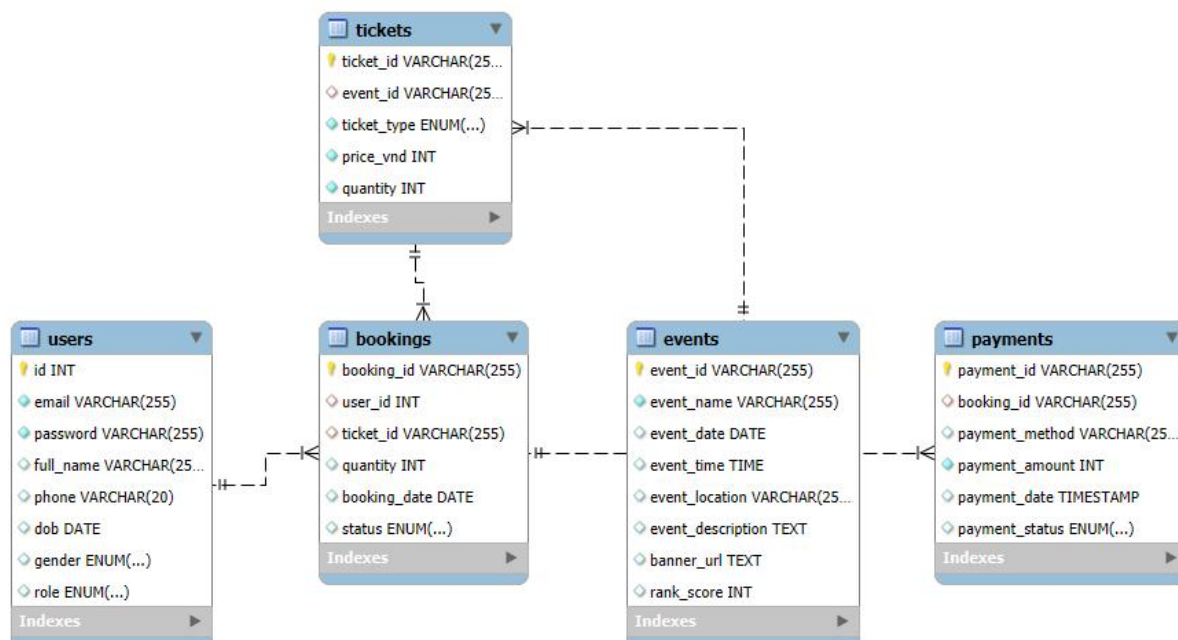
Tên biến	Kiểu dữ liệu	Null	Ràng buộc	Mặc định
booking_id	VARCHAR(255)	Không	PK	
user_id	INT	Không	FK	
ticket_id	VARCHAR(255)	Không	FK	
quantity	INT	Không		
booking_date	DATE	Không		
status	ENUM()			confirmed

5.2.5. Bảng payments:

Tên biến	Kiểu dữ liệu	Null	Ràng buộc	Mặc định
payment_id	VARCHAR(255)	Không	PK	
booking_id	VARCHAR(255)	Không	FK	
payment_method	VARCHAR(255)	Không		
payment_amount	INT	Không		
payment_date	TIMESTAMP			CURRENT_TIM

				ESTAMP
--	--	--	--	--------

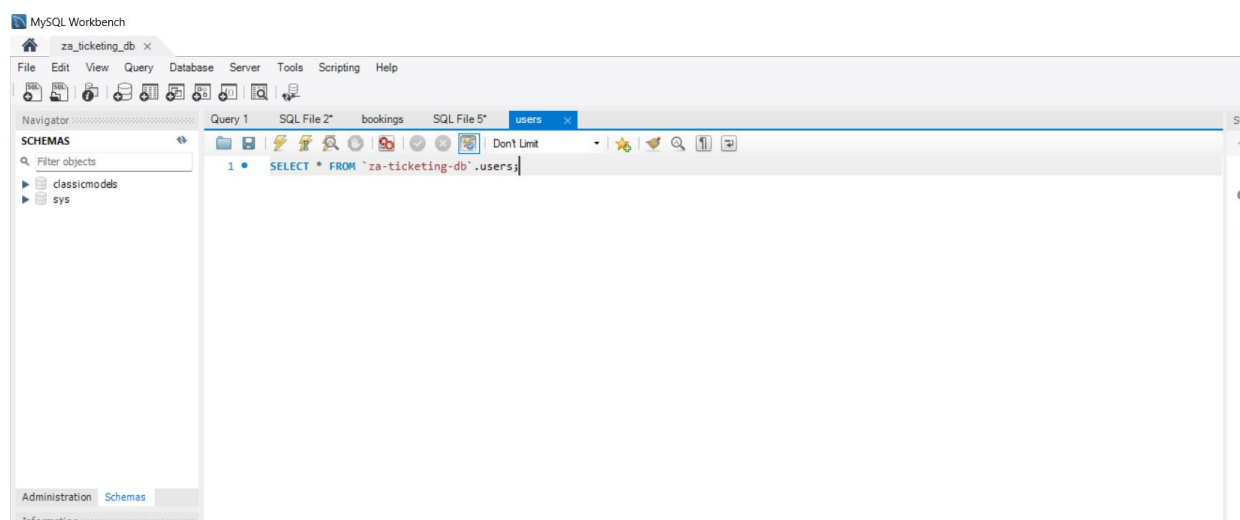
5.3. Mô hình quan hệ giữa các bảng:



Hình 5.3: Mô hình quan hệ giữa các bảng

5.4. Cài đặt cấu hình cơ sở dữ liệu:

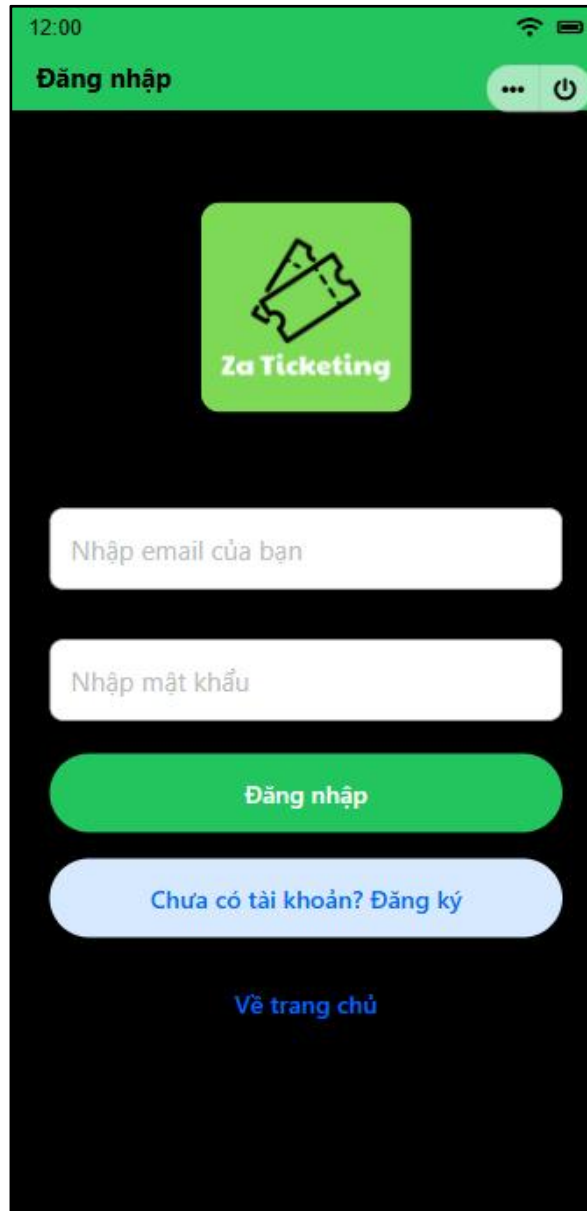
Nhóm chọn sử dụng công cụ MySQL Workbench, một công cụ trực quan để thực hiện dự án này.



Hình 5.4: MySQL Workbench

VI. DEMO CÁC CHỨC NĂNG

6.1. Giao diện đăng nhập và đăng ký tài khoản mới:



Hình 6.1.1: Giao diện đăng nhập Mini App

Người dùng cần nhập địa chỉ email và mật khẩu vào các trường tương ứng, sau đó nhấn nút "Đăng nhập" để tiến hành xác thực và vào trang chủ của ứng dụng. Ngoài ra, màn hình cũng cung cấp tùy chọn cho người dùng chưa có tài khoản có thể "Đăng ký" và nút "Về trang chủ" để điều hướng đến trang chủ mà không cần đăng nhập.

12:00

< Đăng ký

Za Ticketing

Nhập họ tên

Nhập số điện thoại

Nhập địa chỉ email

Tạo mật khẩu (ít nhất 6 ký tự)

Nhập lại mật khẩu

mm/dd/yyyy

☒ Nam ☐ Nữ ☐ Khác

Đăng ký

Hình 6.1.2: Giao diện đăng ký tài khoản mới

Sau khi điền đầy đủ các thông tin cần thiết, người dùng có thể nhấn nút "Đăng ký" để hoàn tất quá trình tạo tài khoản và có thể bắt đầu sử dụng ứng dụng Za Ticketing.

6.2. Giao diện trang chủ khách hàng:

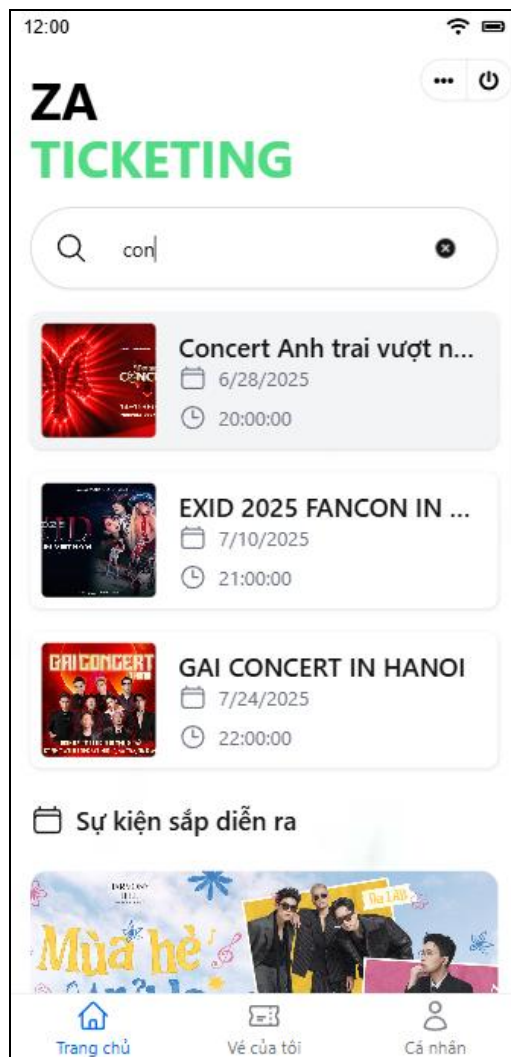


Hình 6.2.1: Giao diện trang chủ khách hàng

Trang chủ là điểm khởi đầu để người dùng khám phá các sự kiện, tìm kiếm thông tin và điều hướng đến các nơi khác của ứng dụng Za Ticketing.

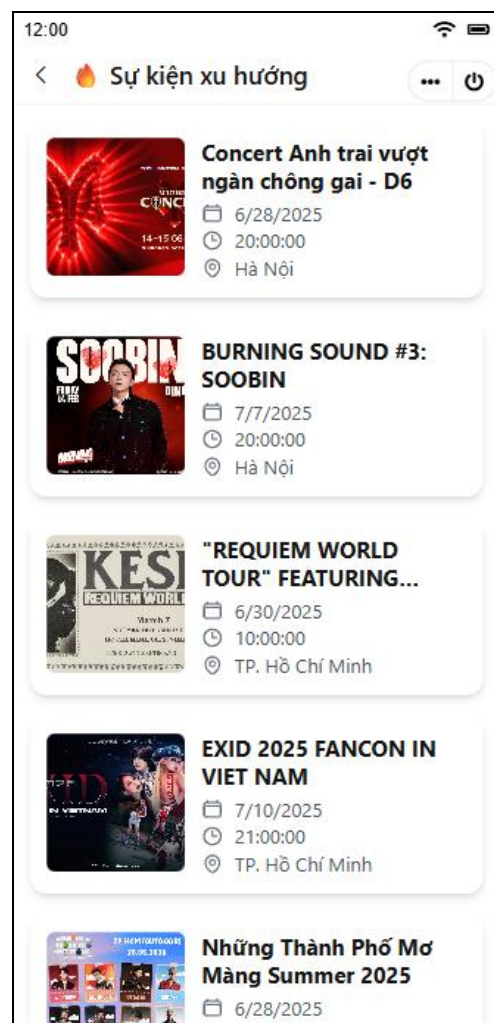
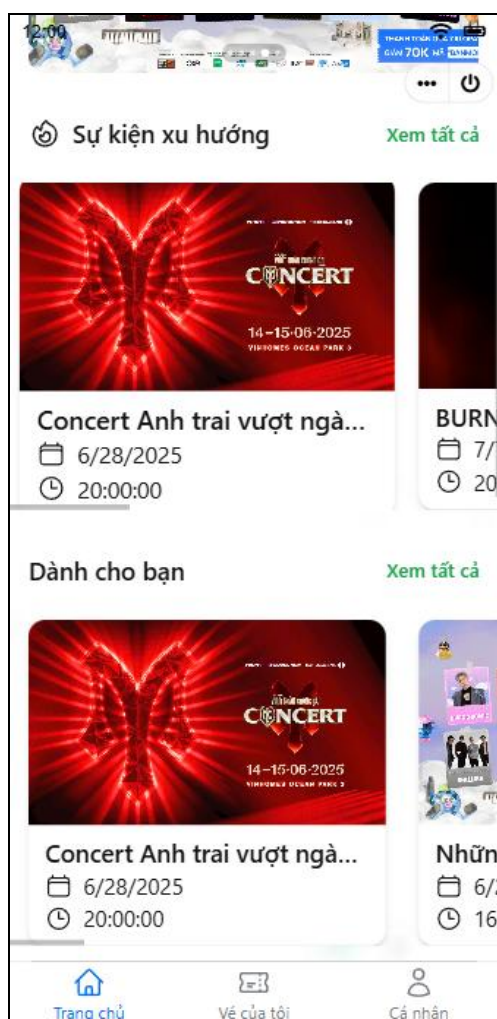
Trang chủ bao gồm các tùy chọn chức năng như:

- Xem danh sách sự kiện theo các phân loại khác nhau.
- Tìm kiếm sự kiện mà người dùng cần.
- Thanh điều hướng đến các trang khác.



Hình 6.2.2: Giao diện tìm kiếm sự kiện theo từ khóa

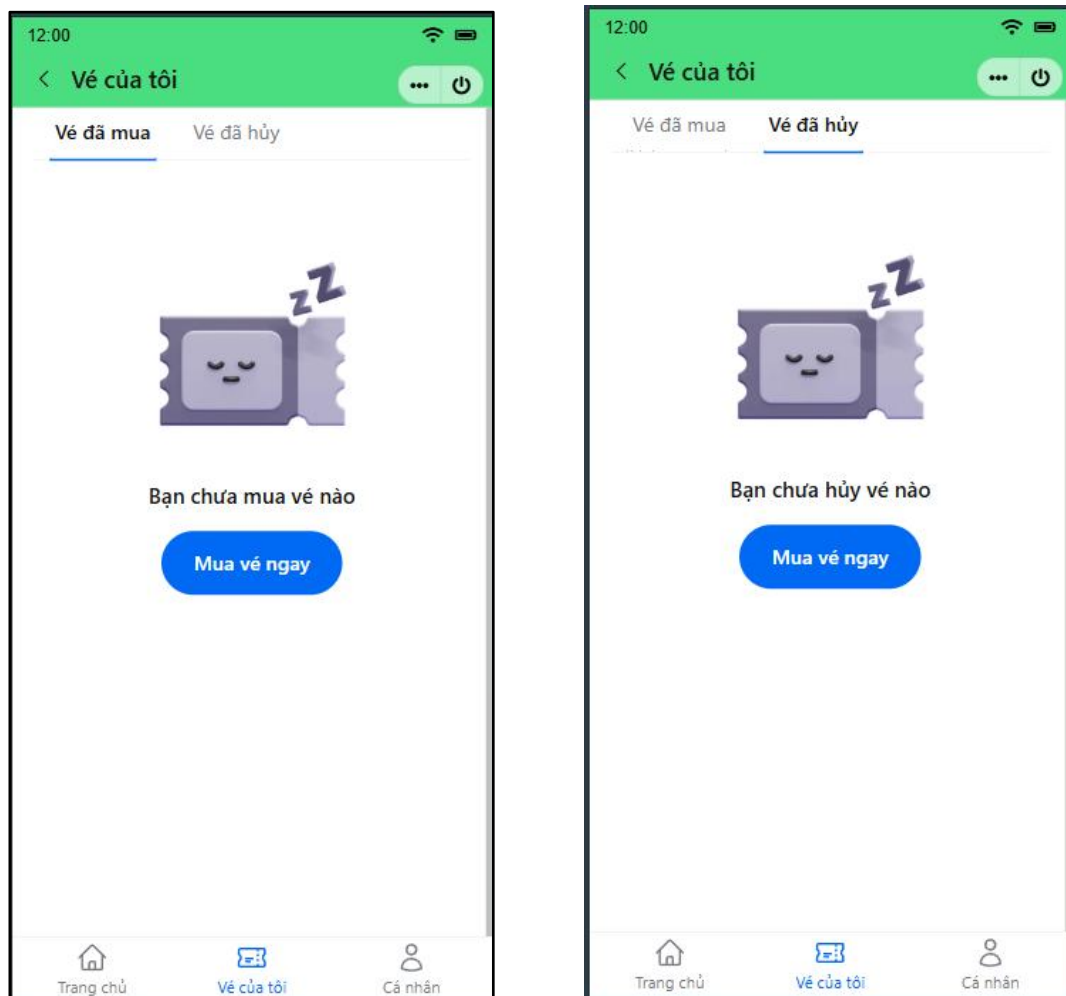
Khi người dùng nhập từ khóa tìm kiếm vào Search bar, hệ thống sẽ trả lại danh sách các sự kiện có từ khóa khớp với từ khóa người dùng nhập vào.



Hình 6.2.3: Danh sách sự kiện theo phân loại nếu người dùng chọn “xem tất cả”

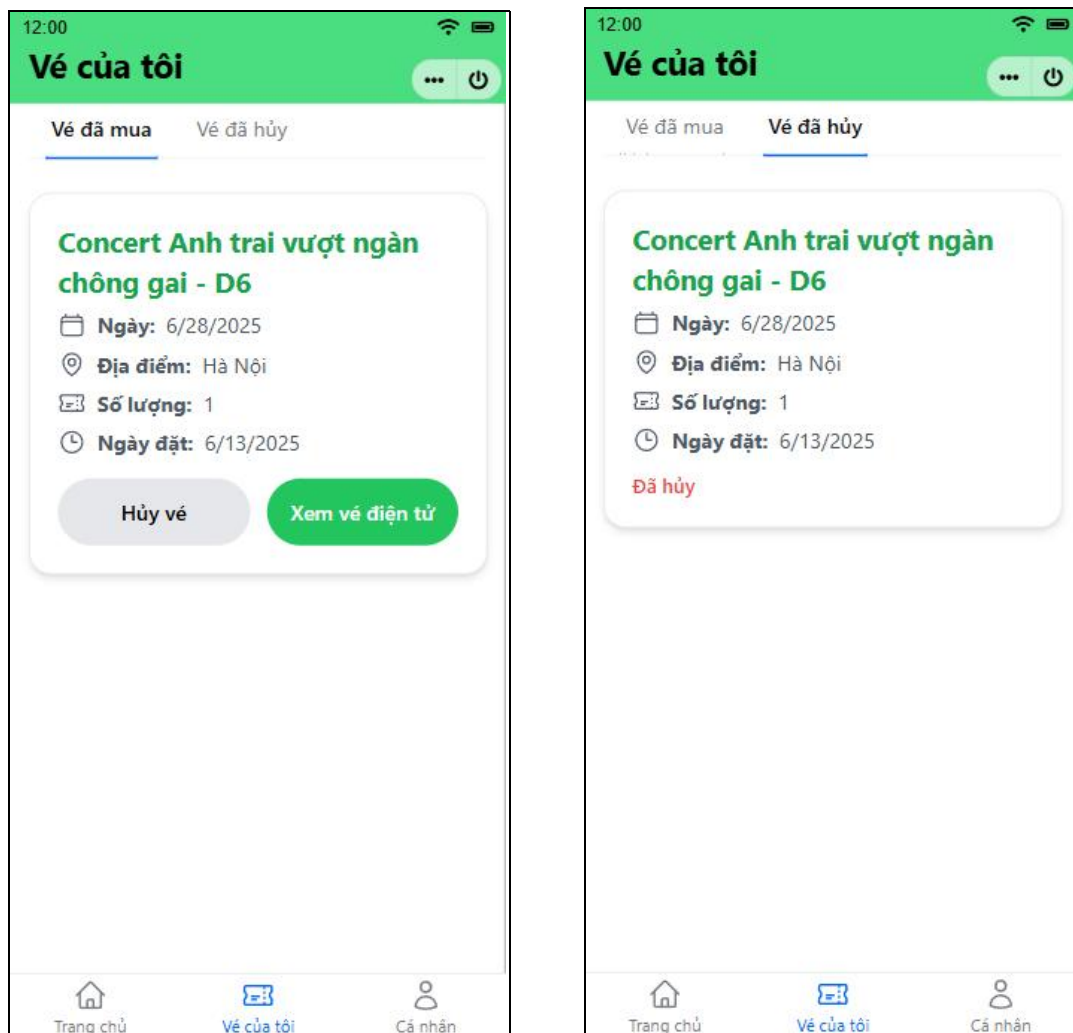
Khi người dùng chọn “xem tất cả” sự kiện theo danh mục tùy thích, hệ thống sẽ trả về danh sách tất cả sự kiện theo phân loại đó

6.3. Giao diện Vé của tôi (Ví đựng vé):



Hình 6.3.1: Giao diện khi ví đựng vé trống

Khi người dùng chưa từng mua vé hoặc hủy vé, trạng thái của tab “Vé đã mua” và “Vé đã hủy” sẽ là trống. Khi người dùng nhấn vào nút “Mua vé ngay” sẽ được điều hướng về trang chủ để mua vé.



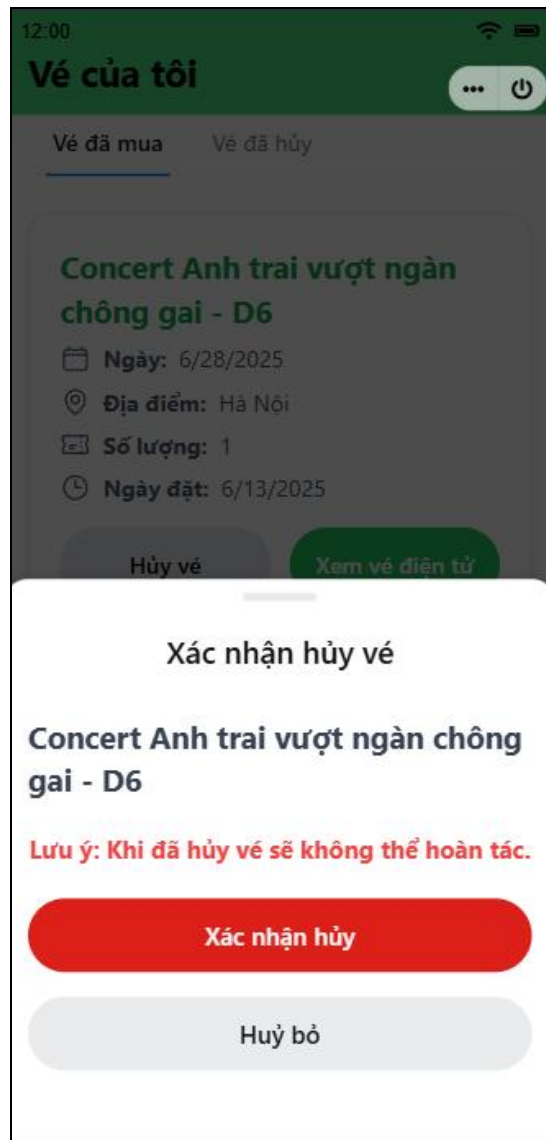
Hình 6.3.2: Giao diện khi ví đựng vé đã được sử dụng

Khi người dùng đã mua vé hoặc hủy vé, tab “Vé đã mua” và “Vé đã hủy” sẽ hiển thị danh sách các vé đã mua và đã hủy.



Hình 6.3.3: Giao diện xem vé điện tử (QR)

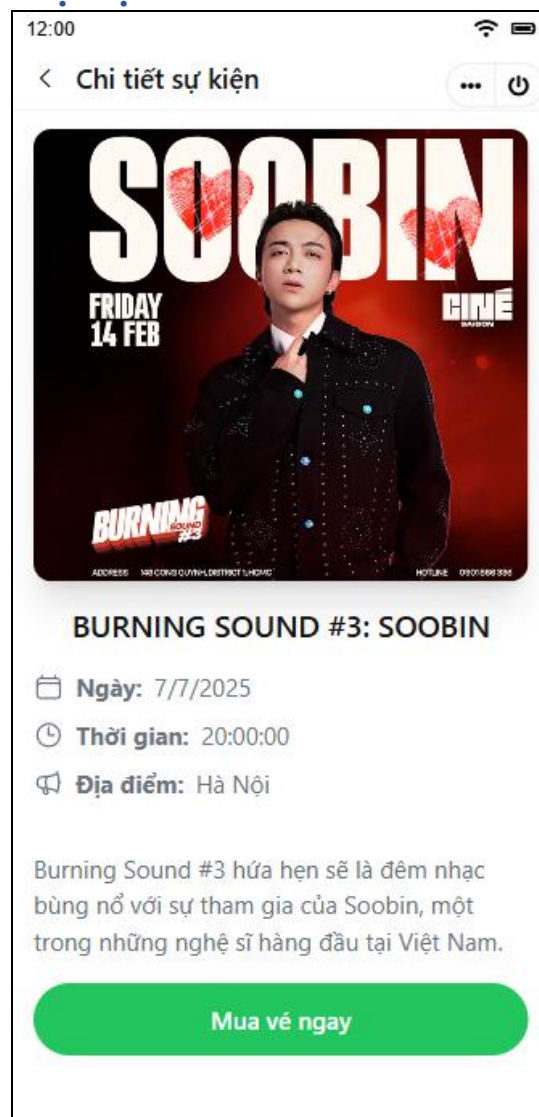
Khi người dùng nhấn vào tùy chọn “Xem vé điện tử” với những vé đã được đặt, hệ thống sẽ hiển thị lên mã QR của những vé đó.



Hình 6.3.4: Giao diện khi hủy vé

Khi người dùng nhấn vào tùy chọn “Hủy vé”, hệ thống sẽ hiển thị lên giao diện **Xác nhận hủy vé**. Người dùng có thể lựa chọn hủy vé hoặc thoát.

6.4. Giao diện Chi tiết sự kiện:



Hình 6.4.1: Giao diện chi tiết sự kiện

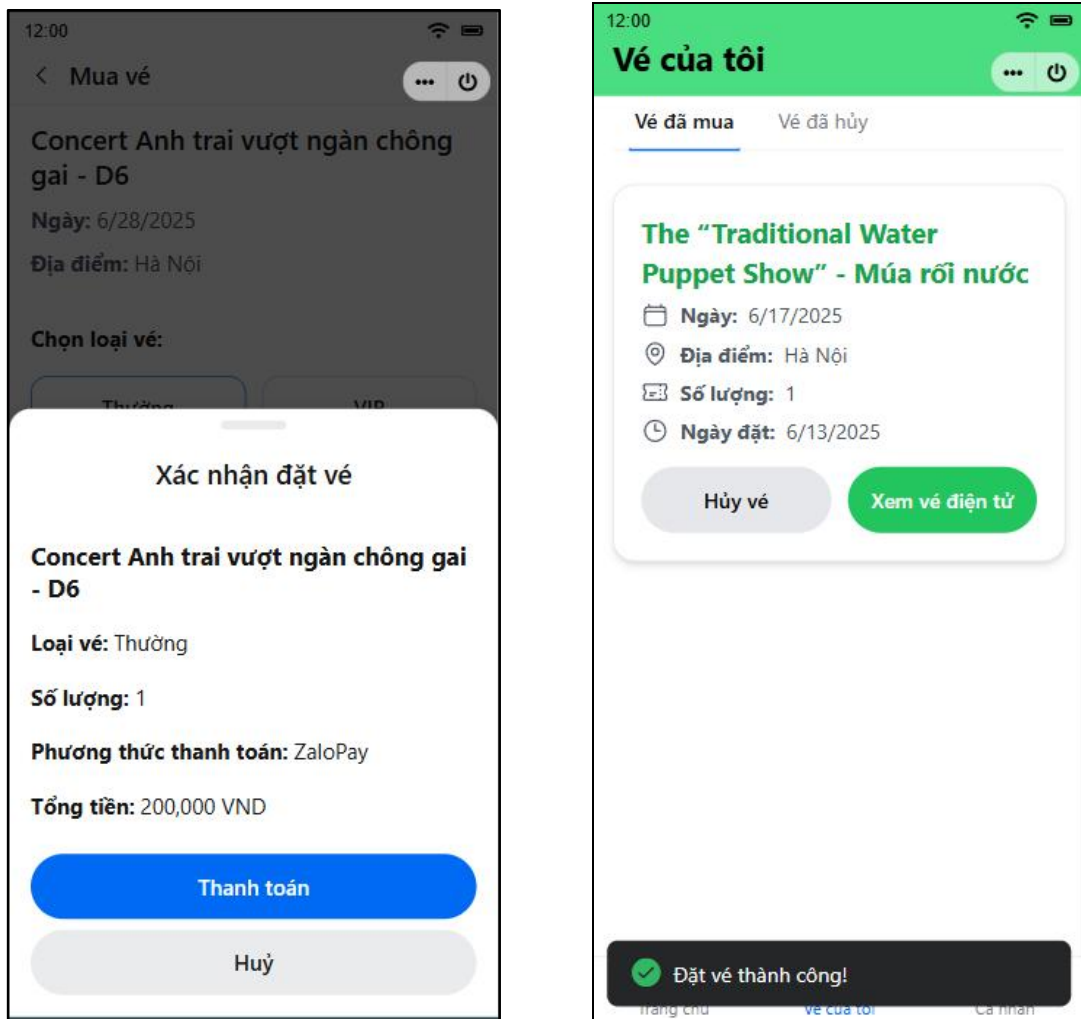
Khi người dùng nhấn vào một sự kiện từ trang chủ, họ sẽ được chuyển đến trang **Chi tiết sự kiện**. Trang này cung cấp thông tin đầy đủ và chi tiết về sự kiện mà người dùng đã chọn, giúp người dùng có cái nhìn tổng quan nhất trước khi quyết định mua vé.

6.5. Giao diện mua vé:



Hình 6.4.2: Giao diện mua vé

Khi người dùng nhấn vào nút "Mua vé ngay" ở trang Chi tiết sự kiện, họ sẽ được chuyển đến trang **Mua vé**. Tại đây cho phép người dùng lựa chọn loại vé, số lượng vé và phương thức thanh toán để đến bước tiếp theo trong quy trình đặt vé.



Hình 6.4.3: Giao diện thanh toán (bên trái)

Khi người dùng nhấn vào nút "Xác nhận đặt vé". Hệ thống sẽ hiển thị giao diện Xác nhận đặt vé, tại đây người dùng sẽ nhấn nút “Thanh toán” để hoàn tất đặt vé, hoặc nhấn nút “Huỷ” để thoát. Sau khi thanh toán thành công, hệ thống sẽ tự động điều hướng người dùng đến **Vé của tôi**.

6.6. Giao diện Thông tin cá nhân:

12:00

< Cá nhân

Za Ticketing

Cung cấp thông tin chính xác giúp bạn mua vé & xác minh dễ dàng hơn

test1

0312345678

test1@gmail.com

Ngày sinh

mm/dd/yyyy

Giới tính

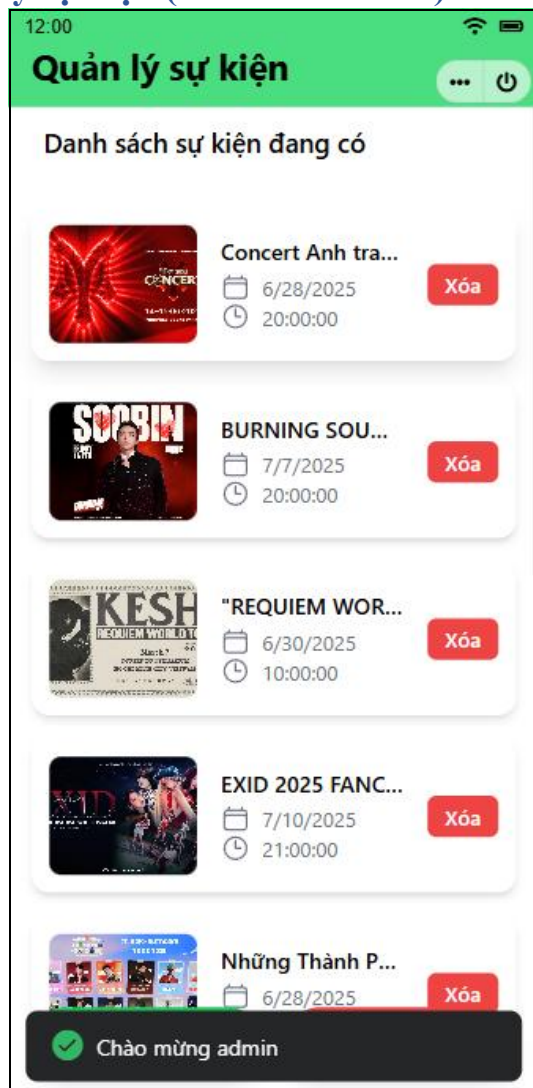
☒ Nam ☐ Nữ ☐ Khác

Hoàn thành

Đăng xuất

Trang **Thông tin cá nhân** cho phép người dùng đã đăng ký xem thông tin tài khoản của mình trên ứng dụng Za Ticketing. Tại đây người dùng có thể đăng xuất khỏi tài khoản của mình.

6.7. Giao diện Quản lý sự kiện (dành cho admin):



Hình 6.6.1: Giao diện Quản lý sự kiện (dành cho admin)

Khi người dùng đăng nhập bằng tài khoản admin được cấp sẵn, hệ thống sẽ điều hướng người dùng đến giao diện **Quản lý sự kiện (admin)**, tại đây admin có thể thực hiện những thao tác quản lý sự kiện như xem danh sách sự kiện, xóa sự kiện đã có và thêm sự kiện mới.

12:00

< Thêm sự kiện mới

Tên sự kiện

Ngày tổ chức

mm/dd/yyyy

Giờ tổ chức

--:-- --

Địa điểm

Mô tả sự kiện

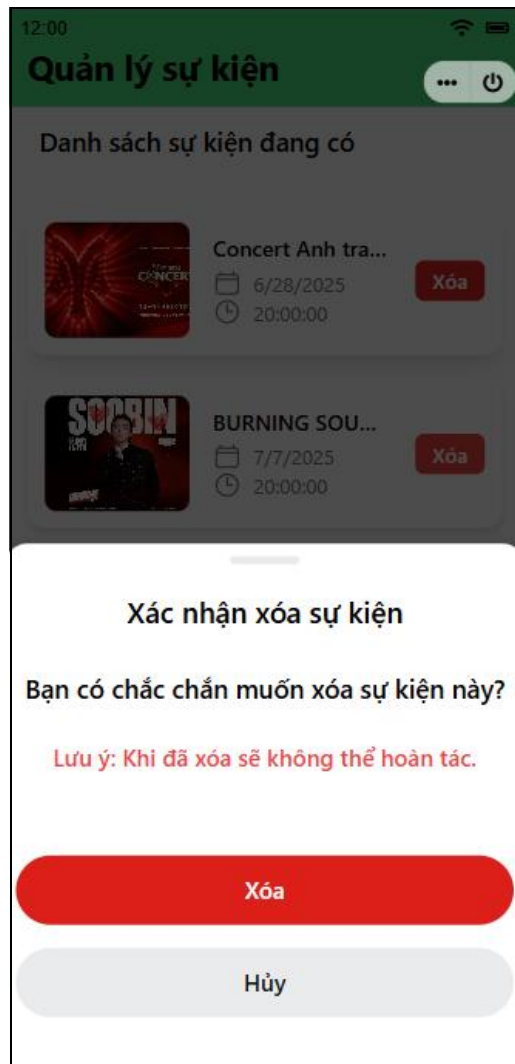
Banner URL

Điểm xếp hạng

0

Hình 6.6.2: Giao diện Thêm sự kiện mới

Khi admin nhấn vào nút “Thêm sự kiện mới”, hệ thống sẽ điều hướng admin đến trang **Thêm sự kiện mới**, tại đây admin có thể nhập các thông tin cần thiết và nhấn tạo sự kiện, sự kiện được tạo xong sẽ tự động thêm vào hệ thống.



Hình 6.6.3: Giao diện xóa sự kiện

Khi admin thực hiện xóa sự kiện trong danh sách các sự kiện đang có, hệ thống sẽ hiển thị giao diện **Xác nhận xóa sự kiện**, tại đây admin sẽ quyết định xóa sự kiện vĩnh viễn hoặc hoàn tác và trở về màn hình chính.

VII. KIỂM THỬ PHẦN MỀM

7.1. Phần đăng ký và đăng nhập

	Step#	Step Detail	Expected Result	Actual Result	Pass/Fail/Not Executed/Suspended
Đăng nhập	1	Đăng nhập vào hệ thống với Gmail và mật khẩu chính xác	Đăng nhập thành công, dẫn đến trang chủ của app bán vé	As Expected	Pass
	2	Đăng nhập vào hệ thống với tên đăng nhập và mật khẩu không chính xác	Thông báo lỗi, người dùng không thể đăng nhập	As Expected	Pass
Đăng kí	3	Tài khoản Admin (được cấp sẵn) khi đăng nhập với gmail và mật khẩu chính xác	Đăng nhập thành công, dẫn đến trang quản lý sự kiện của Admin	As Expected	Pass
	4	Tài khoản User đăng ký với đầy đủ thông tin và yêu cầu, không bị trùng thông tin với tài khoản đã tồn tại trong database	Tạo tài khoản thành công, người dùng có thể đăng nhập với thông tin vừa tạo	As Expected	Pass
	5	Tài khoản User đăng ký với đầy đủ thông tin và yêu cầu, nhưng bị trùng tên hoặc gmail với tài khoản đã tồn tại trong database	Thông báo lỗi tên đăng nhập hoặc gmail đã tồn tại	As Expected	Pass
	6	Tạo tài khoản không đầy đủ thông tin	Thông báo lỗi thiếu thông tin cần thiết, tạo tài khoản không thành công	As Expected	Pass
Đăng	7	Đăng xuất khỏi tài	Đăng xuất thành	As	Pass

xuất		khoản hiện tại	công, dẫn đến màn hình Đăng nhập	Expected	
------	--	----------------	----------------------------------	----------	--

7.2. Phần quản lý sự kiện

	Step#	Step Detail	Expected Result	Actual Result	Pass/Fail/Not Executed/Suspended
Thêm	8	Admin thêm mới các sự kiện sắp tới	Hiện thị form điền thông tin, hình ảnh của sự kiện	As Expected	Pass
Xóa	9	Admin xóa các sự kiện hiện có	Bấm vào nút “Xóa” sẽ hiện thị thông báo xác thực xóa sự kiện, bấm xóa sẽ hoàn tất việc xóa sự kiện	As Expected	Pass

7.3. Phần quy trình mua hàng của người dùng

	Step#	Step Detail	Expected Result	Actual Result	Pass/Fail/Not Executed/Suspended
Xem	10	Khi User nhấp vào các mục sự kiện hiển thị ở trang chủ sẽ hiện thị chi tiết thông tin về vé	Hệ thống chuyển sang trang “Chi tiết sự kiện” để người dùng xem thông tin	As Expected	Pass
Mua	11	Khi User bấm mua vé ở trang “Chi tiết sự kiện” sẽ hiện thị trang xác nhận để mua vé	Hệ thống chuyển sang trang “Mua vé” khi người dùng bấm mua vé. Người dùng chọn các phương thức, số vé, loại vé và xác nhận mua vé.	As Expected	Pass

Hủy	12	Người dùng khi không muốn mua vé thì nhấn nút quay lại	Hệ thống sẽ quay về các trang ban đầu khi người dùng nhấp vào.	As Expected	Pass
Tìm kiếm	13	Người dùng có thể tìm sự kiện thông qua chức năng tìm kiếm	Có thể tìm kiếm sự kiện nhanh chóng bằng cách nhập từ khóa của sự kiện và hiển thị thông tin	As Expected	Pass
Thanh toán	14	Khi người dùng đã xác nhận mua vé, sẽ hiển thị modal chứa thông tin thanh toán	Hệ thống sẽ hiển thị trang để người dùng thấy thông tin thanh toán để hoàn tất việc mua vé	As Expected	Pass

7.4. Phần quy trình xem vé và tùy chỉnh của người dùng

	Step#	Step Detail	Expected Result	Actual Result	Pass/Fail/Not Executed/Suspended
Xem	15	Khi User nhấp vào trang “Vé của tôi” sẽ hiển thị các vé đã mua hoặc đã hủy	Hệ thống chuyển sang trang “Vé của tôi” để người dùng xem thông tin vé	As Expected	Pass
Xem (QR)	16	Khi User bấm “Xem vé điện tử” sẽ hiển thị Qr code	Hệ thống sẽ hiển thị modal gồm Qr code và tên sự kiện	As Expected	Pass

Hủy	17	Khi User bấm hủy vé sẽ hiển thị thông báo “Xác nhận hủy vé” . Khi hoàn tất xác nhận sẽ hiển thị loại vé đã hủy ở tab “Vé đã hủy”	Hệ thống sẽ thêm thông tin vé đã hủy khi người dùng nhấp vào đồng thời cập nhật số lượng vé còn lại trong hệ thống	As Expected	Pass
-----	----	--	--	-------------	------

VIII. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết quả đạt được:

Sau thời gian triển khai, dự án đã hoàn thành và đạt được những kết quả sau:

1. Hiểu rõ quy trình mua vé trực tuyến.
2. Xây dựng Zalo MiniApp “Za Ticketing”
3. Tạo được nhiều tài khoản mua vé khác nhau, không trùng lặp thông tin đăng ký.
4. Hiện thị danh sách sự kiện và đặt vé trực tuyến với giao diện thân thiện, dễ sử dụng với người dùng.
5. Mua vé mới và hủy vé đã đặt đối với user.
6. Xem danh sách sự kiện đang có, quản lý thêm sự kiện mới và xóa sự kiện đang có dành cho admin.
7. Tự động tạo vé mã QR hỗ trợ cho check-in sự kiện.
8. Thời gian đặt vé được rút ngắn so với phương pháp truyền thống.
9. Quản lý thông tin người dùng dễ dàng.
10. Kiểm thử phần mềm được thực hiện ở cấp độ lập trình.
11. Tổ chức dữ liệu theo mô hình quan hệ hợp lý và tách biệt quyền của user với admin.

Hạn chế của đề tài:

Chưa tích hợp được API thanh toán(ví dụ: Zalo Pay, Momo) vào nghiệp vụ mua vé.

Vẫn còn nhiều ý tưởng chưa được phát triển vào dự án: bảo hành, dịch vụ khách hàng, quản lý doanh thu...

Mini App phụ thuộc vào (cookie) của trình duyệt, nếu token hết hạn, người dùng cần đăng nhập lại.

Hướng phát triển của đề tài:

Tuy nhiên, với những ràng buộc về thời gian, kiến thức, kinh nghiệm triển khai thực tế và giới hạn của phạm vi môn học, nhóm chúng em hiện mới chỉ xây dựng một phiên bản Mini App bán vé sự kiện với các tính năng đơn giản, tập trung vào việc cung cấp trải nghiệm mua vé trực tuyến cơ bản. Hệ thống ở giai đoạn hiện tại có thể chưa đạt đến mức độ linh hoạt cao, khả năng xử lý tối ưu cho lượng truy cập lớn và các biện pháp bảo mật chuyên sâu như một ứng dụng thương mại hoàn chỉnh.

Trong tương lai, nhóm dự định phát triển và mở rộng theo chiều hướng như sau:

- Về mặt chức năng, phát triển hệ thống review và đánh giá sau khi tham dự sự kiện nhằm nâng cao trải nghiệm khách hàng về thanh toán, việc bổ sung các phương thức như Momo, Viettel Pay, ... sẽ giúp mở rộng đối tượng sử dụng.
- Phân quyền truy cập chi tiết: Cho phép phân quyền chi tiết hơn cho các vai trò khác nhau trong quá trình quản lý sự kiện (ví dụ: quản lý địa điểm, quản lý khách mời).
- Hệ thống thông báo: Gửi thông báo nhắc nhở về sự kiện đã mua, thông tin cập nhật về sự kiện, hoặc các ưu đãi đặc biệt.
- Hỗ trợ khách hàng trực tuyến: Tích hợp chatbot hoặc kênh hỗ trợ trực tiếp để giải đáp thắc mắc và hỗ trợ người dùng trong quá trình mua vé.
- Nghiên cứu và áp dụng các công nghệ mới: Cập nhật và áp dụng các công nghệ mới nổi như trí tuệ nhân tạo (AI) để cải thiện khả năng gợi ý sự kiện, phân tích xu hướng người dùng và tối ưu hóa trải nghiệm.

TÀI LIỆU THAM KHẢO

- [1] ThS Châu Văn Vân. 2025. Bài giảng Nhập môn Công nghệ phần mềm
- [2] Introduction to database systems - UET: <https://uet.vnu.edu.vn/~chaunh/slide/csdl.htm>
- [3] React docs: <https://react.dev/learn>
- [4] ZaUI docs: <https://mini.zalo.me/documents/zaui/>
- [5] W3Schools (React + MySQL docs):
<https://www.w3schools.com/react/default.asp>
<https://www.w3schools.com/mysql/default.asp>
- [6] npm Docs: <http://docs.npmjs.com/>
- [7] Express Docs: <https://expressjs.com/en/guide/routing.html>
- [8] TailwindCSS Docs: <https://tailwindcss.com/docs/installation/using-vite>

BẢNG PHÂN CÔNG VÀ ĐÁNH GIÁ

Họ và tên	MSSV	Nhiệm vụ	Đánh giá của nhóm	Đánh giá của GV
Trần Nguyễn Bá Huy (leader)	N22DCPT037	Xây dựng giao diện. Xây dựng các tính năng. Thiết kế và triển khai Database. Hỗ trợ vẽ Use case. Viết báo cáo.	10	
Nguyễn Đăng Khoa	N22DCPT045	Xây dựng giao diện. Xây dựng các tính năng. Xây dựng testcase. Kiểm thử phần mềm.	10	
Vũ Hoàng Quang Dũng	N22DCPT011	Phân tích + xây dựng kiến trúc hệ thống. Thiết kế giao diện. Làm slide thuyết trình Viết báo cáo.	10	