

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN
NHẬP MÔN CÔNG NGHỆ PHẦN MỀM
XÂY DỰNG MINI APP HỖ TRỢ BÁN VÉ ĐĂNG KÍ THAM GIA
SỰ KIỆN**

Giáo viên hướng dẫn: Châu Văn Vân

Sinh viên thực hiện:

Nguyễn Văn Hợp N22DCPT034

Lê Đức Hải N22DCPT023

Vũ Quang Long N22DCPT052

Lớp : D22CQPTUD01-N

TPHCM. Ngày 14 tháng 6 năm 2025

LỜI CẢM ƠN

Nhóm chúng em xin được gửi lời cảm ơn chân thành và sâu sắc nhất đến **thầy Châu Văn Vân** – người đã tận tâm giảng dạy và đồng hành cùng chúng em trong suốt học phần **Nhập môn Công nghệ Phần mềm**. Trong thời gian học với thầy, chúng em không chỉ tiếp thu được những kiến thức lý thuyết nền tảng về quy trình phát triển phần mềm mà còn có cơ hội áp dụng thực tế thông qua việc triển khai một dự án nhóm – xây dựng **miniapp hỗ trợ bán vé sự kiện**.

Với sự hướng dẫn tận tình của thầy, nhóm chúng em đã học được cách phân tích yêu cầu người dùng, thiết kế sơ đồ hệ thống, xác định các thực thể và mối quan hệ trong cơ sở dữ liệu, cũng như xây dựng các chức năng quan trọng như tìm kiếm sự kiện, đặt vé, thanh toán và quản lý tài khoản. Qua đó, chúng em nhận ra vai trò quan trọng của từng giai đoạn trong quy trình phát triển phần mềm – từ phân tích, thiết kế đến hiện thực hóa và kiểm thử.

Thầy luôn tạo điều kiện để chúng em làm việc nhóm một cách hiệu quả, khuyến khích chúng em tư duy phản biện, chủ động học hỏi và giải quyết vấn đề. Những buổi học của thầy không chỉ mang lại kiến thức chuyên môn mà còn truyền cảm hứng để chúng em tiếp tục theo đuổi lĩnh vực công nghệ phần mềm một cách nghiêm túc và bài bản.

Chúng em thật sự biết ơn sự hỗ trợ, góp ý và sự nghiêm khắc đúng lúc của thầy trong suốt quá trình học và làm dự án. Môn học này đã giúp chúng em có cái nhìn rõ ràng hơn về công việc của một kỹ sư phần mềm và chuẩn bị tốt hơn cho các môn chuyên ngành sau này.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn thầy và kính chúc thầy luôn dồi dào sức khỏe, hạnh phúc và tiếp tục thành công trong sự nghiệp giảng dạy. Chúng em hy vọng sẽ có thêm nhiều sinh viên được truyền cảm hứng và trưởng thành từ những bài học quý báu mà thầy mang lại.

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the paper.

Giảng viên hướng dẫn

BẢNG PHÂN CÔNG CÔNG VIỆC

HỌ TÊN	MSSV	LỚP	CÔNG VIỆC	ĐÁNH GIÁ
Nguyễn Văn Hợp	N22DCPT034	D22CQPTUD01-N	Code Danh mục, Admin, Tài khoản. Viết báo cáo , viết powerpoint ,thuyết trình.	9đ
Lê Đức Hải	N22DCPT023	D22CQPTUD01-N	Code phân Đăng ký, Đăng nhập, Trang chủ, Thanh toán Chuyển khoản. Tìm tư liệu báo cáo, thuyết trình ,viết powerpoint,chỉnh sửa code, báo cáo lần cuối.	10đ
Vũ Quang Long	N22DCPT052	D22CQPTUD01-N	Code Giỏ hàng, lịch sử, chi tiết sự kiện và Đặt vé ,thuyết trình , viết powerpoint ,chỉnh sửa powerpoint .	9đ

Mục lục

Chương I: Giới thiệu.....	7
1. Giới thiệu đề tài	7
2. Mục đích của miniapp cần thiết kế.....	7
3. Công nghệ sử dụng	8
4. Tìm hiểu rõ hơn về Zalo miniapp	10
Chương 2. Phân tích hệ thống	12
2.1. Thu thập và phân tích yêu cầu.....	12
2.2. Các lớp trong hệ thống.....	14
2.3. Sơ đồ usecase tổng quan	17
2.4. Sơ đồ Usecase chi tiết.....	18
2.4.1. Mô tả:	18
2.4.2. Mối quan hệ giữa các usecase	23
2.4.3. Sơ đồ hoạt động của các usecase chính	25
2.5. Xây dựng danh sách các từ khóa chuyên môn.....	28
2.6. Scenario	30
2.7. Xây dựng sơ đồ ERD	35
2.8. Sơ đồ tuần tự.....	39
Chương 3. Thiết kế Cơ sở dữ liệu.....	43
1. Sơ đồ cơ sở dữ liệu	43
2. Xây dựng CRC dựa trên các lớp điều khiển	44
3. Thiết kế thuộc tính và phương thức cho lớp	45
4. Danh sách mô tả chi tiết các bảng dữ liệu	47
Chương 4. Giao diện và tính năng.....	51
1. Trang đăng nhập, đăng ký	51
2. Trang chủ.....	53
3. Trang chi tiết sự kiện	54

4.Trang danh mục	59
5.Trang lịch sử mua vé.....	60
6.Trang tài khoản người dùng	61
Chương 5.Cài đặt và kiểm thử.....	62
1. Môi trường cài đặt	62
2. Kịch bản thử nghiệm.....	62
3. Kết quả thử nghiệm.....	63
Chương 6.Tổng kết và hướng phát triển	64
1.Kết quả	64
2.Hạn chế.....	64
3.Hướng phát triển	64
Tài liệu tham khảo:	65

Chương I: Giới thiệu

1. Giới thiệu đề tài

Trong thời đại số hóa hiện nay, nhu cầu tham gia các sự kiện giải trí, hội thảo, workshop, concert... ngày càng gia tăng, kéo theo đó là yêu cầu về một hệ thống bán vé tiện lợi, nhanh chóng và hiệu quả. Đề tài "Mini App Bán Vé Sự Kiện Online" được triển khai nhằm xây dựng một nền tảng ứng dụng nhỏ gọn, thân thiện với người dùng, cho phép người tổ chức sự kiện dễ dàng đăng tải thông tin và bán vé, trong khi người dùng có thể tìm kiếm, đặt mua và thanh toán vé sự kiện trực tuyến một cách tiện lợi.

Ứng dụng được thiết kế với các chức năng cơ bản như: hiển thị danh sách sự kiện, chi tiết sự kiện, đặt vé, thanh toán online (hoặc qua mã QR), và quản lý vé đã mua. Ngoài ra, mini app còn hỗ trợ tạo mã vé điện tử (e-ticket) để người dùng có thể sử dụng trực tiếp khi tham gia sự kiện mà không cần in vé giấy.

Đề tài không chỉ mang tính thực tiễn cao mà còn góp phần thúc đẩy quá trình chuyển đổi số trong lĩnh vực tổ chức sự kiện, hướng tới một hệ sinh thái quản lý sự kiện thông minh và hiện đại.

Vì vậy, nhóm chúng em quyết định lựa chọn đề tài này để phát triển một mini app có tính ứng dụng thực tiễn, dễ sử dụng và phù hợp với xu hướng công nghệ hiện nay.

2. Mục đích của miniapp cần thiết kế

Trước khi xây dựng một mini app, chúng ta cần hình dung rõ ràng về mục tiêu và định hướng mà miniapp đó sẽ phục vụ. Nếu không có chủ đích cụ thể ngay từ đầu, việc thiết kế sẽ dễ bị lan man, sai lệch mục tiêu ban đầu và cuối cùng dẫn đến một sản phẩm không thể sử dụng hiệu quả. Việc lên kế hoạch rõ ràng, thiết kế cẩn thận, có định hướng chiến lược là yếu tố then chốt dẫn đến sự thành công của một miniapp.

Trong bối cảnh hiện nay, thương mại điện tử tại Việt Nam đang phát triển mạnh mẽ, tuy nhiên nhiều cửa hàng vẫn chưa tận dụng tốt sức mạnh của Internet và nền tảng app mobile để mở rộng kinh doanh. Đề tài này nhằm mục đích thiết kế một mini app bán vé sự kiện online – một nền tảng giúp các đơn vị tổ chức sự kiện dễ dàng tiếp cận khách hàng, quảng bá chương trình và bán vé nhanh chóng, hiệu quả. Đồng thời, người dùng có thể tìm kiếm các sự kiện phù hợp với nhu cầu, đặt mua vé thuận tiện, thanh toán linh hoạt và nhận vé ngay qua nền tảng số.

Việc xây dựng mini app này không chỉ giúp hiện đại hóa cách thức tiếp cận khách hàng mà còn giảm thiểu nhiều chi phí như: in ấn vé, quảng cáo truyền thống, vận chuyển... Hệ thống còn giúp tiết kiệm thời gian giao dịch, tăng tính minh bạch và chuyên nghiệp trong quá trình vận hành sự kiện. Trong kỷ nguyên số, một mini app hoạt động hiệu quả chính là công cụ quảng bá và kinh doanh mạnh mẽ nhất.

Trước khi bắt tay vào xây dựng, nhóm chúng em đã xác định rõ các yếu tố quan trọng như:

- Đối tượng người dùng: bao gồm cả người tổ chức sự kiện (admin) và người tham gia sự kiện (khách hàng).
- Mục tiêu : cung cấp nền tảng bán vé online tiện lợi, dễ sử dụng, hoạt động ổn định.
- Nội dung chính: danh sách sự kiện, thông tin chi tiết từng sự kiện, chức năng đặt vé, thanh toán, quản lý vé và gửi vé điện tử.
- Hệ thống thông tin: được thiết kế rõ ràng, dễ truy cập, thân thiện với người dùng ở mọi độ tuổi.

Chúng em cũng nhận thức được rằng việc thiết kế một mini app không chỉ là một dự án ngắn hạn, mà là một quá trình dài hạn cần duy trì, cập nhật thường xuyên và cải tiến liên tục để đáp ứng xu hướng thị trường cũng như nhu cầu người dùng.

3.Công nghệ sử dụng

3.1. Visual Studio Code

Visual Studio Code là một trình soạn thảo mã nguồn mở, miễn phí do Microsoft phát triển, hỗ trợ đa nền tảng bao gồm Windows, macOS và Linux. Công cụ này được thiết kế nhằm cung cấp môi trường lập trình nhẹ nhàng, tiện lợi nhưng vẫn đảm bảo đầy đủ các tính năng cần thiết cho việc phát triển phần mềm.

Visual Studio Code hỗ trợ nhiều ngôn ngữ lập trình phổ biến như JavaScript, Python, C++, Java, và nhiều ngôn ngữ khác thông qua hệ thống mở rộng (extensions) đa dạng và phong phú. Ngoài ra, công cụ còn tích hợp sẵn các tính năng quản lý mã nguồn thông qua Git, trình gỡ lỗi (debugger) hiệu quả, cũng như terminal tích hợp, giúp người dùng thực hiện các thao tác lập trình một cách nhanh chóng và thuận tiện.

Đặc biệt, Visual Studio Code có khả năng tùy biến cao về giao diện, phím tắt và cấu hình, đáp ứng tốt nhu cầu của người dùng ở nhiều trình độ khác nhau, từ người mới bắt đầu đến lập trình viên chuyên nghiệp. Nhờ những ưu điểm trên, Visual Studio Code đã trở thành một trong những công cụ lập trình phổ biến và được đánh giá cao trong cộng đồng phát triển phần mềm hiện nay.

3.2.Zalo Mini App

Zalo Mini App – Ứng dụng nhỏ tích hợp trong Zalo

Zalo Mini App là nền tảng cho phép doanh nghiệp và lập trình viên phát triển các ứng dụng nhẹ, hoạt động trực tiếp bên trong ứng dụng Zalo mà không cần người dùng cài đặt thêm. Các Mini App có thể phục vụ nhiều mục đích như đặt lịch, đặt món ăn, chơi game, thanh toán, chăm sóc khách hàng, tra cứu thông tin,...

Mini App sử dụng công nghệ web (HTML, CSS, JavaScript) và được vận hành qua hệ thống API của Zalo để truy cập dữ liệu người dùng, gửi thông báo, và tích hợp với Zalo OA. Người dùng có thể truy cập nhanh qua Zalo OA, mã QR, tin nhắn hoặc các nút menu.

Với lợi thế tiếp cận hơn 70 triệu người dùng Zalo, Zalo Mini App giúp doanh nghiệp triển khai dịch vụ tiện lợi, nhanh chóng, giảm chi phí phát triển và tăng cường tương tác người dùng.

4. Tìm hiểu rõ hơn về Zalo miniapp

4.1. Zalo Mini App là gì?

Zalo Mini App là nền tảng cho phép các nhà phát triển xây dựng các ứng dụng nhỏ (mini-app) chạy bên trong hệ sinh thái Zalo mà không cần người dùng cài đặt thêm ứng dụng mới. Đây là một hướng đi tương tự như WeChat Mini Program (Trung Quốc) hay Apple App Clips, giúp doanh nghiệp nhanh chóng cung cấp dịch vụ cho người dùng.

4.2. Đặc điểm chính của Zalo Mini App

Tính năng	Mô tả
Đa nền tảng	Mini app chạy trên cả iOS, Android và Zalo PC.
Không cần cài đặt	Người dùng truy cập mini app trực tiếp từ Zalo.
Tận dụng tài khoản Zalo	Đăng nhập, chia sẻ thông tin, gửi tin nhắn với tài khoản Zalo.
Giao diện nhẹ	Tối ưu cho tốc độ tải nhanh, tương tác mượt.
Tích hợp với Official Account	Hỗ trợ quảng bá mini app qua OA.

4.3. Kiến trúc tổng quan

Zalo Mini App sử dụng mô hình client-server gồm 3 thành phần chính:

1. Zalo Client (App Zalo)

Chạy giao diện frontend của mini app.

Gọi các API của Zalo Platform hoặc Server của bạn.

2. Backend của Nhà phát triển

Xử lý logic nghiệp vụ, lưu trữ dữ liệu người dùng.

Được triển khai bởi bạn, có thể là Node.js, PHP, Python, v.v.

3. Zalo Mini App Platform

Cung cấp SDK, API, các công cụ tích hợp.

Đảm bảo bảo mật và quản lý truy cập mini app.

4.4. Thành phần của một Mini App

Thành phần	Mô tả
Manifest file	Mô tả thông tin ứng dụng (tên, icon, permissions, routes...).
Pages	Các màn hình giao diện viết bằng ZMP (Zalo Mini Program) – giống Vue.js.
API calls	Giao tiếp với Zalo Platform và backend.
Assets	Hình ảnh, font, CSS...
Router	Quản lý điều hướng giữa các trang.

4.5. Ngôn ngữ & Công nghệ sử dụng

- **Ngôn ngữ giao diện:** Gần giống với **Vue.js**.
- **Ngôn ngữ server:** Không giới hạn – bạn có thể dùng Node.js, PHP, Java,...
- **Dữ liệu truyền:** JSON qua HTTP/HTTPS.
- **Giao tiếp frontend-backend:** Gọi API RESTful hoặc GraphQL.

Chương 2. Phân tích hệ thống

2.1. Thu thập và phân tích yêu cầu

2.1.1. Kỹ thuật thu thập yêu cầu

- Khảo sát người dùng Zalo hiện tại (qua OA chatbot, khảo sát đơn giản).
- Phỏng vấn nhà tổ chức sự kiện để biết quy trình nghiệp vụ.
- Nghiên cứu ứng dụng tương tự như Ticketbox, Eventbrite, MoMo Event,...
- Dùng sơ đồ Use Case và kịch bản người dùng để phân tích luồng hoạt động.

2.1.2. Yêu cầu về chức năng

1. Quản lý sự kiện

- Người dùng có thể xem danh sách các sự kiện đang diễn ra / sắp tới.
- Có thể lọc sự kiện theo địa điểm, thời gian, thể loại (nhạc, hội thảo, thể thao...).
- Có thể xem chi tiết sự kiện: tên, mô tả, thời gian, địa điểm, sơ đồ chỗ ngồi, giá vé.

2. Đặt vé

- Người dùng có thể chọn loại vé (thường, VIP...).
- Có thể chọn vị trí ghế (nếu có sơ đồ).
- Hiển thị thông tin vé, số lượng, giá tiền.

3. Giỏ hàng và thanh toán

- Cho phép người dùng thêm nhiều vé vào giỏ.
- Xác nhận đơn hàng, thanh toán qua ZaloPay hoặc các phương thức được tích hợp.
- Sau thanh toán, gửi thông tin vé qua Zalo OA hoặc mã QR để check-in.

4. Lịch sử đặt vé

- Hiển thị danh sách các vé đã mua.
- Trạng thái vé: Đã thanh toán, Đang xử lý, Đã sử dụng, Hết hạn.
- Có thể xuất hoặc hiển thị mã QR để check-in.

5. Tài khoản người dùng

- Lấy thông tin từ tài khoản Zalo (tên, số điện thoại).
- Cho phép cập nhật thông tin liên hệ (nếu cần).
- Bảo mật và xác minh tài khoản thông qua Zalo ID.

6. Thông báo & tương tác

- Gửi thông báo qua OA khi có vé mới, sự kiện mới, nhắc lịch.
- Cho phép chia sẻ sự kiện lên Zalo chat / Zalo feed.

2.1.3. Yêu cầu phi chức năng

1. Hiệu năng

- Thời gian phản hồi giao diện: dưới 2 giây.
- Tối ưu truy vấn API để giảm độ trễ.

2. Khả năng mở rộng

- Hệ thống có thể mở rộng để thêm nhiều loại sự kiện hoặc nhà tổ chức khác nhau.
- Backend hỗ trợ tăng tải khi số lượng người dùng tăng cao (ví dụ lúc mở bán vé).

3. Tính bảo mật

- Sử dụng token xác thực từ Zalo cho mọi thao tác cá nhân.
- Đảm bảo dữ liệu thanh toán được mã hóa và bảo vệ.
- Chống gian lận vé (check-in trùng, giả vé).

4. Khả năng tương thích

- Mini app phải chạy mượt mà trên các nền tảng Zalo (iOS, Android, Web/PWA nếu có).
- Giao diện thích ứng với nhiều kích thước màn hình.

5. Khả năng bảo trì

- Mã nguồn phải dễ dàng nâng cấp và cập nhật tính năng mới.

- Có log lỗi phía backend để xử lý nhanh khi có sự cố.

6. Trải nghiệm người dùng (UX)

- Giao diện trực quan, dễ thao tác.
- Hạn chế số bước cần thực hiện để hoàn tất đặt vé.
- Cung cấp thông tin rõ ràng về vé và sự kiện.

2.2. Các lớp trong hệ thống

2.2.1.Lớp Thực Thể (Entity Class) - MiniApp Bán Vé Sự Kiện

Tên lớp	Mô tả	Thuộc tính	Phương thức
NguoiDung	Lưu thông tin cá nhân và tài khoản người dùng.	- ID - Họ tên - Email - Số điện thoại - Mật khẩu đã mã hóa - Vai trò (user/admin)	- dangNhap() - dangKy() - capNhatThongTin()
SuKien	Thông tin chi tiết về sự kiện được tổ chức.	- ID - Tên sự kiện - Mô tả - Thời gian - Địa điểm - Hình ảnh - Sơ đồ chỗ ngồi	- capNhatThongTinSuKien() - xoaSuKien()
VeSuKien	Thông tin vé tương ứng với từng sự kiện.	- ID vé - Mã QR - Loại vé - Giá vé - Trạng thái (chưa thanh toán/đã thanh toán/đã check-in)	- taoMaQR() -kiemTraTinhTrangVe()
DonHang	Lưu thông tin đơn hàng của người dùng khi mua vé.	- ID đơn hàng - Ngày đặt - Trạng thái thanh toán - Tổng tiền - ID người dùng - Danh sách vé	- tinhTongTien() - capNhatTrangThaiThanhToan())

ThongBao	Gửi thông báo hệ thống cho người dùng.	<ul style="list-style-type: none"> - ID - Nội dung - Loại thông báo - Ngày gửi - Trạng thái đã đọc 	<ul style="list-style-type: none"> - guiThongBao() - danhDauDaDoc()
LichSuDatVe	Lưu trữ các đơn đặt vé trong quá khứ của người dùng.	<ul style="list-style-type: none"> - ID lịch sử - ID người dùng - Danh sách đơn hàng - Thời gian đặt vé 	<ul style="list-style-type: none"> - hienThiLichSu()

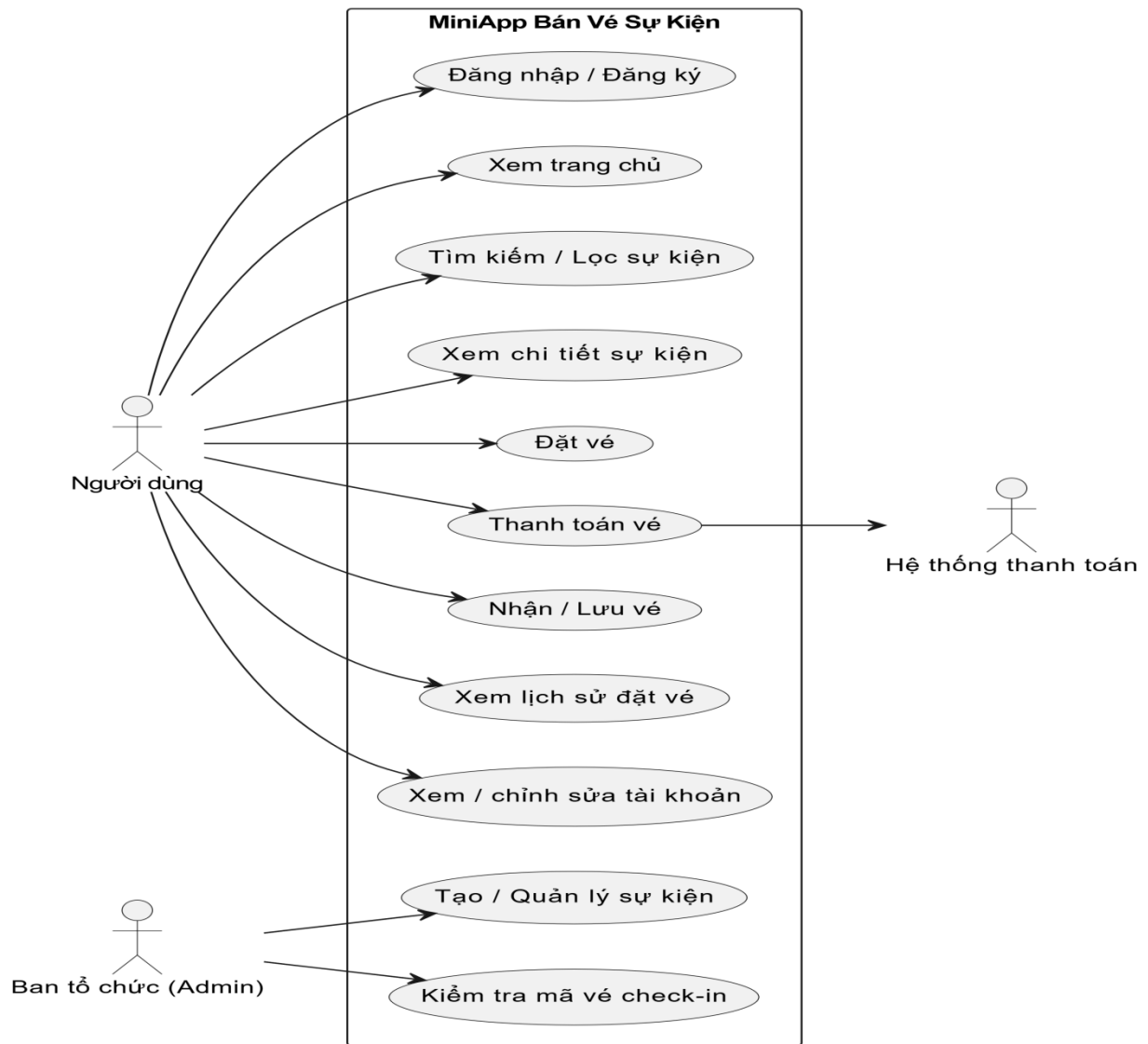
2.2.2.Lớp Điều Khiển (Control Class) - MiniApp Bán Vé Sự Kiện

Tên lớp	Mô tả	Chức năng chính
QuanLyNguoiDungControl	Xử lý đăng nhập, đăng ký và cập nhật thông tin người dùng.	<ul style="list-style-type: none"> - xác thực thông tin đăng nhập - xử lý đăng ký mới - gọi cập nhật thông tin người dùng
QuanLySuKienControl	Điều phối việc hiển thị và cập nhật thông tin sự kiện.	<ul style="list-style-type: none"> - lấy danh sách sự kiện - xử lý tìm kiếm - cập nhật thông tin sự kiện
DatVeControl	Xử lý logic đặt vé và lưu đơn hàng.	<ul style="list-style-type: none"> - tạo đơn đặt vé - tính tổng tiền - xác thực số lượng vé - tạo mã QR cho vé
ThanhToanControl	Điều phối quy trình thanh toán và cập nhật trạng thái đơn hàng.	<ul style="list-style-type: none"> - tích hợp với ZaloPay/VNPay - kiểm tra trạng thái giao dịch - cập nhật đơn hàng sau thanh toán
ThongBaoControl	Quản lý việc gửi thông báo cho người dùng.	<ul style="list-style-type: none"> - tạo nội dung thông báo - gửi thông báo qua ZNS - đánh dấu thông báo đã đọc
LichSuControl	Quản lý và truy xuất lịch sử đặt vé của người dùng.	<ul style="list-style-type: none"> - lấy danh sách đơn hàng theo người dùng - hiển thị thông tin chi tiết lịch sử đặt vé

2.2.3.Lớp biên

Tên lớp	Mô tả	Chức năng giao tiếp
ManHinhDangNhap	Giao diện người dùng để đăng nhập vào hệ thống.	<ul style="list-style-type: none"> - Nhập email và mật khẩu - Hiện thị lỗi xác thực - Gọi xử lý đăng nhập
ManHinhDangKy	Giao diện đăng ký tài khoản người dùng mới.	<ul style="list-style-type: none"> - Nhập thông tin cá nhân - Gửi yêu cầu đăng ký - Hiện thị thông báo kết quả
ManHinhDanhSachSuKien	Hiện thị danh sách các sự kiện theo danh mục hoặc tìm kiếm.	<ul style="list-style-type: none"> - Hiện thị danh sách sự kiện - Hỗ trợ tìm kiếm, lọc - Chọn sự kiện để xem chi tiết
ManHinhChiTietSuKien	Hiện thị thông tin chi tiết của một sự kiện cụ thể.	<ul style="list-style-type: none"> - Hiện thị chi tiết sự kiện, sơ đồ chỗ ngồi - Chọn vé - Nhấn nút đặt vé
ManHinhThanhToan	Giao diện xử lý thanh toán đơn hàng.	<ul style="list-style-type: none"> - Chọn phương thức thanh toán - Xác nhận thông tin - Xử lý trạng thái giao dịch
ManHinhLichSuDatVe	Giao diện hiển thị lịch sử đặt vé của người dùng.	<ul style="list-style-type: none"> - Hiện thị danh sách đơn hàng đã đặt - Xem chi tiết từng vé
ThongBaoUI	Giao diện hiển thị thông báo từ hệ thống.	<ul style="list-style-type: none"> - Hiện thị danh sách thông báo - Đánh dấu đã đọc - Chuyển đến màn liên quan

2.3.Sơ đồ usecase tổng quan



Hình1.Sơ đồ usecase tổng quan

2.4. Sơ đồ Usecase chi tiết

2.4.1. Mô tả:

1. Đăng nhập / Đăng ký

- **Tác nhân:** Người dùng
- **Điều kiện trước:** Người dùng chưa đăng nhập
- **Dòng sự kiện chính:**
 1. Người dùng truy cập miniapp lần đầu.
 2. Hệ thống yêu cầu đăng nhập qua tài khoản Zalo.
 3. Nếu chưa có tài khoản, người dùng tiến hành đăng ký (liên kết tài khoản Zalo hoặc nhập thông tin).
 4. Hệ thống xác thực và chuyển hướng vào trang chủ.
- **Điều kiện sau:** Người dùng đã đăng nhập và có thể sử dụng chức năng của ứng dụng.
- **Dòng thay thế:**
 - 3a. Nếu đăng ký thất bại → hiển thị lỗi.
 - 2a. Người dùng từ chối quyền truy cập → không vào được hệ thống.

2. Xem trang chủ

- **Tác nhân:** Người dùng
- **Điều kiện trước:** Người dùng đã đăng nhập
- **Dòng sự kiện chính:**
 1. Người dùng truy cập ứng dụng.
 2. Hệ thống hiển thị danh sách sự kiện nổi bật, sắp diễn ra, đề xuất.
- **Điều kiện sau:** Người dùng thấy danh sách các sự kiện.
- **Ngoại lệ:** Không có sự kiện nào → hiển thị thông báo "Chưa có sự kiện".

3. Tìm kiếm / Lọc sự kiện

- **Tác nhân:** Người dùng
- **Dòng sự kiện chính:**
 1. Người dùng nhập từ khóa tìm kiếm hoặc chọn bộ lọc (theo thể loại, địa điểm, thời gian).
 2. Hệ thống hiển thị kết quả phù hợp.

- **Điều kiện trước:** Có dữ liệu sự kiện
- **Điều kiện sau:** Người dùng chọn được sự kiện muốn xem.
- **Ngoại lệ:** Không tìm thấy kết quả → thông báo “Không tìm thấy sự kiện phù hợp”.

4. Xem chi tiết sự kiện

- **Tác nhân:** Người dùng
- **Dòng sự kiện chính:**
 1. Người dùng chọn một sự kiện trong danh sách.
 2. Hệ thống hiển thị thông tin chi tiết: tên, mô tả, thời gian, địa điểm, sơ đồ chỗ ngồi, giá vé.
- **Điều kiện trước:** Có sự kiện để hiển thị.
- **Điều kiện sau:** Người dùng có thể tiếp tục đặt vé.

5. Đặt vé

- **Tác nhân:** Người dùng
- **Dòng sự kiện chính:**
 1. Người dùng chọn loại vé, số lượng và vị trí chỗ ngồi (nếu có).
 2. Hệ thống kiểm tra số lượng vé còn lại.
 3. Thêm vào giỏ hàng.
- **Điều kiện trước:** Sự kiện còn vé.
- **Điều kiện sau:** Vé được thêm vào giỏ hàng.
- **Ngoại lệ:** Hết vé hoặc số lượng không đủ → thông báo lỗi.

6. Thanh toán vé

- **Tác nhân:** Người dùng, Hệ thống thanh toán
- **Dòng sự kiện chính:**
 1. Người dùng xác nhận giỏ hàng.
 2. Chọn phương thức thanh toán (ZaloPay,...).
 3. Hệ thống chuyển hướng sang cổng thanh toán.
 4. Nhận kết quả từ hệ thống thanh toán.
- **Điều kiện trước:** Có vé trong giỏ hàng.

- **Điều kiện sau:** Thanh toán thành công → cập nhật trạng thái vé.
- **Ngoại lệ:** Thanh toán thất bại → thông báo lỗi và cho phép thử lại.

7. Nhận / Lưu vé

- **Tác nhân:** Người dùng
- **Dòng sự kiện chính:**
 1. Sau khi thanh toán thành công, hệ thống gửi vé đến người dùng.
 2. Vé bao gồm thông tin, mã QR, thời gian sự kiện.
- **Điều kiện trước:** Thanh toán thành công.
- **Điều kiện sau:** Người dùng có thể sử dụng vé để check-in.

8. Xem lịch sử đặt vé

- **Tác nhân:** Người dùng
- **Dòng sự kiện chính:**
 1. Người dùng mở mục "Lịch sử vé".
 2. Hệ thống hiển thị danh sách vé đã mua: trạng thái (đã dùng, chưa dùng).
- **Điều kiện trước:** Người dùng đã từng đặt vé.
- **Điều kiện sau:** Có thể chọn vé để xem hoặc tải lại mã QR.

9. Xem / chỉnh sửa tài khoản

- **Tác nhân:** Người dùng
- **Dòng sự kiện chính:**
 1. Người dùng truy cập mục tài khoản.
 2. Cập nhật thông tin liên hệ (email, SĐT, tên...).
- **Điều kiện trước:** Người dùng đã đăng nhập.
- **Điều kiện sau:** Thông tin tài khoản được cập nhật.

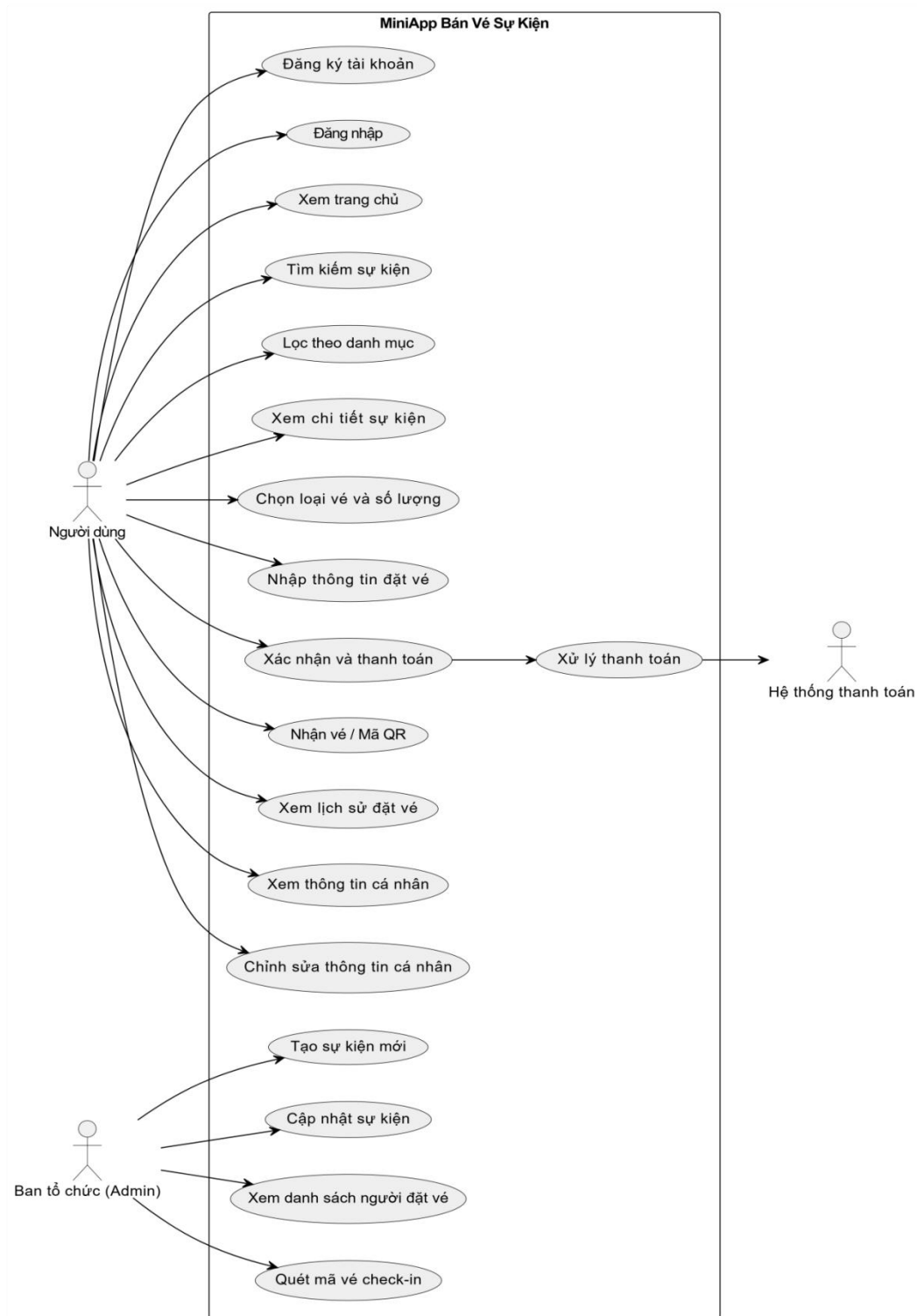
10. Tạo / Quản lý sự kiện

- **Tác nhân:** Ban tổ chức (Admin)
- **Dòng sự kiện chính:**
 1. Admin đăng nhập hệ thống.

2. Tạo sự kiện: nhập tên, mô tả, thời gian, số lượng vé, giá vé, sơ đồ chỗ ngồi.
 3. Cập nhật hoặc xoá sự kiện nếu cần.
- **Điều kiện trước:** Có quyền Admin.
 - **Điều kiện sau:** Sự kiện được hiển thị cho người dùng.
 - **Ngoại lệ:** Thông tin không hợp lệ → không lưu được.

11. Kiểm tra mã vé check-in

- **Tác nhân:** Ban tổ chức (Admin)
- **Dòng sự kiện chính:**
 1. Người dùng đến sự kiện, cung cấp mã QR.
 2. Admin quét mã bằng hệ thống check-in.
 3. Hệ thống xác thực mã và cập nhật trạng thái vé.
- **Điều kiện trước:** Vé chưa được sử dụng.
- **Điều kiện sau:** Người dùng được vào sự kiện.
- **Ngoại lệ:** Mã không hợp lệ hoặc đã dùng → từ chối check-in.



Hình2.Sơ đồ usecase chi tiết

2.4.2.Mối quan hệ giữa các usecase

Use Case chính	Quan hệ	Use Case liên quan	Giải thích
Đăng ký / Đăng nhập	Include	Quản lý tài khoản người dùng	Người dùng phải đăng nhập để cập nhật thông tin cá nhân.
	Include	Đặt vé, Lưu vé, Xem lịch sử, v.v.	Là điều kiện cần để truy cập các chức năng liên quan tài khoản.
Xem trang chủ	Include	Tìm kiếm / Lọc sự kiện	Từ trang chủ người dùng có thể thực hiện tìm kiếm, lọc sự kiện.
Tìm kiếm / Lọc sự kiện	Include	Đăng nhập	Cần xác định người dùng để hiển thị thông tin cá nhân hóa (ưu đãi, lịch sử).
Xem chi tiết sự kiện	Include	Tìm kiếm / Lọc sự kiện	Phải tìm được sự kiện trước khi xem chi tiết.
	Extend	Đặt vé	Người dùng có thể đặt vé từ màn hình chi tiết.
Đặt vé	Include	Xem chi tiết sự kiện	Phải xem thông tin sự kiện trước khi đặt vé.
	Extend	Thanh toán vé	Sau khi đặt vé có thể tiến hành thanh toán.
Thanh toán vé	Include	Đặt vé	Phải có thông tin vé trước khi thanh toán.
	Extend	Gửi thông báo /	Sau thanh toán có thể gửi xác nhận vé,

		nhắc nhở	QR code, nhắc nhở sự kiện.
Nhận / Lưu vé	Include	Thanh toán vé	Chỉ có thể lưu vé sau khi đã thanh toán thành công.
Xem lịch sử đặt vé	Include	Đăng nhập	Lịch sử đặt vé thuộc về tài khoản cá nhân.
Xem / chỉnh sửa tài khoản	Include	Đăng nhập	Chỉ người dùng đăng nhập mới chỉnh sửa được thông tin.
Tạo / Quản lý sự kiện	Include	Đăng nhập (Admin)	Ban tổ chức cần xác thực để tạo, sửa, xóa sự kiện.
Kiểm tra mã vé (check-in)	Include	Nhận / Lưu vé	Mỗi vé chứa mã, chỉ có thể quét kiểm tra khi đã lưu vé.

Tổng quan các nhóm Use Case và quan hệ

1. Hệ thống người dùng (User):

- Luồng chính: Đăng ký/Đăng nhập → Tìm kiếm sự kiện → Xem chi tiết → Đặt vé → Thanh toán → Lưu vé → Check-in
- Luồng phụ: Xem lịch sử, Chỉnh sửa tài khoản

2. Hệ thống ban tổ chức (Admin):

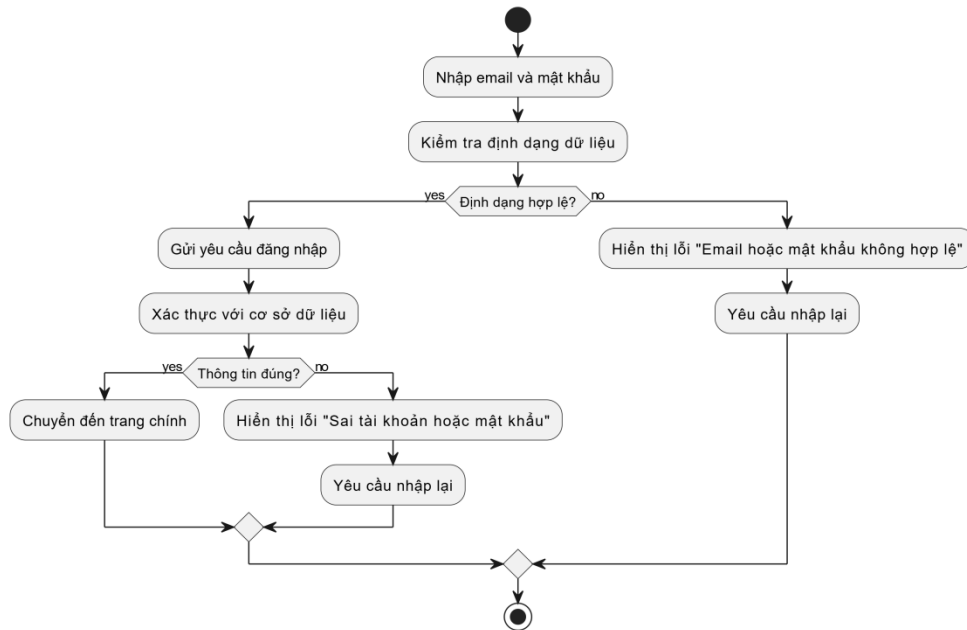
- Đăng nhập → Tạo / Quản lý sự kiện → Check-in (quét vé)

3. Hệ thống hỗ trợ:

- Gửi thông báo, nhắc nhở là use case mở rộng (Extend) từ Thanh toán, có thể gửi xác nhận, QR code hoặc nhắc sự kiện sắp diễn ra

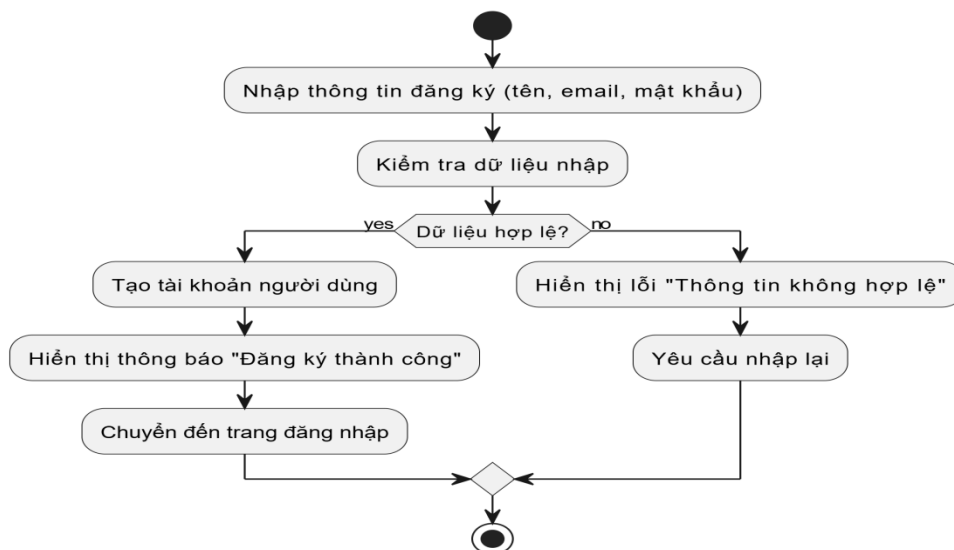
2.4.3.Sơ đồ hoạt động của các usecase chính

1.Usecase quy trình đăng nhập



Hình3 .Sơ đồ usecase quy trình đăng nhập

2.Usecase quy trình đăng ký



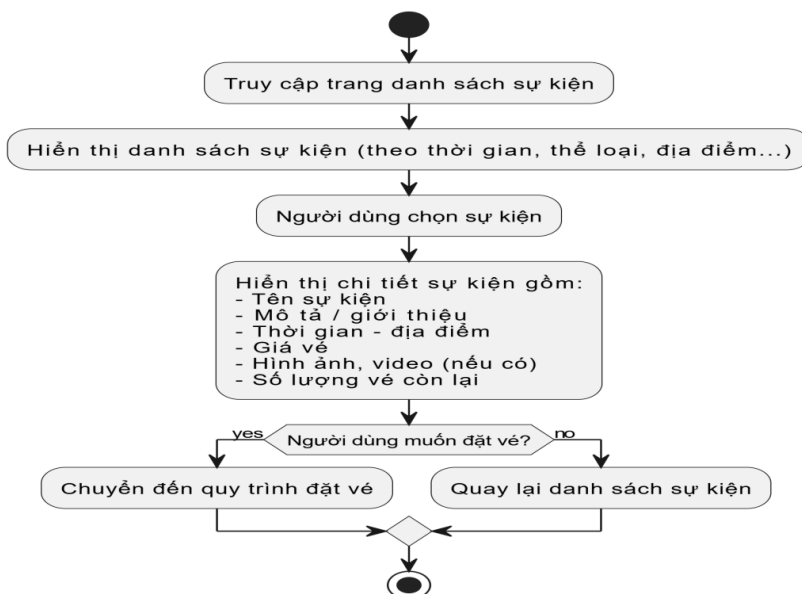
Hình4 .Sơ đồ usecase quy trình đăng ký

3.Usecase Quy trình mua vé sự kiện



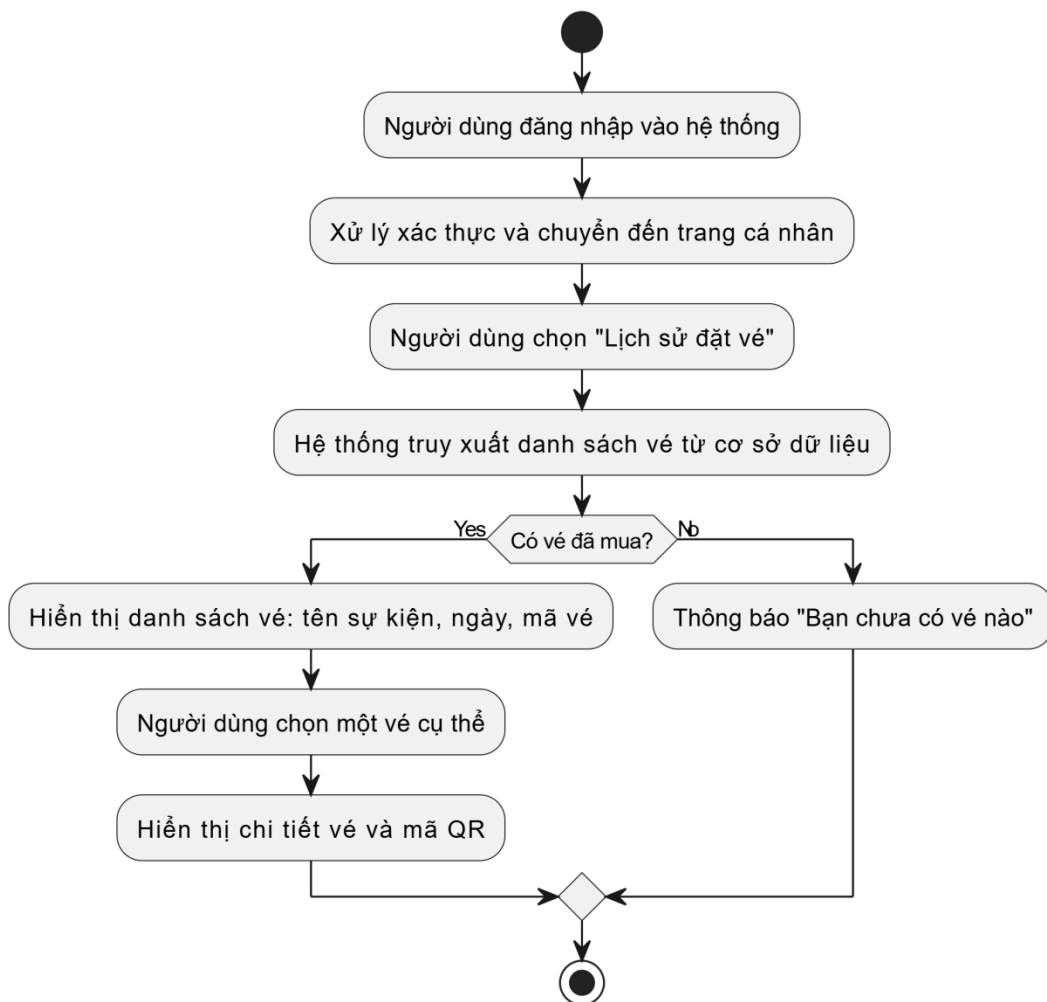
Hình5 .Sơ đồ usecase quy trình mua vé sự kiện

4.Usecase quy trình xem chi tiết danh sách sự kiện



Hình6 .Sơ đồ usecase quy trình xem chi tiết danh sách sự kiện

5.Usecase Quy trình Kiểm tra vé đã mua



Hình7.Sơ đồ usecase quy trình kiểm tra vé đã mua

2.5. Xây dựng danh sách các từ khóa chuyên môn

1. Từ khóa về phân tích & thiết kế hệ thống

Từ khóa	Giải nghĩa
Use Case	Trường hợp sử dụng – mô tả chức năng mà hệ thống cung cấp cho người dùng.
Actor (Tác nhân)	Thực thể tương tác với hệ thống (người dùng, admin, hệ thống khác).
Activity Diagram	Biểu đồ hoạt động – mô tả luồng xử lý nghiệp vụ.
Class Diagram	Biểu đồ lớp – mô tả các thực thể, thuộc tính và mối quan hệ.
Functional Requirements	Yêu cầu chức năng – những gì hệ thống phải thực hiện.
Non-functional Requirements	Yêu cầu phi chức năng – các tiêu chí như hiệu năng, bảo mật, khả dụng.
Include / Extend	Mối quan hệ mở rộng hoặc bắt buộc giữa các Use Case.
Sequence Diagram	Biểu đồ trình tự – mô tả cách các đối tượng tương tác theo thời gian.

2. Từ khóa về MiniApp (Zalo)

Từ khóa	Giải nghĩa
Zalo Mini App	Ứng dụng nhỏ chạy trong nền tảng Zalo.
Zalo OA (Official Account)	Tài khoản chính thức dùng để quản lý và phát hành mini app.
ZMP SDK	Bộ công cụ lập trình cho Zalo Mini App (Zalo Mini Program SDK).
ZMP Page	Trang giao diện trong Zalo Mini App.
ZMP API	Giao diện lập trình ứng dụng tích hợp các chức năng (login, gửi thông báo).

ZNS (Zalo Notification Service)	Dịch vụ gửi tin nhắn thông báo cho người dùng qua Zalo.
Zalo Pay Integration	Tích hợp thanh toán qua ZaloPay.

3. Từ khóa về nghiệp vụ bán vé sự kiện

Từ khóa	Giải nghĩa
Đặt vé (Booking)	Quy trình người dùng chọn và xác nhận tham gia sự kiện.
Mã vé / QR code	Mã định danh duy nhất dùng để check-in tham dự sự kiện.
Check-in	Quét mã vé tại sự kiện để xác thực người tham gia.
Lịch sử đặt vé	Danh sách các vé người dùng đã mua trong quá khứ.
Sơ đồ chỗ ngồi	Bản đồ tương tác hiển thị vị trí ghế tại sự kiện (nếu có).

4. Từ khóa về thanh toán

Từ khóa	Giải nghĩa
Payment Gateway	Cổng thanh toán – trung gian xử lý thanh toán online.
ZaloPay / Momo / VNPAY	Các ví điện tử được tích hợp làm phương thức thanh toán.
Order ID	Mã đơn hàng duy nhất trong hệ thống thanh toán.
Transaction	Giao dịch tài chính – được ghi nhận sau khi thanh toán.

5. Từ khóa về UI/UX và trải nghiệm người dùng

Từ khóa	Giải nghĩa
Responsive Design	Giao diện tương thích nhiều thiết bị.
Onboarding	Hướng dẫn người dùng lần đầu sử dụng.
User Flow	Luồng thao tác của người dùng trong hệ thống.
CTA (Call to Action)	Thành phần kêu gọi người dùng hành động (mua vé, đặt vé,...).
Toast / Popup	Cửa sổ thông báo nhanh cho người dùng.

2.6.Scenario

1. Scenario chính: Mua vé sự kiện thành công

Mô tả: Người dùng truy cập miniapp, chọn sự kiện, chọn loại vé, nhập thông tin cá nhân, thực hiện thanh toán, và nhận vé điện tử thành công.

Điều kiện tiên quyết:

- Người dùng đã đăng nhập vào miniapp (hoặc sử dụng chế độ khách với thông tin tạm thời).
- Sự kiện đang trong giai đoạn mở bán, còn vé khả dụng trong hệ thống.
- Người dùng có kết nối internet ổn định và thiết bị tương thích (web/mobile).
- Hệ thống thanh toán đã được tích hợp (ví dụ: Stripe, Momo, ZaloPay).
- Miniapp có thời gian chờ tối đa 10 phút để giữ vé trong giỏ hàng trước khi thanh toán.

Luồng sự kiện chính:

1. Truy cập miniapp:

- Người dùng mở miniapp qua nền tảng (web, ứng dụng di động, hoặc tích hợp trong siêu ứng dụng như Zalo/WeChat).
- Màn hình chính hiển thị danh sách sự kiện nổi bật (hình ảnh, tên, ngày, địa điểm) với bộ lọc (theo ngày, thể loại: âm nhạc, hội thảo, thể thao).
- Người dùng nhấn vào sự kiện "Hòa nhạc Rock 2025" để xem chi tiết.

2. Xem chi tiết sự kiện:

- Miniapp hiển thị thông tin sự kiện: thời gian (ví dụ: 20:00, 15/06/2025), địa điểm (Sân vận động Mỹ Đình), mô tả, và các loại vé (VIP: 2.000.000 VND, Thường: 800.000 VND).
- Hiển thị số lượng vé còn lại (ví dụ: "Còn 50 vé VIP, 200 vé Thường").
- Người dùng nhấn nút "Mua vé".

3. Chọn loại vé và số lượng:

- Miniapp hiển thị giao diện chọn vé với các tùy chọn: loại vé (VIP, Thường), số lượng (tối đa 6 vé mỗi giao dịch).
 - Người dùng chọn 2 vé VIP, miniapp cập nhật tổng tiền (4.000.000 VND) và hiển thị thời gian giữ vé (10 phút).
 - Người dùng nhấn "Tiếp tục" để sang bước nhập thông tin.
- 4. Nhập thông tin người mua:**
- Miniapp yêu cầu thông tin: họ tên, email, số điện thoại, và tùy chọn thêm thông tin người tham gia (nếu vé yêu cầu danh tính).
 - Giao diện hiển thị các trường nhập với gợi ý định dạng (ví dụ: email phải có "@", số điện thoại 10 chữ số).
 - Người dùng nhập: "Nguyễn Văn A, nguyen.a@gmail.com, 0987654321" và nhấn "Xác nhận".
 - Hệ thống kiểm tra định dạng thông tin và xác nhận hợp lệ.
- 5. Chọn phương thức thanh toán:**
- Miniapp hiển thị các phương thức thanh toán: thẻ tín dụng (Visa/Mastercard), ví điện tử (Momo, ZaloPay), chuyển khoản ngân hàng.
 - Người dùng chọn "Thẻ tín dụng" và nhập thông tin: số thẻ, ngày hết hạn, CVV.
 - Miniapp mã hóa dữ liệu thanh toán và gửi đến cổng thanh toán.
- 6. Thanh toán:**
- Miniapp chuyển hướng đến cổng thanh toán (nếu cần) hoặc xử lý trực tiếp.
 - Người dùng xác nhận thanh toán (ví dụ: nhập OTP từ ngân hàng).
 - Hệ thống nhận xác nhận thanh toán thành công từ cổng thanh toán trong vòng 30 giây.
 - Miniapp lưu giao dịch vào cơ sở dữ liệu với trạng thái "Thành công".
- 7. Nhận vé điện tử:**
- Miniapp tạo vé điện tử với mã QR duy nhất, thông tin sự kiện, và thông tin người mua.
 - Vé được gửi qua email (nguyen.a@gmail.com) và hiển thị trong mục "Vé của tôi" trên miniapp.
 - Người dùng nhận thông báo: "Mua vé thành công! Kiểm tra email hoặc mục 'Vé của tôi' để xem vé."
 - Người dùng có thể tải vé (PDF) hoặc lưu mã QR để sử dụng tại cổng sự kiện.

Kết quả:

- Người dùng nhận được vé điện tử và xác nhận giao dịch qua email/SMS.
- Hệ thống cập nhật số lượng vé còn lại (48 vé VIP).
- Giao dịch được ghi nhận với mã giao dịch duy nhất (ví dụ: TXN_20250615_123456).

2. Scenario ngoại lệ: Các trường hợp lỗi hoặc điều kiện bất thường

2.1. Hết vé sự kiện

Mô tả: Người dùng cố gắng mua vé nhưng sự kiện đã hết vé trước khi hoàn tất giao dịch.

Luồng sự kiện:

1. Người dùng chọn 2 vé VIP và nhấn "Tiếp tục".
2. Hệ thống kiểm tra kho vé và phát hiện vé VIP đã hết (do người dùng khác mua đồng thời).
3. Miniapp hiển thị thông báo: "Rất tiếc, vé VIP cho 'Hòa nhạc Rock 2025' đã hết. Vui lòng chọn loại vé khác hoặc sự kiện khác."
4. Miniapp đề xuất:
 - o Chọn vé Thường (nếu còn).
 - o Đăng ký nhận thông báo qua email nếu có vé bổ sung.
 - o Gợi ý các sự kiện tương tự (ví dụ: "Hòa nhạc Pop 2025" còn 100 vé).
5. Người dùng chọn xem sự kiện khác hoặc thoát miniapp.

Kết quả: Người dùng không mua được vé nhưng được hướng dẫn các tùy chọn thay thế.

2.2. Lỗi thanh toán

Mô tả: Thanh toán thất bại do lỗi từ phía người dùng (thẻ không đủ số dư) hoặc cổng thanh toán (lỗi hệ thống).

Luồng sự kiện:

1. Người dùng hoàn thành bước chọn vé và nhập thông tin thanh toán (thẻ tín dụng).
2. Khi nhấn "Thanh toán", cổng thanh toán trả về lỗi (ví dụ: "Số dư không đủ" hoặc "Lỗi kết nối với ngân hàng").
3. Miniapp hiển thị thông báo: "Thanh toán thất bại. Lý do: Số dư không đủ. Vui lòng kiểm tra thẻ hoặc chọn phương thức khác."
4. Miniapp giữ giỏ hàng trong 5 phút và đề xuất:
 - o Nhập lại thông tin thẻ hoặc chọn phương thức khác (ví dụ: ZaloPay).
 - o Liên hệ ngân hàng để kiểm tra vấn đề.
5. Người dùng thử lại với phương thức khác:
 - o Nếu thành công, quay lại luồng chính.
 - o Nếu thất bại 3 lần liên tiếp, miniapp lưu giỏ hàng và hiển thị: "Vui lòng liên hệ hỗ trợ qua hotline 19001234 hoặc email support@ticketing.com."
6. Hệ thống gửi email thông báo trạng thái giao dịch: "Giao dịch TXN_20250615_123456 thất bại. Vui lòng thử lại."

Kết quả: Người dùng được hướng dẫn khắc phục lỗi hoặc liên hệ hỗ trợ, giỏ hàng được giữ tạm thời.

2.3. Mất kết nối internet

Mô tả: Người dùng mất kết nối trong quá trình chọn vé hoặc thanh toán.

Luồng sự kiện:

1. Người dùng đang chọn vé hoặc thanh toán thì mất kết nối internet.
2. Miniapp phát hiện lỗi và hiển thị: "Mất kết nối. Vui lòng kiểm tra mạng và thử lại."
3. Hệ thống lưu tạm thời thông tin giỏ hàng (loại vé, số lượng) và thông tin người mua vào bộ nhớ cache (local storage).
4. Khi kết nối được khôi phục:

- Miniapp tự động tải lại giỏ hàng và hiển thị thông báo: "Đã khôi phục giỏ hàng. Vui lòng tiếp tục."
- Nếu ở bước thanh toán, miniapp kiểm tra trạng thái giao dịch qua API cổng thanh toán:
 - Nếu giao dịch chưa hoàn tất, yêu cầu thử lại.
 - Nếu giao dịch đã thành công, chuyển sang bước nhận vé.
- 5. Nếu người dùng thoát miniapp, hệ thống gửi email nhắc nhở: "Bạn có giao dịch chưa hoàn tất. Vui lòng quay lại miniapp để tiếp tục."

Kết quả: Người dùng có thể tiếp tục giao dịch khi mạng ổn định, dữ liệu được bảo toàn.

2.4. Thông tin người mua không hợp lệ

Mô tả: Người dùng nhập thông tin không đúng định dạng (email sai, số điện thoại không hợp lệ).

Luồng sự kiện:

1. Người dùng nhập thông tin: "Nguyễn Văn A, nguyen.a, 12345" và nhấn "Xác nhận".
2. Hệ thống kiểm tra và phát hiện lỗi:
 - Email "nguyen.a" thiếu "@" và tên miền.
 - Số điện thoại "12345" không đủ 10 chữ số.
3. Miniapp hiển thị thông báo: "Lỗi: Email không hợp lệ, số điện thoại phải có 10 chữ số." Các trường lỗi được đánh dấu đỏ với gợi ý sửa.
4. Người dùng nhập lại: "nguyen.a@gmail.com, 0987654321".
5. Hệ thống kiểm tra lại:
 - Nếu hợp lệ, tiếp tục luồng chính.
 - Nếu lỗi tiếp diễn (tối đa 3 lần), miniapp hiển thị: "Thông tin không hợp lệ. Vui lòng kiểm tra hoặc liên hệ hỗ trợ qua hotline 19001234."
6. Hệ thống ghi log lỗi để hỗ trợ phân tích (ví dụ: số lần nhập sai, loại lỗi).

Kết quả: Người dùng được hướng dẫn sửa thông tin hoặc liên hệ hỗ trợ, giao dịch không bị gián đoạn nếu sửa kịp thời.

2.5. Hủy giao dịch trước khi thanh toán

Mô tả: Người dùng quyết định hủy giao dịch trong quá trình mua vé.

Luồng sự kiện:

1. Người dùng đang ở bước chọn vé, nhập thông tin, hoặc thanh toán thì nhấn nút "Hủy".
2. Miniapp hiển thị cửa sổ xác nhận: "Bạn có chắc chắn muốn hủy giao dịch? Vé trong giỏ hàng sẽ được giải phóng."
3. Người dùng chọn:
 - **Xác nhận hủy:** Miniapp xóa giỏ hàng, giải phóng vé về kho, và chuyển về màn hình chính. Hệ thống ghi log: "Giao dịch TXN_20250615_123456 bị hủy."
 - **Tiếp tục:** Miniapp quay lại bước hiện tại.
4. Nếu vé đã giữ quá 10 phút mà không thanh toán, hệ thống tự động hủy giỏ hàng và thông báo: "Giỏ hàng đã hết hạn. Vui lòng chọn lại vé."

Kết quả: Người dùng hủy giao dịch thành công, vé được giải phóng cho người dùng khác.

2.6. Lỗi hệ thống nội bộ

Mô tả: Hệ thống gặp lỗi không mong muốn (database, API thất bại).

Luồng sự kiện:

1. Người dùng đang thực hiện giao dịch (ví dụ: thanh toán) thì hệ thống gặp lỗi (timeout, server quá tải).
2. Miniapp hiển thị: "Lỗi hệ thống. Vui lòng thử lại sau vài phút hoặc liên hệ hỗ trợ qua hotline 19001234."
3. Hệ thống lưu trạng thái giao dịch và gửi thông báo qua email: "Giao dịch TXN_20250615_123456 gặp lỗi. Chúng tôi đang xử lý, vui lòng kiểm tra lại sau."
4. Miniapp tự động thử lại API call (tối đa 2 lần) trước khi thông báo lỗi cuối cùng.
5. Nếu lỗi được khắc phục, miniapp khôi phục giao dịch và hướng dẫn người dùng tiếp tục.

Kết quả: Người dùng được thông báo về lỗi và hướng dẫn thử lại, giao dịch được bảo toàn nếu có thể.

Lưu ý

- **Giao diện người dùng:** Thiết kế tối giản, nút bấm rõ ràng (ví dụ: "Mua vé", "Thanh toán", "Hủy"), hỗ trợ đa ngôn ngữ (Việt Nam, Anh, v.v.).
- **Bảo mật:** Mã hóa thông tin thanh toán (AES-256), sử dụng HTTPS, và xác thực OTP cho giao dịch nhạy cảm.
- **Hiệu suất:** Hệ thống cần xử lý đồng thời tối thiểu 1.000 giao dịch/phút để tránh quá tải trong giờ cao điểm.
- **Hỗ trợ khách hàng:** Tích hợp chatbot hoặc hotline trong miniapp để hỗ trợ tức thì.
- **Log hệ thống:** Ghi lại mọi lỗi và giao dịch (thành công/thất bại) để phân tích và cải thiện.

2.7. Xây dựng sơ đồ ERD

1.Xác định các thực thể

Bảng Thực Thể và Các Thuộc Tính Chính

STT	Tên thực thể	Thuộc tính chính	PK	FK	Ghi chú về loại
1	Người dùng (User)	user_id, full_name, email, phone, password	user_id	Không có	Người dùng đăng ký trên hệ thống
2	Sự kiện (Event)	event_id, name, date, location, organizer	event_id	Không có	Sự kiện do tổ chức hoặc cá nhân tạo ra
3	Vé (Ticket)	ticket_id, event_id, user_id, price, status, qr_code	ticket_id	event_id, user_id	Vé tham gia sự kiện.
4	Thanh toán (Payment)	payment_id, ticket_id, user_id, amount, payment_method, payment_status	payment_id	ticket_id, user_id	Giao dịch thanh toán của người dùng
5	Kiểm tra vé (Ticket Check)	check_id, ticket_id, event_id, check_time, check_status	check_id	ticket_id, event_id	Kiểm tra vé khi tham gia sự kiện
6	Lịch sử giao dịch (Transaction History)	transaction_id, payment_id, user_id, transaction_date, transaction_details	transaction_id	payment_id, user_id	Lưu lại lịch sử thanh toán của người dùng

Dưới đây là cách các thực thể trong sơ đồ ERD liên kết với nhau thông qua các mối quan hệ:

1. Người dùng (User) - Vé (Ticket):

- Một người dùng có thể mua nhiều vé tham gia các sự kiện khác nhau.
- Mỗi vé thuộc về một người dùng duy nhất.
- **Quan hệ:** 1:N (Một người dùng - Nhiều vé).

2. Sự kiện (Event) - Vé (Ticket):

- Mỗi sự kiện có thể có nhiều vé được bán ra.
- Một vé chỉ thuộc về một sự kiện duy nhất.
- **Quan hệ:** 1:N (Một sự kiện - Nhiều vé).

3. Người dùng (User) - Thanh toán (Payment):

- Mỗi người dùng có thể thực hiện nhiều giao dịch thanh toán.
- Mỗi thanh toán gắn liền với một người dùng cụ thể.
- **Quan hệ:** 1:N (Một người dùng - Nhiều thanh toán).

4. Vé (Ticket) - Thanh toán (Payment):

- Một vé có thể được thanh toán một hoặc nhiều lần (trong trường hợp hoàn tiền, thay đổi phương thức thanh toán).
- Mỗi thanh toán chỉ liên quan đến một vé.
- **Quan hệ:** 1:N (Một vé - Nhiều thanh toán).

5. Sự kiện (Event) - Kiểm tra vé (Ticket Check):

- Mỗi sự kiện sẽ có nhiều lần kiểm tra vé cho các người tham gia.
- Một lần kiểm tra vé chỉ thuộc về một sự kiện duy nhất.
- **Quan hệ:** 1:N (Một sự kiện - Nhiều kiểm tra vé).

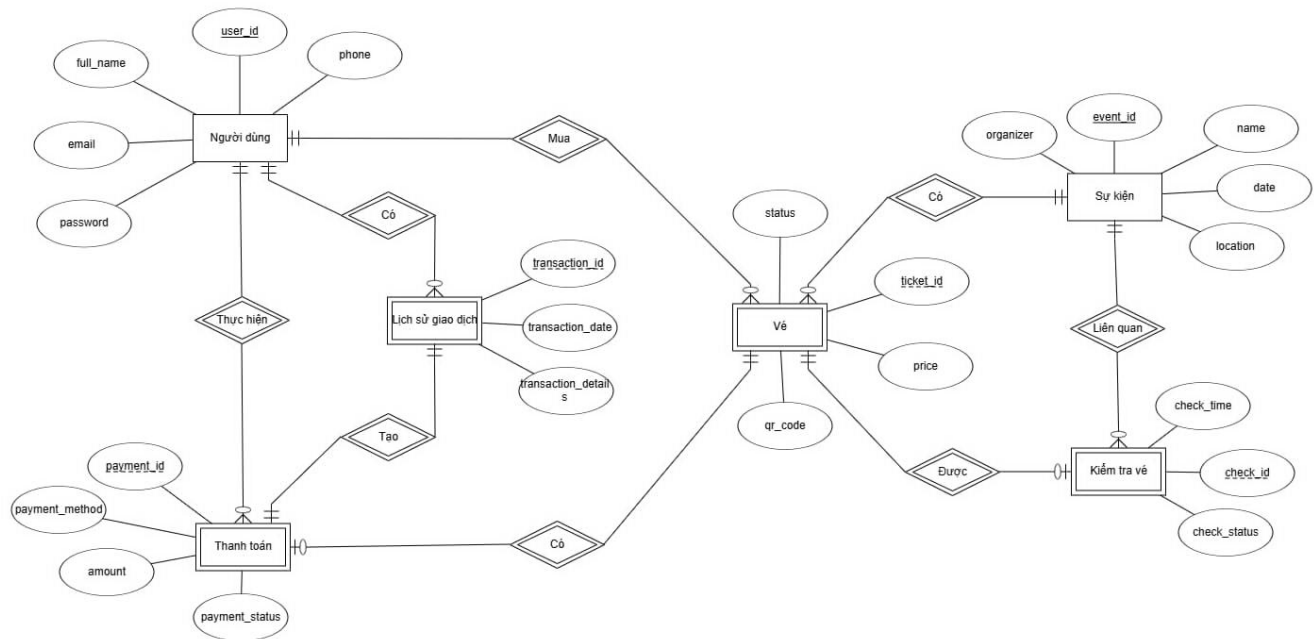
6. Vé (Ticket) - Kiểm tra vé (Ticket Check):

- Mỗi vé sẽ được kiểm tra tại sự kiện trước khi vào cửa.
- Một lần kiểm tra vé gắn liền với một vé duy nhất.
- **Quan hệ:** 1:1 (Một vé - Một lần kiểm tra).

7. Thanh toán (Payment) - Lịch sử giao dịch (Transaction History):

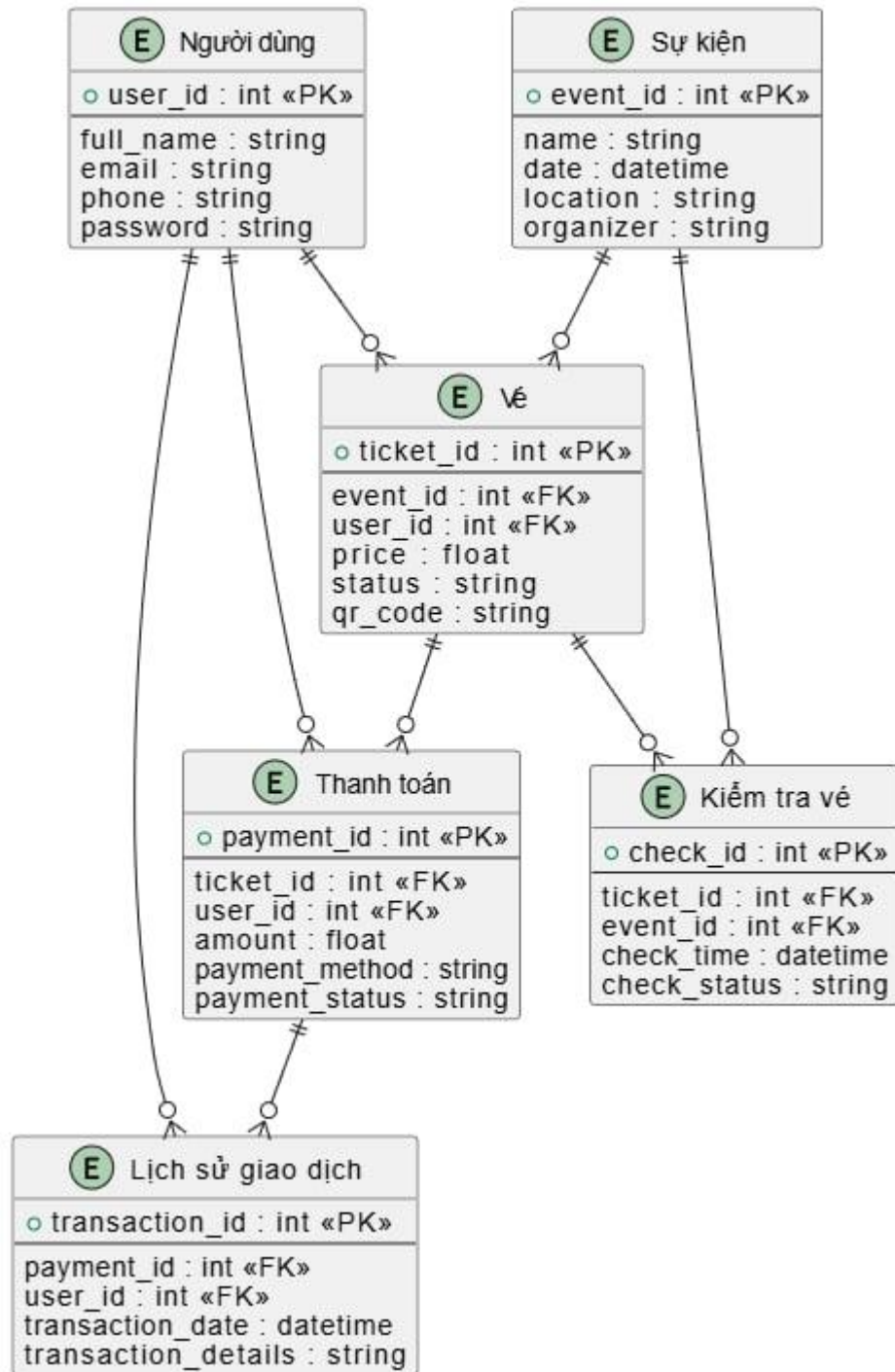
- Mỗi giao dịch thanh toán sẽ được ghi lại trong lịch sử giao dịch.
- Một thanh toán có thể tạo ra nhiều bản ghi trong lịch sử giao dịch để theo dõi trạng thái.
- **Quan hệ:** 1:N (Một thanh toán - Nhiều bản ghi lịch sử giao dịch).

Sơ đồ ERD



Hình 8. Sơ đồ ERD

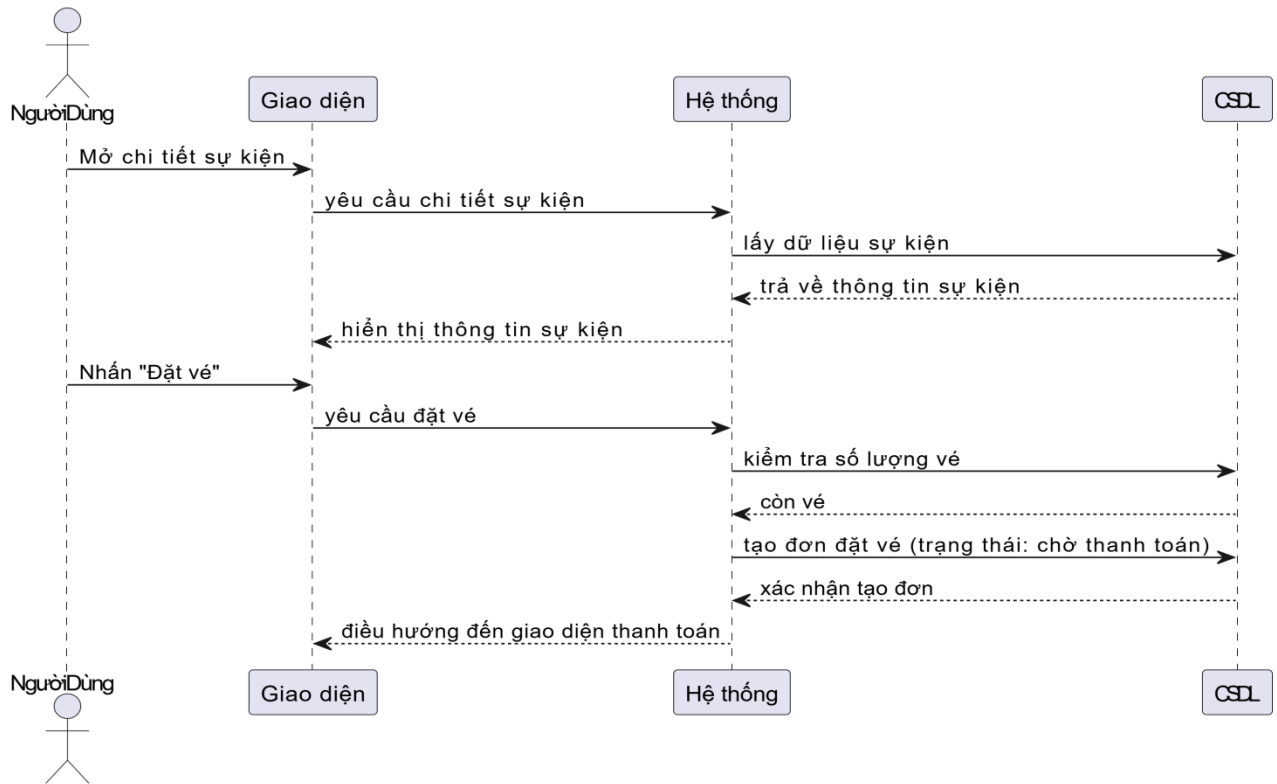
Sơ đồ UML



Hình 9. Sơ đồ UML

2.8.Sơ đồ tuần tự

1. Quy trình đặt vé sự kiện



Hình 9.Sơ đồ tuần tự quy trình đặt vé sự kiện

Mô tả tiên trình:

Người dùng truy cập vào chi tiết một sự kiện từ danh sách hoặc kết quả tìm kiếm.

Giao diện gửi yêu cầu lên **hệ thống** để lấy dữ liệu chi tiết về sự kiện.

Hệ thống truy vấn cơ sở dữ liệu và trả về thông tin: tên, thời gian, địa điểm, loại vé, giá vé, tình trạng còn chỗ,...

Người dùng quyết định đặt vé → giao diện gửi yêu cầu đặt vé.

Hệ thống kiểm tra số lượng vé trong cơ sở dữ liệu

Nếu còn vé:

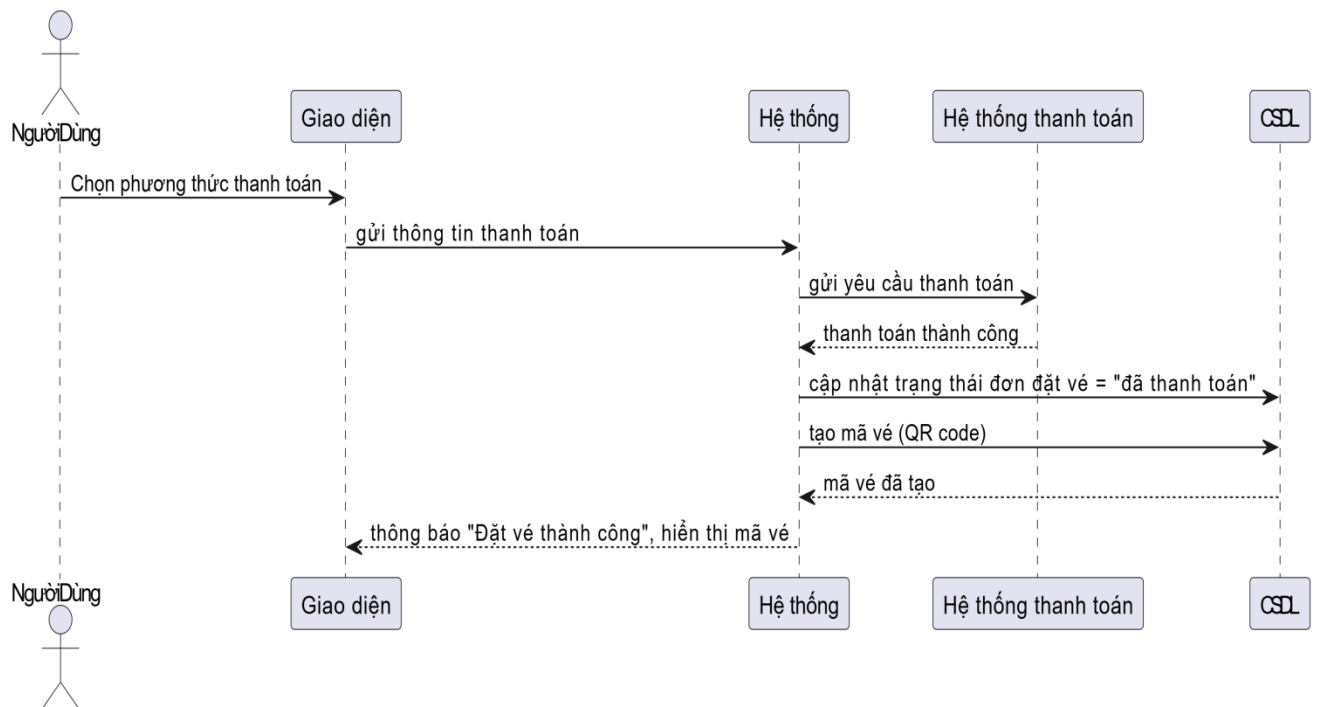
-Hệ thống tạo một **đơn đặt vé mới** trong CSDL với trạng thái "chờ thanh toán".

Người dùng được điều hướng đến giao diện thanh toán.

Mục tiêu:

Tạo đơn đặt vé với thông tin chính xác và đảm bảo vé còn chỗ.

2.Quy trình Thanh toán và nhận vé



Hình 10.Sơ đồ tuần tự quy trình Thanh toán và xác nhận vé

Mô tả tiến trình:

Người dùng chọn **phương thức thanh toán** và xác nhận thanh toán.

Giao diện gửi thông tin thanh toán đến **hệ thống**.

Hệ thống chuyển thông tin đến bên thứ ba là **Hệ thống thanh toán** (VD: VNPay, Momo...).

Hệ thống thanh toán phản hồi kết quả: **thành công** hoặc **thất bại**.

Nếu thanh toán thành công

-Hệ thống cập nhật trạng thái đơn đặt vé thành "đã thanh toán".

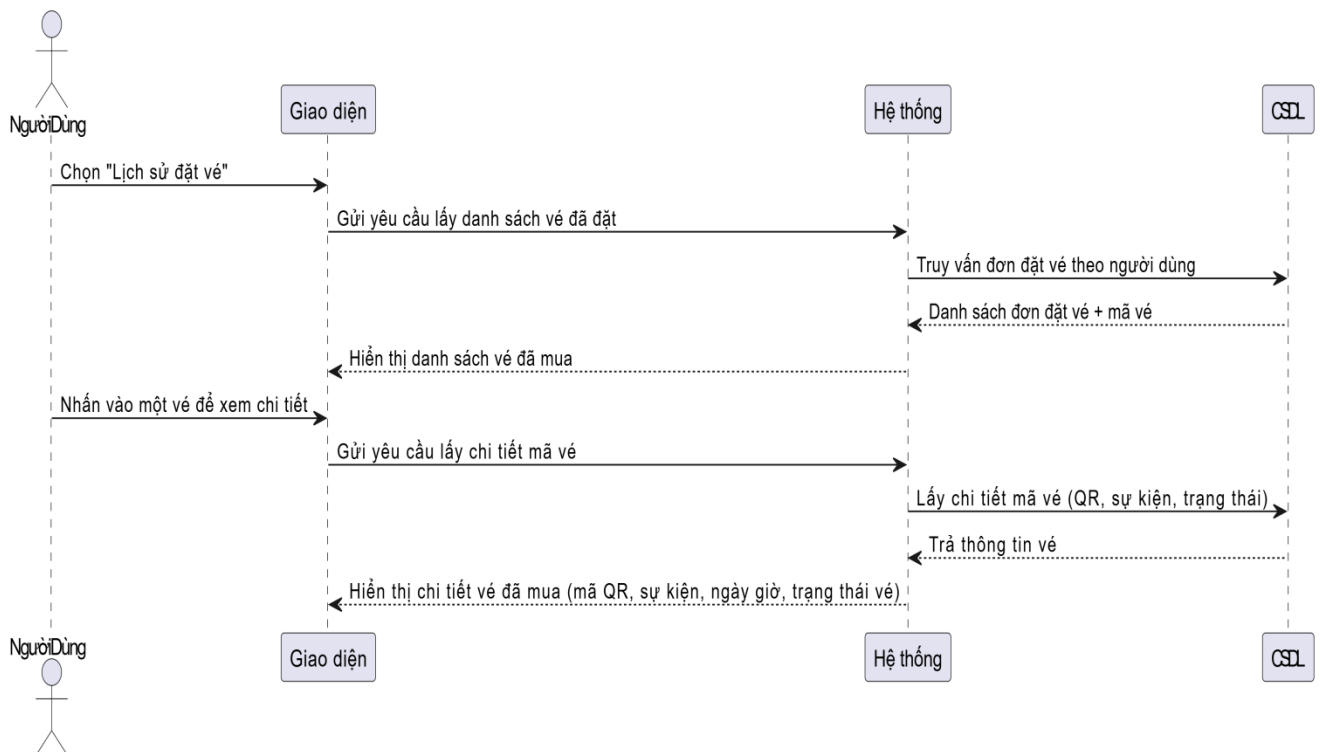
-Sinh mã vé QR hoặc mã số vé riêng cho người dùng.

Giao diện hiển thị thông báo: **“Đặt vé thành công”**, kèm theo mã vé để lưu và sử dụng khi đến sự kiện.

Mục tiêu:

Xác nhận thanh toán thành công, lưu trữ đơn đặt vé và cung cấp vé cho người dùng.

3.Quy trình Kiểm tra vé đã mua



Hình 11.Sơ đồ tuần tự quy trình Kiểm tra vé đã mua

Mô tả tiến trình:

Người dùng mở app và chọn mục **“Lịch sử đặt vé”** trên giao diện chính hoặc trong tài khoản cá nhân.

Giao diện sẽ gửi yêu cầu đến **Hệ thống** để truy vấn danh sách các vé đã đặt.

Hệ thống tìm trong **cơ sở dữ liệu (CSDL)** các đơn đặt vé của người dùng đó, bao gồm:

-Tên sự kiện

-Ngày giờ diễn ra

-Mã vé (QR code hoặc mã số)

-Trạng thái vé: Đã thanh toán / Chưa thanh toán / Đã sử dụng

Giao diện hiển thị danh sách các vé mà người dùng đã mua.

Người dùng có thể **click vào một vé cụ thể** để xem thông tin chi tiết.

Giao diện tiếp tục gửi yêu cầu chi tiết về mã vé.

Hệ thống truy vấn chi tiết mã vé trong CSDL và trả về:

-Mã QR (nếu có)

-Loại vé, khu vực ngồi (nếu áp dụng)

-Thời gian sự kiện

-Trạng thái sử dụng

Giao diện hiển thị đầy đủ thông tin vé để người dùng tiện sử dụng hoặc xuất trình khi đến sự kiện

Mục tiêu:

Cho phép người dùng xem lại toàn bộ vé đã mua.

Cung cấp đầy đủ thông tin chi tiết để:

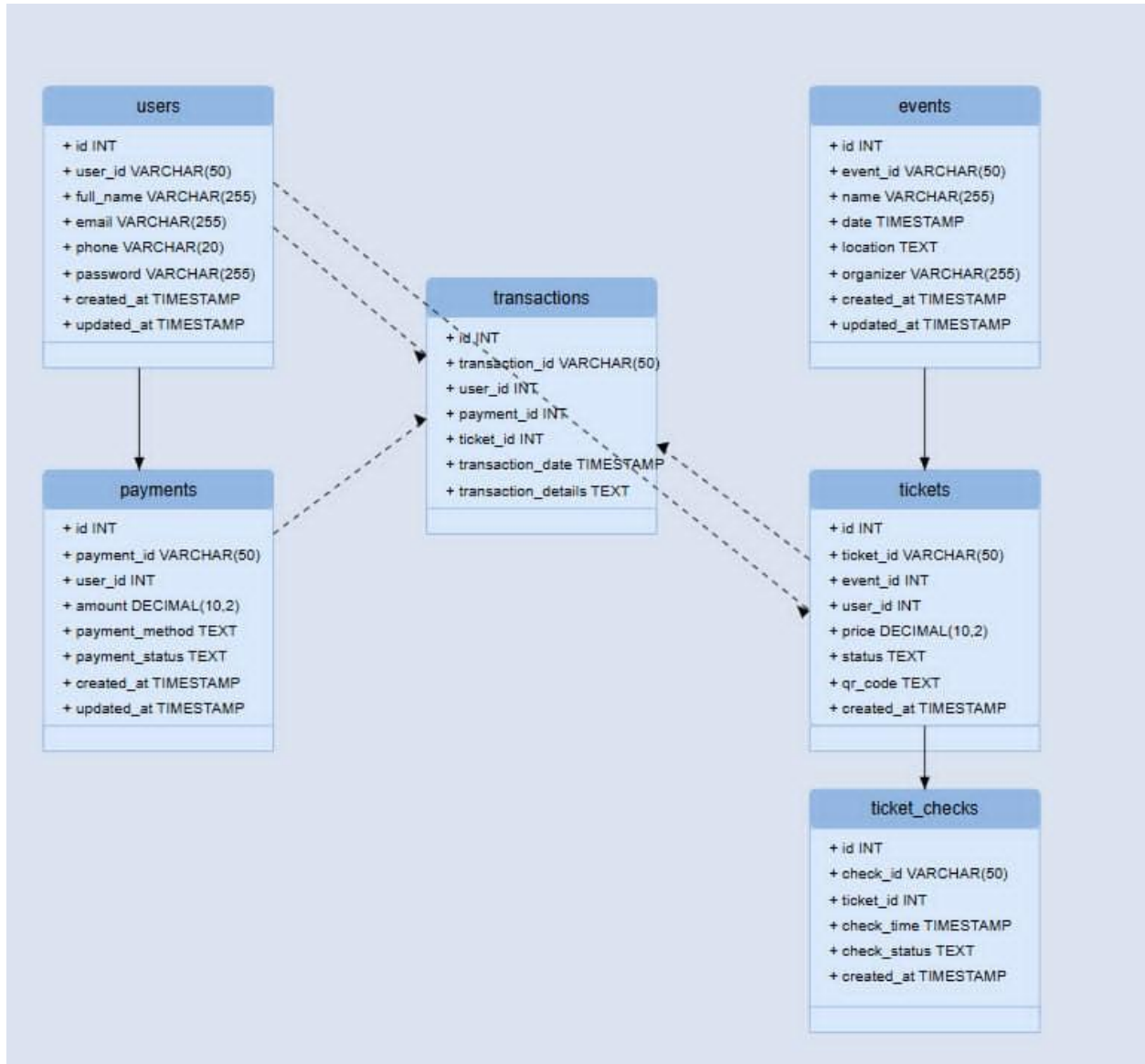
-Xác nhận vé đã thanh toán

-Chuẩn bị cho check-in tại sự kiện

-Lưu trữ làm bằng chứng mua vé (nếu cần)

Chương 3. Thiết kế Cơ sở dữ liệu

1. Sơ đồ cơ sở dữ liệu




Hình 10. Sơ đồ Cơ sở dữ liệu

2. Xây dựng CRC dựa trên các lớp điều khiển


Lớp học	Trách nhiệm	Hợp tác
QuanLyNguoiDungControl	<ul style="list-style-type: none"> - Xác thực thông tin đăng nhập của người dùng. - Xử lý mới tài khoản đăng nhập. - Cập nhật thông tin cá nhân người dùng. 	NgườiDung, ManHinhDangNhap, ManHinhDangKy, ThongBaoControl
QuanLySuKienControl	<ul style="list-style-type: none"> - Lấy danh sách sự kiện để hiển thị - Xử lý tìm kiếm và lọc sự kiện - Cập nhật sự kiện thông tin (sửa, xóa). 	SuKien, ManHinhDanhSachSuKien, ManHinhChiTietSuKien
DatVeControl	<ul style="list-style-type: none"> - Tạo đơn đặt hàng cho người dùng. - Tính tổng tiền dựa trên vé đã chọn. - Xác thực lại số lượng vé còn lại. - Tạo mã QR cho vé. 	SuKien, VeSuKien, DonHang, NguoiDung, ManHinhChiTietSuKien
ThanhToanControl	<ul style="list-style-type: none"> - Tích hợp với cổng thanh toán (ZaloPay/VNPay) - Kiểm tra thanh toán giao dịch trạng thái - Cập nhật trạng thái đơn hàng sau thanh toán. 	DonHang, ManHinhThanhToan, ThongBaoControl
ThongBaoControl	<ul style="list-style-type: none"> - Tạo nội dung thông báo cho người dùng. - Gửi thông báo qua ZNS (Dịch vụ thông báo Zalo). - Đánh dấu đã đọc thông báo. 	ThongBao, ThongBaoUI, NguoiDung
Kiểm soát LịchSu	<ul style="list-style-type: none"> - Lấy danh sách đơn hàng đã được người dùng thiết lập. - Hiển thị thông tin chi tiết lịch sử đặt vé. 	LichSuDatVe, DonHang, NguoiDung, ManHinhLichSu

3. Thiết kế thuộc tính và phương thức cho lớp


1. QuanLyNguoiDungControl

 QuanLyNguoiDungControl
<ul style="list-style-type: none">❑ <code>userId: int</code>❑ <code>email: string</code>❑ <code>role: string</code>
<ul style="list-style-type: none">● <code>authenticate(email: string, password: string) : bool</code>● <code>register(name: string, email: string, phone: string, password: string) : bool</code>● <code>updateUserInfo(userId: int, name: string, phone: string) : bool</code>


2. QuanLySuKienControl

 QuanLySuKienControl
<ul style="list-style-type: none">❑ <code>eventId: int</code>❑ <code>searchQuery: string</code>❑ <code>category: string</code>
<ul style="list-style-type: none">● <code>getEventList(category: string) : List<SuKien></code>● <code>searchEvents(query: string) : List<SuKien></code>● <code>updateEventInfo(eventId: int, details: SuKien) : bool</code>


3. DatVeControl

 DatVeControl
<ul style="list-style-type: none">❑ <code>orderId: int</code>❑ <code>userId: int</code>❑ <code>ticketList: List<VeSuKien></code>
<ul style="list-style-type: none">● <code>createOrder(userId: int, ticketList: List<VeSuKien>) : DonHang</code>● <code>calculateTotal(ticketList: List<VeSuKien>) : float</code>● <code>validateTicketAvailability(eventId: int, quantity: int) : bool</code>● <code>generateQRCode(ticketId: int) : string</code>


4.ThanhToanControl

 ThanhToanControl
<ul style="list-style-type: none">□ paymentId: int□ orderId: int□ amount: float□ paymentMethod: string
<ul style="list-style-type: none">● processPayment(orderId: int, amount: float, paymentMethod: string) : bool● checkPaymentStatus(paymentId: int) : string● updateOrderStatus(orderId: int, status: string) : bool

5. ThongBaoControl

 ThongBaoControl
<ul style="list-style-type: none">□ notificationId: int□ userId: int□ message: string
<ul style="list-style-type: none">● sendNotification(userId: int, message: string) : bool● markAsRead(notificationId: int) : bool● generateNotificationContent(orderId: int) : string

6.LichSuControl

 LichSuControl
<ul style="list-style-type: none">□ historyId: int□ userId: int□ orderHistory: List<DonHang>
<ul style="list-style-type: none">● getOrderHistory(userId: int) : List<DonHang>● displayOrderDetails(orderId: int) : string

4. Danh sách mô tả chi tiết các bảng dữ liệu

1.Bảng NguoiDung (Customer)

Column Name	DataType	Description	Constraint
ID	INT	Mã định danh duy nhất của người dùng	PRIMARY KEY, AUTO_INCREMENT
HoTen	VARCHAR(100)	Họ và tên của người dùng	NOT NULL
Email	VARCHAR(100)	Địa chỉ email của người dùng	NOT NULL, UNIQUE
SoDienThoai	VARCHAR(15)	Số điện thoại của người dùng	NOT NULL, UNIQUE
MatKhau	VARCHAR(255)	Mật khẩu đã mã hóa của người dùng	NOT NULL
VaiTro	VARCHAR(20)	Vai trò của người dùng (user/admin)	NOT NULL, DEFAULT 'user'

2.Bảng SuKien (Event)

Column Name	DataType	Description	Constraint
ID	INT	Mã định danh duy nhất của sự kiện	PRIMARY KEY, AUTO_INCREMENT
TenSuKien	VARCHAR(200)	Tên của sự kiện	NOT NULL
MoTa	TEXT	Mô tả chi tiết về sự kiện	NULL
ThoiGian	DATETIME	Thời gian diễn ra sự kiện	NOT NULL
DiaDiem	VARCHAR(200)	Địa điểm tổ chức sự kiện	NOT NULL
HinhAnh	VARCHAR(255)	Đường dẫn đến hình ảnh sự kiện	NULL
SoDoChoNgoni	TEXT	Dữ liệu sơ đồ chỗ ngồi (JSON hoặc text)	NULL

3. Bảng VeSuKien (Ticket)

Column Name	Data Type	Description	Constraint
IDVe	INT	Mã định danh duy nhất của vé	PRIMARY KEY, AUTO_INCREMENT
MaQR	VARCHAR(255)	Mã QR để check-in vé	NOT NULL, UNIQUE
LoaiVe	VARCHAR(50)	Loại vé (VIP, thường, v.v.)	NOT NULL
GiaVe	FLOAT	Giá của vé	NOT NULL
TrangThai	VARCHAR(20)	Trạng thái vé (chưa thanh toán/đã thanh toán/đã check-in)	NOT NULL, DEFAULT 'chưa thanh toán'
IDSukien	INT	Mã sự kiện mà vé thuộc về	FOREIGN KEY (IDSukien), REFERENCES SuKien(ID)

4. Bảng DonHang (Order)

Column Name	Data Type	Description	Constraint
IDDonHang	INT	Mã định danh duy nhất của đơn hàng	PRIMARY KEY, AUTO_INCREMENT
NgayDat	DATETIME	Thời gian đặt đơn hàng	NOT NULL
TrangThaiThanhToan	VARCHAR(20)	Trạng thái thanh toán (đã thanh toán/chưa thanh toán)	NOT NULL, DEFAULT 'chưa thanh toán'
TongTien	FLOAT	Tổng tiền của đơn hàng	NOT NULL
IDNguoiDung	INT	Mã người dùng đặt đơn hàng	FOREIGN KEY (IDNguoiDung) REFERENCES NguoiDung(ID)

DanhSachVe	TEXT	Danh sách ID vé trong đơn hàng (JSON)	NOT NULL
------------	------	---------------------------------------	----------

5.Bảng ThôngBao (Notification)

Column Name	Data Type	Description	Constraint
ID	INT	Mã định danh duy nhất của thông báo	PRIMARY KEY, AUTO_INCREMENT
NoiDung	TEXT	Nội dung thông báo	NOT NULL
LoaiThongBao	VARCHAR(50)	Loại thông báo (email, push, ZNS)	NOT NULL
NgayGui	DATETIME	Thời gian gửi thông báo	NOT NULL
TrangThaiDaDoc	BOOLEAN	Trạng thái đã đọc (true/false)	NOT NULL, DEFAULT FALSE
IDNguoiDung	INT	Mã người dùng nhận thông báo	FOREIGN KEY(IDNguoiDung) REFERENCES NguoiDungID

6.Bảng LịchSuDatVe (Order History)

Column Name	Data Type	Description	Constraint
IDLichSu	INT	Mã định danh duy nhất của lịch sử	PRIMARY KEY, AUTO_INCREMENT
IDNguoiDung	INT	Mã người dùng sở hữu lịch sử	FOREIGN KEY (IDNguoiDung) REFERENCES NguoiDung(ID)
DanhSachDonHang	TEXT	Danh sách ID đơn hàng (JSON)	NOT NULL
ThoiGianDatVe	DATETIME	Thời gian đặt vé	NOT NULL

Giải thích

- **Column Name:** Tên cột trong cơ sở dữ liệu, dựa trên thuộc tính của từng lớp thực thể.
- **DataType:** Kiểu dữ liệu phù hợp (INT, VARCHAR, TEXT, FLOAT, DATETIME, BOOLEAN).
- **Description:** Mô tả ý nghĩa của cột.
- **Constraint:** Các ràng buộc, bao gồm:
 - PRIMARY KEY: Khóa chính, duy nhất và tự động tăng (AUTO_INCREMENT).
 - NOT NULL: Không được để trống.
 - UNIQUE: Giá trị duy nhất (như email, số điện thoại, mã QR).
 - FOREIGN KEY: Khóa ngoại liên kết đến bảng khác (ví dụ: IDSukien trong VeSuKien liên kết với SuKien).
 - DEFAULT: Giá trị mặc định nếu không được cung cấp.

Chương 4. Giao diện và tính năng

1. Trang đăng nhập, đăng ký

Giao diện:

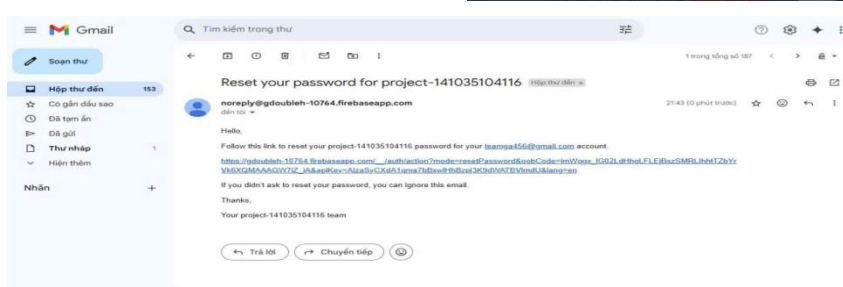
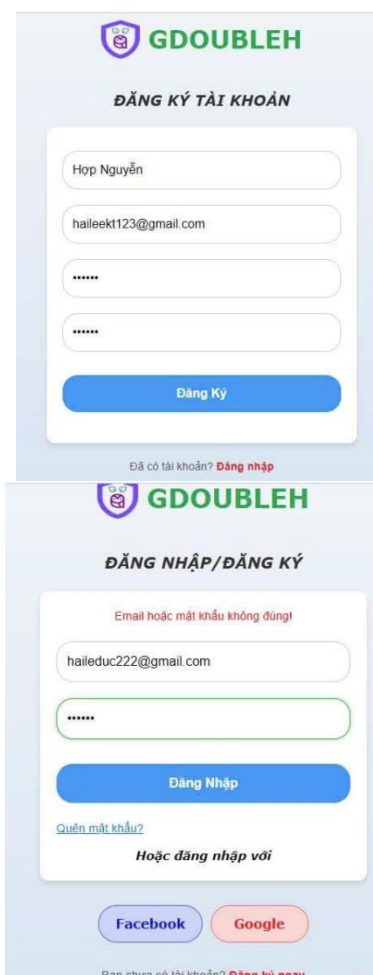
· **Giao diện trực quan, dễ sử dụng:** Người dùng nhập **email** và **mật khẩu** để đăng nhập hoặc chọn đăng nhập nhanh bằng **Facebook** hoặc **Google**. Các trường nhập liệu và nút bấm được bố trí rõ ràng, dễ thao tác trên thiết bị di động.

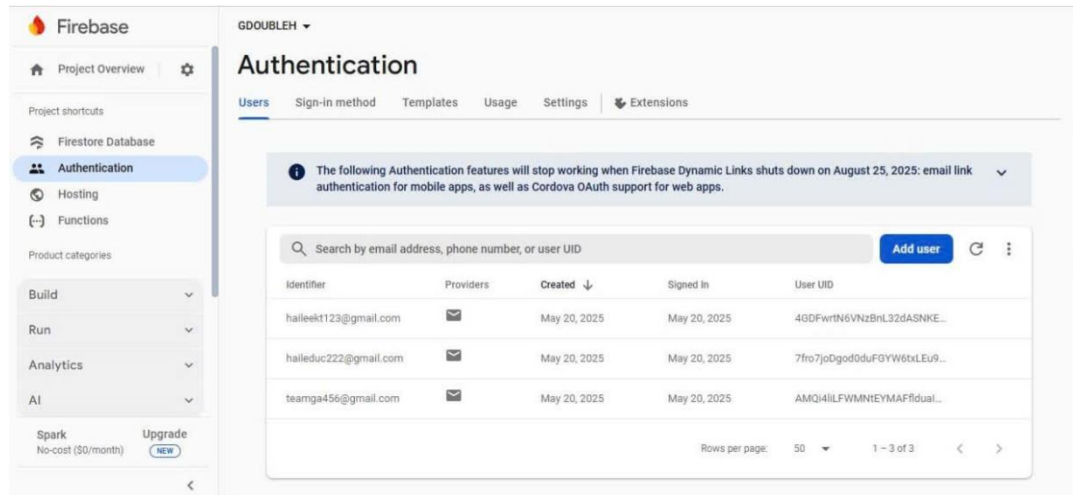
· **Thiết kế hiện đại:** Logo thương hiệu hiển thị nổi bật, kết hợp bố cục tối giản, phù hợp với trải nghiệm người dùng trên điện thoại thông minh.

Tính năng:

Form đăng ký: Yêu cầu nhập đủ thông tin, nếu email đã được đăng ký thì khi nhấn đăng ký sẽ hiện thông báo email đã được đăng ký, và email người dùng sẽ được lưu trong authentication của firebase

Form đăng nhập: Chỉ cho phép tài khoản đã đăng ký đăng nhập, Nhập sai mk hoặc email cũng hiện thông báo. Nếu người dùng quên mk thì khi nhấn label quên mk thì sẽ hiện ra cửa sổ yêu cầu nhập email và gửi. Sau khi gửi người dùng sẽ được gửi link để reset lại mk thông qua email. Nhập mk mới nhấn lưu và quay lại trang login đăng nhập bình thường





Firestore

Mục đích: Lưu trữ thông tin người dùng (tên, email, UID) sau khi đăng ký. Kiểm tra xem email hoặc tên đã tồn tại trước khi đăng ký. Các API sử dụng: collection: Truy cập một bộ sưu tập (collection) trong Firestore. addDoc: Thêm tài liệu mới vào Firestore.

Authentication

Mục đích: Xác thực người dùng bằng email và mật khẩu. Hỗ trợ đăng nhập qua các nhà cung cấp bên thứ ba như Google, Facebook. Gửi email đặt lại mật khẩu. Các API sử dụng:

createUserWithEmailAndPassword: Đăng ký tài khoản mới với email và mật khẩu.

signInWithEmailAndPassword: Đăng nhập bằng email và mật khẩu.

updateProfile: Cập nhật thông tin người dùng (ví dụ: tên hiển thị).

sendPasswordResetEmail: Gửi email đặt lại mật khẩu.

signInWithRedirect: Đăng nhập bằng Google hoặc các nhà cung cấp khác.

2.Trang chủ

Giao diện:

1. Thanh tìm kiếm & tiện ích (trên cùng):

- **Ô tìm kiếm:** cho phép người dùng nhanh chóng tra cứu tên sự kiện yêu thích.
- **Biểu tượng giỏ hàng & thông báo:** hiển thị các vé đã chọn và cập nhật thông tin mới từ hệ thống.

2. Danh mục thể loại sự kiện:

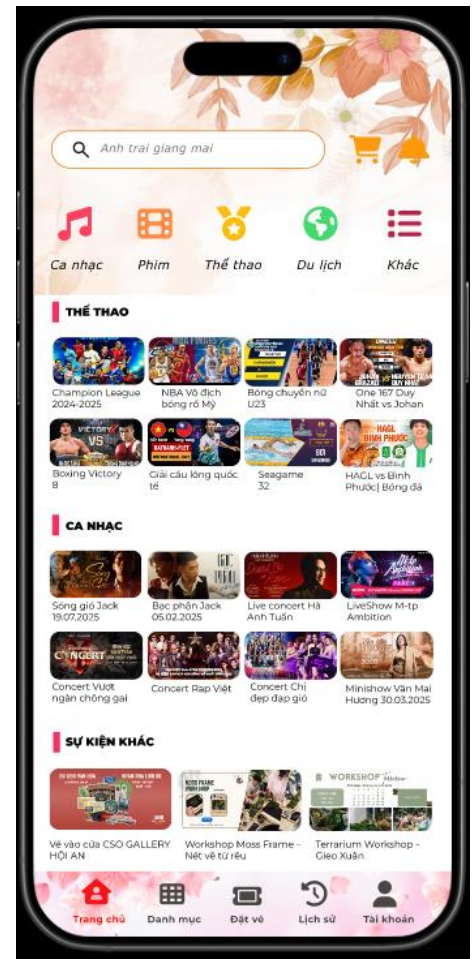
- Gồm các biểu tượng và nhãn: **Ca nhạc, Phim, Thể thao, Du lịch, Khác.**
- Giúp người dùng lọc nhanh các sự kiện theo sở thích cá nhân.

3. Khu vực nổi bật - danh sách sự kiện:

- Các sự kiện được chia thành nhóm rõ ràng:
 - Thể thao** (VD: Bóng đá, bóng rổ...)
 - Ca nhạc** (Live show, minishow, concert...)
 - Sự kiện khác** (Workshop, triển lãm...)
- Mỗi sự kiện hiển thị thumbnail sinh động, tiêu đề, ngày giờ cụ thể.

4. Thanh điều hướng dưới cùng:

- Gồm các mục: **Trang chủ, Danh mục, Đặt vé, Lịch sử, Tài khoản.**
- Giúp người dùng dễ dàng điều hướng giữa các chức năng chính của app.



Tính năng

Cho phép người dùng có thể xem và chọn những sự kiện yêu thích theo từng chủ đề để tham gia. Khi người dùng click vào một sự kiện bất kì thì sẽ mở sang trang chi tiết sự kiện

3. Trang chi tiết sự kiện

Giao diện

1. Phần đầu – Tổng quan sự kiện:

- Giao diện nổi bật với hình ảnh poster bắt mắt của sự kiện ở phía trên cùng.
- Hiện thị **tên sự kiện, ngày tổ chức, giờ bắt đầu, và địa điểm cụ thể** (bao gồm tên địa điểm và địa chỉ).
- Thống kê **xếp hạng sao, số lượng đánh giá, và số lượng vé đã được đặt**, giúp người dùng đánh giá mức độ phổ biến và uy tín của sự kiện.

2. Thông tin chi tiết & Ưu đãi:

- Mục “**Giới thiệu sự kiện**” cung cấp lịch trình cụ thể, thông tin sân khấu, thời lượng, đối tượng khán giả phù hợp.
- Bao gồm chi tiết về các **loại vé, mức giá** (ví dụ: vé thường, vé VIP), và các **ưu đãi hiện có** như giảm giá %.
- Phần **thông tin liên hệ** (hotline, điều khoản mua vé) hỗ trợ người dùng cần tư vấn hoặc hỗ trợ thêm.

3. Giao diện hành động – Đặt vé:

- Nút “**MUA NGAY**” được thiết kế nổi bật với màu cam giúp người dùng dễ dàng nhận biết và hành động nhanh chóng.
- Giá vé được đặt ở vị trí dễ thấy (“Từ 1.500.000đ”), kèm theo biểu tượng giỏ hàng hoặc chuyển hướng tới trang thanh toán sau khi click.



Tính năng

-Khi nhân chọn sự kiện sẽ mở ra trang như hình bên. Người dùng sẽ có thể chọn ngày để tham gia sự kiện (tùy vào từng sự kiện sẽ diễn ra nhiều ngày hoặc chỉ 1 ngày).

-Cho phép người dùng chọn số lượng vé theo người lớn và trẻ em, giá vé sẽ thay đổi dựa trên kiểu người và kiểu vé, vé trong tuần sẽ có giá khác so với vé cuối tuần và lễ.

-Người dùng cũng có thể chọn chỗ ngồi thích hợp dựa vào sơ đồ ghế. Hệ thống cũng sẽ hỗ trợ người dùng chọn theo loại ghế vip hoặc thường.

Nếu người dùng chọn sai số lượng ghế theo số lượng vé đã mua thì hệ thống sẽ thông báo không hợp lệ và cho phép chọn lại

Sinh nhật Jack

20-04-2025 - 19:00

Vui lòng chọn ngày tham quan

21/05/2025

Người lớn - Vé Trong Tuần

Còn lại: 49

Giá: 350.000 đ

- 1 +

Trẻ em (100 - 139 cm) - Vé Trong Tuần

Còn lại: 50

Giá: 250.000 đ

- 0 +

Người lớn - Vé Cuối Tuần và Lễ

Còn lại: 50

Giá: 450.000 đ

- 0 +

Trẻ em (100 - 139cm) - Vé Cuối Tuần & Lễ

Còn lại: 50

Giá: 400.000 đ

- 0

Sơ đồ ghế

Ghế VIP +200,000đ

Ghế thường +100,000đ

Số ghế đã chọn không khớp với số lượng vé. Vui lòng chọn đủ ghế.

A1	A2	A3	A4
B1	B2	B3	B4

250.000 đ

Thêm vào giỏ hàng

Đặt ngay

-Sau khi người dùng đã thực hiện thành công chọn vé và chỗ ngồi, hệ thống sẽ chuyển sang trang tiếp theo để hoàn tất đặt hàng(2/2)

-Người dùng sẽ có thể xem lại chi tiết sự kiện mà mình đã đặt bao gồm thông tin sự kiện, chi tiết vé và chi tiết chỗ ngồi

-Khi đã kiểm tra thành công, hệ thống sẽ thông báo người dùng hãy xác nhận thông tin liên lạc của mình, bao gồm họ tên, số điện thoại và email.

-Sẽ có những ưu đãi dành cho người dùng để giảm giá tiền, tùy vào từng người dùng thì sẽ có những giảm giá khác nhau.

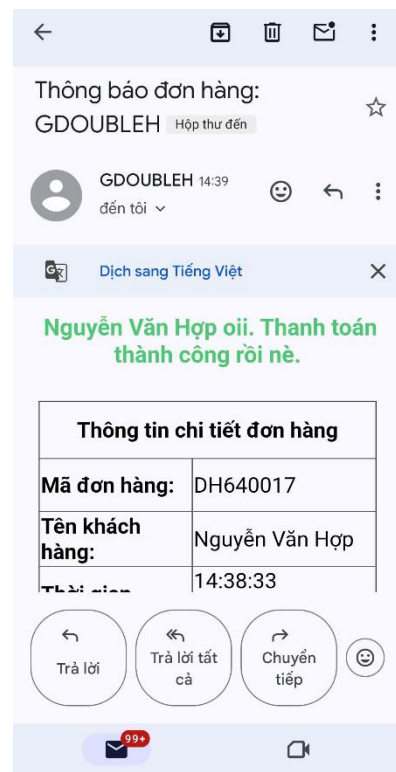
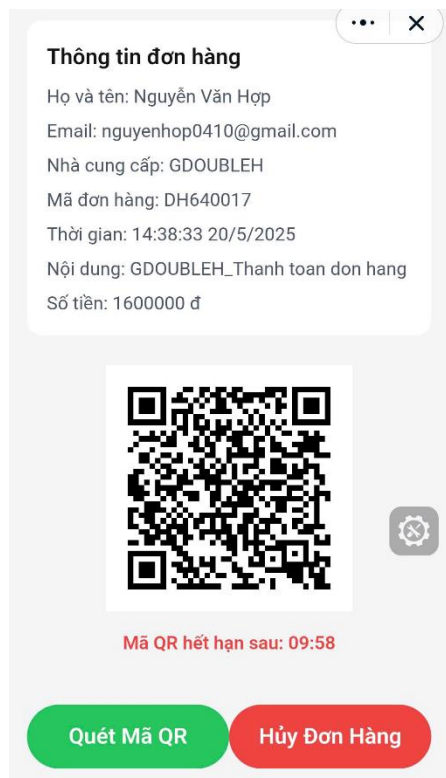
This is a screenshot of a modal window titled "Thông tin sự kiện" (Event Information). It contains the following details: "Sự kiện: Sinh nhật Jack" (Event: Jack's Birthday), "Ngày: 2025-05-21" (Date: 2025-05-21), "Thời gian: 19:00" (Time: 19:00), and "Địa điểm: Bến Tre" (Location: Bến Tre). There is a red button labeled "Xem chi tiết" (View details). Below this, there is a section titled "CHI TIẾT" (Details) with a blue "Đóng" (Close) button. Under "CHI TIẾT", there is a sub-section "Thông tin sự kiện" (Event Information) with the same details as above. Below that is a sub-section "Chi tiết vé" (Ticket Details) with the following information: "Loại vé: Trẻ em - Trong tuần" (Ticket type: Child - In week), "Số lượng: 1" (Quantity: 1), "Giá: 250.000 đ" (Price: 250,000 VND), and "Tổng: 250.000 đ" (Total: 250,000 VND). At the bottom, there is a sub-section "Chi tiết ghế" (Seat Details).

This is a screenshot of a form titled "Thông tin liên lạc" (Contact Information). It has a red "Đóng" (Close) button in the top right corner. The form contains four input fields: "Họ" (Last name) with the value "Nguyễn", "Tên" (First name) with the value "Văn Hợp", "Số điện thoại" (Phone number) with the value "0862782841", and "Email" with the value "nguyenhop0410@gmail.com". There is a red "Lưu thông tin" (Save information) button at the bottom. A settings icon is visible next to the phone number field.

This is a screenshot of a modal window titled "Chọn mã giảm giá" (Select discount code). It has a red "Đóng" (Close) button in the top right corner. The modal contains a list of discount codes: "Không dùng mã giảm giá" (Do not use discount code), "Giảm 10%" (10% off), "Giảm 20%" (20% off), and "Giảm 25%" (25% off). A settings icon is visible next to the "Không dùng mã giảm giá" option.

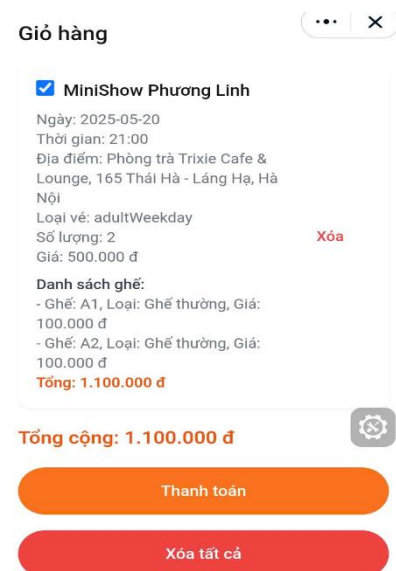
-Khi người dùng đã nhập đầy đủ thông tin ,sau đó nhấn thanh toán hệ thống thì hệ thống sẽ chuyển tới phần thanh toán gồm có mã QR để quét.

-Quét mã thành công sẽ có 1 email gửi về chính email mà người dùng cung cấp thông báo rằng bạn đã thực hiện thanh toán thành công.



Thêm vào giỏ hàng

-Nếu người dùng chưa muốn thực hiện mua ngay thì có thể thêm vào giỏ hàng



4. Trang danh mục

Giao diện:

- Phân nhóm: Chia thành "Vé ca nhạc" (concert, live show), "Vé du lịch," "Workshop," và "Văn hóa nghệ thuật" (triển lãm, biểu diễn).
- Hiển thị: Mỗi sự kiện có thumbnail bắt mắt, tiêu đề, ngày giờ (ví dụ: 14/10/2025), địa điểm, giá vé (từ 450.000 VNĐ), giúp nắm thông tin nhanh và thu hút người dùng.

Tính năng:

-Cho phép người dùng tìm kiếm và chọn lọc những sự kiện theo từng mục nhất định.

-Khi chọn một sự kiện nào đó thì sẽ quay về trang chi tiết sự kiện và cho phép người dùng có thể đặt vé liền tay hoặc thêm vào giỏ hàng tùy thích.



5. Trang lịch sử mua vé

Giao diện

- Tiêu đề và bộ lọc trạng thái:

Phần tiêu đề "**Lịch sử đơn hàng**" được in đậm. Ngay bên dưới là **ô lọc trạng thái** với placeholder dạng dropdown, viền bo tròn nhẹ, nền trắng, chữ xám nhạt. Giao diện lọc đơn giản nhưng trực quan, giúp người dùng dễ dàng chọn lọc theo các trạng thái đơn hàng (ví dụ: Đã thanh toán, Đã hủy,...)

- Danh sách đơn hàng:

Các đơn hàng được trình bày theo dạng **thẻ thông tin**, mỗi thẻ có bố cục rõ ràng:

Tiêu đề sự kiện in đậm (ví dụ: *MiniShow Thùy Chi*, *Sinh nhật Jack*).

Trạng thái đơn hàng được làm nổi bật:

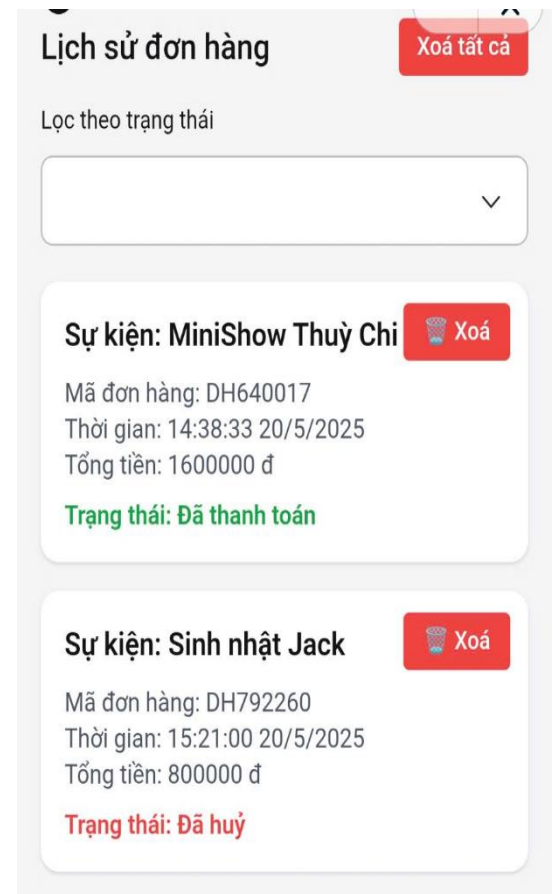
Đã thanh toán: hiển thị bằng chữ **màu xanh lá cây**.

Đã hủy: hiển thị bằng chữ **màu đỏ tươi**.

Nút "Xóa" nằm ở góc trên bên phải của mỗi thẻ, có màu đỏ nổi bật với biểu tượng thùng rác, dễ dàng thao tác.

Tính năng

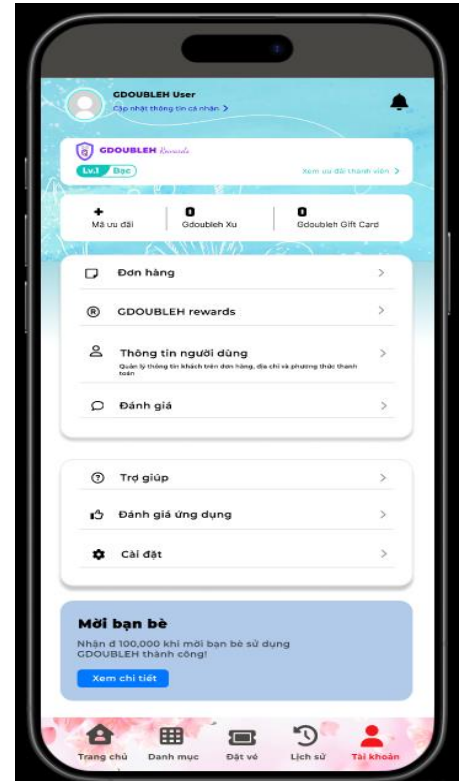
-Cho phép người dùng có thể xem được những sự kiện đã đặt vé rồi hoặc đã hủy



6. Trang tài khoản người dùng

Giao diện

- **Thông tin người dùng:** Phần trên cùng hiển thị tên "GDOUBLEH User" in đậm, bên dưới là cấp độ "GDOUBLEH Rewards" (Lv.1 Bạc) với nền xanh dương nhạt, chữ trắng, và nút "Xem ưu đãi thanh toán" màu xanh, chữ nhỏ, tạo điểm nhấn.
- **Tùy chọn tài khoản:** Gồm ba nút "Mã ưu đãi," "GDOUBLEH Xu," "GDOUBLEH Gift Card" với viền xám nhạt, chữ đen, bố trí ngang hàng, dễ thao tác. Danh sách bên dưới gồm "Đơn hàng," "GDOUBLEH rewards," "Thông tin người dùng," "Đánh giá," "Trợ giúp," "Đánh giá ứng dụng," "Cài đặt," mỗi mục có biểu tượng nhỏ bên trái, chữ đen trên nền trắng, kèm mũi tên phải màu xám.
- **Mời bạn bè:** Phần cuối là banner "Mời bạn bè" với text "Nhận 100.000 khi mời bạn bè sử dụng GDOUBLEH thành công," nền xanh nhạt, chữ trắng, nút "Xem chi tiết" màu xanh đậm, chữ trắng, bo góc, tạo cảm giác thu hút.



Tính năng

-Người dùng có thể quản lý đơn hàng, thông tin cá nhân, đánh giá sự kiện và truy cập các mục hỗ trợ. Ngoài ra, trang còn có tính năng mời bạn bè để nhận ưu đãi. Giao diện rõ ràng, dễ thao tác, tập trung vào trải nghiệm người dùng.

Chương 5. Cài đặt và kiểm thử

1. Môi trường cài đặt

Thành phần	Công cụ / Công nghệ sử dụng
Ngôn ngữ lập trình	Javascript
Frontend Framework	ReactJS (Kết hợp Vite)
Backend Framework	Firebase Cloud Functions (Node.js)
Cơ sở dữ liệu	Firebase Firestore
Xác thực	Firebase Authentication (Google, Facebook)
Hệ điều hành	Window 10, 11
IDE	Visual Studio Code
Công cụ thiết kế CSDL	Firebase Console (Firestore UI)
Thư viện hỗ trợ và utility	Tailwind CSS, React Icons
Mail thông báo	EmailJs

2. Kịch bản thử nghiệm

Chức năng	Kịch bản kiểm thử	Kết quả mong đợi
Đăng ký tài khoản	- Vào trang Đăng ký - Nhập họ tên, email, mật khẩu hợp lệ. - Nhấn “Đăng ký”	Tạo user mới, chuyển về trang Đăng nhập, hiển thị “Đăng ký thành công”
Đăng nhập	- Vào trang Đăng nhập - Nhập email + mật khẩu đúng. - Nhấn “Đăng nhập”	Chuyển đến trang chủ, hiển thị avatar và tên user
Xem danh sách sự kiện	- Truy cập trang chủ. - Scroll để xem slider & danh mục. - Nhấn vào 1 item	Chuyển đến Danh mục hoặc chi tiết event, hiển thị dữ liệu đúng

Xem chi tiết sự kiện	Tại trang danh sách, nhấn event bất kỳ. Đảm bảo URL /chitietsukien/:id	Hiển thị ảnh, tên, ngày giờ, địa điểm, mô tả, giá vé & đánh giá
Chọn vé / Tùy chọn	<ul style="list-style-type: none"> - Nhấn “CHỌN” ở chi tiết event. - Chuyển đến Tùy chọn - Chọn hạng vé, số lượng, nhấn tiếp tục 	Thông tin vé được đẩy vào state context, tiếp đến Giỏ hàng
Xem giỏ hàng	<ul style="list-style-type: none"> - Vào trang Giỏ hàng. - Kiểm tra list vé đã chọn. - Thay đổi số lượng / xoá item 	Cập nhật tổng tiền chính xác, phản hồi ngay lập tức trên UI
Thanh toán	<ul style="list-style-type: none"> - Nhấn “Thanh toán” trên /cart. - Chọn phương thức (chuyển khoản. - Xác nhận 	Chuyển đến Chuyển khoản, hiện thông tin chuyển khoản & QR code
Lịch sử đơn hàng	Vào trang Lịch sử.	Danh sách đơn đúng user, trạng thái & số tiền hiển thị chính xác
Đánh giá sự kiện	Mở phần “Đánh giá” tại chi tiết event sau khi thanh toán. Nhập text và nhấn “Đăng”	Bình luận mới lưu trong Firestore (hoặc local nếu quản lý cmt.js), hiển thị thêm
Đăng xuất	Nhấn logout ở header/profile. Ấn Xác nhận	Đăng xuất Firebase Auth, quay về trang Đăng nhập

3. Kết quả thử nghiệm

- Hệ thống chạy ổn định trong môi trường cài đặt.
- Các chức năng được thử nghiệm hoạt động đúng như thiết kế.
- Không phát hiện lỗi nghiêm trọng.
- Các tương tác giữa người dùng và hệ thống đều phản hồi đúng.

Chương 6. Tổng kết và hướng phát triển

1. Kết quả

Mini app bán vé sự kiện có giao diện thân thiện, trực quan với các trang chính như danh mục, đặt vé, lịch sử mua vé, và tài khoản. Trang danh mục phân loại rõ ràng (Ca nhạc, Du lịch, Workshop, Văn hóa nghệ thuật), hỗ trợ tìm kiếm và lọc nhanh. Trang đặt vé thiết kế đơn giản, dễ nhập thông tin, hiển thị giá vé rõ ràng. Lịch sử mua vé cung cấp thông tin chi tiết, dễ theo dõi. Trang tài khoản tích hợp nhiều chức năng như quản lý đơn hàng, ưu đãi, và cài đặt, tăng tính tiện lợi.

2. Hạn chế

- **Tương tác hạn chế:** Thiếu tính năng tương tác như đánh giá sự kiện hoặc gợi ý cá nhân hóa dựa trên sở thích người dùng.
- **Hỗ trợ đa ngôn ngữ:** Chỉ hỗ trợ tiếng Việt, chưa phù hợp với người dùng quốc tế hoặc khách du lịch.
- **Hiệu suất:** Với số lượng sự kiện lớn, tốc độ tải có thể bị ảnh hưởng nếu không tối ưu hóa dữ liệu.

3. Hướng phát triển

- **Cá nhân hóa trải nghiệm:** Thêm gợi ý sự kiện dựa trên lịch sử mua vé và sở thích, tích hợp tính năng đánh giá, bình luận sau sự kiện.
- **Hỗ trợ đa ngôn ngữ:** Bổ sung ngôn ngữ (Anh, Trung, Hàn) để thu hút khách quốc tế, kèm bản đồ định vị địa điểm sự kiện.
- **Tối ưu hiệu suất:** Sử dụng phân trang hoặc tải dữ liệu theo nhu cầu, tích hợp AI để dự đoán và ưu tiên hiển thị sự kiện phù hợp.
- **Tích hợp thanh toán đa dạng:** Hỗ trợ ví điện tử quốc tế (PayPal, thẻ tín dụng), thêm mã QR để check-in sự kiện nhanh chóng.

Tài liệu tham khảo:

1. Object-Oriented and Classical Software Engineering. 8th edition, McGraw Hill 2010.
2. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, Data Structures and Algorithms, Addison – Wesley, 1983.
3. Niklas Wirth, Data Structures and Algorithms, Prentice Hall, 2004.
4. Russ Miles, Kim Hamilton, Learning UML 2.0, O'Reilly Media, 2006.
5. Website: <http://uml-diagrams.org/>
6. Một số bài tiểu luận và nguồn khác trên internet.
7. Kiến thức trong quá trình học môn Nhập môn Công nghệ phần mềm của thầy Châu Văn Vân