|  |
| --- |
| ĐẠI HỌC THÁI NGUYÊN  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG** |
|  |
| **HÀ MINH ĐỨC** |
|  |
| **XÂY DỰNG ỨNG DỤNG ĐẶT ĐỒ UỐNG CHO CỬA HÀNG MỠ COFFEE & TEA TẠI TỈNH BẮC KẠN** |
|  |
| **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC** |
| **NGÀNH CÔNG NGHỆ THÔNG TIN** |
|  |

**THÁI NGUYÊN, NĂM 2023**

|  |  |  |  |
| --- | --- | --- | --- |
| ĐẠI HỌC THÁI NGUYÊN  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG** | | | |
|  | | | |
| D:\DATA\Picture\1, LOGO\TNU-CNTT_2.png |  | |  |
|  | | | |
| **ĐỒ ÁN**  **TỐT NGHIỆP ĐẠI HỌC** | | | |
| **NGÀNH CÔNG NGHỆ THÔNG TIN** | | | |
|  | | | |
| ***Đề tài:*** | | | |
| **XÂY DỰNG ỨNG DỤNG ĐẶT ĐỒ UỐNG CHO CỬA HÀNG MỠ COFFEE & TEA TẠI TỈNH BẮC KẠN** | | | |
|  | | | |
| **Sinh viên thực hiện:**  **Lớp:**  **Giáo viên hướng dẫn:** | | **Hà Minh Đức**  **CNTT K18K**  **TS. Ngô Hữu Huy** | |
|  | | | |

**THÁI NGUYÊN, NĂM 2023**

**LỜI CAM ĐOAN**

Tôi xin cam đoan rằng đồ án này đã được thực hiện bởi tôi và được hướng dẫn bởi Tiến sĩ Ngô Hữu Huy. Các phần lý thuyết được trình bày dựa trên các tài liệu đã được công bố và được trích dẫn đầy đủ. Những kết quả từ các chương trình thực nghiệm trong đồ án này được thực hiện một cách trung thực và tôi chịu trách nhiệm hoàn toàn nếu có bất kỳ sai sót nào.

*Thái Nguyên, ngày …… tháng …… năm 2023*

|  |  |
| --- | --- |
|  | **Sinh viên** |
|  |  |
|  | **Hà Minh Đức** |

**LỜI CẢM ƠN**

Lời đầu tiên, Em muốn bày tỏ lòng biết ơn sâu sắc đến TS. Ngô Hữu Huy vì sự tận tình hướng dẫn, chỉ bảo và giúp đỡ em suốt quá trình thực hiện đồ án này.

Em cũng xin gửi lời tri ân đến các thầy cô giáo tại Trường Đại học Công nghệ Thông tin và Truyền thông - Đại học Thái Nguyên, đã chia sẻ kiến thức và hỗ trợ em trong suốt quá trình học tập.

Em xin gửi lời cảm ơn đến bạn bè, đồng nghiệp và gia đình đã luôn động viên, khích lệ và tạo điều kiện giúp đỡ em trong quá trình học tập, thực hiện và hoàn thành đồ án này.

Xin chân thành cảm ơn!

*Thái Nguyên, ngày …… tháng …… năm 2023*

|  |  |
| --- | --- |
|  | **Sinh viên** |
|  |  |
|  | **Hà Minh Đức** |

MỤC LỤC

[LỜI CAM ĐOAN i](#_Toc164435895)

[LỜI CẢM ƠN ii](#_Toc164435896)

[MỤC LỤC 1](#_Toc164435897)

[DANH SÁCH CÁC BẢNG 3](#_Toc164435898)

[DANH SÁCH CÁC HÌNH 4](#_Toc164435899)

[MỞ ĐẦU 6](#_Toc164435900)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 7](#_Toc164435901)

[1.1. React Native 7](#_Toc164435902)

[1.1.1. Giới thiệu về React Native 7](#_Toc164435903)

[1.1.2. Nguyên lý hoạt động 7](#_Toc164435904)

[1.1.3. Ưu điểm 8](#_Toc164435905)

[1.1.4. Nhược điểm 8](#_Toc164435906)

[1.2. React Redux 9](#_Toc164435907)

[1.2.1. Giới thiệu về React Redux 9](#_Toc164435908)

[1.2.2. Nguyên lý hoạt động 9](#_Toc164435909)

[1.2.3. Kết hợp Redux với React 11](#_Toc164435910)

[1.3. Firebase 12](#_Toc164435911)

[1.3.1. Giới thiệu Firebase 12](#_Toc164435912)

[1.3.2. Cách thức hoạt động 12](#_Toc164435913)

[1.3.3. Ưu điểm 14](#_Toc164435914)

[1.3.4. Nhược điểm 15](#_Toc164435915)

[1.4. Về cửa hàng 15](#_Toc164435916)

[1.4.1. Đặc điểm nổi bật của cửa hàng 15](#_Toc164435917)

[1.4.2. Phương thức hoạt động 16](#_Toc164435918)

[1.4.3. Dịch vụ cung cấp 17](#_Toc164435919)

[CHƯƠNG 2. KHẢO SÁT, PHÂN TÍCH & THIẾT KẾ HỆ THỐNG 18](#_Toc164435920)

[2.1. Khảo sát yêu cầu 18](#_Toc164435921)

[2.1.1. Xác định nhu cầu của người dùng 18](#_Toc164435922)

[2.1.2. Thu thập yêu cầu chức năng và phi chức năng 19](#_Toc164435923)

[2.2. Khảo sát hệ thống 20](#_Toc164435924)

[2.2.2. Mô hình hệ thống 20](#_Toc164435925)

[2.2.3. Yêu cầu phần cứng 20](#_Toc164435926)

[2.3. Phân tích các chức năng chính của ứng dụng 21](#_Toc164435927)

[2.4. Xác định tác nhân của hệ thống và User Case 22](#_Toc164435928)

[2.4.1. Liệt kê các tác nhân của Hệ thống (Actor) 22](#_Toc164435929)

[2.4.2. Liệt kê các User Case 22](#_Toc164435930)

[2.4.3. Đặc tả User Case 23](#_Toc164435931)

[2.4.4. Biểu đồ User Case 27](#_Toc164435932)

[2.4.5. Biểu đồ hoạt động 34](#_Toc164435933)

[2.4.6. Biểu đồ tuần tự 40](#_Toc164435934)

[2.4.7. Cơ sở dữ liệu 41](#_Toc164435935)

[CHƯƠNG 3. KẾT QUẢ THỰC HIỆN 49](#_Toc164435936)

[3.1. Mô tả bài toán 49](#_Toc164435937)

[3.2. Về kiến thức 50](#_Toc164435938)

[3.3. Về chương trình 51](#_Toc164435939)

[KẾT LUẬN 55](#_Toc164435940)

[TÀI LIỆU THAM KHẢO 57](#_Toc164435941)

DANH SÁCH CÁC BẢNG

[Bảng 2.1. Minh họa Danh mục sản phẩm (Tên bảng: Categories) 43](#_Toc164875503)

[Bảng 2.2: Minh họa Menu sản phẩm (Tên bảng: MenuMoC&T) 45](#_Toc164875504)

[Bảng 2.3: Minh họa Thông tin người dùng (Tên bảng: Users) 47](#_Toc164875505)

[Bảng 2.4: Minh họa Danh sách đơn đặt hàng (Tên bảng: OrdersConfirmation) 48](#_Toc164875506)

DANH SÁCH CÁC HÌNH

[Hình 1: Cách hoạt động của Redux 11](#_Toc164875685)

[Hình 2: Cửa hàng Mỡ Coffee & Tea 16](#_Toc164875686)

[Hình 3: Mô hình hệ thống 20](#_Toc164875687)

[Hình 4: UC Tổng quát 28](#_Toc164875688)

[Hình 5: UC Xem thông tin sản phẩm 29](#_Toc164875689)

[Hình 6: UC Quản lý thông tin sản phẩm 29](#_Toc164875690)

[Hình 7: UC Đăng nhập, đăng ký 30](#_Toc164875691)

[Hình 8: UC Quản lý giỏ hàng 30](#_Toc164875692)

[Hình 9: UC Thanh toán 31](#_Toc164875693)

[Hình 10: UC Quản lý danh mục sản phẩm 31](#_Toc164875694)

[Hình 11: UC Xét duyệt hóa đơn 32](#_Toc164875695)

[Hình 12: UC Quản lý doanh thu 32](#_Toc164875696)

[Hình 13: UC Quản lý nhân viên 33](#_Toc164875697)

[Hình 14: UC Quản lý khách hàng 33](#_Toc164875698)

[Hình 15: Biểu đồ thêm sản phẩm vào giỏ hàng 34](#_Toc164875699)

[Hình 16: Biểu đồ hoạt động đăng nhập 35](#_Toc164875700)

[Hình 17: Biểu đồ hoạt động đăng ký 35](#_Toc164875701)

[Hình 18: Biểu đồ hoạt động xoá sản phẩm khỏi giỏ hàng 36](#_Toc164875702)

[Hình 19: Biểu đồ hiển thị thông tin sản phẩm 36](#_Toc164875703)

[Hình 20: Biểu đồ thanh toán 37](#_Toc164875704)

[Hình 21: Biểu đồ hoạt động quản lý thông tin cá nhân 38](#_Toc164875705)

[Hình 22: Biểu đồ hoạt động quản lý khách hàng 38](#_Toc164875706)

[Hình 23: Biểu đồ hoạt động quản lý danh mục sản phẩm 39](#_Toc164875707)

[Hình 24: Biểu đồ hoạt động quản lý nhân viên 39](#_Toc164875708)

[Hình 25: Biểu đồ tuần tự UC Sửa thông tin cá nhân 40](#_Toc164875709)

[Hình 26: Biểu đồ tuần tự UC Quản lý danh sách thành viên 40](#_Toc164875710)

[Hình 27: Biểu đồ tuần tự UC Mua hàng (Cập nhật giỏ hàng, thanh toán) 41](#_Toc164875711)

[Hình 28: Danh mục sản phẩm (Tên bảng: Categories) 41](#_Toc164875712)

[Hình 29: Menu sản phẩm (Tên bảng: MenuMoC&T) 42](#_Toc164875713)

[Hình 30: Thông tin người dùng (Tên bảng: Users) 42](#_Toc164875714)

[Hình 31: Danh sách đơn đặt hàng (Tên bảng: OrdersConfirmation) 43](#_Toc164875715)

[Hình 32: Giao diện đăng nhập, đăng ký 51](#_Toc164875716)

[Hình 33: Giao diện trang chủ 52](#_Toc164875717)

[Hình 34: Giao diện sản phẩm và chi tiết sản phẩm 52](#_Toc164875718)

[Hình 35: Giao diện tài khoản và chỉnh sửa hồ sơ 53](#_Toc164875719)

[Hình 36: Giao diện quản trị viên, danh sách sản phẩm 53](#_Toc164875720)

[Hình 37: Giao diện giỏ hàng, đơn hàng và đơn đặt hàng 54](#_Toc164875721)

[Hình 38: Kết quả kiểm tra đạo văn 58](#_Toc164875722)

MỞ ĐẦU

Ngày nay, việc phát triển công nghệ thông tin đồng nghĩa với sự phát triển của các ứng dụng phần mềm. Trong đó, việc ***“Xây dựng ứng dụng đặt đồ uống cho cửa hàng Mỡ Coffee & Tea tại tỉnh Bắc Kạn”*** là một nhu cầu cấp thiết. Ứng dụng này sẽ giúp khách hàng dễ dàng đặt đồ uống từ xa thông qua điện thoại di động.

Các ứng dụng đặt đồ uống đã trở nên phổ biến trong các cửa hàng cà phê và trà trên khắp thế giới, giúp tối ưu hóa quá trình đặt hàng và tạo ra trải nghiệm thuận lợi cho khách hàng. Tại tỉnh Bắc Kạn, việc áp dụng công nghệ này cũng sẽ mang lại nhiều lợi ích cho cả cửa hàng và khách hàng.

Mặc dù đã cố gắng hết sức nhưng vì thời gian có hạn nên đề tài không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự quan tâm và những góp ý cần thiết từ phía thầy cô và các bạn để cải thiện và hoàn thiện hơn đề tài này.

Để hoàn thiện đề tài này, em muốn bày tỏ lòng biết ơn sâu sắc đến giáo viên hướng dẫn đã dành thời gian và công sức để hướng dẫn, chỉ bảo và chia sẻ kiến thức thực tế rất hữu ích cho tôi trong suốt làm đề tài. Nhờ sự hỗ trợ tận tình của thầy, tôi mới có thể hoàn thành được đề tài này.

Em xin chân thành cảm ơn!

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. React Native

1.1.1. Giới thiệu về React Native

React Native là một framework mã nguồn mở được sáng tạo bởi Facebook, React Native được sử dụng để phát triển ứng dụng di động Android, iOS, Web và UWP bằng cách cho phép các nhà phát triển sử dụng React cùng với môi trường ứng dụng gốc (native). Framework này còn cho phép các nhà phát triển tận dụng các tính năng cục bộ trên các thiết bị di động như camera, định vị GPS, cảm biến và nhiều hơn nữa, tạo ra trải nghiệm người dùng tương tác và phong phú.

* Native App là thuật ngữ chỉ những ứng dụng được xây dựng và phát triển bằng các công cụ do nhà phát triển nền tảng cung cấp cho lập trình viên. Trong số đó, Android và iOS là hai hệ điều hành phổ biến nhất cho việc phát triển ứng dụng di động.
* Hybrid App là sự kết hợp giữa ứng dụng Web và ứng dụng di động. Hybrid App cho phép người dùng cài đặt ứng dụng lên điện thoại giống như các ứng dụng Native thông thường, nhưng vẫn có thể truy cập vào những tính năng và dịch vụ trên mạng. Điều này làm cho Hybrid App trở nên phổ biến và thuận tiện cho người dùng. Hybrid App thường có thể tìm thấy ở những kho ứng dụng trả phí

1.1.2. Nguyên lý hoạt động

React Native được viết bằng sự kết hợp của JavaScript và JSX, một ngôn ngữ đánh dấu đặc biệt giống với XML. Framework này có khả năng thao tác với cả hai luồng là main thread và JS thread. Mỗi luồng đều có vai trò riêng biệt:

* Main thread: Đảm nhiệm vai trò cập nhật giao diện người dùng và xử lý tương tác người dùng.
* JS thread: Đảm bảo hệ thống hoạt động hiệu quả thông qua việc thực thi và xử lý mã JavaScript.

Nguyên lý hoạt động của React Native gần như tương tự với React, tuy nhiên, React Native không sử dụng thao tác với DOM và HTML mà thực thi một quá trình xử lý nền với nền tảng gốc. Để kết nối giữa JavaScript và native, React Native sử dụng một tính năng gọi là Bridge (cầu nối). Mặc dù hai thread JavaScript và native được viết bằng các ngôn ngữ hoàn toàn khác nhau, nhưng cầu nối giúp thao tác hai chiều có thể thực hiện dễ dàng hơn.

1.1.3. Ưu điểm

***Khả năng tái sử dụng code:*** React Native cho phép tái sử dụng code giữa các nền tảng, giúp tiết kiệm thời gian và công sức trong việc phát triển ứng dụng cho cả iOS và Android.

***Cộng đồng hỗ trợ lớn mạnh:*** React Native sở hữu một cộng đồng hỗ trợ cực kỳ lớn, giúp người dùng dễ dàng tìm kiếm giải pháp cho các vấn đề phát sinh trong quá trình phát triển ứng dụng.

***Tính ổn định và khả năng tối ưu cao:*** React Native được xây dựng trên cơ sở của React, là một framework phổ biến và đã được thử nghiệm qua nhiều dự án lớn, do đó có tính ổn định cao và khả năng tối ưu hóa tốt.

***Thời gian học không quá dài:*** Với người đã có kiến thức về JavaScript và React, việc học React Native không quá phức tạp và đòi hỏi thời gian học không quá dài, giúp người dùng nhanh chóng bắt đầu phát triển ứng dụng di động.

1.1.4. Nhược điểm

Thiếu một số component quan trọng và không phù hợp cho việc phát triển các trò chơi có đồ họa phức tạp.

Hiệu suất thấp hơn so với native app, đặc biệt là trong các ứng dụng có khả năng tính toán cao.

Sự bảo mật còn chưa được đảm bảo do việc sử dụng JavaScript.

Một số module có sẵn trong React Native không cho phép tùy biến nhiều.

1.2. React Redux

1.2.1. Giới thiệu về React Redux

React Redux là một thư viện JavaScript được sử dụng để quản lý trạng thái của ứng dụng. Nó được xây dựng trên cơ sở của kiến trúc Flux, một kiến trúc được giới thiệu bởi Facebook. React Redux là một sự kết hợp hoàn hảo cho cả React JS và React Native.

Redux giúp tạo ra một lớp quản lý trạng thái của ứng dụng, giúp cho việc quản lý và cập nhật trạng thái trở nên dễ dàng hơn. Bằng cách sử dụng Redux, các thành phần của ứng dụng có thể truy cập vào trạng thái một cách đồng nhất và linh hoạt. Điều này giúp tạo ra một ứng dụng có cấu trúc dễ bảo trì và mở rộng.

Với việc kết hợp Redux và React, việc quản lý trạng thái của các thành phần React trở nên dễ dàng hơn bao giờ hết. React Redux cung cấp các công cụ và cơ chế giúp kết nối các thành phần React với trạng thái được quản lý bởi Redux một cách thuận tiện và hiệu quả. Điều này giúp tạo ra các ứng dụng React có trải nghiệm người dùng tốt hơn và mã nguồn dễ bảo trì hơn.

1.2.2. Nguyên lý hoạt động

Redux dựa trên 3 thành phần chính là: Actions, Store, Reducers.

***Action:*** Là các đối tượng mô tả những gì xảy ra trong ứng dụng. Mỗi action có một thuộc tính `type` để xác định loại action và có thể chứa các dữ liệu khác để truyền thông tin cần thiết cho việc thay đổi trạng thái.

***- Store:*** Là nơi chứa toàn bộ trạng thái của ứng dụng. Store được tạo ra bằng cách kết hợp giữa reducers và initialState. Nó cung cấp các phương thức để truy cập trạng thái, gửi action và đăng ký các hàm lắng nghe (listener) để theo dõi sự thay đổi của trạng thái. (Giống như một ngăn kéo lớn chứa toàn bộ trạng thái của ứng dụng & Lưu trữ thông tin về cách mà dữ liệu của ứng dụng thay đổi theo thời gian.)

***- Reducer:*** Là các hàm thuần túy (pure functions) nhận vào hai tham số là trạng thái hiện tại và action, và trả về trạng thái mới. Reducer xác định cách thay đổi trạng thái của ứng dụng dựa trên action được tạo ra. (Nhận các hành động `actions` và trạng thái hiện tại, sau đó trả về một trạng thái mới dựa trên hành động đó)

Ngoài ra còn 1 thành phần là Middleware

***- Middleware:*** Là các lớp trung gian giữa việc gửi action và khi action đến reducers. Middleware cho phép thực hiện các chức năng như ghi log, xử lý side effects, hoặc thậm chí là thay đổi action trước khi nó đến reducers.

*\* Nguyên lý hoạt động Redux*

***Nguyên lý thứ 1: Quản lý trạng thái***

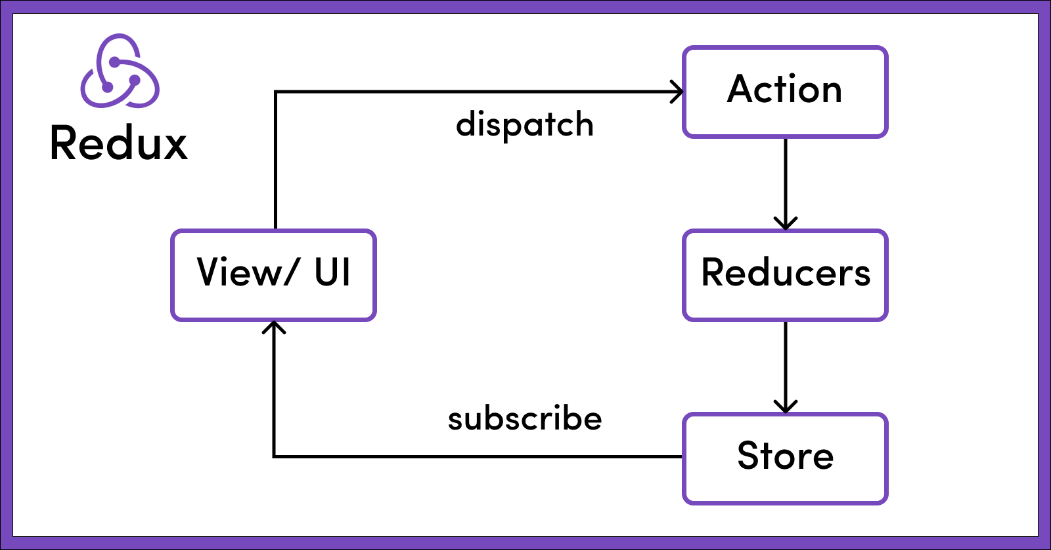
Trong Redux, mọi thứ thay đổi trong ứng dụng, bao gồm cả dữ liệu và trạng thái giao diện (UI state), đều được lưu trữ trong một đối tượng được gọi là state hoặc state tree. Ứng dụng phụ thuộc vào nhiều nguồn dữ liệu khác nhau như dữ liệu từ máy chủ, tương tác người dùng và dữ liệu tính toán. Redux giải quyết vấn đề phức tạp này bằng cách tạo ra một nguồn dữ liệu duy nhất, tin cậy và dễ kiểm soát.

***Nguyên lý thứ 2: Hành động là cách duy nhất thay đổi trạng thái***

Trạng thái của ứng dụng không thể thay đổi trực tiếp. Chỉ có thể thay đổi thông qua việc phát ra một hành động (action) - một đối tượng mô tả những gì xảy ra trong ứng dụng. Redux yêu cầu mọi thay đổi trạng thái phải thông qua một hành động được phát ra, điều này giúp duy trì tính nhất quán và dễ theo dõi của trạng thái ứng dụng.

***Nguyên lý thứ 3: Sử dụng reducers để thay đổi trạng thái***

Redux sử dụng các hàm thuần túy (pure functions) gọi là reducers để thay đổi trạng thái của ứng dụng. Mỗi reducer nhận vào hai tham số là trạng thái hiện tại và hành động, và trả về trạng thái mới của ứng dụng. Các hàm reducer này quản lý các thay đổi trạng thái một cách nhỏ gọn và có thể kết hợp với nhau để tạo ra chuỗi sự kiện. Điều này giúp Redux duy trì tính nhất quán và dễ bảo trì của ứng dụng.



Hình 1: Cách hoạt động của Redux

1.2.3. Kết hợp Redux với React

Redux không chỉ làm việc tốt với React, mà còn là một trong những lựa chọn phổ biến để quản lý trạng thái ứng dụng khi sử dụng React. Dưới đây là một số cách mà Redux có thể được kết hợp với React để tạo ra ứng dụng hiệu quả:

***- Quản lý UI state với Redux:*** Redux lưu trữ tất cả các trạng thái của ứng dụng, bao gồm cả dữ liệu và trạng thái giao diện. Mỗi khi có sự thay đổi, Redux cập nhật trạng thái thông qua các action tương ứng.

***- Xử lý các action không đồng bộ:*** Redux hỗ trợ xử lý các action không đồng bộ, chẳng hạn như gọi API từ máy chủ. Bằng cách sử dụng các action đặc biệt, Redux thông báo với reducer khi request bắt đầu, kết thúc thành công hoặc thất bại.

***- Sử dụng middleware:*** Middleware như redux-thunk và redux-promise giúp xử lý các action không đồng bộ. Chúng cho phép gửi đi nhiều hơn một action, có thể là hàm hoặc promise, và có thể được sử dụng để ghi log, báo lỗi, routing và nhiều công việc khác.

***- Kết hợp với react-router:*** Redux và react-router có thể được kết hợp để quản lý routing trong ứng dụng. Redux quản lý trạng thái của ứng dụng, trong khi react-router quản lý URL. Sự kết hợp này giúp đồng bộ hóa trạng thái và URL của ứng dụng.

***Ứng dụng các ứng dụng quan trọng của Redux:*** Redux cung cấp ba ứng dụng quan trọng: quản lý trạng thái, tăng tốc phát triển và hỗ trợ ứng dụng offline. Với Redux, bạn có thể quản lý trạng thái của ứng dụng như một bản ghi lịch sử có thể undo/redo, tăng tốc phát triển với Hot Module Replacement và hỗ trợ ứng dụng offline thông qua việc lưu trữ tất cả các thao tác của người dùng vào một cây trạng thái.

1.3. Firebase

1.3.1. Giới thiệu Firebase

Firebase là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây của Google, cung cấp cho người dùng một hệ thống máy chủ mạnh mẽ. Chức năng chính của Firebase là giúp người dùng lập trình ứng dụng một cách đơn giản thông qua việc đơn giản hóa các thao tác với cơ sở dữ liệu. Firebase cung cấp các giao diện lập trình ứng dụng API đơn giản để tạo ra ứng dụng một cách nhanh chóng và dễ dàng.

Mục đích chính của Firebase là tăng số lượng người dùng và tạo ra nhiều lợi nhuận hơn cho người sử dụng. Đặc biệt, Firebase cung cấp các tính năng đa năng và bảo mật cực kỳ tốt, đảm bảo an toàn cho dữ liệu của người dùng. Firebase hỗ trợ cả hai nền tảng Android và IOS, làm cho nó trở thành lựa chọn phổ biến đối với lập trình viên muốn xây dựng ứng dụng cho hàng triệu người dùng trên toàn thế giới.

1.3.2. Cách thức hoạt động

Firebase hoạt động dựa trên các máy chủ được quản lý bởi Google Cloud Platform. Khi một ứng dụng được phát triển bằng Firebase, dữ liệu của ứng dụng sẽ được lưu trữ trên các cơ sở dữ liệu của Firebase, bao gồm Firebase Realtime Database và Firestore.

Khi người dùng sử dụng ứng dụng, Firebase sẽ xử lý yêu cầu từ người dùng và truy vấn cơ sở dữ liệu để trả về kết quả. Đồng thời, Firebase cũng hỗ trợ các tính năng như phân tích, báo cáo lỗi và truyền thông xã hội.

Cụ thể, Firebase cung cấp các dịch vụ như Firebase Realtime Database, Cloud Firestore, Authentication, Hosting, giúp cho việc phát triển ứng dụng trở nên dễ dàng và hiệu quả hơn.

***- Firebase Realtime Database:*** là một cơ sở dữ liệu thời gian thực, lưu trữ dữ liệu dưới dạng JSON và tự động đồng bộ hóa dữ liệu giữa các thiết bị người dùng trong thời gian thực. Điều này giúp xây dựng các ứng dụng có khả năng phản hồi nhanh chóng và đồng bộ hoá dữ liệu hiệu quả. Dữ liệu được truyền tải qua kết nối SSL có chứng nhận 2048 bit, đảm bảo tính bảo mật và an toàn. Trong các ứng dụng đa nền tảng, một database được chia sẻ cho tất cả các máy khách. Nếu kết nối Internet bị gián đoạn, dữ liệu sẽ được lưu trữ trên local và tự động cập nhật khi đường truyền ổn định trở lại.

***- Firebase Cloud Firestore:*** là một cơ sở dữ liệu linh hoạt, mạnh mẽ, dựa trên tài liệu, được thiết kế để phát triển ứng dụng di động, web và IoT. Firestore cho phép lưu trữ và truy vấn dữ liệu theo thời gian thực, hỗ trợ tối ưu hóa hiệu suất và mở rộng.

Đặc biệt, Cloud Firestore là một cloud-hosted, NoSQL database mà các ứng dụng phía client có thể trực tiếp truy cập thông qua native SDKs. Nó lưu dữ liệu theo mô hình dữ liệu NoSQL, trong đó dữ liệu được tổ chức trong các tài liệu và tập hợp. Điều này giúp tổ chức và truy vấn dữ liệu một cách hiệu quả. Cloud Firestore hỗ trợ rất nhiều kiểu dữ liệu từ đơn giản như String, Integer đến những kiểu dữ liệu phức tạp như các nested object.

Tương tự như Firebase Realtime Database, Cloud Firestore cũng cho phép đồng bộ dữ liệu giữa các ứng dụng phía client một cách nhanh chóng (Realtime) và hỗ trợ lưu trữ offline data trong ứng dụng của bạn.

***- Firebase Authentication:*** là dịch vụ xác thực người dùng, cung cấp các phương thức đăng nhập an toàn và dễ sử dụng như đăng nhập bằng email/password, đăng nhập bằng Google, Facebook, Twitter, GitHub, và nhiều hơn nữa. Firebase Authentication giúp bảo vệ thông tin người dùng và quản lý quyền truy cập vào ứng dụng.

***- Firebase Hosting:*** là dịch vụ cung cấp máy chủ web tĩnh và nội dung động cho các ứng dụng web và trang web. Với Firebase Hosting, bạn có thể dễ dàng triển khai và quản lý ứng dụng của mình một cách hiệu quả. Dịch vụ này cung cấp các công cụ tối ưu hóa và giảm thiểu thời gian phản hồi, giúp tiết kiệm thời gian thiết kế và phát triển.

Firebase Hosting sử dụng SSL từ mạng CDN, giúp đảm bảo tính bảo mật và tốc độ truy cập cho các ứng dụng của bạn. Điều này giúp người dùng truy cập trang web của bạn một cách nhanh chóng và an toàn.

1.3.3. Ưu điểm

* Dễ dàng tạo tài khoản và sử dụng.
* Tốc độ phát triển nhanh nhờ vào các công cụ và dịch vụ tiện ích.
* Đa dịch vụ trong một nền tảng, giúp tiết kiệm thời gian và công sức.
* Có tính năng theo dõi lỗi, giúp dễ dàng phát hiện và sửa chữa lỗi trong ứng dụng.
* Được cung cấp bởi Google, đảm bảo sự ổn định và tin cậy.
* Tập trung vào phát triển giao diện người dùng, giúp tăng trải nghiệm người dùng.
* Firebase không có máy chủ riêng, giúp tiết kiệm chi phí về hạ tầng.
* Hỗ trợ tích hợp các tính năng học máy (Machine Learning), giúp tối ưu hóa trải nghiệm người dùng và phân tích dữ liệu.
* Tạo lưu lượng truy cập một cách linh hoạt và hiệu quả.
* Cung cấp tính năng sao lưu dữ liệu, đảm bảo an toàn và bảo vệ thông tin của người dùng.

1.3.4. Nhược điểm

* Không phải là mã nguồn mở, do đó người dùng không có quyền truy cập mã nguồn.
* Firebase không hoạt động ở nhiều quốc gia, điều này có thể gây hạn chế cho các ứng dụng có người dùng toàn cầu.
* Chỉ hoạt động với Cơ sở dữ liệu NoSQL, điều này có thể là một hạn chế đối với các ứng dụng cần sử dụng SQL.
* Truy vấn có thể chậm đối với các cơ sở dữ liệu lớn hoặc có nhiều kết nối.
* Không phải tất cả các dịch vụ Firebase đều miễn phí, điều này có thể tạo ra chi phí không đáng kể cho các dự án lớn.
* Firebase có thể đắt và giá không ổn định, đặc biệt là đối với các dự án phát triển nhanh.
* Chỉ chạy trên Google Cloud, điều này có thể hạn chế sự linh hoạt của ứng dụng.
* Thiếu Dedicated Servers và hợp đồng doanh nghiệp, làm giảm khả năng tùy chỉnh và điều khiển của người dùng.
* Không cung cấp các API GraphQL, điều này có thể làm giảm sự linh hoạt trong việc truy vấn và quản lý dữ liệu.

1.4. Về cửa hàng

1.4.1. Đặc điểm nổi bật của cửa hàng

Cửa hàng Mỡ Coffee & Tea có không gian trang nhã và thoải mái, mang lại cảm giác thư thái cho khách hàng ngay từ lúc bước chân vào. Không chỉ có không gian trang trí đẹp mắt, mà cửa hàng còn nổi bật với không gian mở rộng và ánh sáng tự nhiên dồi dào, tạo điều kiện thuận lợi cho việc làm việc và học tập. Bố trí của cửa hàng cũng được thiết kế một cách thông minh, với các khu vực riêng biệt phù hợp cho nhóm hoặc cá nhân, từ khu vực ngồi thoải mái với sofa mềm mại đến bàn làm việc riêng tư. Không gian này cũng được trang bị các tiện nghi hiện đại như Wi-Fi miễn phí và ổ cắm điện tiện lợi, tạo điều kiện thuận lợi cho khách hàng muốn làm việc hoặc học tập từ xa. Đồ uống được phục vụ tại cửa hàng cũng là điểm nhấn đáng chú ý, từ cà phê đậm đà đến các loại trà thảo mộc và đồ uống sữa đặc biệt, mỗi món đều được chế biến tỉ mỉ và tinh tế. Tất cả những đặc điểm này kết hợp tạo nên một không gian ấm áp và đầy cảm hứng, là nơi lý tưởng để thư giãn, sáng tạo và kết nối với cộng đồng.



Hình 2: Cửa hàng Mỡ Coffee & Tea

1.4.2. Phương thức hoạt động

Cửa hàng hoạt động linh hoạt bằng cách cho phép khách hàng đặt và thanh toán đồ uống trực tuyến hoặc đến uống tại cửa hàng. Khách hàng có thể chọn món qua ứng dụng hoặc website, sau đó thanh toán trực tuyến. Hoặc họ có thể đến trực tiếp và đặt hàng với nhân viên. Điều này mang lại sự thuận tiện cho khách hàng và giúp cửa hàng phục vụ mọi nhu cầu một cách linh hoạt.

1.4.3. Dịch vụ cung cấp

Cửa hàng luôn tự tin về việc cung cấp một loạt đồ uống chất lượng, bao gồm cà phê và trà, được chế biến từ nguyên liệu chất lượng cao. Đội ngũ nhân viên chuyên nghiệp và thân thiện luôn sẵn lòng phục vụ khách hàng, tạo ra trải nghiệm mua sắm và thưởng thức đồ uống tốt nhất có thể. Em cam kết đảm bảo đồ uống luôn tươi mới và hương vị đặc trưng, cũng như giao hàng nhanh chóng và đáp ứng mọi yêu cầu của khách hàng một cách tận tình và chuyên nghiệp.

CHƯƠNG 2. KHẢO SÁT, PHÂN TÍCH & THIẾT KẾ HỆ THỐNG

2.1. Khảo sát yêu cầu

2.1.1. Xác định nhu cầu của người dùng

Để xác định nhu cầu và mong đợi của người dùng, em đã tiến hành một loạt các phương pháp nghiên cứu, bao gồm khảo sát trực tuyến, phỏng vấn người dùng tiềm năng và phân tích các dữ liệu sẵn có. Dựa trên các phản hồi và thông tin thu thập được, tôi đã rút ra những kết luận sau:

***Yêu cầu cơ bản:*** Người dùng mong đợi ứng dụng cung cấp tính năng đặt đồ uống một cách thuận tiện và nhanh chóng, bao gồm lựa chọn sản phẩm, thanh toán dễ dàng và giao hàng đến địa chỉ chỉ định.

***Tương tác thân thiện:*** Người dùng mong đợi giao diện ứng dụng dễ sử dụng, thân thiện và trực quan, giúp họ tìm kiếm và chọn lựa các sản phẩm một cách dễ dàng mà không gặp phải khó khăn.

***Tính linh hoạt và cá nhân hóa:*** Người dùng mong muốn có khả năng tùy chỉnh đơn hàng của họ theo các yêu cầu cụ thể, bao gồm việc thêm bớt thành phần, điều chỉnh số lượng và sự lựa chọn về phương thức giao hàng.

***Đáng tin cậy và an toàn:*** Người dùng quan tâm đến tính bảo mật của thông tin cá nhân và thanh toán, mong muốn ứng dụng đảm bảo an toàn cho dữ liệu của họ.

***Hỗ trợ khách hàng:*** Người dùng mong đợi có một cơ chế hỗ trợ khách hàng hiệu quả, hotline hỗ trợ và một hệ thống giải quyết khiếu nại nhanh chóng và công bằng.

***Tính năng đa nền tảng:*** Người dùng mong đợi ứng dụng có sẵn trên nhiều nền tảng, bao gồm di động và web, để họ có thể tiếp cận và sử dụng dễ dàng từ mọi thiết bị.

2.1.2. Thu thập yêu cầu chức năng và phi chức năng

Trong giai đoạn này, em đã thu thập các yêu cầu về chức năng và phi chức năng của ứng dụng dựa trên nhu cầu và mong đợi của người dùng. Cụ thể em đã thu thập các yêu cầu sau:

***\*Yêu cầu chức năng:***

- ***Đặt đồ uống:*** Người dùng có thể dễ dàng tìm kiếm và đặt đồ uống một cách nhanh chóng thông qua ứng dụng.

***- Quản lý tài khoản:*** Người dùng có thể đăng ký, đăng nhập và quản lý thông tin cá nhân, địa chỉ giao hàng và lịch sử đặt hàng của họ.

***- Tìm kiếm và lọc sản phẩm:*** Người dùng cần có khả năng tìm kiếm và lọc sản phẩm dễ dàng theo các tiêu chí như loại đồ uống, giá cả, đánh giá của người dùng, và các tính năng khác.

***- Đánh giá và nhận xét:*** Hệ thống cần cho phép người dùng đánh giá và viết nhận xét về sản phẩm mà họ đã mua để cung cấp thông tin hữu ích cho người khác.

***- Hỗ trợ khách hàng:*** Cung cấp kênh hỗ trợ khách hàng để giải đáp thắc mắc, xử lý khiếu nại và cung cấp hỗ trợ kỹ thuật khi cần thiết.

***- Thanh toán an toàn:*** Hệ thống thanh toán phải đảm bảo an toàn và tiện lợi cho người dùng, bao gồm nhiều phương thức thanh toán như thẻ tín dụng, thanh toán khi nhận hàng, và ví điện tử.

***- Thông báo đơn hàng:*** Người dùng sẽ nhận được thông báo về trạng thái của đơn hàng, bao gồm xác nhận đặt hàng, xử lý đơn hàng, và giao hàng thành công.

***\*Yêu cầu phi chức năng:***

***- Hiệu suất:*** Ứng dụng phải có hiệu suất cao và thời gian phản hồi nhanh để cung cấp trải nghiệm người dùng tốt nhất.

***- Bảo mật:*** Bảo mật thông tin cá nhân và giao dịch của người dùng là ưu tiên hàng đầu, đảm bảo rằng dữ liệu được lưu trữ và truyền tải một cách an toàn.

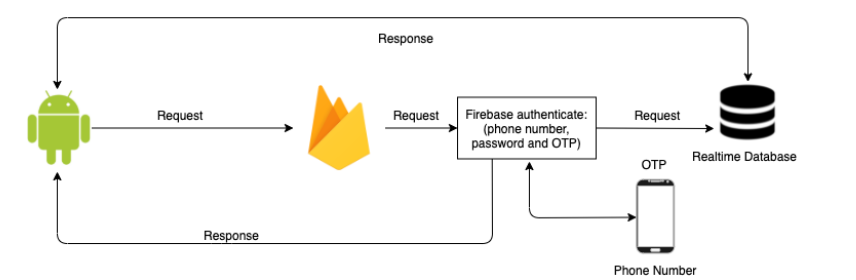
***- Giao diện người dùng thân thiện:*** Giao diện người dùng phải được thiết kế một cách đơn giản, dễ sử dụng và hấp dẫn để tăng cường trải nghiệm người dùng.

***- Dễ dàng sao lưu và khôi phục dữ liệu:*** Hệ thống cần cung cấp tính năng sao lưu và khôi phục dữ liệu để đảm bảo an toàn cho dữ liệu người dùng trong trường hợp có sự cố xảy ra.

***- Tuân thủ quy định và luật pháp:*** Đảm bảo rằng hệ thống tuân thủ các quy định và luật pháp liên quan đến bảo mật dữ liệu, quyền riêng tư và giao dịch thương mại điện tử.

2.2. Khảo sát hệ thống

2.2.2. Mô hình hệ thống



Hình 3: Mô hình hệ thống

2.2.3. Yêu cầu phần cứng

***Admin:***

- Smartphone chạy hệ điều hành Android hoặc iOS, kết nối internet

- Cài đặt phiên bản mới nhất của ứng dụng

- Máy tính có trình duyệt được kết nối internet

***User:***

- Smartphone chạy hệ điều hành Android hoặc iOS, kết nối internet

- Cài đặt phiên bản mới nhất của ứng dụng

2.3. Phân tích các chức năng chính của ứng dụng

Các chức năng chính mà ứng dụng cần có để đáp ứng nhu cầu của người dùng và đạt được mục tiêu kinh doanh

***Chức năng đặt đồ uống:*** Người dùng có thể chọn và đặt đồ uống một cách dễ dàng từ danh sách sản phẩm có sẵn trên ứng dụng. Hiển thị vào giỏ hang, nhập số lượng cần mua, nhập mã giảm giá (nếu có)

***Chức năng tìm kiếm và lọc sản phẩm:*** Người dùng có thể tìm kiếm sản phẩm theo tên, loại, giá cả và các tiêu chí khác. Họ cũng có thể sử dụng các bộ lọc để thu hẹp kết quả tìm kiếm.

***Chức năng thanh toán và giao hàng:*** Người dùng có thể thanh toán đơn hàng của mình thông qua các phương thức thanh toán trực tuyến và chọn phương thức giao hàng phù hợp.

***Chức năng đánh giá và nhận xét:*** Người dùng có thể đánh giá và viết nhận xét về sản phẩm mà họ đã mua, cung cấp thông tin hữu ích cho người khác.

***Chức năng hỗ trợ khách hàng:*** Hệ thống cung cấp kênh hỗ trợ khách hàng để giải đáp thắc mắc, xử lý khiếu nại và cung cấp hỗ trợ kỹ thuật khi cần thiết.

***Chức năng quản lý tài khoản:*** Người dùng có thể quản lý thông tin cá nhân, địa chỉ giao hàng, cập nhật mật khẩu và kiểm tra lịch sử đơn hàng của họ.

***Chức năng thông báo và cập nhật:*** Hệ thống sẽ cung cấp thông báo và cập nhật về tình trạng đơn hàng, ưu đãi mới, thông tin khuyến mãi và các thông báo quan trọng khác đến người dùng.

***Chức năng đăng nhập/đăng ký:*** Người dùng có khả năng đăng nhập vào hệ thống bằng tài khoản đã đăng ký để đặt hang, mua hàng trên ứng dụng.

2.4. Xác định tác nhân của hệ thống và User Case

2.4.1. Liệt kê các tác nhân của Hệ thống (Actor)

***Khách Xem:*** Người chưa có tài khoản trong hệ thống, chỉ có thể xem thông tin sản phẩm. Có thể đăng ký làm thành viên.

***Khách Hàng:*** Người đã có tài khoản trong hệ thống và có thể thực hiện các giao dịch như đặt hàng, chọn sản phẩm, địa điểm và thời gian giao hàng.

***Quản lý:*** Người điều hành, quản lý và theo dõi mọi hoạt động của hệ thống.

***Nhân Viên:*** Người tiếp nhận và xử lý các đơn hàng, yêu cầu bảo hành do người quản lý giao.

2.4.2. Liệt kê các User Case

***\* Quản lý Tài khoản và Người dùng:***

* UC Đăng nhập
* UC Đăng ký
* UC Quản lý thông tin cá nhân
* UC Quản lý khách hàng
* UC Quản lý nhân viên

***\* Quản lý Sản phẩm và Danh mục:***

* UC Thêm sản phẩm vào giỏ hàng
* UC Loại sản phẩm khỏi giỏ hàng
* UC Xem thông tin sản phẩm
* UC Quản lý danh mục, thể loại

***\* Quản lý Đơn hàng và Thanh toán:***

* UC Thanh toán hóa đơn
* UC Xét duyệt hóa đơn
* UC Quản lý doanh thu

2.4.3. Đặc tả User Case

***\* Đăng ký:***

* ***Tác nhân****:* Khách xem
* ***Mục tiêu:*** Cho phép khách xem đăng ký làm thành viên của hệ thống.
* ***Tiền điều kiện*:** Khách xem chưa có tài khoản thành viên của hệ thống.
* ***Hậu điều kiện:*** Khách hàng trở thành thành viên của hệ thống.
* ***Mô tả:*** Khách xem điền vào form đăng ký. Sau khi hệ thống lưu các thông tin cần thiết, khách hàng có thể đăng nhập vào hệ thống.

***\* Đăng nhập:***

* ***Tác nhân*:** Khách hàng, quản trị, nhân viên.
* ***Mục tiêu:*** Cho phép thành viên đăng nhập vào hệ thống.
* ***Tiền điều kiện:*** Thành viên chưa đăng nhập vào hệ thống.
* ***Hậu điều kiện:*** Thành viên đã đăng nhập thành công và có thể sử dụng các chức năng mà hệ thống cung cấp.
* ***Mô tả:*** Thành viên điền thông tin vào form đăng nhập. Nếu thông tin đúng, khách hàng có thể đặt hàng trên hệ thống. Nếu không đúng, hệ thống sẽ yêu cầu khách hàng kiểm tra lại tài khoản.

***\* Xem thông tin sản phẩm:***

* ***Tác nhân*:** Quản trị, nhân viên, khách hàng, khách xem.
* ***Mục tiêu:*** Cho phép xem thông tin về các sản phẩm trong cửa hàng.
* ***Mô tả:*** Thành viên trong hệ thống xem các thuộc tính của từng sản phẩm. Nếu thành viên là người quản l**ý** thì có thể thay đổi thông tin sản phẩm. Nếu thành viên là khách hàng thì có thể thêm sản phẩm vào giỏ hàng.

***\* Quản lý thông tin cá nhân:***

* ***Tác nhân*:** Quản trị, nhân viên, khách hàng.
* ***Mục tiêu:*** Cho phép xem thông tin cá nhân và thay đổi chúng.
* ***Mô tả:*** Thành viên trong hệ thống xem các thông tin cá nhân đã lưu trên hệ thống. Hệ thống sẽ lưu những thay đổi thông tin cá nhân của thành viên đó.

***\* Thêm sản phẩm vào giỏ hàng:***

* ***Tác nhân*:** Khách hàng.
* ***Mục tiêu:*** Cho phép khách hàng đưa sản phẩm đã chọn vào giỏ hàng.
* ***Tiền điều kiện:*** Sản phẩm đã được chọn.
* ***Hậu điều kiện:*** Thông tin sản phẩm đã được chọn phải được lưu trữ.
* ***Mô tả:*** Khi muốn mua một sản phẩm nào đó, khách hàng tiến hành nhấp vào chức năng thêm sản phẩm vào giỏ hàng. Lúc này, sản phẩm đó sẽ được chuyển đến giỏ hàng.

***\* Loại sản phẩm vào giỏ hàng:***

* ***Tác nhân*:** Khách hàng.
* ***Mục tiêu****:* Cho phép khách hàng loại sản phẩm đã chọn khỏi giỏ hàng.
* ***Tiền điều kiện:*** Sản phẩm đã trong giỏ hàng.
* ***Hậu điều kiện****:* Sản phẩm đã được chọn đã loại bỏ khỏi giỏ hàng.
* ***Mô tả:*** Khi cảm thấy không còn muốn mua một sản phẩm nào đó, khách hàng tiến hành nhấp vào nút xóa sản phẩm. Lúc này sản phẩm đó sẽ loại khỏi giỏ hàng.

***\* Thanh toán:***

* ***Tác nhân*:** Khách hàng.
* ***Mục tiêu:*** Cho phép khách hàng tạo đơn hàng để tiến hành thanh toán.
* ***Tiền điều kiện:*** Khách hàng đã chọn sản phẩm vào giỏ hàng.
* ***Hậu điều kiện:*** Đơn hàng đã được gửi.
* ***Mô tả:*** Khách hàng lựa chọn những sản phẩm mà mình muốn thanh toán, sau đó nhấn vào chức năng thanh toán. Nếu tài khoản của khách hàng đủ để thanh toán, hệ thống sẽ kiểm tra mọi thông tin, nếu chính xác sẽ hiện thông báo đã thành công. Nếu không đủ, hệ thống sẽ thông báo khách hàng nạp thêm vào tài khoản.

***\* Quản lý danh mục:***

* ***Tác nhân*:** Quản trị.
* ***Mục tiêu****:* Cho phép thêm, xóa, cho phép ẩn hiện danh mục và thể loại hiện có trong hệ thống.
* ***Tiền điều kiện:*** Người quản trị cần quản lý danh mục và thể loại.
* ***Mô tả:*** Người quản trị vào mục “quản lý danh mục và thể loại” trong trang quản trị thực hiện các thao tác thêm, xóa, ẩn hiện. Hệ thống sẽ lưu những thay đổi của người quản trị vào cơ sở dữ liệu.

***\* Quản lý khách hàng:***

* ***Tác nhân*:** Quản trị.
* ***Mục tiêu:*** Cho phép xem và sửa thông tin của khách hàng nếu cần thiết.
* ***Tiền điều kiện:*** Người quản trị cần kiểm tra thông tin của khách hàng.
* ***Mô tả:*** Người quản trị vào mục “quản lý khách hàng” trong trang quản trị thực hiện các thao tác xem, sửa thông tin. Hệ thống sẽ lưu những thay đổi của người quản trị vào cơ sở dữ liệu.

***\* Xét duyệt hóa đơn:***

* ***Tác nhân*:** Nhân viên.
* ***Mục tiêu:*** Cho phép xem thông tin đơn hàng và duyệt đơn hàng.
* ***Tiền điều kiện:*** Nhân viên có đơn hàng cần được duyệt.
* ***Hậu điều kiện:*** Đơn hàng đã được duyệt.
* ***Mô tả:*** Nhân viên vào mục “quản lý đơn hàng” trong trang quản trị thực hiện xử lý các đơn hàng chưa được duyệt.

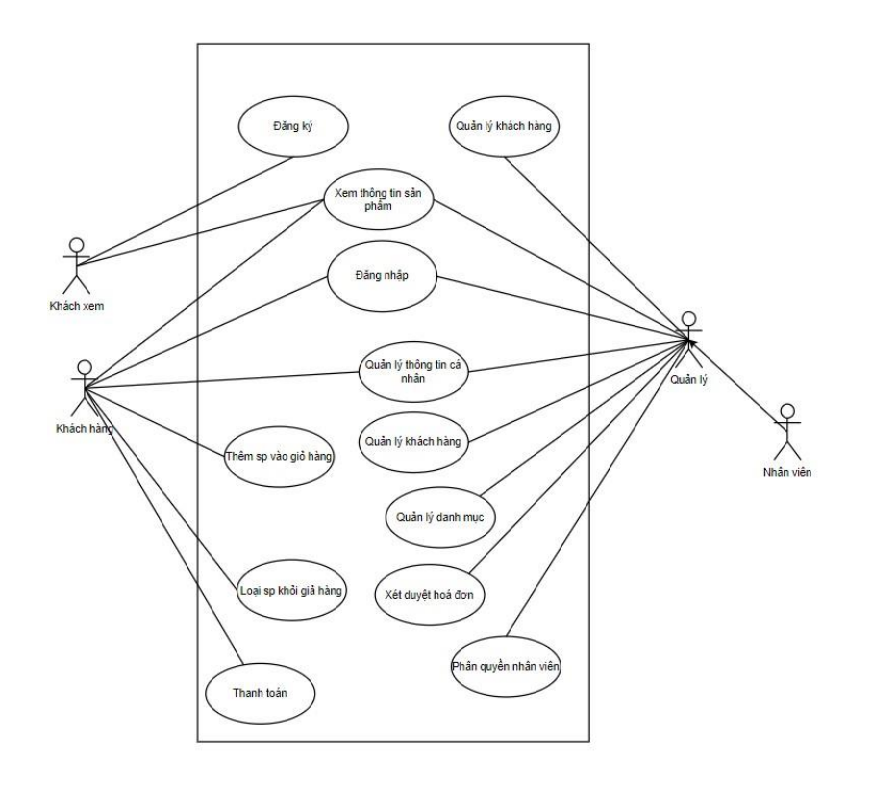
***\* Quản lý nhân viên:***

* ***Tác nhân*:** Quản trị.
* ***Mục tiêu:*** Cho phép người quản trị thêm, sửa, xoá nhân viên.
* ***Mô tả****:* Người quản trị vào mục “nhân viên” trong trang quản trị thực hiện điều chỉnh chức vụ cho nhân viên. Hệ thống sẽ lưu những thay đổi của người quản trị vào cơ sở dữ liệu.

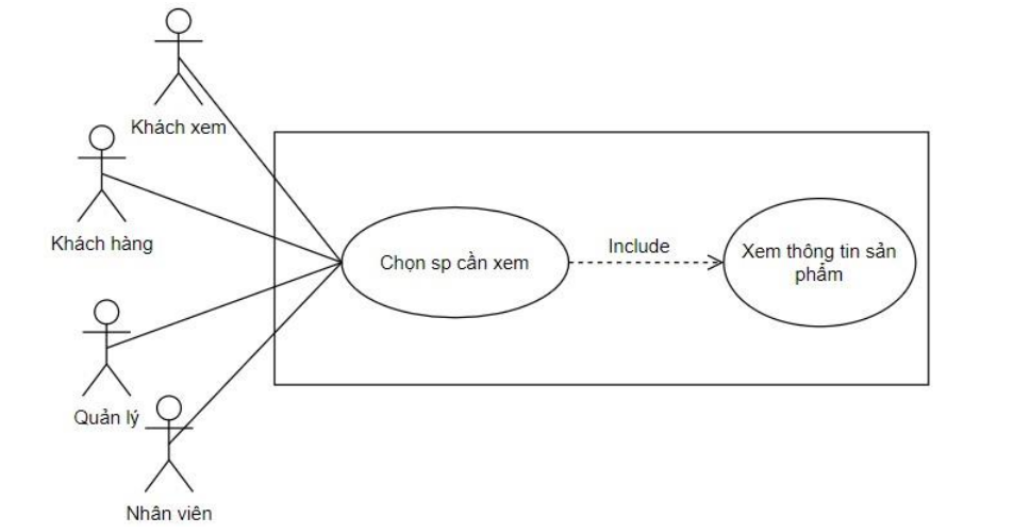
***\* Quản lý doanh thu:***

* ***Tác nhân*:** Quản trị, nhân viên.
* ***Mục tiêu:*** Cho phép người quản trị, nhân viên thống kê doanh thu của cửa hàng theo thời gian nhất định.
* ***Mô tả:*** Người quản trị và nhân viên vào mục “dashboard” trong trang quản trị để xem các số liệu thống kê.

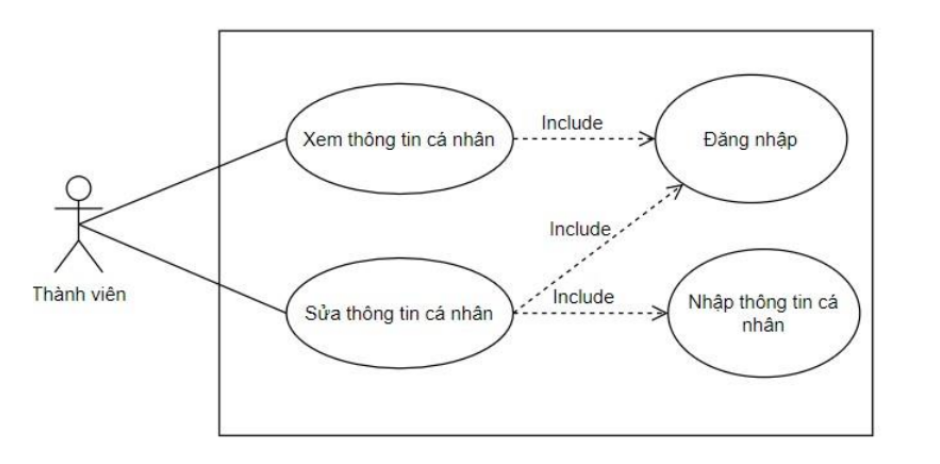
2.4.4. Biểu đồ User Case



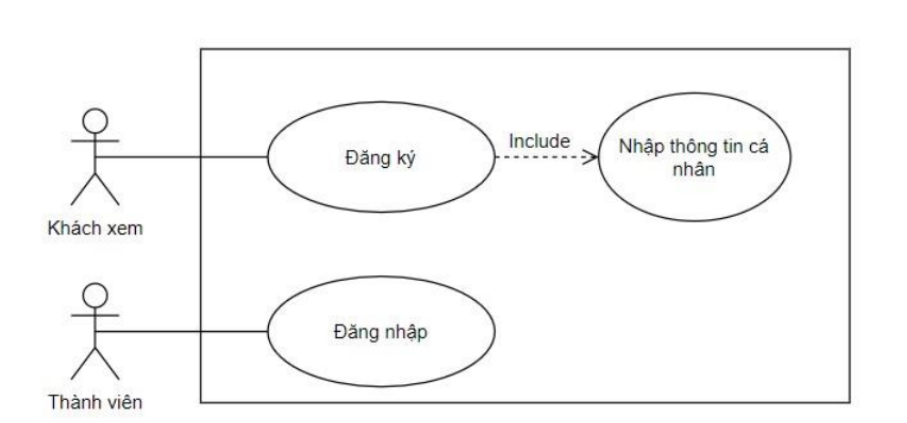
Hình 4: UC Tổng quát



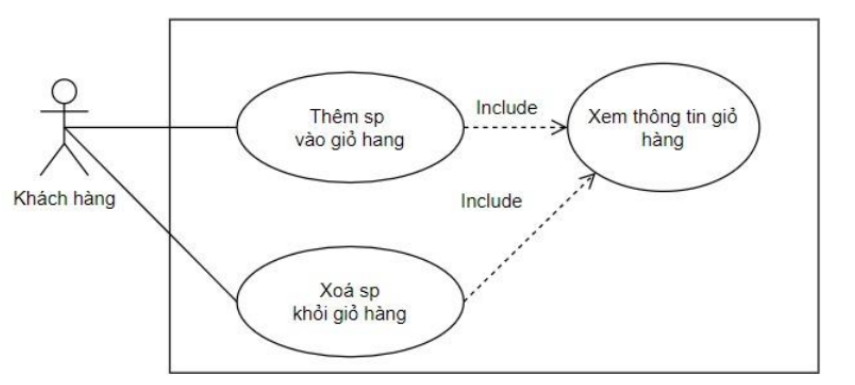
Hình 5: UC Xem thông tin sản phẩm



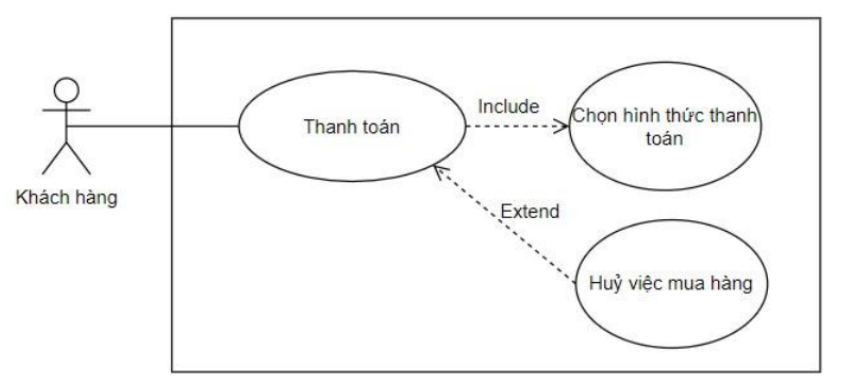
Hình 6: UC Quản lý thông tin sản phẩm



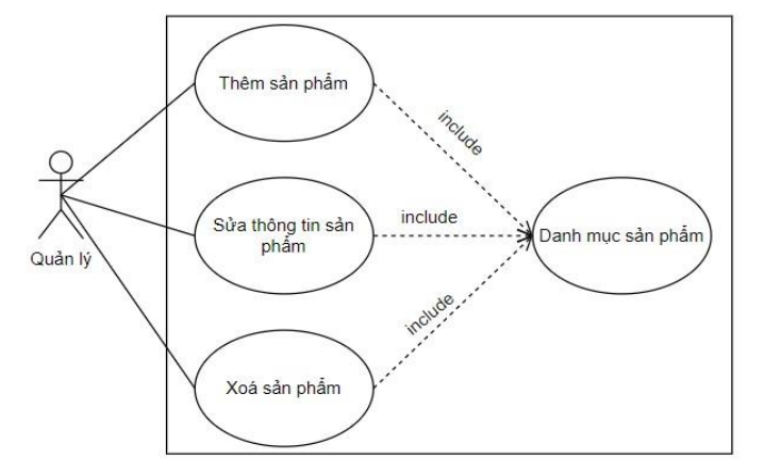
Hình 7: UC Đăng nhập, đăng ký



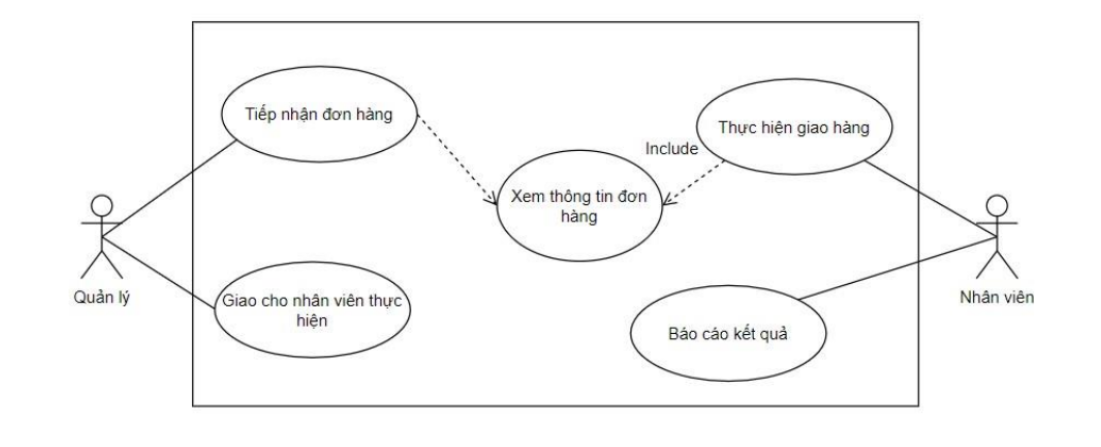
Hình 8: UC Quản lý giỏ hàng



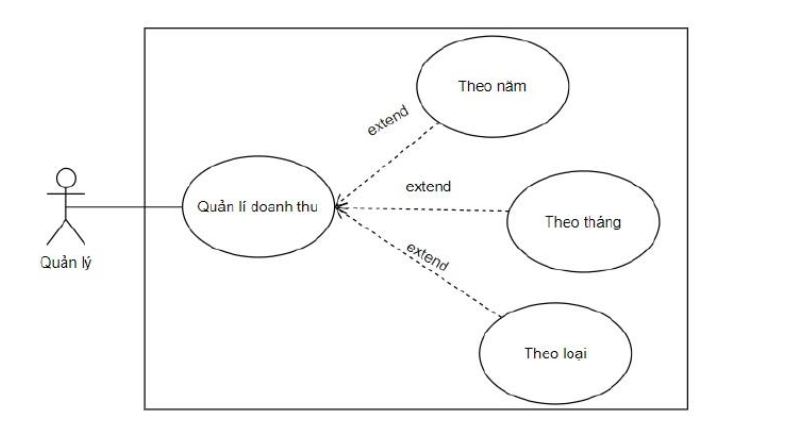
Hình 9: UC Thanh toán



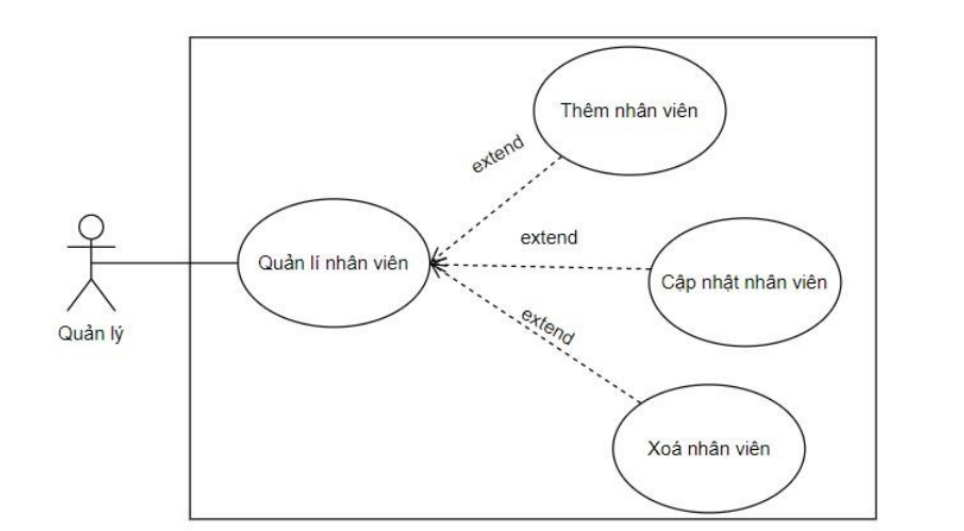
Hình 10: UC Quản lý danh mục sản phẩm



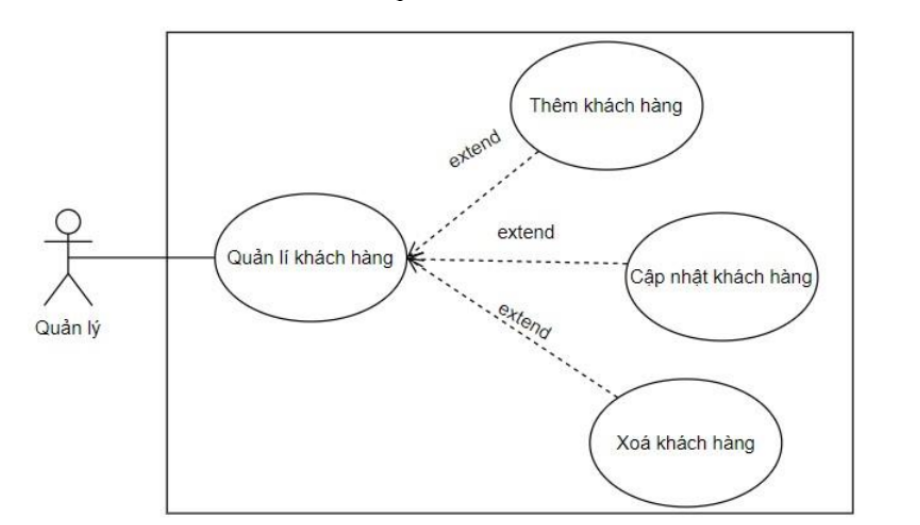
Hình 11: UC Xét duyệt hóa đơn



Hình 12: UC Quản lý doanh thu

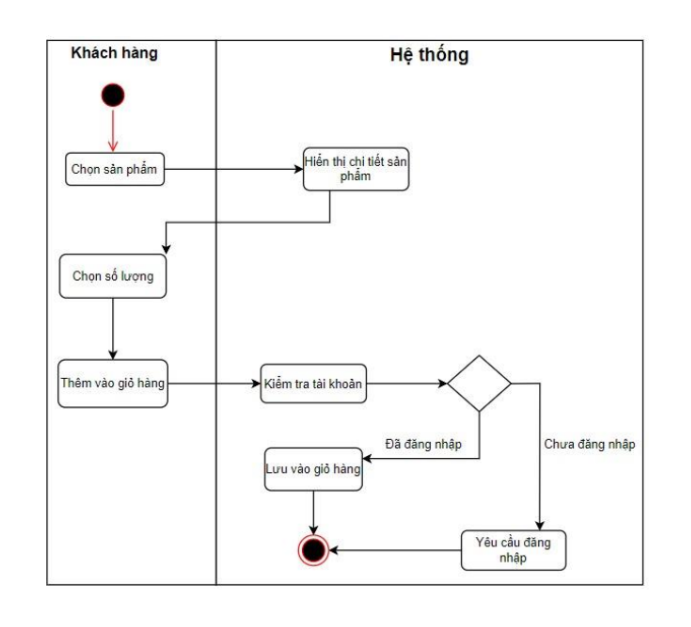


Hình 13: UC Quản lý nhân viên

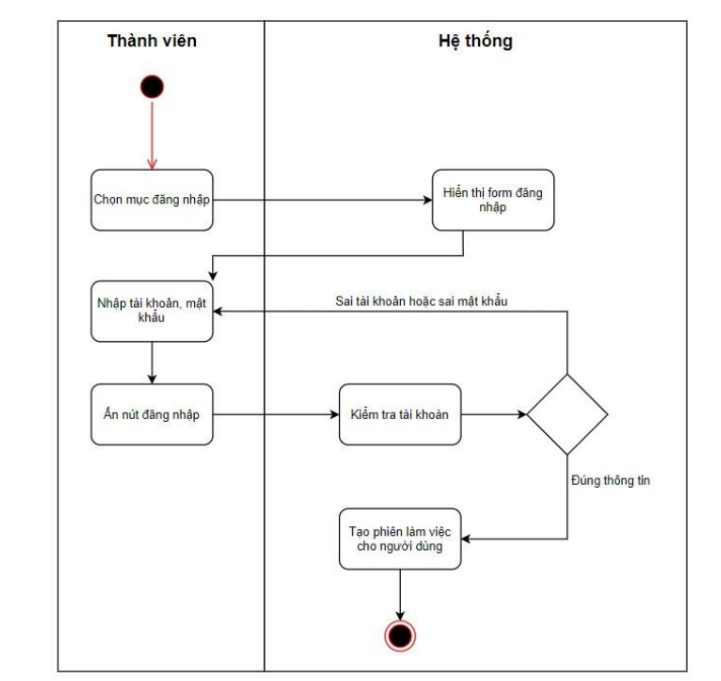


Hình 14: UC Quản lý khách hàng

2.4.5. Biểu đồ hoạt động



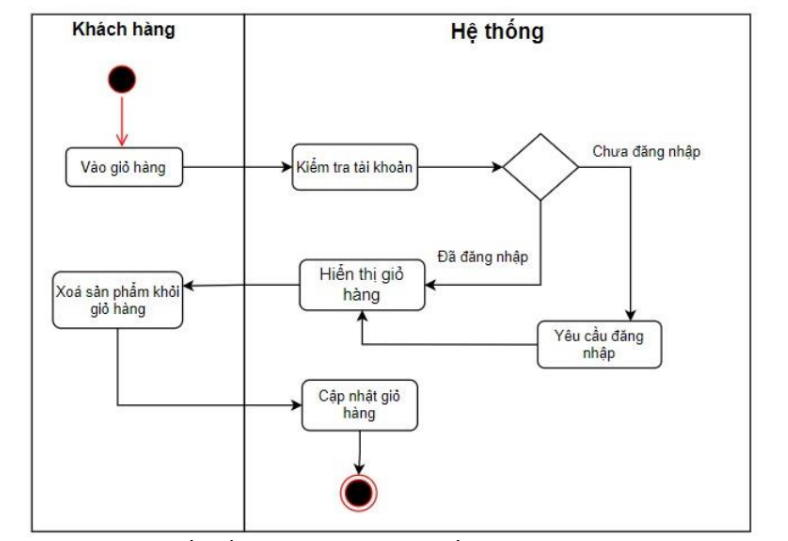
Hình 15: Biểu đồ thêm sản phẩm vào giỏ hàng



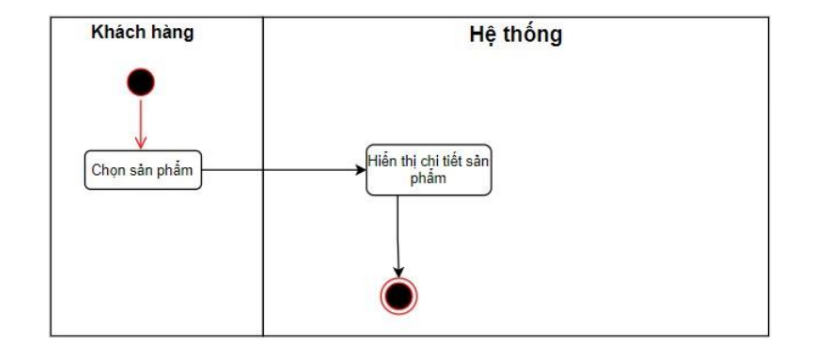
Hình 16: Biểu đồ hoạt động đăng nhập



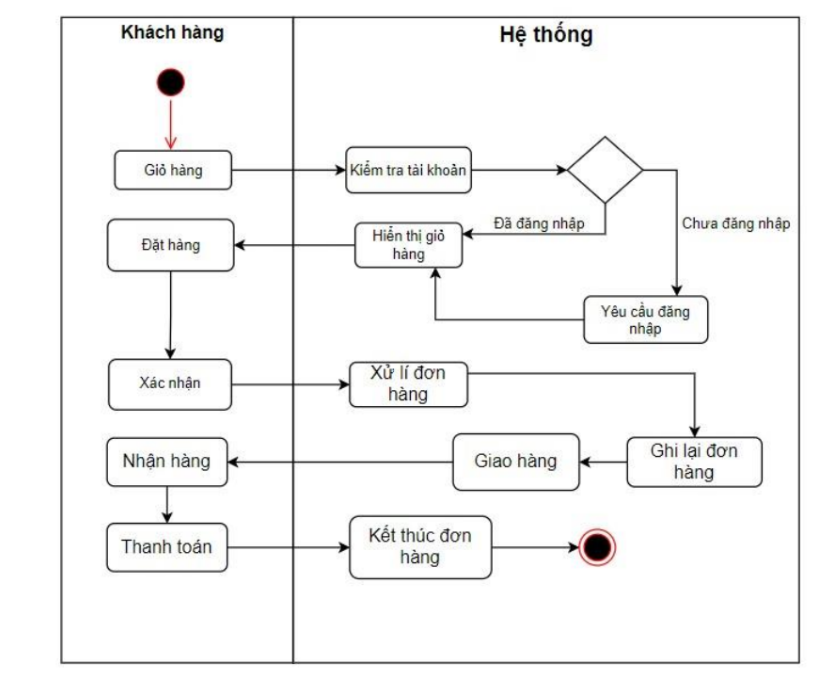
Hình 17: Biểu đồ hoạt động đăng ký



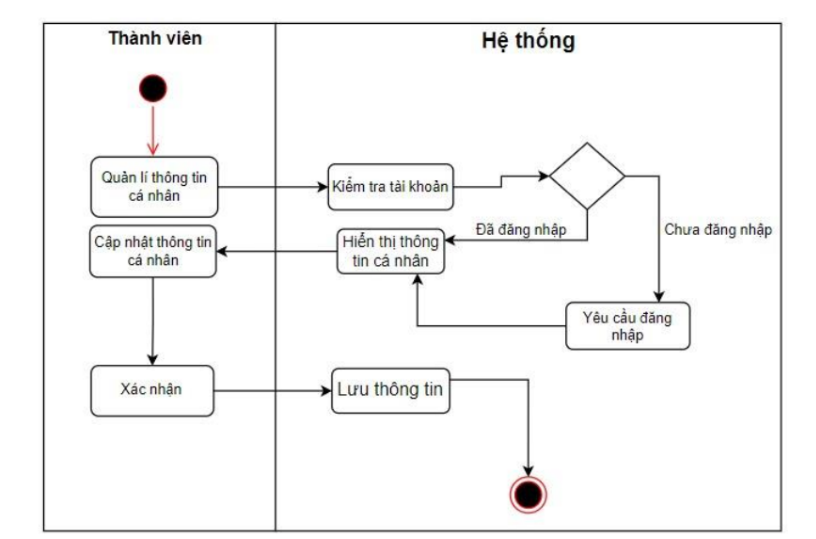
Hình 18: Biểu đồ hoạt động xoá sản phẩm khỏi giỏ hàng



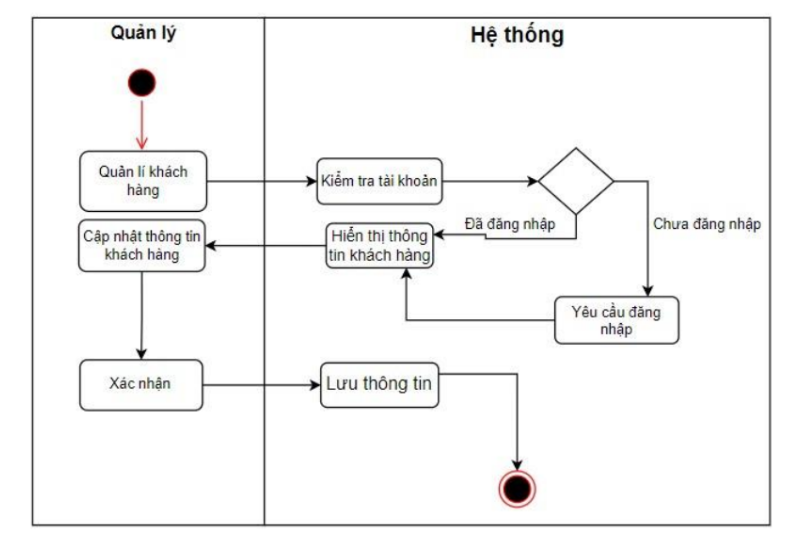
Hình 19: Biểu đồ hiển thị thông tin sản phẩm



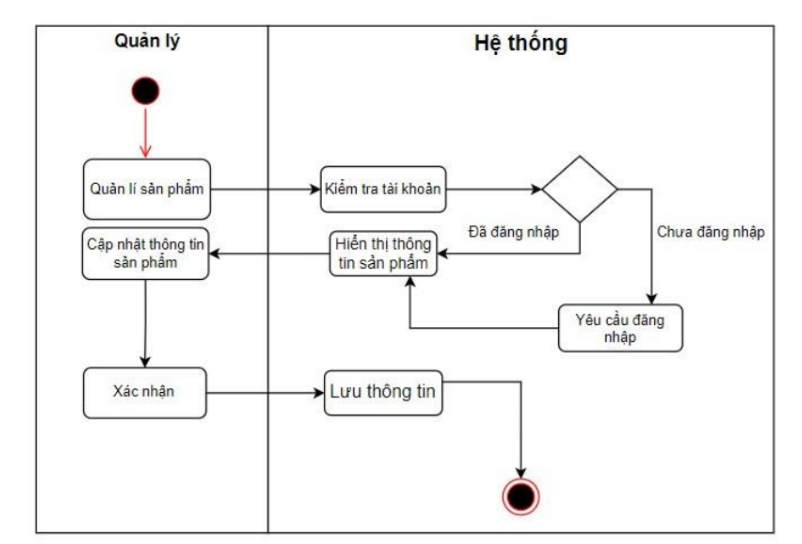
Hình 20: Biểu đồ thanh toán



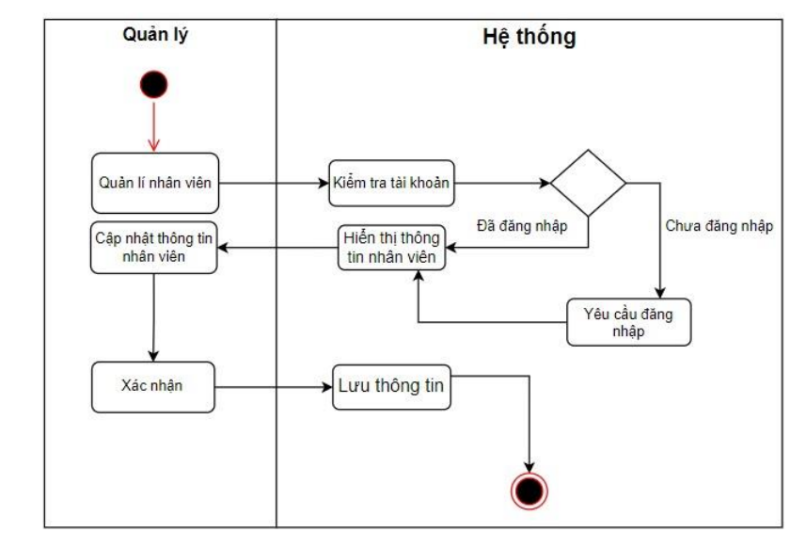
Hình 21: Biểu đồ hoạt động quản lý thông tin cá nhân



Hình 22: Biểu đồ hoạt động quản lý khách hàng

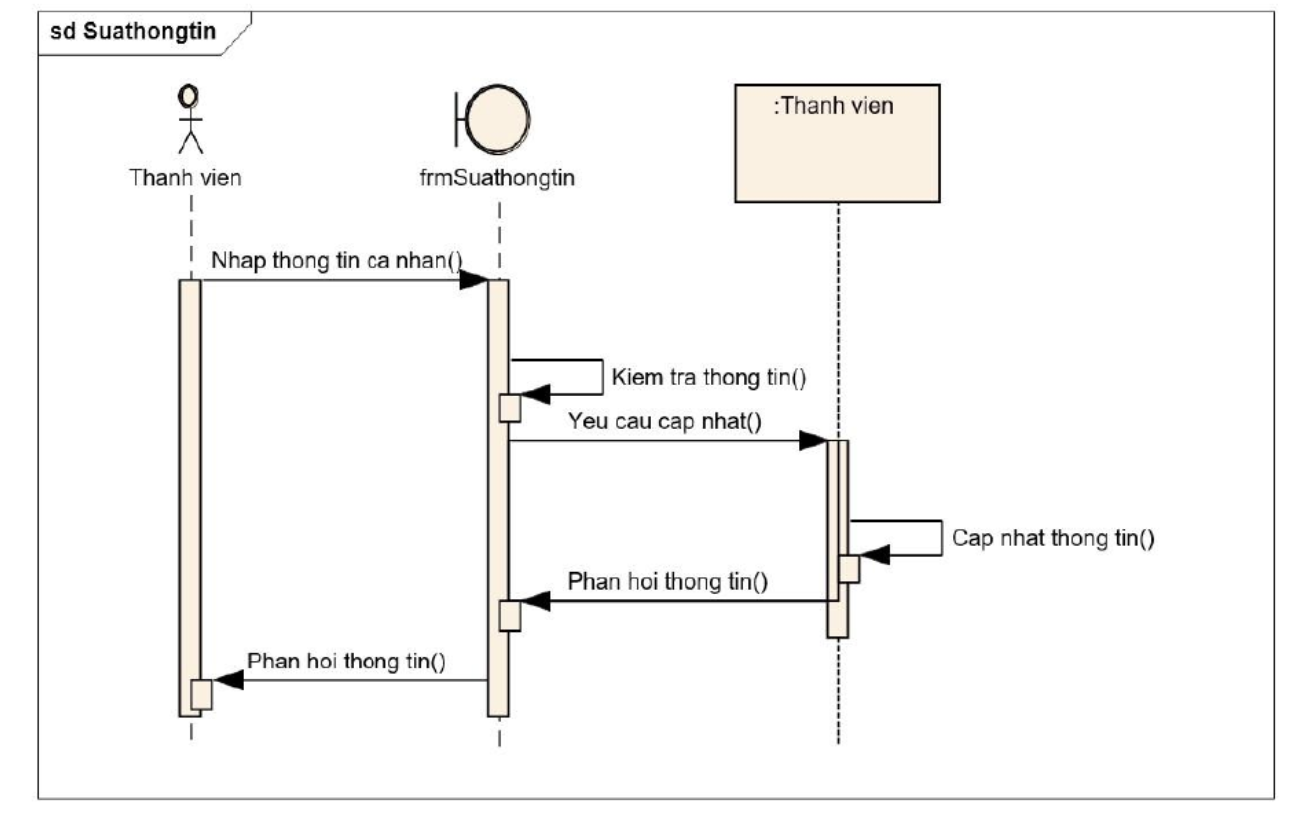


Hình 23: Biểu đồ hoạt động quản lý danh mục sản phẩm

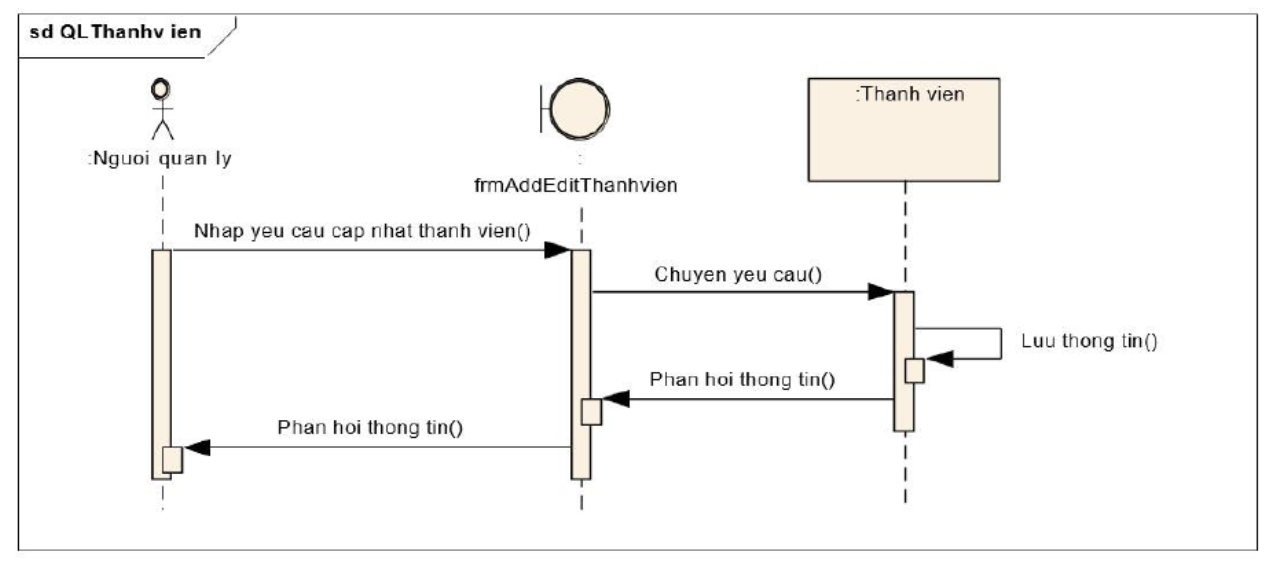


Hình 24: Biểu đồ hoạt động quản lý nhân viên

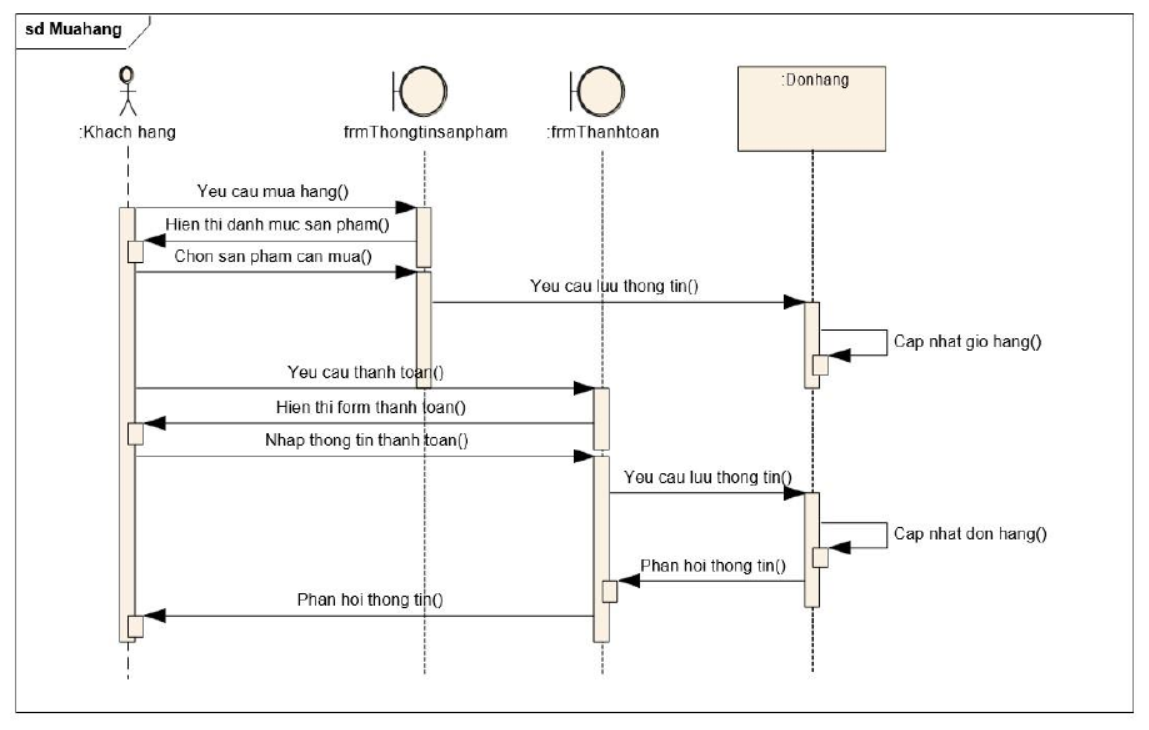
2.4.6. Biểu đồ tuần tự



Hình 25: Biểu đồ tuần tự UC Sửa thông tin cá nhân



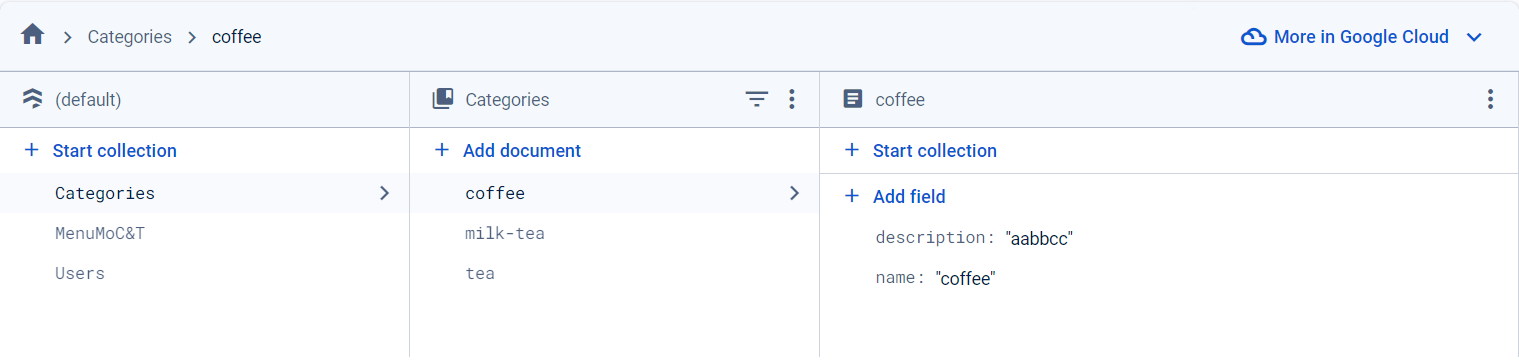
Hình 26: Biểu đồ tuần tự UC Quản lý danh sách thành viên



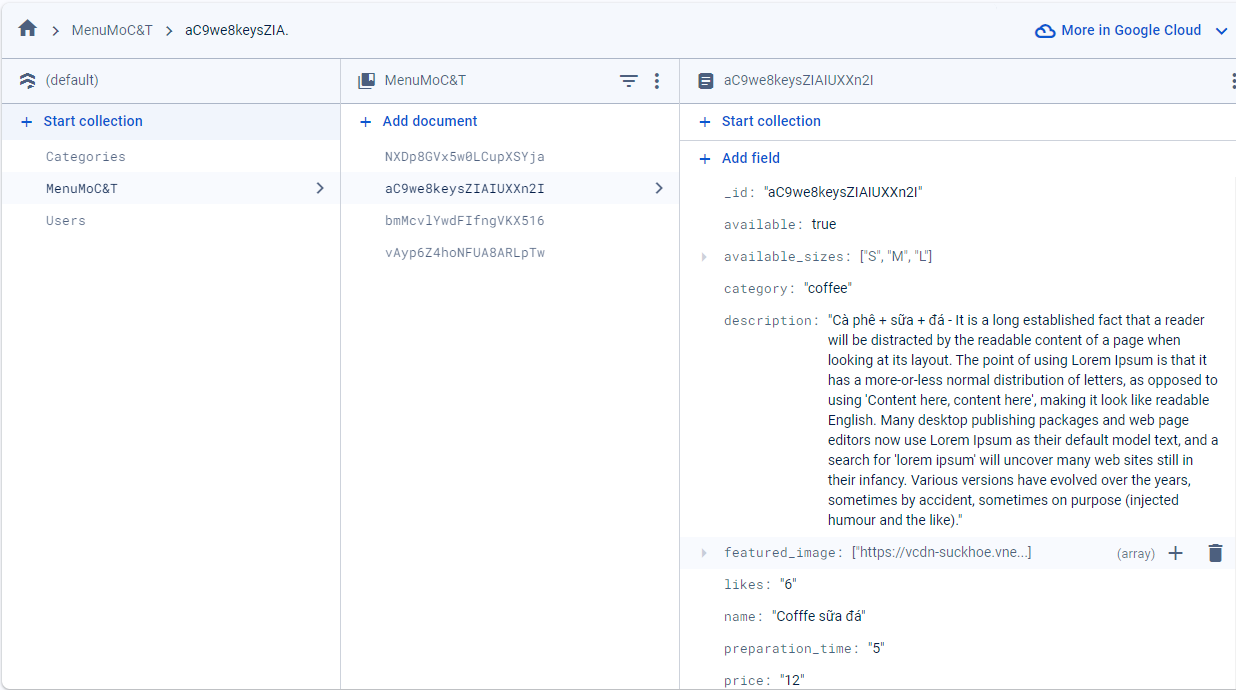
Hình 27: Biểu đồ tuần tự UC Mua hàng (Cập nhật giỏ hàng, thanh toán)

2.4.7. Cơ sở dữ liệu

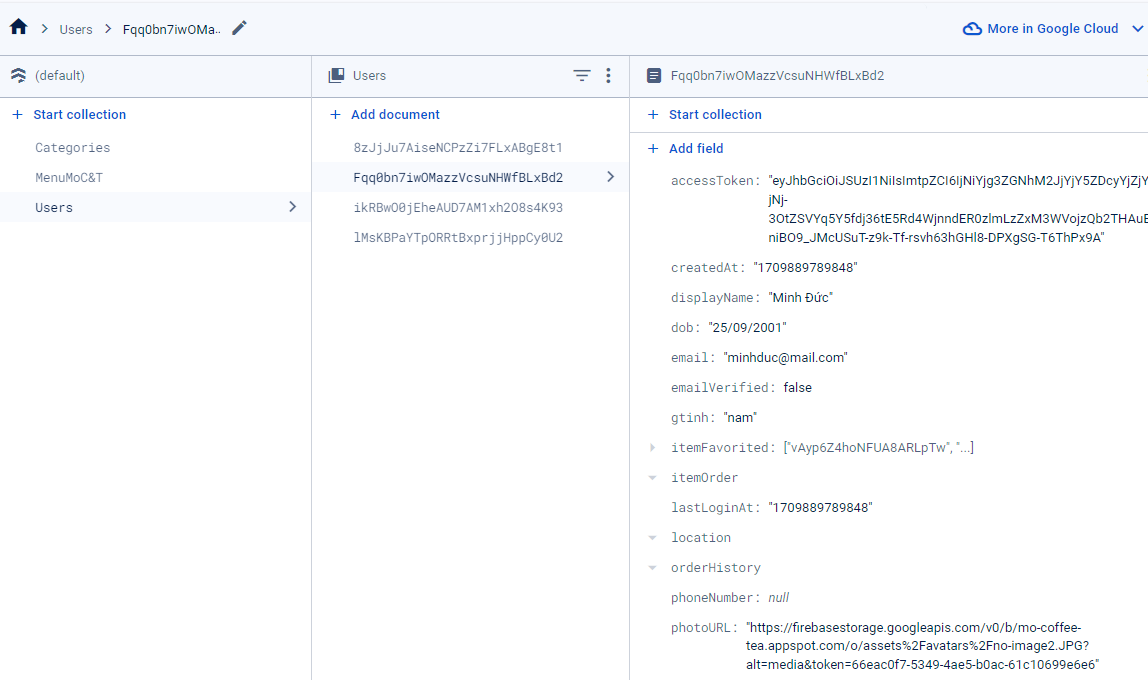
Firebase Cloud Firestore được sử dụng làm cơ sở dữ liệu cho ứng dụng. Cấu trúc cơ sở dữ liệu được tổ chức theo mô hình tài liệu (Collection), giúp quản lý dữ liệu một cách linh hoạt và dễ dàng mở rộng.



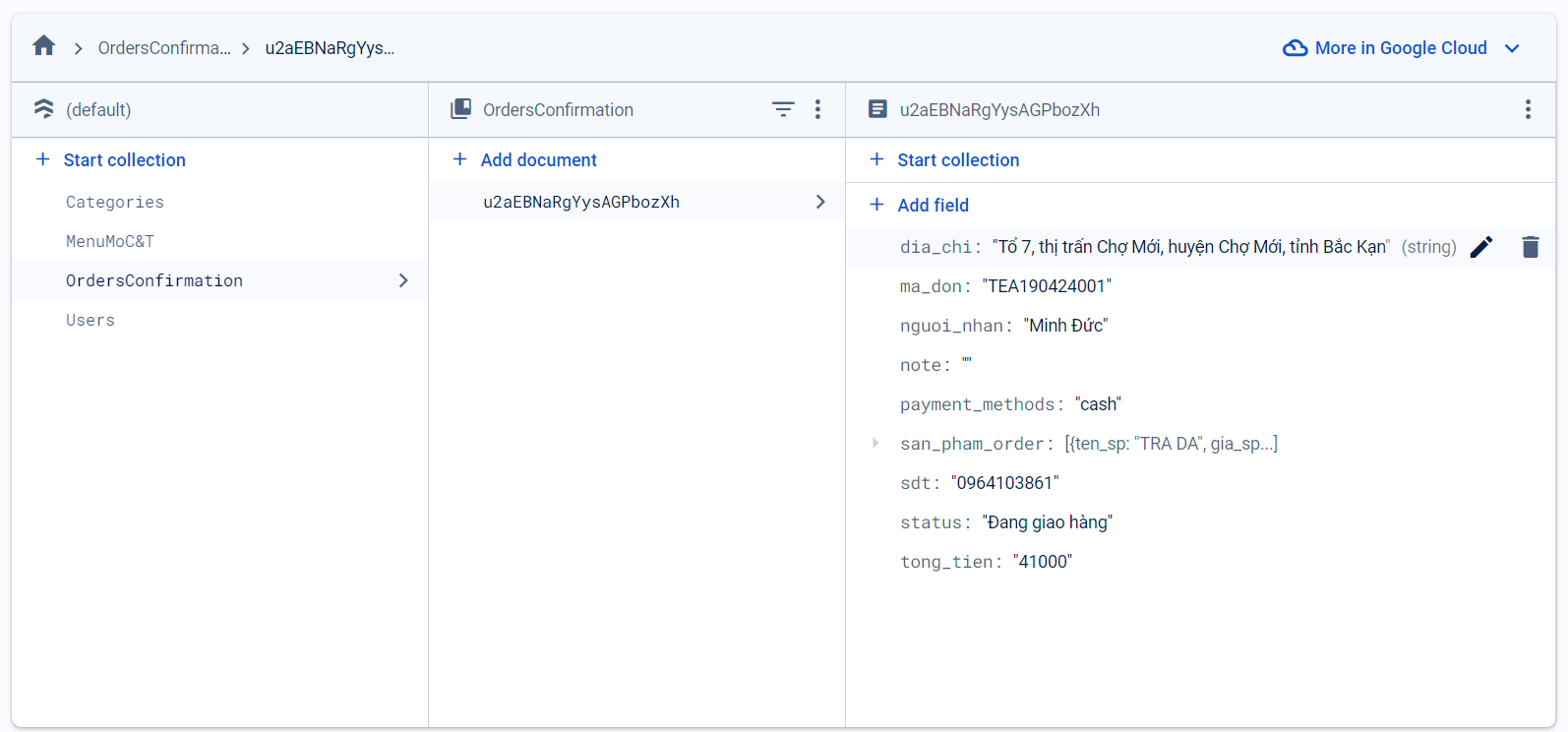
Hình 28: Danh mục sản phẩm (Tên bảng: Categories)



Hình 29: Menu sản phẩm (Tên bảng: MenuMoC&T)



Hình 30: Thông tin người dùng (Tên bảng: Users)



Hình 31: Danh sách đơn đặt hàng (Tên bảng: OrdersConfirmation)

Dưới đây là cấu trúc chi tiết của cơ sở dữ liệu:

* ***Danh mục sản phẩm (Categories):*** 
  + Mỗi danh mục sản phẩm được lưu trữ dưới dạng một *collection*.
  + Mỗi *key* trong danh mục sản phẩm là tên của danh mục.
  + Trong mỗi *key*:

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | Mô tả |
| **name** | String | Tên của danh mục sản phẩm. |
| **description** | String | Mô tả cho danh mục sản phẩm. |

Bảng 2.1. Minh họa Danh mục sản phẩm (Tên bảng: Categories)

* ***Menu sản phẩm (MenuMoC&T):***
  + Mỗi sản phẩm trong menu được lưu trữ dưới dạng một *collection*.
  + Mỗi *key* trong menu là một ID duy nhất đại diện cho sản phẩm.
  + Trong mỗi *key*:

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | Mô tả |
| **name** | String | Tên của sản phẩm. |
| **category** | String | Danh mục của sản phẩm, tham chiếu đến tên của danh mục sản phẩm trong Categories. |
| **price** | String | Giá của sản phẩm. |
| **available\_sizes** | Array | Các kích thước có sẵn cho sản phẩm. |
| **preparation\_time** | String | Thời gian chuẩn bị sản phẩm. |
| **ratings** | Object | Thông tin về đánh giá của sản phẩm, bao gồm "**total\_ratings**" (String) và "**average\_rating**" (String). |
| **available** | Boolean | Trạng thái của sản phẩm (có sẵn hay không). |
| **sold\_count** | String | Số lượng sản phẩm đã bán. |
| **featured\_image** | Array | Đường dẫn đến hình ảnh nổi bật của sản phẩm. |
| **likes** | String | Số lượt thích của sản phẩm. |
| **description** | String | Mô tả về sản phẩm. |

Bảng 2.2: Minh họa Menu sản phẩm (Tên bảng: MenuMoC&T)

* ***Thông tin người dùng (Users):***
  + Mỗi người dùng được lưu trữ dưới dạng một *collection*.
  + Mỗi *key* trong thông tin người dùng là một ID duy nhất đại diện cho người dùng.
  + Trong mỗi *key*:

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | Mô tả |
| **displayName** | String | Tên hiển thị của người dùng. |
| **role** | String | Vai trò của người dùng. |
| **itemOrder** | Array | Đơn hàng của người dùng. |
| **orderHistory** | Array | Lịch sử đơn hàng của người dùng. |
| **accessToken** | String | Mã thông báo truy cập của người dùng. |
| **emailVerified** | Boolean | Trạng thái xác minh email của người dùng. |
| **photoURL** | String | Đường dẫn đến ảnh đại diện của người dùng. |
| **createdAt** | String | Thời điểm tạo tài khoản. |
| **phoneNumber** | String | Số điện thoại của người dùng. |
| **lastLoginAt** | String | Thời điểm đăng nhập cuối cùng. |
| **dob** | String | Ngày sinh của người dùng. |
| **gtinh** | String | Giới tính của người dùng. |
| **location** | Array | Thông tin vị trí của người dùng. |
| **email** | String | Địa chỉ email của người dùng. |
| **itemFavorited** | Array | Danh sách các sản phẩm mà người dùng đã thích. |

Bảng 2.3: Minh họa Thông tin người dùng (Tên bảng: Users)

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | Mô tả |
| **ma\_don** | String | Mã đơn hàng đã đặt |
| **status** | String | Tình trạng đơn hàng |
| **nguoi\_nhan** | String | Tên người nhận hàng |
| **dia\_chi** | String | Địa chỉ người nhận hàng |
| **sdt** | String | Số điện thoại người nhận hàng |
| **note** | String | Ghi chú của người dùng cho cửa hàng |
| **san\_pham\_order** | Array | Sản phẩm mà người dùng đã đặt và bao gồm các thông tin như giá sản phẩm và số lượng sản phẩm |
| **tong\_tien** | String | Số tiền mà người dùng phải thanh toán |

Bảng 2.4: Minh họa Danh sách đơn đặt hàng (Tên bảng: OrdersConfirmation)

CHƯƠNG 3. KẾT QUẢ THỰC HIỆN

3.1. Mô tả bài toán

Trong thời đại công nghệ ngày nay, việc áp dụng các tiến bộ của công nghệ thông tin vào việc xây dựng các ứng dụng trở nên vô cùng phổ biến. Trong xu hướng này, việc phát triển ứng dụng đặt đồ uống qua mạng trở nên ngày càng quan trọng và phát triển mạnh mẽ. Trong bối cảnh mà nhu cầu về ăn uống của con người không ngừng tăng cao, song lại gặp phải sự bận rộn của cuộc sống hiện đại, những ứng dụng đặt đồ uống qua mạng đóng vai trò quan trọng, giúp khách hàng tiết kiệm thời gian và thuận lợi trong việc đặt món.

Đề tài này tập trung vào việc xây dựng một ứng dụng đặt đồ uống cho cửa hàng Mỡ Coffee & Tea tại tỉnh Bắc Kạn. Mục tiêu là tạo ra một nền tảng di động dễ sử dụng, hiệu quả, và giúp người dùng dễ dàng đặt đồ uống từ cửa hàng Mỡ Coffee & Tea thông qua thiết bị di động của họ.

Một cửa hàng kinh doanh cả cà phê và trà đặt ra nhiều thách thức đặc biệt. Đầu tiên, cần phải đảm bảo nguồn cung cấp ổn định cho cả hai loại sản phẩm này, với chất lượng và giá cả phù hợp. Không chỉ vậy, việc quản lý kho hàng cũng trở nên phức tạp hơn với sự đa dạng của các sản phẩm.

Ngoài ra, việc quản lý hoạt động kinh doanh như đặt hàng, lập hóa đơn, và theo dõi doanh thu cũng đòi hỏi sự chính xác và tổ chức. Một hệ thống bán hàng hiệu quả sẽ giúp giải quyết các thách thức này bằng cách tối ưu hóa quy trình làm việc và tăng cường khả năng theo dõi và điều chỉnh các hoạt động kinh doanh.

Như vậy, việc sử dụng các phần mềm bán hàng có thể giúp cửa hàng cà phê và trà giải quyết nhiều khó khăn trong quản lý hàng hóa, nhân viên và hoạt động kinh doanh tổng thể, từ đó tiết kiệm thời gian và nâng cao hiệu suất làm việc.

3.2. Về kiến thức

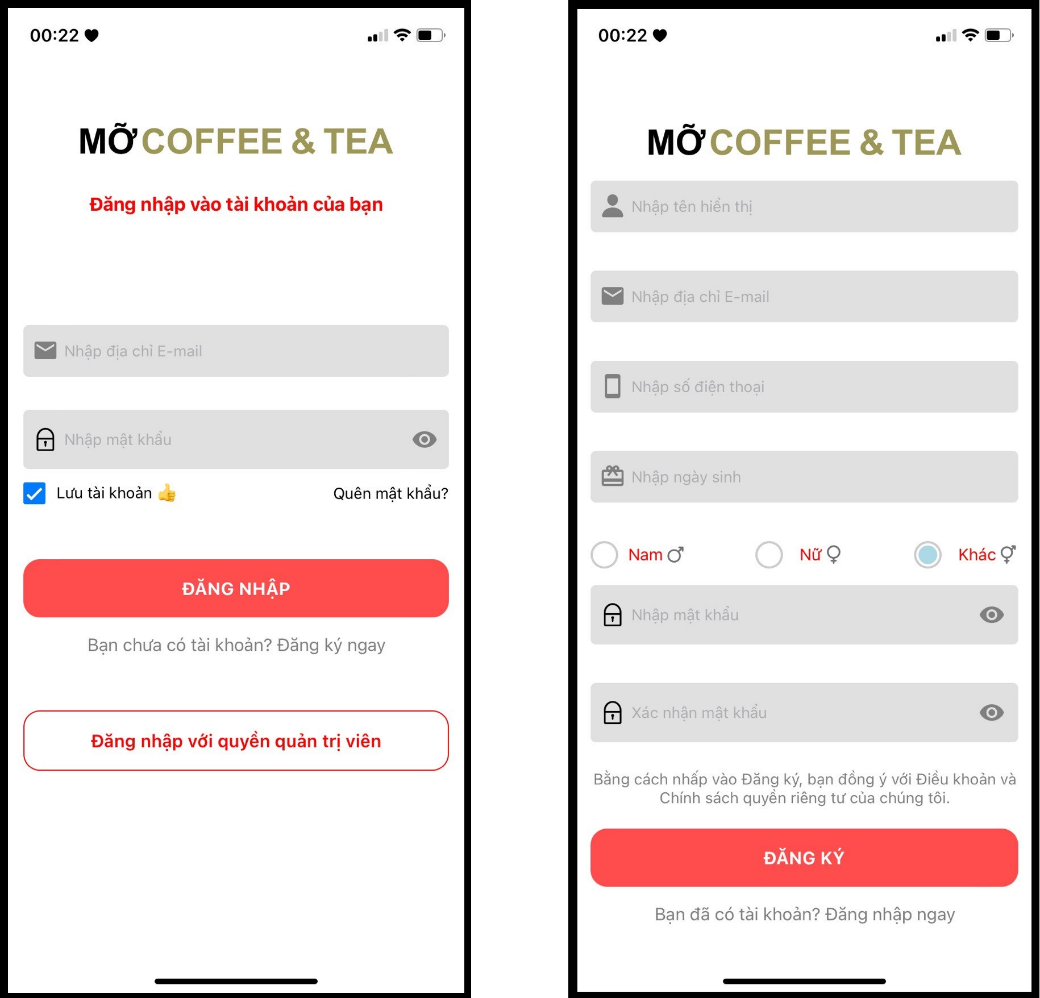
Sau thời gian thực hiện đồ án tốt nghiệp em đã đạt được nhiều tiến bộ cả về

mặt tìm hiểu, nghiên cứu lý thuyết lẫn kỹ năng lập trình. Có thể nói, thông qua

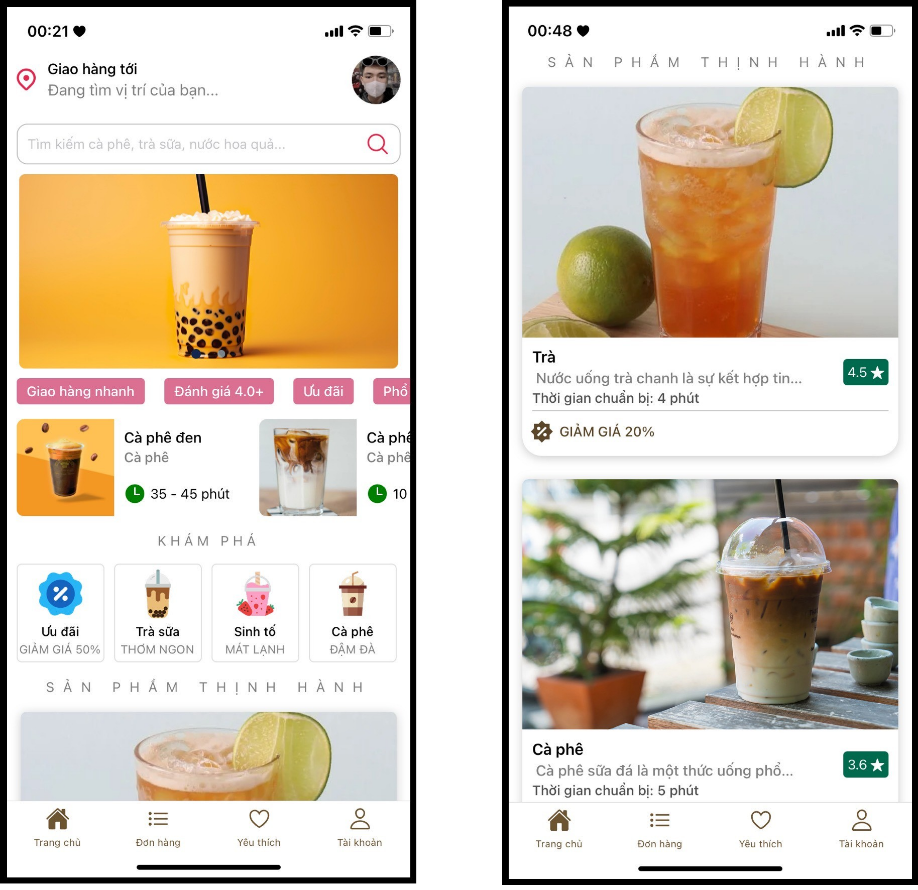
đồ án tốt nghiệp, em đã đạt được:

* Hiểu biết sâu hơn về các kỹ thuật và cách thức hoạt động của React Native trong việc phát triển ứng dụng di động.
* Nắm vững phương pháp phân tích và thiết kế hệ thống theo hướng đối tượng.
* Có kiến thức sâu rộng về nghiệp vụ trong lĩnh vực phát triển ứng dụng di động.
* Nâng cao khả năng tự học, tự nghiên cứu để giải quyết các vấn đề trong quá trình phát triển ứng dụng.
* Sử dụng nhiều thư viện mới trong cộng đồng React Native, mở rộng phạm vi công việc và cải thiện hiệu suất lập trình.

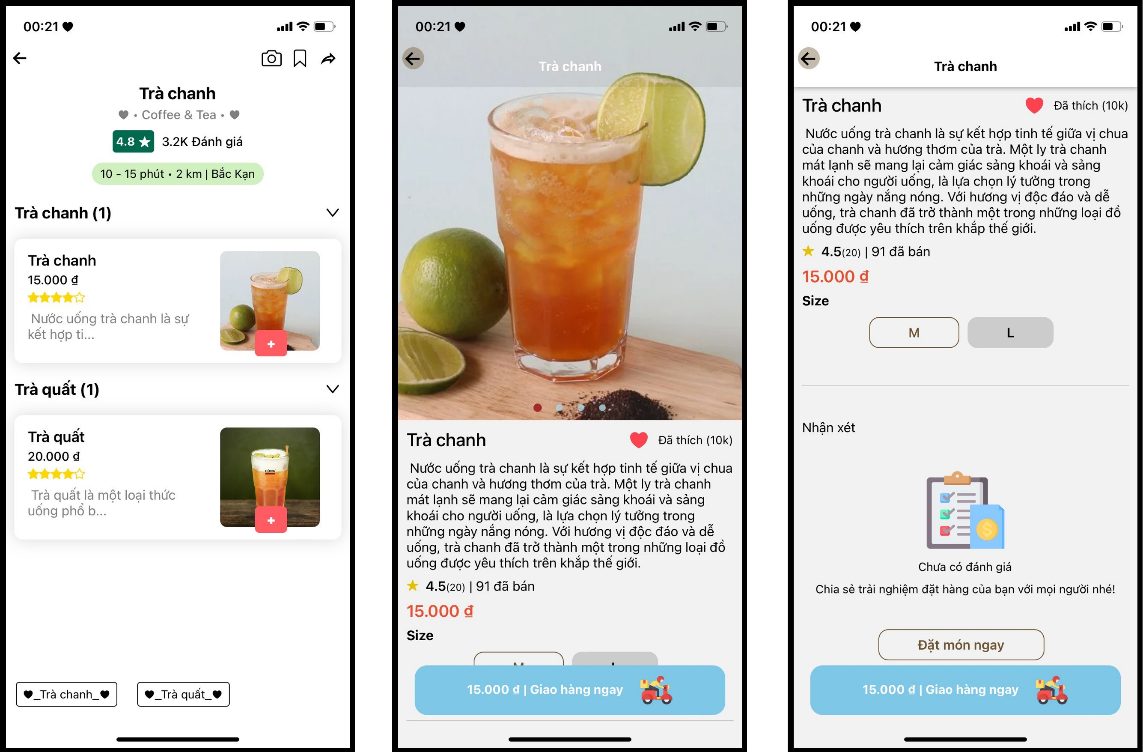
3.3. Về chương trình



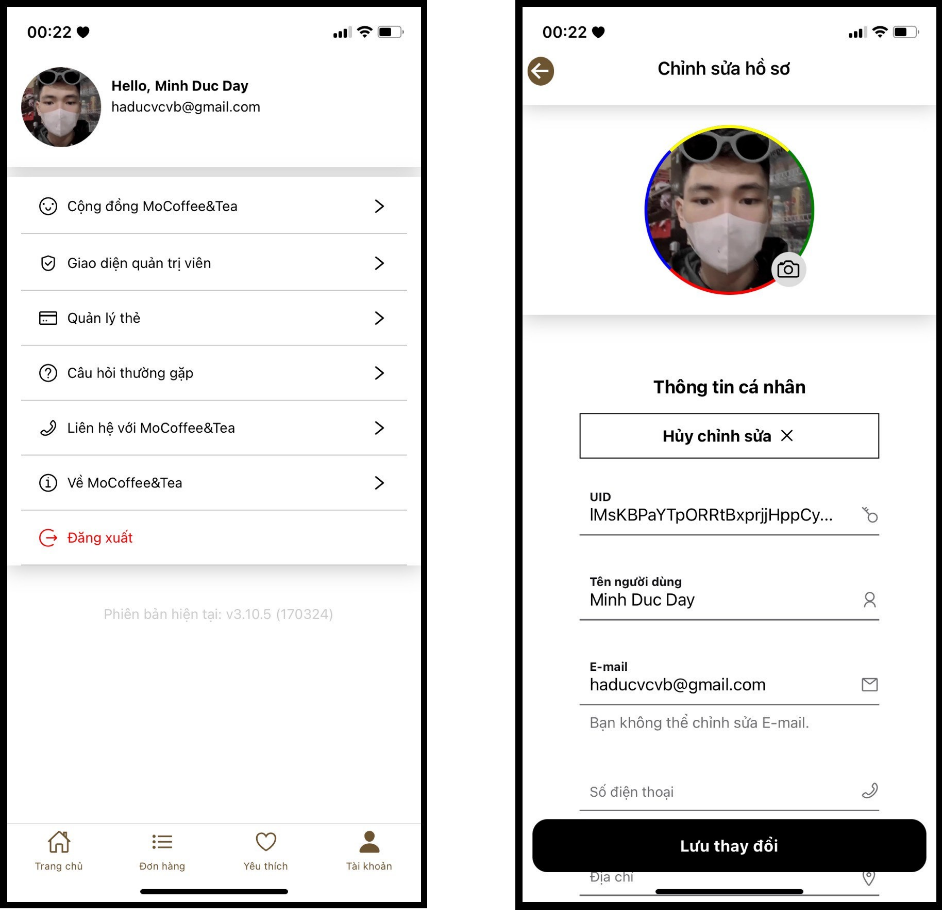
Hình 32: Giao diện đăng nhập, đăng ký



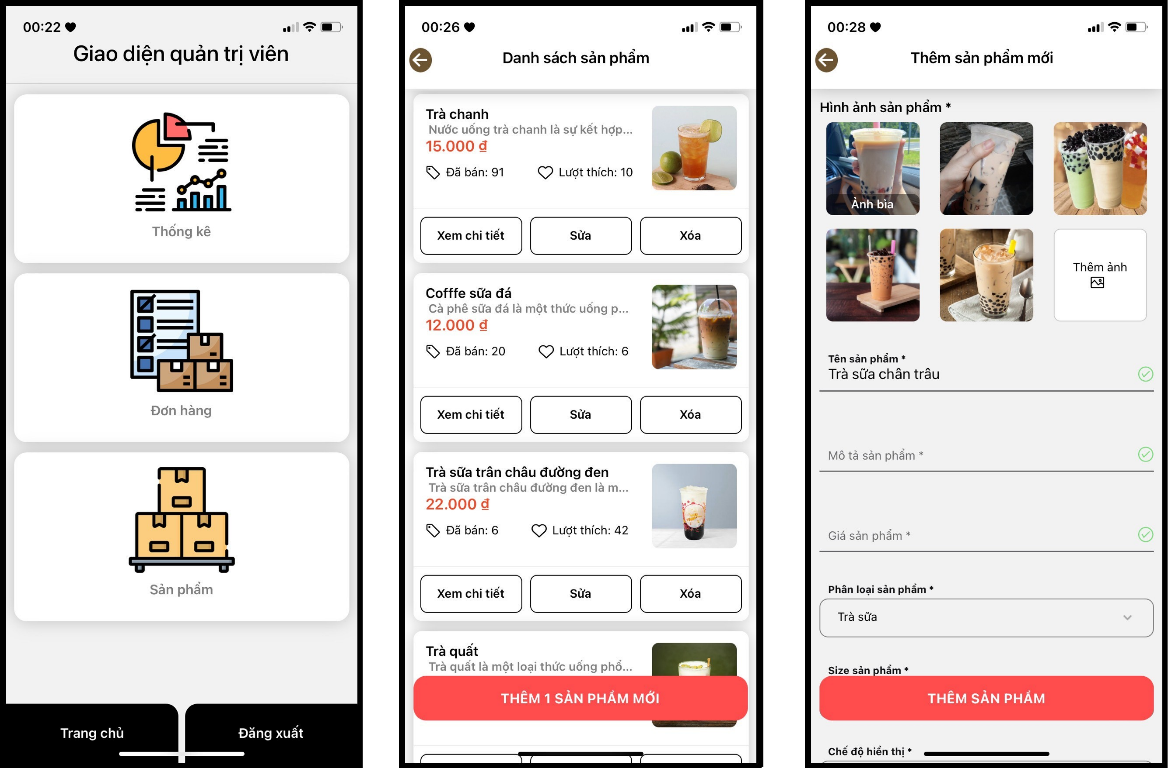
Hình 33: Giao diện trang chủ



Hình 34: Giao diện sản phẩm và chi tiết sản phẩm



Hình 35: Giao diện tài khoản và chỉnh sửa hồ sơ



Hình 36: Giao diện quản trị viên, danh sách sản phẩm



Hình 37: Giao diện giỏ hàng, đơn hàng và đơn đặt hàng

Do thời gian thực hiện phân tích và thiết kế hệ thống là tương đối hạn chế so

với một đề tài tương đối rộng và phong phú nên không tránh khỏi những thiếu

sót nhất định. Bên cạnh đó, chương trình còn một số chức năng chưa hoàn thiện

và chính xác.

Sự hạn chế trong việc tích hợp và tương thích giữa React Native và Firebase có thể gây ra một số thách thức trong quá trình phát triển và triển khai ứng dụng, đặc biệt là khi cần xử lý các tác vụ phức tạp hoặc yêu cầu tính toán lớn trên máy chủ.

Việc quản lý và bảo trì mã nguồn của ứng dụng sử dụng React Native và Firebase có thể phức tạp hơn so với việc sử dụng các công nghệ khác, đặc biệt là khi cần phải đảm bảo tính ổn định và bảo mật của hệ thống.

KẾT LUẬN

Dưới sự hướng dẫn và hỗ trợ của TS. Ngô Hữu Huy, em đã hoàn thành đồ án "Xây dựng ứng dụng đặt đồ uống cho cửa hàng Mỡ Coffee & Tea" sử dụng React Native và Firebase.

Những kết quả đạt được:

* Nắm vững kiến thức về React Native và Firebase.
* Phát triển giao diện người dùng mượt mà và thân thiện.
* Tích hợp hệ thống đặt hàng và thanh toán an toàn qua Firebase.
* Bản thân tích lũy được kinh nghiệm trong việc làm việc nhóm và dựa án.
* Ứng dụng thao tác được với những chức năng cơ bản cần thiết của một ứng dụng đặt đồ uống online.

Những thuận lợi & khó khăn:

1. Thuận lợi

* Nhiều nguồn tài liệu tham khảo.
* Nắm vững các kiến thức đã học trên trường để vận dụng vào dự án.

1. Khó khăn

* Áp lực thời gian.
* Ứng dụng gặp một số lỗi và mất nhiều thời gian để sửa.
* Một số chức năng vẫn chưa sử dụng được.
* Chưa được tối ưu

Hướng phát triển:

* Cải thiện về mặt giao diện người dùng.
* Hoàn thiện và nâng cấp các chức năng.
* Nâng cao tính bảo mật cho ứng dụng.
* Thanh toán trực tuyến bằng mã QR, ví điện tử...
* Mở rộng phạm vi ứng dụng và doanh nghiệp.
* Cải thiện trải nghiệm người dùng.

Em hy vọng sẽ tiếp tục nhận được sự hỗ trợ và hướng dẫn để phát triển ứng dụng thành công hơn trong tương lai.

TÀI LIỆU THAM KHẢO

[1] [NPMJS Docs](https://docs.npmjs.com/)

[2] [React Native Docs](https://reactnative.dev/docs/getting-started)

[3] [Firebase CRUD](https://firebase.google.com/docs/database/web/read-and-write)

[4] [React Native for Mobile Development](https://dl.ebooksworld.ir/motoman/Apress.React.Native.for.Mobile.Development.2nd.Edition.www.EBooksWorld.ir.pdf)

[5] [Firebase Authentication](https://firebase.google.com/docs/auth)

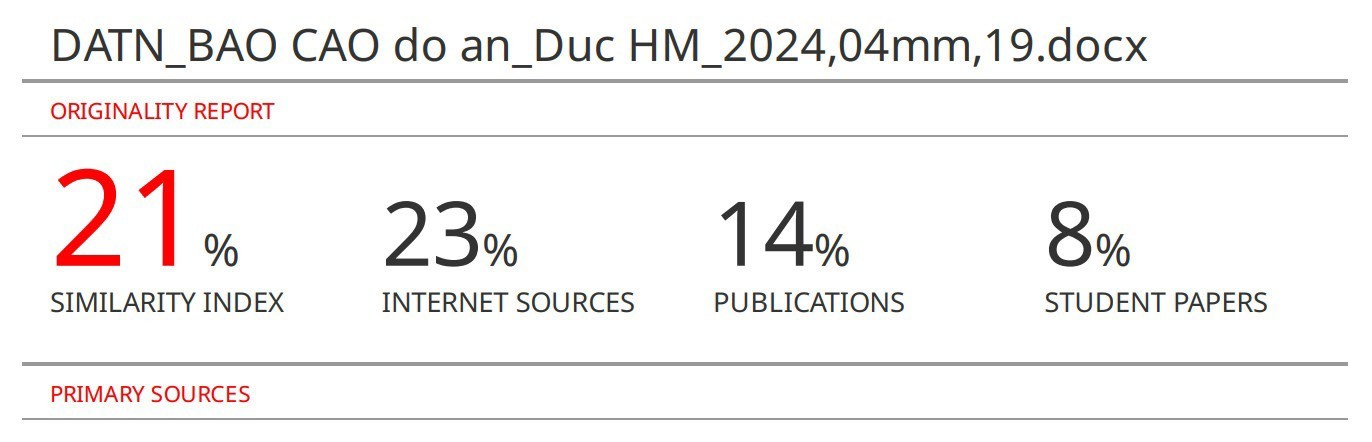
[6] [Cloud Storage for Firebase](https://firebase.google.com/docs/storage)

[7] [React Native Components & API](https://reactnative.dev/docs/components-and-apis)

[8] [Stackoverflow](https://stackoverflow.com/)

[9] [Expo Documentation](https://docs.expo.dev/)

[10] [React Redux design lession learned](https://assets.ctfassets.net/n2fn2e44lsiz/3xBFbsdYJ7DFytHolHTI3/17983f523d695e507b4d4eb9fdfe9ad2/React-Redux-Design-Lessons-Learned.pdf)



Hình 38: Kết quả kiểm tra đạo văn