



**HUTECH**  
Đại học Công nghệ Tp.HCM

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM**

## **ĐỒ ÁN CHUYÊN NGÀNH**

**ĐỀ TÀI**

**S-Wallet Ví Điện Tử Sinh Viên**

Giảng viên hướng dẫn: **Trần Văn Hùng**

Sinh viên thực hiện: **2280603283 - Đặng Doanh Toại**

**2280603036 - Phan Thanh Thiên**

TP. HỒ CHÍ MINH – THÁNG 11 NĂM 2025

# Mục lục

<b>LỜI MỞ ĐẦU</b>	<b>1</b>
<b>LỜI CAM ĐOAN</b>	<b>2</b>
<b>Chương 1. TỔNG QUAN VỀ NỀN TẢNG VÍ ĐIỆN TỬ SINH VIÊN</b>	<b>3</b>
1.1 Giới thiệu . . . . .	3
1.2 Tổng quan hệ thống . . . . .	3
1.2.1 Mục đích hệ thống . . . . .	3
1.2.2 Khảo sát hiện trạng và giải pháp . . . . .	3
1.2.3 Yêu cầu hoạt động của ứng dụng . . . . .	4
1.3 Thiết kế tương tác . . . . .	5
1.4 Phương pháp tiếp cận và giải quyết vấn đề . . . . .	5
1.4.1 Mô hình tổng quát hệ thống . . . . .	5
1.4.2 Kiến trúc phần mềm và Công nghệ . . . . .	5
<b>Chương 2. CƠ SỞ LÝ THUYẾT</b>	<b>7</b>
2.1 Giới thiệu . . . . .	7
2.2 Mô hình Client-Server . . . . .	7
2.3 Công nghệ Backend . . . . .	8
2.3.1 Node.js . . . . .	8
2.3.2 Express.js . . . . .	8
2.4 Công nghệ Cơ sở dữ liệu . . . . .	9
2.4.1 MongoDB . . . . .	9
2.5 Công nghệ Mobile . . . . .	9
2.5.1 Flutter . . . . .	9
2.6 Công nghệ Web Frontend . . . . .	10
2.6.1 React.js . . . . .	10
2.7 Công nghệ Tích hợp . . . . .	10
2.7.1 MoMo Payment Gateway . . . . .	10
2.7.2 Google Gemini API . . . . .	11
2.8 Giao thức và Bảo mật . . . . .	11
2.8.1 RESTful API . . . . .	11
2.8.2 JWT (JSON Web Token) . . . . .	11
2.8.3 NFC (Near Field Communication) . . . . .	11
<b>Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG</b>	<b>12</b>

3.1 Phân Tích Hệ Thống . . . . .	12
3.1.1 Biểu đồ Use Case . . . . .	12
3.1.2 Đặc tả các Use Case chính . . . . .	13
3.2 Thiết Kế Hệ Thống . . . . .	16
3.2.1 Biểu đồ Activity (Hoạt động) . . . . .	16
3.2.2 Biểu đồ Sequence (Tuần tự) . . . . .	18
<b>Chương 4. THIẾT KẾ CƠ SỞ DỮ LIỆU</b>	<b>19</b>
4.1 Thiết kế sơ đồ . . . . .	19
4.2 Thiết kế chi tiết Collection (MongoDB) . . . . .	19
4.3 Mô hình quan hệ (Schema Diagram) . . . . .	22
<b>Chương 5. KẾT QUẢ THỰC NGHIỆM</b>	<b>23</b>
5.1 Môi trường và Công cụ thực nghiệm . . . . .	23
5.2 Kết quả triển khai Ứng dụng Sinh viên (Mobile App) . . . . .	23
5.2.1 1. Đăng nhập và Màn hình chính . . . . .	23
5.2.2 2. Chức năng Nạp tiền qua MoMo . . . . .	24
5.2.3 3. Chức năng Thanh toán NFC . . . . .	24
5.2.4 4. Chatbot AI Tư vấn tài chính . . . . .	26
5.3 Kết quả triển khai Web Quản trị (Admin Portal) . . . . .	27
5.3.1 1. Dashboard Thông kê . . . . .	27
5.3.2 2. Quản lý Giao dịch và Sinh viên . . . . .	27
<b>Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN</b>	<b>28</b>
6.1 Kết luận . . . . .	28
6.2 Hướng phát triển . . . . .	29
6.2.1 1. Mở rộng tính năng và Nền tảng . . . . .	29
6.2.2 2. Hệ sinh thái tài chính mở rộng . . . . .	29
6.2.3 3. Ứng dụng Blockchain . . . . .	29
<b>TÀI LIỆU THAM KHẢO</b>	<b>30</b>

# Danh sách hình vẽ

1	Sơ đồ Kiến trúc Hệ thống . . . . .	6
2	Sơ đồ Use Case Tổng quát . . . . .	13
3	Sơ đồ hoạt động chức năng Thanh toán NFC . . . . .	16
4	Sơ đồ hoạt động chức năng Nạp tiền MoMo . . . . .	17
5	Sơ đồ tuần tự chức năng Chat AI . . . . .	18
6	Sơ đồ quan hệ cơ sở dữ liệu . . . . .	22
7	Giao diện người dùng cơ bản . . . . .	23
8	Quy trình nạp tiền qua ví MoMo . . . . .	24
9	Mô phỏng thanh toán một chạm NFC . . . . .	25
10	Tương tác với trợ lý ảo AI . . . . .	26
11	Dashboard quản trị viên . . . . .	27

## Danh sách bảng

1	Đặc tả Use Case: Thanh toán NFC . . . . .	13
2	Đặc tả Use Case: Nạp tiền qua MoMo . . . . .	14
3	Đặc tả Use Case: Chat với AI . . . . .	15
4	Cấu trúc Collection Users . . . . .	19
5	Cấu trúc Collection Wallets . . . . .	20
6	Cấu trúc Collection Transactions . . . . .	20
7	Cấu trúc Collection ChatSessions . . . . .	21

## LỜI MỞ ĐẦU

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trên toàn cầu, việc ứng dụng công nghệ vào đời sống hàng ngày đã trở thành xu thế tất yếu. Đặc biệt trong lĩnh vực tài chính, thanh toán không dùng tiền mặt đang dần thay thế các phương thức truyền thống nhờ sự tiện lợi, an toàn và tốc độ vượt trội. Tại môi trường giáo dục đại học, nhu cầu về một hệ thống thanh toán thông minh, tích hợp đa tiện ích cho sinh viên ngày càng trở nên cấp thiết.

Nắm bắt được nhu cầu đó, đồ án này tập trung nghiên cứu và phát triển “S-Wallet Ví Điện Tử Sinh Viên” - một giải pháp tài chính số toàn diện. Hệ thống không chỉ giúp sinh viên đơn giản hóa việc thanh toán các dịch vụ trong khuôn viên trường thông qua công nghệ NFC và Mobile App, mà còn cung cấp công cụ quản lý chi tiêu hiệu quả tích hợp trí tuệ nhân tạo.

Với sự kết hợp giữa ứng dụng di động hiện đại và cổng thông tin quản trị website mạnh mẽ, dự án mong muốn kiến tạo một hệ sinh thái thanh toán không tiền mặt văn minh, minh bạch và tối ưu hóa trải nghiệm cho cả sinh viên lẫn nhà trường, góp phần thúc đẩy tiến trình số hóa trong môi trường giáo dục.

## LỜI CAM ĐOAN

Nhóm chúng em xin cam đoan rằng đồ án “**S-Wallet Ví Điện Tử Sinh Viên**” là kết quả nghiên cứu và thực hiện của tất cả thành viên trong nhóm, dưới sự hướng dẫn khoa học của giảng viên.

Toàn bộ nội dung, số liệu và kết quả trình bày trong đồ án này là trung thực, được chúng tôi tìm hiểu, phân tích và xây dựng dựa trên kiến thức chuyên môn và các tài liệu tham khảo uy tín. Các chức năng của hệ thống như thanh toán NFC, tích hợp cổng thanh toán MoMo, và chatbot AI đều được nhóm tự nghiên cứu và phát triển. Chúng tôi cam kết không sao chép nguyên văn hoặc sử dụng trái phép nội dung từ bất kỳ nguồn nào mà không có trích dẫn rõ ràng.

Nếu phát hiện có bất kỳ hành vi đạo văn hoặc gian lận nào trong quá trình thực hiện đồ án này, chúng tôi xin chịu hoàn toàn trách nhiệm trước Nhà trường và Hội đồng đánh giá.

Chúng tôi xin chân thành cảm ơn!

# **Chương 1. TỔNG QUAN VỀ NỀN TẢNG VÍ ĐIỆN TỬ SINH VIÊN**

## **1.1. Giới thiệu**

Trong môi trường đại học hiện đại, các giao dịch tài chính hàng ngày của sinh viên như thanh toán tại căn tin, cửa hàng tiện lợi, hay đóng các khoản phí nhỏ thường vẫn phụ thuộc nhiều vào tiền mặt. Điều này không chỉ gây bất tiện về mặt thời gian xử lý, khó khăn trong việc quản lý chi tiêu cá nhân mà còn tiềm ẩn rủi ro mất mát. Mặc dù các ví điện tử phổ biến như MoMo hay ZaloPay đã phát triển mạnh, nhưng vẫn thiếu một giải pháp chuyên biệt được tùy biến sâu cho hệ sinh thái của trường đại học, tích hợp chặt chẽ với thẻ sinh viên và các dịch vụ nội bộ.

Đồ án “Nền tảng Ví điện tử Sinh viên” được xây dựng nhằm giải quyết các vấn đề trên thông qua một hệ thống thanh toán khép kín, hiện đại. Dự án cung cấp một ứng dụng di động đa nền tảng cho sinh viên và một cổng quản trị web cho nhà trường, tích hợp các công nghệ tiên tiến như NFC cho thanh toán một chạm và AI Chatbot để tư vấn tài chính. Chương này sẽ trình bày tổng quan về bối cảnh, mục tiêu, yêu cầu chức năng và kiến trúc kỹ thuật của hệ thống.

## **1.2. Tổng quan hệ thống**

### **1.2.1. Mục đích hệ thống**

Hệ thống được thiết kế với mục tiêu cốt lõi là hiện đại hóa trải nghiệm thanh toán trong khuôn viên trường học. Các mục tiêu cụ thể bao gồm:

- **Tiện lợi hóa thanh toán:** Thay thế tiền mặt và thẻ vật lý bằng thanh toán NFC một chạm.
- **Quản lý tài chính thông minh:** Cung cấp cho sinh viên công cụ theo dõi lịch sử giao dịch chi tiết và nhận tư vấn chi tiêu từ trợ lý ảo AI.
- **Tối ưu hóa quản lý:** Cung cấp cho nhà trường công cụ quản lý tập trung các giao dịch, doanh thu và tài khoản sinh viên.
- **An toàn và bảo mật:** Đảm bảo tính toàn vẹn của giao dịch và dữ liệu người dùng thông qua các chuẩn bảo mật hiện đại.

### **1.2.2. Khảo sát hiện trạng và giải pháp**

Hiện tại, sinh viên thường gặp các vấn đề:

- **Tiền mặt:** Tốn thời gian thối tiền, dễ rơi rớt, khó thống kê chi tiêu.

- **Ví điện tử đại chúng:** Chưa tích hợp sâu với các dịch vụ đặc thù của trường (ví dụ: tích hợp thẻ thư viện, vé xe, giảm giá nội bộ).

Giải pháp “Ví điện tử Sinh viên” mang lại sự khác biệt:

- Tích hợp công nghệ NFC, tận dụng hạ tầng thẻ hoặc điện thoại có sẵn.
- Hệ thống nạp/rút tiền linh hoạt, kết nối trực tiếp với cổng thanh toán MoMo.
- Chatbot AI (Google Gemini) tích hợp sẵn để hỗ trợ giải đáp thắc mắc và tư vấn tài chính cá nhân hóa.

### 1.2.3. Yêu cầu hoạt động của ứng dụng

#### Phần dành cho Sinh viên (Mobile App)

Ứng dụng di động là cổng giao tiếp chính của sinh viên với hệ thống, bao gồm các chức năng:

- **Quản lý tài khoản:** Đăng ký, đăng nhập bảo mật, cập nhật thông tin cá nhân.
- **Ví điện tử:** Xem số dư thời gian thực, nạp tiền vào ví thông qua cổng MoMo.
- **Thanh toán NFC:** Thực hiện thanh toán không tiếp xúc tại các điểm chấp nhận trong trường.
- **Lịch sử giao dịch:** Xem danh sách chi tiết các giao dịch nạp/rút/thanh toán với bộ lọc thời gian.
- **Trợ lý ảo AI:** Chat với AI để hỏi về số dư, lịch sử chi tiêu hoặc nhận lời khuyên quản lý tài chính.

#### Phần dành cho Quản trị viên (Web Portal)

Cổng thông tin web dành cho ban quản lý nhà trường hoặc bộ phận tài chính:

- **Dashboard:** Xem biểu đồ thống kê tổng quan về dòng tiền, số lượng giao dịch và người dùng mới.
- **Quản lý sinh viên:** Thêm, sửa, xóa hoặc khóa tài khoản sinh viên; xem chi tiết hồ sơ.
- **Quản lý giao dịch:** Tra cứu và kiểm soát toàn bộ lịch sử giao dịch trong hệ thống để đối soát.

## 1.3. Thiết kế tương tác

Trải nghiệm người dùng (UX) là ưu tiên hàng đầu trong thiết kế hệ thống:

- **Mobile App (Flutter)**: Giao diện Material Design hiện đại, tối ưu cho thao tác một tay. Các luồng thanh toán được tối giản hóa để hoàn tất trong dưới 3 bước.
- **Web Admin (React.js)**: Giao diện Dashboard trực quan, sử dụng các biểu đồ và bảng dữ liệu dynamic để hỗ trợ ra quyết định nhanh chóng. Responsive design giúp quản trị viên có thể truy cập từ nhiều loại thiết bị.

## 1.4. Phương pháp tiếp cận và giải quyết vấn đề

### 1.4.1. Mô hình tổng quát hệ thống

Hệ thống hoạt động theo mô hình Client-Server với kiến trúc vi dịch vụ (Microservices) đơn giản hóa:

- **Mobile Client**: Ứng dụng Android xây dựng bằng Flutter, giao tiếp với Server qua RESTful API.
- **Web Client**: Trang quản trị React.js, giao tiếp với cùng hệ thống API.
- **Backend Server**: Node.js/Express đóng vai trò trung tâm xử lý logic nghiệp vụ.
- **Database**: MongoDB lưu trữ dữ liệu phi cấu trúc (NoSQL) đảm bảo khả năng mở rộng.

### 1.4.2. Kiến trúc phần mềm và Công nghệ

Dự án sử dụng bộ công nghệ (Tech Stack) hiện đại và phổ biến:

- **Backend**: Node.js và Express.js cung cấp hiệu năng cao cho các tác vụ I/O. Xác thực người dùng an toàn bằng JWT (JSON Web Token).
- **Database**: MongoDB (triển khai trên MongoDB Atlas) cho phép lưu trữ dữ liệu giao dịch và người dùng linh hoạt.
- **Mobile**: Flutter (Dart) cho phép phát triển ứng dụng đa nền tảng với hiệu suất gần như native. Sử dụng thư viện flutter\_nfc\_kit cho tính năng NFC và riverpod cho quản lý trạng thái.

– **Frontend Web:** React.js kết hợp với Material-UI mang lại giao diện quản trị chuyên nghiệp.

– **Tích hợp bên thứ 3:**

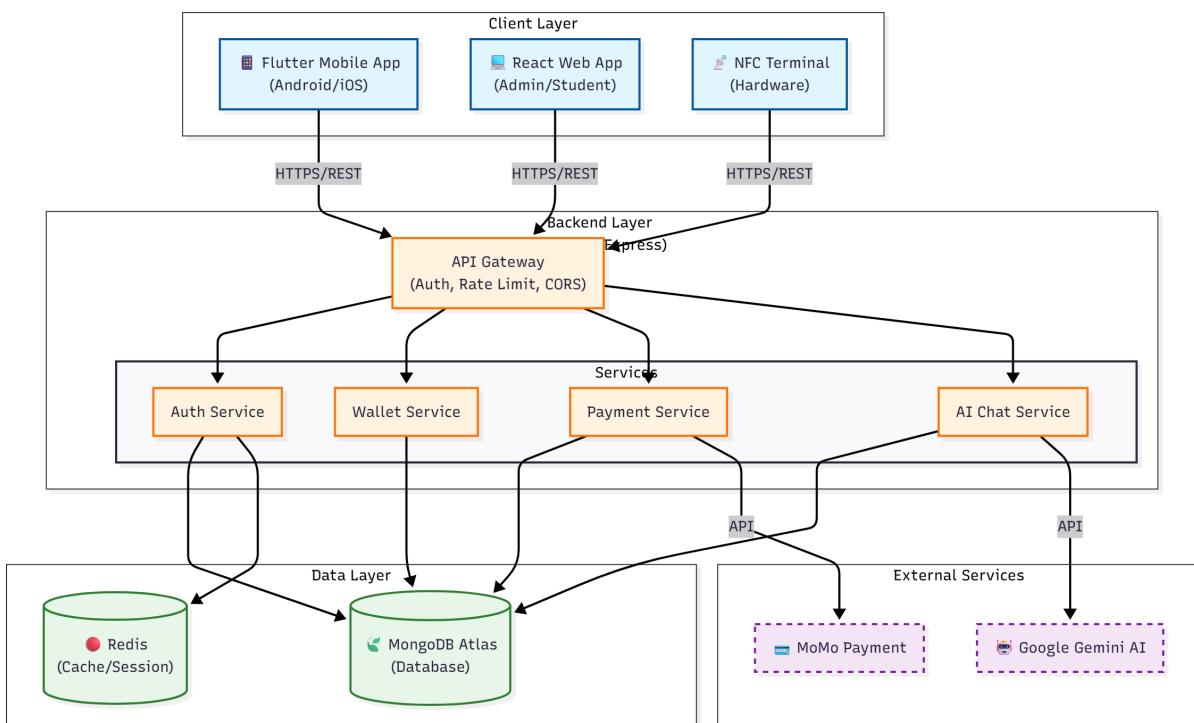
+ **MoMo API:** Cổng thanh toán để nạp tiền vào ví.

+ **Google Gemini API:** Cung cấp trí tuệ nhân tạo cho tính năng Chatbot.

– **Hạ tầng và Triển khai (DevOps):** Quy trình triển khai được tự động hóa (CI/CD) thông qua GitHub.

+ **Backend (Railway):** Tự động build và deploy container mỗi khi có cập nhật mã nguồn, cung cấp môi trường serverless linh hoạt và HTTPS mặc định.

+ **Frontend (Vercel):** Tối ưu hóa việc phân phối nội dung tĩnh qua mạng lưới Edge Network toàn cầu, giúp giảm độ trễ truy cập cho người dùng.



Hình 1. Sơ đồ Kiến trúc Hệ thống

## **Chương 2. CƠ SỞ LÝ THUYẾT**

### **2.1. Giới thiệu**

Chương này trình bày cơ sở lý thuyết của các công nghệ được sử dụng trong đồ án "Nền tảng Ví điện tử Sinh viên". Hệ thống được xây dựng dựa trên sự kết hợp giữa các công nghệ web và di động hiện đại, đảm bảo hiệu suất cao, khả năng mở rộng và trải nghiệm người dùng tối ưu. Các công nghệ chính bao gồm Node.js và Express.js cho Backend, MongoDB cho cơ sở dữ liệu, Flutter cho Mobile App, và React.js cho Web Portal. Bên cạnh đó, hệ thống tích hợp các dịch vụ bên thứ ba quan trọng như MoMo Payment Gateway cho thanh toán và Google Gemini API cho trí tuệ nhân tạo.

### **2.2. Mô hình Client-Server**

**Khái niệm:** Mô hình Client-Server (Khách-Chủ) là mô hình kiến trúc mạng máy tính trong đó hệ thống được chia thành hai thành phần chính: Client (Máy khách) gửi yêu cầu và Server (Máy chủ) xử lý yêu cầu và phản hồi kết quả. Đây là mô hình cơ bản cho hầu hết các ứng dụng web và di động hiện đại.

#### **Cơ chế hoạt động:**

- Client:** (Mobile App hoặc Web Browser) gửi yêu cầu (HTTP Request) đến Server thông qua Internet. Yêu cầu chứa thông tin về hành động người dùng muốn thực hiện (ví dụ: lấy số dư, thanh toán).
- Server:** (Backend API) nhận yêu cầu, xử lý logic nghiệp vụ, truy xuất dữ liệu từ Database nếu cần, và tạo phản hồi.
- Database:** Lưu trữ và cung cấp dữ liệu cho Server.
- Server:** Gửi phản hồi (HTTP Response) trả về cho Client, thường dưới dạng JSON.
- Client:** Nhận phản hồi và hiển thị kết quả cho người dùng.

#### **Ưu điểm:**

- Tập trung hóa:** Dữ liệu được quản lý tập trung tại Server, dễ dàng bảo mật và sao lưu.
- Khả năng mở rộng:** Có thể nâng cấp Server hoặc thêm nhiều Server để phục vụ lượng lớn Client.

- **Đa nền tảng:** Một Server có thể phục vụ nhiều loại Client khác nhau (Web, Android, iOS).

## 2.3. Công nghệ Backend

### 2.3.1. Node.js

**Khái niệm:** Node.js là một môi trường chạy mã JavaScript phía máy chủ (server-side runtime environment), được xây dựng trên V8 JavaScript engine của Google Chrome. Node.js cho phép phát triển các ứng dụng mạng nhanh chóng và dễ mở rộng.

#### Đặc điểm nổi bật:

- **Non-blocking I/O:** Node.js sử dụng mô hình nhập/xuất không chặn, cho phép xử lý nhiều kết nối đồng thời mà không cần tạo luồng mới cho mỗi yêu cầu, giúp tiết kiệm tài nguyên hệ thống.
- **Event-driven:** Hoạt động dựa trên cơ chế sự kiện, giúp xử lý các tác vụ bất đồng bộ hiệu quả.
- **Single Language:** Sử dụng JavaScript cho cả Client và Server, giúp đồng bộ hóa logic và tăng tốc độ phát triển.

**Ứng dụng trong dự án:** Node.js đóng vai trò là nền tảng cốt lõi cho Backend Server, xử lý toàn bộ logic nghiệp vụ, kết nối Database và tích hợp các API bên thứ ba.

### 2.3.2. Express.js

**Khái niệm:** Express.js là một framework web tối giản và linh hoạt dành cho Node.js, cung cấp bộ tính năng mạnh mẽ để xây dựng các ứng dụng web và mobile.

#### Vai trò:

- **Routing:** Quản lý các đường dẫn API (Endpoint) để Client gửi yêu cầu (ví dụ: /api/users, /api/transactions).
- **Middleware:** Xử lý các tác vụ trung gian như xác thực người dùng (Authentication), ghi log (Logging), xử lý lỗi (Error Handling) trước khi đến controller chính.
- **Hiệu suất:** Giúp xây dựng RESTful API nhanh chóng và cấu trúc code rõ ràng.

## 2.4. Công nghệ Cơ sở dữ liệu

### 2.4.1. MongoDB

**Khái niệm:** MongoDB là hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, lưu trữ dữ liệu dưới dạng văn bản (Document) theo định dạng JSON/BSON linh hoạt, thay vì dạng bảng (Table) như SQL truyền thống.

**Lý do lựa chọn:**

- **Schema linh hoạt:** Phù hợp với dữ liệu thay đổi thường xuyên như thông tin người dùng, log giao dịch.
- **Hiệu năng cao:** Tốc độ đọc/ghi nhanh, đặc biệt với lượng dữ liệu lớn.
- **Khả năng mở rộng:** Hỗ trợ Horizontal Scaling (Sharding) dễ dàng.
- **Tương thích tốt:** Dữ liệu JSON của MongoDB tương thích hoàn hảo với JavaScript (Node.js/React/React Native).

**Ứng dụng trong dự án:** Lưu trữ thông tin sinh viên, ví điện tử, lịch sử giao dịch, log hệ thống và cấu hình.

## 2.5. Công nghệ Mobile

### 2.5.1. Flutter

**Khái niệm:** Flutter là bộ công cụ phát triển phần mềm giao diện người dùng (UI toolkit) mã nguồn mở do Google phát triển, cho phép xây dựng ứng dụng biên dịch gốc (natively compiled) cho di động, web và desktop từ một codebase duy nhất.

**Ngôn ngữ Dart:** Flutter sử dụng ngôn ngữ lập trình Dart, được thiết kế tối ưu cho phát triển giao diện người dùng với tính năng "Hot Reload" giúp xem thay đổi code ngay lập tức mà không cần khởi động lại ứng dụng.

**Đặc điểm kỹ thuật:**

- **Widget-based:** Mọi thành phần giao diện trong Flutter đều là Widget, giúp tùy biến cao và nhất quán trên các nền tảng.
- **Hiệu suất Native:** Flutter biên dịch code Dart thành mã máy (ARM/x86), không thông qua cầu nối (bridge) JavaScript như React Native, mang lại hiệu suất gần như ứng dụng gốc.

**Thư viện sử dụng:**

- **flutter\_nfc\_kit**: Thư viện hỗ trợ đọc/ghi thẻ NFC đa nền tảng, là nòng cốt cho tính năng thanh toán chạm.
- **riverpod**: Giải pháp quản lý trạng thái (State Management) hiện đại, an toàn và dễ test.

## 2.6. Công nghệ Web Frontend

### 2.6.1. React.js

**Khái niệm:** React.js là thư viện JavaScript phổ biến nhất hiện nay để xây dựng giao diện người dùng, được phát triển bởi Facebook.

**Đặc điểm:**

- **Component-based**: Chia nhỏ giao diện thành các thành phần độc lập, có thể tái sử dụng.
- **Virtual DOM**: Tối ưu hóa việc cập nhật giao diện, giúp ứng dụng chạy mượt mà.
- **Hệ sinh thái phong phú**: Rất nhiều thư viện hỗ trợ như Material-UI (giao diện), React Router (điều hướng), Axios (gọi API).

**Ứng dụng trong dự án:** Xây dựng trang Dashboard quản trị (Admin Portal) cho nhà trường, giúp quản lý sinh viên và thống kê giao dịch trực quan.

## 2.7. Công nghệ Tích hợp

### 2.7.1. MoMo Payment Gateway

**Giới thiệu:** MoMo là ví điện tử hàng đầu tại Việt Nam. Cổng thanh toán MoMo cho phép các ứng dụng bên thứ ba tích hợp để thực hiện thanh toán trực tuyến an toàn.

**Cơ chế tích hợp (IPN - Instant Payment Notification):** Hệ thống sử dụng cơ chế IPN để xử lý giao dịch nạp tiền tự động:

1. Người dùng tạo yêu cầu nạp tiền trên App.
2. App gọi API MoMo để lấy link thanh toán (QR Code/Deeplink).
3. Người dùng thanh toán trên App MoMo.
4. Server MoMo gửi thông báo kết quả (IPN) về Backend Server của dự án.
5. Backend Server xác thực chữ ký số (Signature) và cập nhật số dư ví sinh viên.

## **2.7.2. Google Gemini API**

**Giới thiệu:** Google Gemini là mô hình ngôn ngữ lớn (LLM) đa phương thức tiên tiến của Google. API cho phép tích hợp khả năng hiểu và tạo ngôn ngữ tự nhiên vào ứng dụng.

### **Ứng dụng trong Chatbot Tài chính:**

- **Context-aware:** Chatbot được cung cấp ngữ cảnh là lịch sử giao dịch và số dư hiện tại của sinh viên (dữ liệu được ẩn danh/bảo mật).
- **Tư vấn:** Dựa trên dữ liệu, AI phân tích thói quen chi tiêu và đưa ra lời khuyên tài chính, cảnh báo chi tiêu quá đà hoặc gợi ý tiết kiệm.
- **Tương tác tự nhiên:** Hỗ trợ giao tiếp bằng tiếng Việt tự nhiên, thân thiện.

## **2.8. Giao thức và Bảo mật**

### **2.8.1. RESTful API**

Hệ thống sử dụng kiến trúc REST (Representational State Transfer) để thiết kế API, đảm bảo tính thống nhất và dễ dàng tích hợp giữa Mobile, Web và Backend. Các phương thức HTTP chuẩn (GET, POST, PUT, DELETE) được sử dụng để thực hiện các thao tác CRUD.

### **2.8.2. JWT (JSON Web Token)**

Sử dụng JWT cho cơ chế xác thực (Authentication) và phân quyền (Authorization). Khi người dùng đăng nhập thành công, Server cấp một Token mã hóa. Client gửi Token này trong header của mỗi yêu cầu tiếp theo để xác minh danh tính mà không cần gửi lại mật khẩu.

### **2.8.3. NFC (Near Field Communication)**

Công nghệ giao tiếp trường gần cho phép hai thiết bị kết nối khi đặt gần nhau (dưới 4cm). Trong dự án, NFC được dùng để mô phỏng thẻ thanh toán: điện thoại đóng vai trò thẻ (Card Emulation) hoặc đầu đọc (Reader) để thực hiện giao dịch nhanh chóng, bảo mật.

## **Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

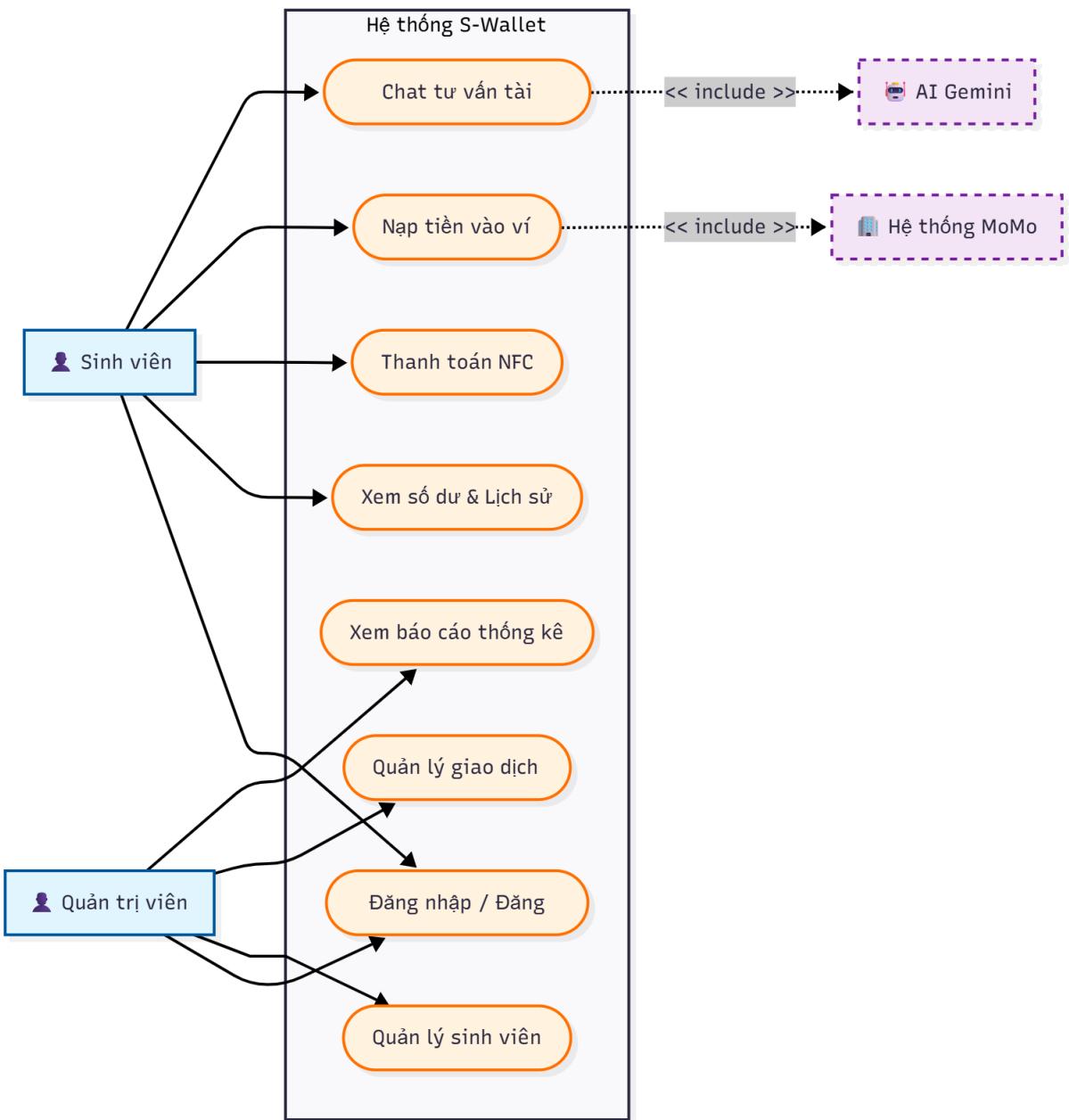
### **3.1. Phân Tích Hệ Thống**

#### **3.1.1. Biểu đồ Use Case**

##### **Use Case Tổng quát**

Hệ thống "Nền tảng Ví điện tử Sinh viên" bao gồm các tác nhân chính và các chức năng tổng quát như sau:

- **Sinh viên:** Đăng ký, Đăng nhập, Thanh toán NFC, Nạp tiền (MoMo), Xem lịch sử giao dịch, Chat với AI.
- **Quản trị viên:** Đăng nhập, Quản lý sinh viên (CRUD), Xem thống kê Dashboard, Quản lý giao dịch.
- **Hệ thống thanh toán (MoMo):** Xử lý yêu cầu nạp tiền, Gửi thông báo kết quả (IPN).
- **Hệ thống AI (Gemini):** Phản hồi câu hỏi, Phân tích chi tiêu.



Hình 2. Sơ đồ Use Case Tổng quát

### 3.1.2. Đặc tả các Use Case chính

#### 1. Thanh toán NFC

Cho phép sinh viên thanh toán tại các điểm chấp nhận (căn tin, thư viện) bằng cách chạm điện thoại vào thiết bị đọc thẻ.

Bảng 1. Đặc tả Use Case: Thanh toán NFC

<b>Tên Use Case</b>	Thanh toán NFC
<b>Mô tả</b>	Sinh viên sử dụng điện thoại có hỗ trợ NFC để thanh toán hóa đơn.

<b>Tác nhân</b>	Sinh viên
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>– Sinh viên đã đăng nhập vào ứng dụng Mobile.</li> <li>– Điện thoại có hỗ trợ NFC và NFC đang bật.</li> <li>– Số dư ví đủ để thanh toán.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>– Số dư ví bị trừ tương ứng với số tiền thanh toán.</li> <li>– Giao dịch được lưu vào lịch sử.</li> <li>– Thông báo thành công hiển thị trên thiết bị.</li> </ul>
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Sinh viên chọn chức năng "Thanh toán" hoặc đưa điện thoại lại gần đầu đọc.</li> <li>2. Ứng dụng phát hiện tín hiệu NFC từ đầu đọc.</li> <li>3. Ứng dụng hiển thị thông tin giao dịch (Số tiền, Đơn vị thu hưởng).</li> <li>4. Sinh viên xác nhận thanh toán (có thể yêu cầu PIN/Vân tay).</li> <li>5. Hệ thống xử lý trừ tiền và gửi xác nhận.</li> <li>6. Thông báo "Thanh toán thành công".</li> </ol>
<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>– <b>Số dư không đủ:</b> Thông báo lỗi và gợi ý nạp tiền.</li> <li>– <b>Lỗi kết nối NFC:</b> Yêu cầu thử lại hoặc nhập mã thủ công.</li> </ul>

## 2. Nạp tiền qua MoMo

Quy trình nạp tiền vào ví điện tử sinh viên thông qua cổng thanh toán MoMo.

Bảng 2. Đặc tả Use Case: Nạp tiền qua MoMo

<b>Tên Use Case</b>	Nạp tiền qua MoMo
---------------------	-------------------

<b>Mô tả</b>	Sinh viên nạp tiền vào tài khoản ví từ ứng dụng MoMo.
<b>Tác nhân</b>	Sinh viên, Hệ thống MoMo
<b>Tiền điều kiện</b>	Sinh viên đã đăng nhập và liên kết hoặc cài đặt ứng dụng MoMo.
<b>Hậu điều kiện</b>	Số dư ví sinh viên tăng lên tương ứng số tiền nạp.
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Sinh viên chọn "Nạp tiền" trên màn hình chính.</li> <li>2. Nhập số tiền cần nạp và chọn nguồn tiền "MoMo".</li> <li>3. Hệ thống chuyển hướng sang ứng dụng MoMo.</li> <li>4. Sinh viên xác nhận thanh toán trên MoMo.</li> <li>5. MoMo xử lý và gửi thông báo (IPN) về Server.</li> <li>6. Server cập nhật số dư và thông báo cho Sinh viên.</li> </ol>
<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>– <b>Hủy giao dịch:</b> Sinh viên hủy thanh toán trên MoMo → Quay về app, hiển thị "Giao dịch bị hủy".</li> <li>– <b>Lỗi mạng:</b> Thông báo "Vui lòng kiểm tra kết nối".</li> </ul>

### 3. Chat với Trợ lý ảo AI

Sinh viên tương tác với chatbot để hỏi về tài chính cá nhân.

Bảng 3. Đặc tả Use Case: Chat với AI

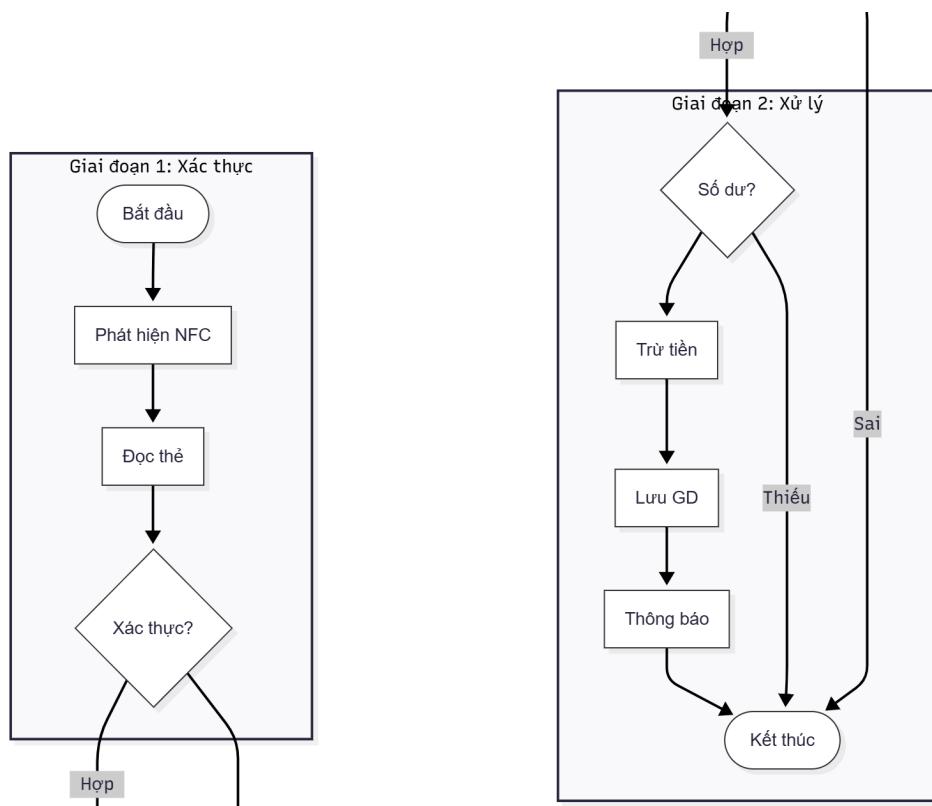
<b>Tên Use Case</b>	Chat với AI Financial Advisor
<b>Mô tả</b>	Sinh viên hỏi đáp với AI về lịch sử giao dịch và nhận lời khuyên tài chính.
<b>Tác nhân</b>	Sinh viên, Hệ thống AI (Gemini)
<b>Tiền điều kiện</b>	Sinh viên đã đăng nhập.
<b>Hậu điều kiện</b>	Sinh viên nhận được câu trả lời từ AI.

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Sinh viên mở màn hình Chatbot.</li> <li>2. Nhập câu hỏi (ví dụ: "Tháng này tôi tiêu bao nhiêu tiền?").</li> <li>3. Hệ thống gửi câu hỏi kèm dữ liệu ngữ cảnh (lịch sử giao dịch ẩn danh) đến Gemini API.</li> <li>4. Gemini phân tích và trả về câu trả lời văn bản.</li> <li>5. Ứng dụng hiển thị câu trả lời cho Sinh viên.</li> </ol>
----------------------------	--

### 3.2. Thiết Kế Hệ Thống

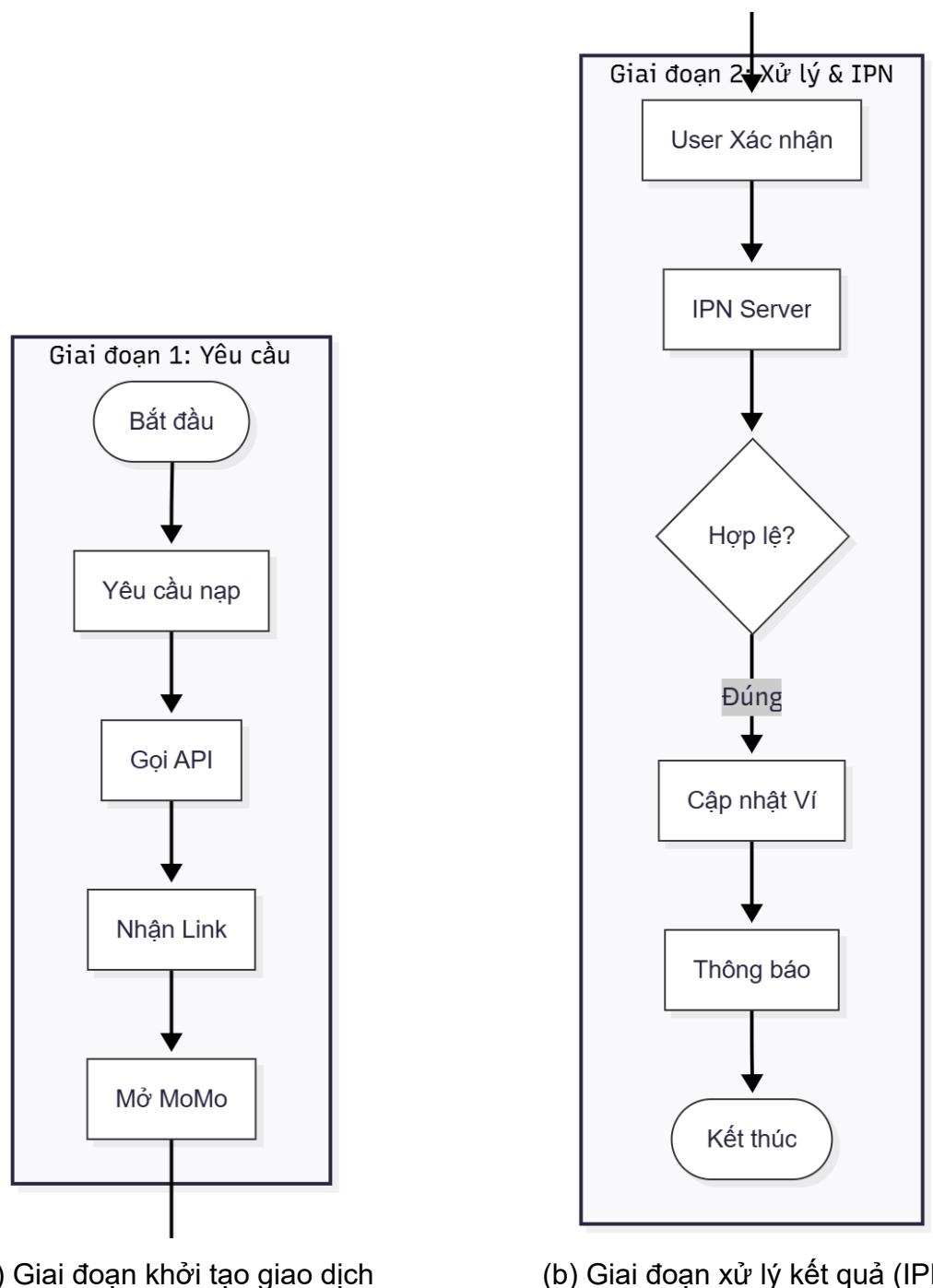
#### 3.2.1. Biểu đồ Activity (Hoạt động)

##### Quy trình Thanh toán NFC



Hình 3. Sơ đồ hoạt động chức năng Thanh toán NFC

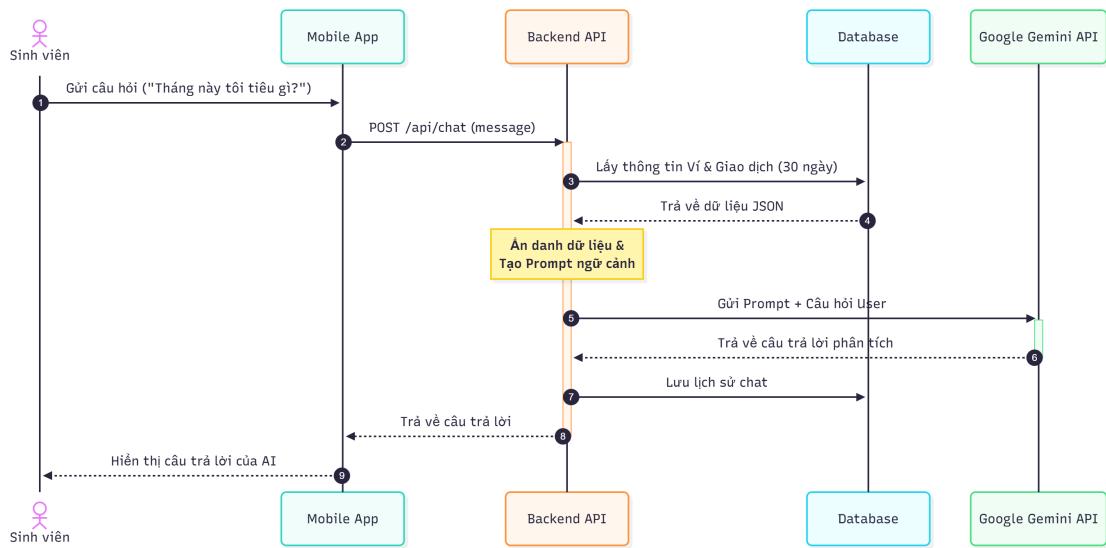
## Quy trình Nạp tiền MoMo



Hình 4. Sơ đồ hoạt động chức năng Nạp tiền MoMo

### 3.2.2. Biểu đồ Sequence (Tuần tự)

#### Luồng tương tác Chatbot AI



Hình 5. Sơ đồ tuần tự chức năng Chat AI

## Chương 4. THIẾT KẾ CƠ SỞ DỮ LIỆU

### 4.1. Thiết kế sơ đồ

Dựa trên các yêu cầu phân tích, hệ thống bao gồm các thực thể chính: Người dùng (Users), Ví (Wallets), Giao dịch (Transactions) và Phiên chat (ChatSessions). Mỗi quan hệ giữa chúng được mô tả như sau:

- Một **User** có duy nhất một **Wallet**.
- Một **Wallet** có thể thực hiện nhiều **Transactions**.
- Một **User** có thể có nhiều **ChatSessions** với AI.

### 4.2. Thiết kế chi tiết Collection (MongoDB)

**1. Users (Người dùng)** Lưu trữ thông tin cá nhân và tài khoản của sinh viên và quản trị viên.

Bảng 4. Cấu trúc Collection Users

Tên trường	Kiểu dữ liệu	Bắt buộc	Mô tả
_id	ObjectId	Có	Khóa chính
studentCode	String	Có	Mã số sinh viên (Unique)
fullName	String	Có	Họ và tên đầy đủ
email	String	Có	Email trường cấp
password	String	Có	Mật khẩu đã mã hóa (bcrypt)
phoneNumber	String	Có	Số điện thoại liên lạc
role	String	Có	Vai trò: "STUDENT" hoặc "ADMIN"
isActive	Boolean	Có	Trạng thái kích hoạt tài khoản
createdAt	Date	Có	Thời gian tạo

**2. Wallets (Ví điện tử)** Lưu trữ thông tin số dư và trạng thái ví của sinh viên.

Bảng 5. Cấu trúc Collection Wallets

Tên trường	Kiểu dữ liệu	Bắt buộc	Mô tả
_id	ObjectId	Có	Khóa chính
userId	ObjectId	Có	Tham chiếu đến Users
balance	Number	Có	Số dư hiện tại (VNĐ)
currency	String	Có	Đơn vị tiền tệ (Default: "VND")
status	String	Có	Trạng thái: "ACTIVE", "LOCKED"
lastTransaction	Date	Không	Thời gian giao dịch gần nhất
updatedAt	Date	Có	Thời gian cập nhật số dư cuối cùng

**3. Transactions (Giao dịch)** Lưu trữ lịch sử nạp tiền và thanh toán.

Bảng 6. Cấu trúc Collection Transactions

Tên trường	Kiểu dữ liệu	Bắt buộc	Mô tả
_id	ObjectId	Có	Khóa chính
walletId	ObjectId	Có	Tham chiếu đến Wallets
type	String	Có	Loại GD: "DEPOSIT_MOMO", "PAYMENT_NFC"
amount	Number	Có	Số tiền giao dịch
status	String	Có	Trạng thái: "PENDING", "SUCCESS", "FAILED"
description	String	Có	Nội dung giao dịch
metadata	Object	Không	Dữ liệu chi tiết (Mã GD MoMo, ID thiết bị POS)
createdAt	Date	Có	Thời gian thực hiện

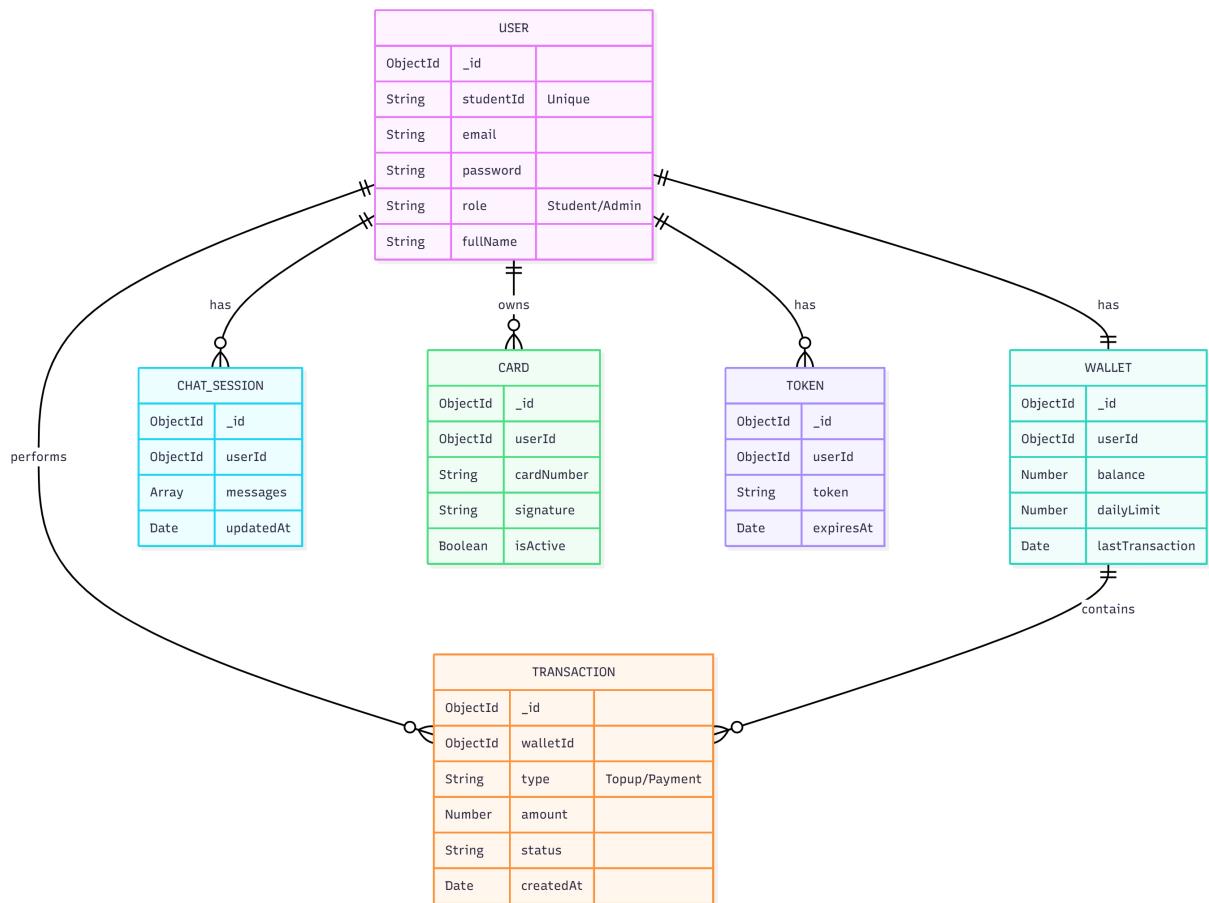
**4. ChatSessions (Hội thoại AI)** Lưu trữ lịch sử trò chuyện giữa sinh viên và trợ lý ảo Gemini để duy trì ngữ cảnh.

Bảng 7. Cấu trúc Collection ChatSessions

Tên trường	Kiểu dữ liệu	Bắt buộc	Mô tả
_id	ObjectId	Có	Khóa chính
userId	ObjectId	Có	Tham chiếu đến Users
messages	Array	Có	Mảng các tin nhắn (role: user/-model, content)
summary	String	Không	Tóm tắt nội dung hội thoại
createdAt	Date	Có	Thời gian bắt đầu
updatedAt	Date	Có	Thời gian tin nhắn cuối cùng

### 4.3. Mô hình quan hệ (Schema Diagram)

Hình dưới đây mô tả mối quan hệ tham chiếu giữa các collection trong cơ sở dữ liệu MongoDB.



Hình 6. Sơ đồ quan hệ cơ sở dữ liệu

## Chương 5. KẾT QUẢ THỰC NGHIỆM

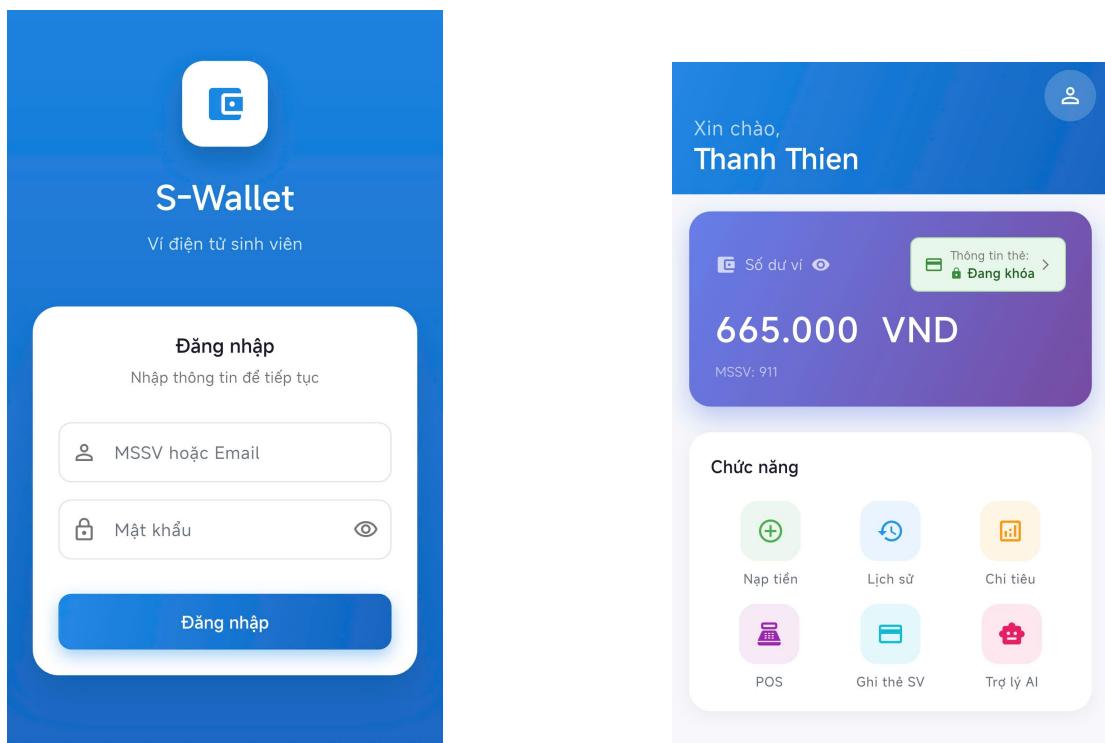
### 5.1. Môi trường và Công cụ thực nghiệm

- **Thiết bị Mobile:** Redmi K70E.
- **Môi trường giả lập:** BlueStacks.
- **Backend Server:** Node.js v18 chạy trên Railway App.
- **Database:** MongoDB Atlas (Cluster Tier M0).
- **Cổng thanh toán:** Môi trường Sandbox của MoMo Developer.

### 5.2. Kết quả triển khai Ứng dụng Sinh viên (Mobile App)

#### 5.2.1. 1. Đăng nhập và Màn hình chính

Giao diện đăng nhập cho phép sinh viên truy cập bằng Mã số sinh viên và mật khẩu. Màn hình chính (Dashboard) hiển thị trực quan số dư hiện tại, các phím tắt chức năng (Nạp tiền, Thanh toán, Lịch sử) và tin tức từ nhà trường.



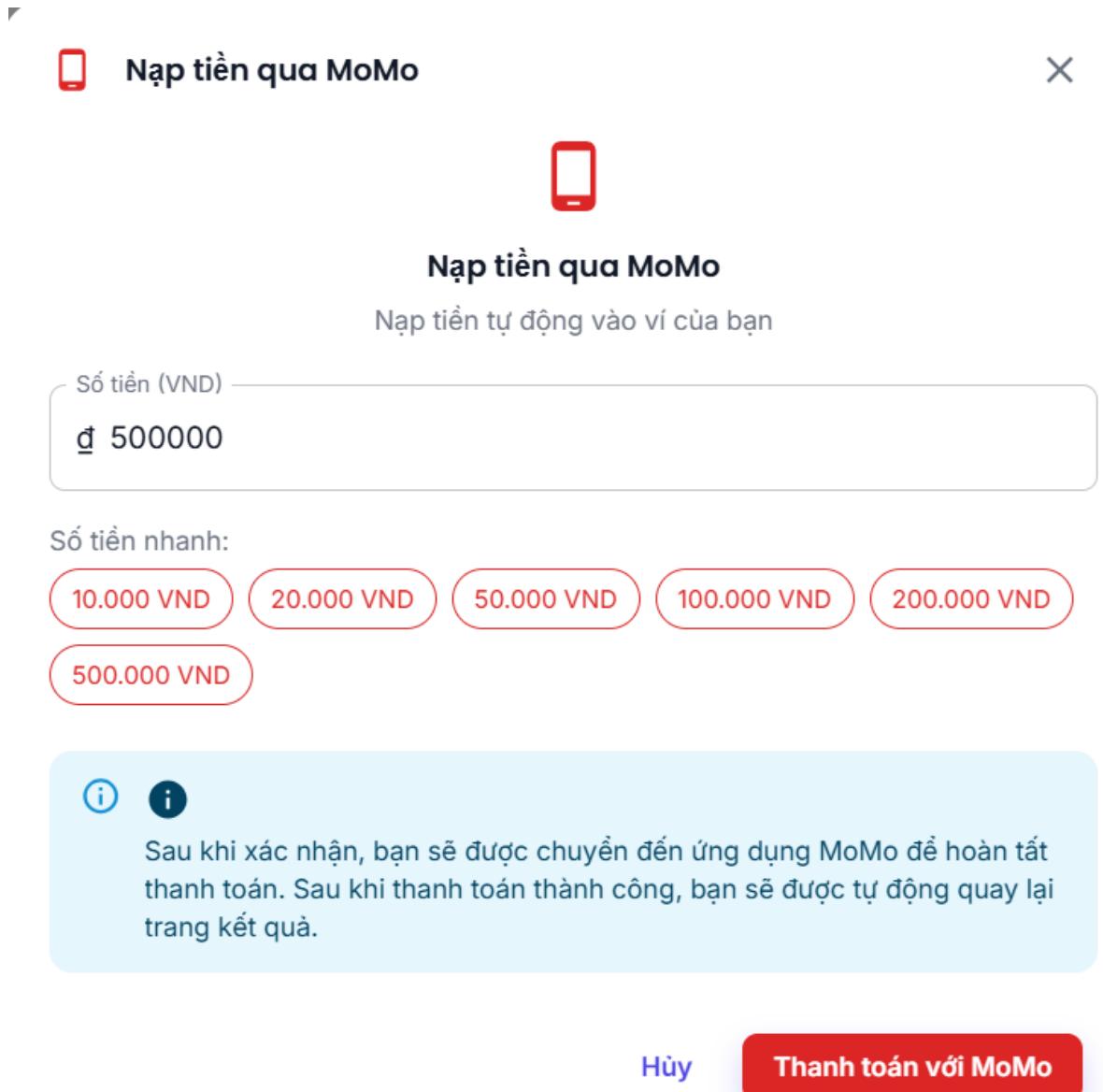
(a) Màn hình Đăng nhập

(b) Màn hình Chính (Dashboard)

Hình 7. Giao diện người dùng cơ bản

### 5.2.2. 2. Chức năng Nạp tiền qua MoMo

Sinh viên nhập số tiền cần nạp, ứng dụng sẽ tự động chuyển hướng sang App MoMo để xác nhận thanh toán. Sau khi thành công, số dư được cập nhật tức thì nhờ cơ chế IPN.



Hình 8. Quy trình nạp tiền qua ví MoMo

### 5.2.3. 3. Chức năng Thanh toán NFC

Sinh viên đưa điện thoại lại gần thiết bị thanh toán (POS giả lập hoặc điện thoại khác đóng vai trò máy POS). Giao dịch được xử lý trong vòng 1-2 giây.



## Xác nhận thanh toán

## Chi tiết đơn hàng

Cơm sườn x1 35.000 VND

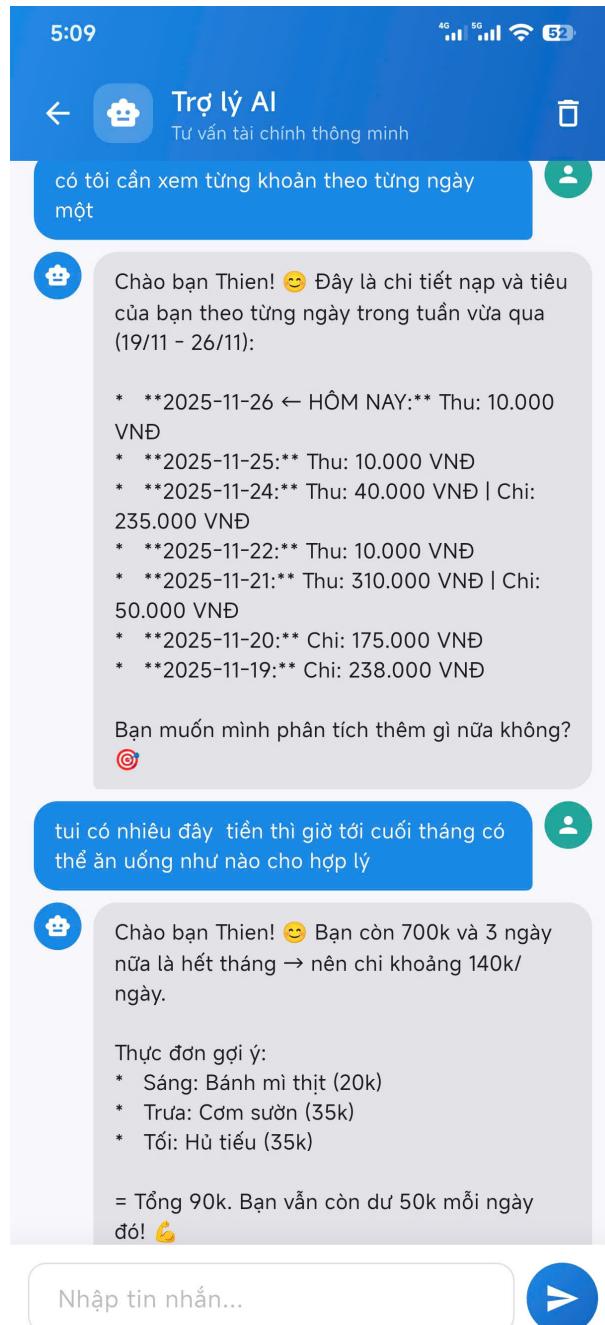
Tổng cộng 35.000 VND

Vui lòng đưa thẻ sinh viên gần điện thoại...

Hình 9. Mô phỏng thanh toán một chạm NFC

#### 5.2.4. 4. Chatbot AI Tư vấn tài chính

Giao diện chat tích hợp Google Gemini. AI có thể trả lời các câu hỏi như "Số dư của tôi còn bao nhiêu?", "Tháng này tôi đã nạp bao nhiêu tiền?" dựa trên dữ liệu giao dịch thực tế.

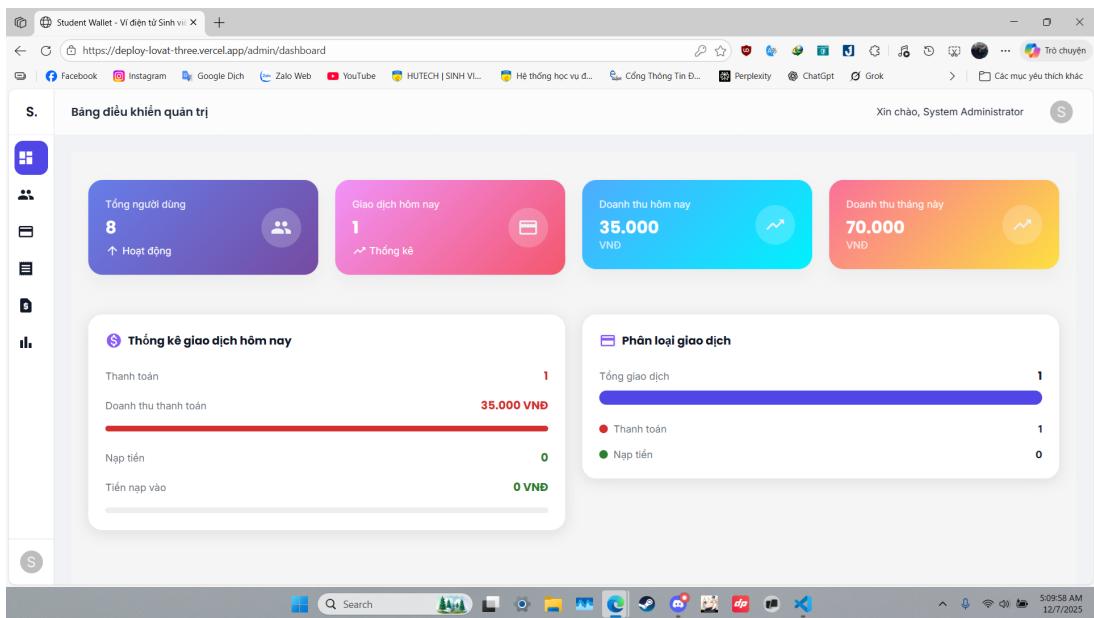


Hình 10. Tương tác với trợ lý ảo AI

## 5.3. Kết quả triển khai Web Quản trị (Admin Portal)

### 5.3.1. 1. Dashboard Thông kê

Trang tổng quan dành cho nhà trường hiển thị biểu đồ dòng tiền vào/ra, tổng số lượng giao dịch trong ngày và số lượng người dùng mới.



Hình 11. Dashboard quản trị viên

### 5.3.2. 2. Quản lý Giao dịch và Sinh viên

Quản trị viên có thể xem danh sách toàn bộ sinh viên, tìm kiếm theo mã số, xem lịch sử giao dịch chi tiết để hỗ trợ giải quyết khiếu nại.

## **Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **6.1. Kết luận**

Sau quá trình nghiên cứu và thực hiện, đồ án "Nền tảng Ví điện tử Sinh viên" đã hoàn thành các mục tiêu đề ra ban đầu, xây dựng thành công một hệ sinh thái thanh toán số khép kín phục vụ môi trường giáo dục đại học.

**Các kết quả đạt được bao gồm:**

#### **1. Về mặt công nghệ:**

- Xây dựng thành công ứng dụng di động đa nền tảng bằng **Flutter**, hoạt động mượt mà trên Android.
- Làm chủ công nghệ **NFC**, cho phép biến điện thoại thành thẻ thanh toán tiện lợi.
- Tích hợp thành công cổng thanh toán **MoMo** và trí tuệ nhân tạo **Google Gemini**, mang lại trải nghiệm hiện đại cho sinh viên.
- Hệ thống Backend **Node.js/MongoDB** hoạt động ổn định, xử lý tốt các tác vụ đồng thời.

#### **2. Về mặt thực tiễn:**

- Giải quyết được bài toán thanh toán không dùng tiền mặt trong trường học (căn tin, gửi xe, đóng phạt thư viện).
- Cung cấp công cụ quản lý tài chính cá nhân hiệu quả cho sinh viên.
- Giúp nhà trường quản lý dòng tiền và dữ liệu sinh viên minh bạch, chính xác hơn.

**Hạn chế tồn tại:**

- Tính năng NFC hiện tại hoạt động tốt nhất trên Android; khả năng hỗ trợ iOS còn hạn chế do chính sách bảo mật của Apple (chỉ hỗ trợ đọc, hạn chế giả lập thẻ).
- Tốc độ phản hồi của Chatbot đôi khi còn phụ thuộc vào độ trễ của API Google Gemini.
- Chưa tích hợp các phương thức xác thực sinh trắc học (Vân tay/FaceID) để tăng cường bảo mật cho giao dịch giá trị lớn.

## 6.2. Hướng phát triển

Để đưa sản phẩm vào ứng dụng thực tế quy mô lớn, nhóm đề xuất các hướng phát triển trong tương lai:

### 6.2.1. 1. Mở rộng tính năng và Nền tảng

- **Hỗ trợ iOS toàn diện:** Nghiên cứu các giải pháp thay thế như QR Code động hoặc Bluetooth Low Energy (BLE) để phục vụ người dùng iPhone nếu rào cản NFC chưa được gỡ bỏ.
- **Bảo mật sinh trắc học:** Tích hợp xác thực vân tay và nhận diện khuôn mặt khi thực hiện thanh toán để thay thế mã PIN truyền thống.

### 6.2.2. 2. Hệ sinh thái tài chính mở rộng

- **Thanh toán đa dịch vụ:** Mở rộng liên kết thanh toán học phí, bảo hiểm y tế và các dịch vụ xung quanh trường (nhà trọ, tiệm in ấn).
- **Gamification:** Tích hợp tính năng tích điểm đổi quà, xếp hạng chi tiêu hợp lý để khuyến khích sinh viên sử dụng.

### 6.2.3. 3. Ứng dụng Blockchain

- Nghiên cứu ứng dụng Blockchain (Private Chain) để lưu trữ log giao dịch, đảm bảo tính minh bạch tuyệt đối và không thể sửa đổi, giúp việc đống soát tài chính giữa nhà trường và sinh viên trở nên tin cậy hơn.

Đồ án này là bước khởi đầu quan trọng, minh chứng cho khả năng ứng dụng công nghệ mới vào giải quyết các vấn đề thực tiễn trong môi trường giáo dục số. Nhóm thực hiện hy vọng sản phẩm sẽ tiếp tục được hoàn thiện và có cơ hội triển khai thực tế trong tương lai gần.

# TÀI LIỆU THAM KHẢO

## Tài liệu

- [1] Google Developers, "Flutter Documentation - Build apps for any screen." [Online].  
Available: <https://flutter.dev/docs>
- [2] MoMo for Business, "MoMo Payment API Integration." [Online].  
Available: <https://developers.momo.vn/>
- [3] Google AI for Developers, "Gemini API Overview." [Online].  
Available: <https://ai.google.dev/docs>
- [4] OpenJS Foundation, "Node.js Documentation." [Online].  
Available: <https://nodejs.org/en/docs/>
- [5] MongoDB, Inc., "MongoDB Manual." [Online].  
Available: <https://www.mongodb.com/docs/manual/>
- [6] NFC Forum, "Near Field Communication Technology Standards." [Online].  
Available: <https://nfc-forum.org/>
- [7] Auth0, "Introduction to JSON Web Tokens." [Online].  
Available: <https://jwt.io/introduction>