

TRƯỜNG ĐẠI HỌC KỸ THUẬT - CÔNG NGHỆ CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN



LUẬN VĂN TỐT NGHIỆP

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CÁC DỊCH VỤ
CUNG CẤP CHO SINH VIÊN CỦA CTUT
TRÊN NỀN TẢNG DI ĐỘNG**

GIẢNG VIÊN HƯỚNG DẪN

Th.S Trần Thị Kim Khánh

SINH VIÊN THỰC HIỆN

Nguyễn Dương Thái Ngọc

(MSSV: 1800724)

Ngành: Hệ thống Thông tin – Khóa 2018

Cần Thơ – 2022

TRƯỜNG ĐẠI HỌC KỸ THUẬT - CÔNG NGHỆ CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN



LUẬN VĂN TỐT NGHIỆP

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CÁC DỊCH VỤ
CUNG CẤP CHO SINH VIÊN CỦA CTUT
TRÊN NỀN TẢNG DI ĐỘNG**

GIẢNG VIÊN HƯỚNG DẪN

Th.S Trần Thị Kim Khánh

SINH VIÊN THỰC HIỆN

Nguyễn Dương Thái Ngọc

(MSSV: 1800724)

Ngành: Hệ thống Thông tin - Khóa 2018

Cần Thơ - 2022

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 page.

Giảng viên hướng dẫn

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

[illegible]

Cần Thơ, ngày tháng..... năm 2022

Giảng viên phản biện

LỜI CAM ĐOAN

Em xin cam đoan: Luận văn tốt nghiệp “Xây dựng ứng dụng quản lý các dịch vụ cung cấp cho sinh viên của CTUT trên nền tảng di động” này được là quá trình tự nghiên cứu và thực hiện dưới sự hướng dẫn của giảng viên Th.S Trần Thị Kim Khánh. Đây là quá trình tự tìm hiểu, không thuê bất kì ai viết hộ, không sao chép toàn bộ hay ăn cắp ý tưởng từ các đề tài khác. Đảm bảo toàn bộ tài liệu tham khảo đều có trích dẫn rõ ràng, ghi chú đầy đủ. Nếu phát hiện bất kì sự gian dối nào em xin hoàn toàn chịu trách nhiệm.

Cần Thơ, ngày 23 tháng 06 năm 2022

Sinh viên thực hiện

Nguyễn Dương Thái Ngọc

LỜI CẢM ƠN

Đầu tiên, em xin chân thành gửi lời cảm ơn sâu sắc đến cô ThS. Trần Thị Kim Khánh đã tận tình hướng dẫn, giúp đỡ em trong suốt quá trình tìm hiểu và hoàn thiện đề tài này.

Em xin chân thành cảm ơn toàn thể giảng viên và cán bộ trường Đại học Kỹ Thuật – Công Nghệ Cần Thơ đã giảng dạy và tạo điều kiện học tập tốt nhất trong những năm qua.

Trong quá trình thực hiện, em đã cố gắng tìm hiểu và từng bước hoàn thiện đề tài nhưng do thời gian và kiến thức có giới hạn khó tránh khỏi những thiếu sót nên mong quý thầy cô thông cảm và đưa ra những đóng góp ý quý báu.

Xin chân thành cảm ơn.

DANH MỤC HÌNH

Hình 3.1 Sơ đồ use case tổng quát	22
Hình 3.2 Use case đăng nhập.....	23
Hình 3.3 Use case xem thông tin cá nhân.....	24
Hình 3.4 Use case quản lý ví.....	26
Hình 3.5 Use case Tạo hóa đơn	27
Hình 3.6 Use case Quản lý sản phẩm	29
Hình 3.7 Use case quản lý danh mục	31
Hình 3.9 Use case Quản lý tài khoản	35
Hình 3.10 Use case Quản lý dịch vụ	38
Hình 4.1 Giao diện đăng nhập	43
Hình 4.2 Chấp nhận yêu cầu quyền truy cập và chọn tài khoản email	44
Hình 4.3 Giao diện bổ sung thông tin.....	45
Hình 4.4 Giao diện đăng nhập username và password	46
Hình 4.5 Trang chủ sinh viên	47
Hình 4.6 Giao diện thông tin và chỉnh sửa thông tin cá nhân	48
Hình 4.7 Giao diện tạo hóa đơn	49
Hình 4.8 Giao diện sửa sản phẩm hóa đơn	50
Hình 4.9 Giao diện xác nhận hóa đơn và chi tiết hóa đơn	51
Hình 4.11 Giao diện ví và chi tiết giao dịch	52
Hình 4.12 Giao diện trang chủ quản lý	53
Hình 4.13 Giao diện tạo sản phẩm.....	54
Hình 4.14 Giao diện quản lý sản phẩm	55
Hình 4.15 Giao diện cập nhật hệ thống.....	56

Hình 4.16	Giao diện quản lý danh mục.....	57
Hình 4.17	Cửa sổ tạo danh mục.....	58
Hình 4.18	Giao diện chi tiết danh mục.....	59
Hình 4.19	Giao diện đơn hàng và chi tiết đơn hàng	60
Hình 4.20	Giao diện báo cáo thu nhập.....	61
Hình 4.21	Trang chủ quản trị viên và danh sách sinh viên.....	62
Hình 4.22	Giao diện danh sách nhân viên và thêm tài khoản nhân viên.....	63
Hình 4.23	Giao diện quản lý dịch vụ	64

DANH MỤC BẢNG

Bảng 1. Bảng chữ cái viết tắt.....	xi
Bảng 2. Các tác nhân hệ thống.....	20
Bảng 3. Các chức năng hệ thống.....	21
Bảng 4: Bảng cơ sở dữ liệu	42

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	i
NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN	ii
LỜI CAM ĐOAN	iii
LỜI CẢM ƠN	iv
DANH MỤC HÌNH.....	v
DANH MỤC BẢNG.....	vii
MỤC LỤC.....	viii
BẢNG VIẾT CHỮ CÁI VIẾT TẮT	xi
MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN.....	3
1.1 Giới thiệu đề tài	3
1.2 Đối tượng và phạm vi nghiên cứu	4
1.3 Phương pháp nghiên cứu	4
1.4 Ý nghĩa thực tiễn và khoa học của đề tài	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	6
2.1 Các lý thuyết được sử dụng	6
2.1.1 Lý thuyết về phân tích và thiết kế hệ thống thông tin	6
2.1.2 Lý thuyết về cơ sở dữ liệu	6
2.1.3 Mobile app.....	7
2.1.4 Cơ sở dữ liệu Firebase	7
2.2 NoSQL.....	10
2.2.1 Giới thiệu NoSQL	10
2.2.2 Các loại cơ sở dữ liệu NoSQL.....	11

2.3 Giới thiệu sơ lược về Cloud Firestore	11
2.3.1 Firestore là gì.....	11
2.3.2 Các tính năng của Firestore	11
2.4 Các công cụ hỗ trợ.....	12
2.4.1 Visual Studio Code (VS Code)	12
2.4.2 Expo.....	13
2.4.3 NodeJs	14
2.5 Các ngôn ngữ lập trình được sử dụng cho đề tài:.....	14
2.5.1 React native	14
2.5.2 Javascript	15
2.6 Giới thiệu về ví điện tử.....	16
2.6.1 Khái niệm ví điện tử.....	16
2.6.2 Lợi ích của ví điện tử.....	16
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ	18
3.1. Đặc tả hệ thống:.....	18
3.2 Đặc tả yêu cầu	19
3.2.1 Yêu cầu chức năng	19
3.2.2 Yêu cầu phi chức năng	19
3.3 Phân tích yêu cầu hệ thống.....	20
3.3.1 Các tác nhân của hệ thống.....	20
3.3.2 Các chức năng của hệ thống	21
3.3.3 Sơ đồ use case.....	22
3.3.4 Sơ đồ lớp.....	41
CHƯƠNG 4 CÀI ĐẶT VÀ HƯỚNG DẪN SỬ DỤNG PHẦN MỀM.....	43

4.1 Chức năng đăng nhập	43
4.2 Thông tin cá nhân	48
4.3 Tạo hóa đơn	49
4.4 Ví	52
4.5 Quản lý sản phẩm	53
4.5.1 Tạo sản phẩm	53
4.5.2 Cập nhật và xóa sản phẩm	56
4.5.2 Tạo danh mục	57
4.6 Quản lý đơn hàng	60
4.7 Báo cáo thu nhập:	61
4.8 Quản lý sinh viên:	62
4.9 Quản lý nhân viên	63
4.10 Quản lý dịch vụ	64
KẾT LUẬN	65
TÀI LIỆU THAM KHẢO	66

BẢNG VIẾT CHỮ CÁI VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1	CSDL	Cơ sở dữ liệu
2	HTML	HyperText Markup Language
3	NoSQL	Not Only Structured Query Language
4	SQL	Structured Query Language
5	UI	User Interface
6	XML	Extensible Markup Language

Bảng 1. Bảng chữ cái viết tắt

MỞ ĐẦU

1. Lý do chọn đề tài

Thời đại 4.0 đang bùng nổ trong cuộc sống hằng ngày của chúng ta, công nghệ thông tin xuất hiện trong gần như mọi lĩnh vực, mọi ngành nghề. Với nhiều công nghệ mới ra đời như: các trang web, công nghệ di động, điện toán đám mây, Trong số đó công nghệ di động ngày càng phát triển mạnh mẽ và rộng rãi. Có thể nói công nghệ di động là xu hướng tương lai trên toàn thế giới.

Với sự đa dụng của công nghệ di động, nhiều ứng dụng được sinh ra nhằm đáp ứng nhu cầu hàng ngày của con người trong nhiều lĩnh vực, mọi lúc, mọi nơi. Một trong những nơi đã và đang áp dụng công nghệ di động là các trường đại học. Việc tra cứu thông tin cá nhân hay thông tin các học phần bằng ứng dụng trên điện thoại không còn quá xa lạ đối với sinh viên. Bên cạnh việc quản lý các thông tin sinh viên, nhà trường còn cần phải quản lý các dịch vụ dành cho sinh viên như ăn uống, in tài liệu. Sẽ rất khó khăn khi quản lý các dịch vụ sinh viên một cách thủ công, tốn nhiều thời gian. Về phía sinh viên mỗi lần sử dụng dịch vụ đều phải trả tiền mặt và không thể xem lại các khoản phí đã chi tiêu cho các dịch vụ trong trường.

Nhằm giải quyết các vấn đề khó khăn trên em chọn đề tài “Xây dựng ứng dụng quản lý các dịch vụ cung cấp cho sinh viên của CTUT trên nền tảng di động” để quản lý các dịch vụ cho sinh viên trong trường dễ dàng và thuận tiện.

2. Mục tiêu đề tài

Đề tài luận văn tập trung thực hiện các mục tiêu sau:

- Hiểu được lý thuyết về ứng dụng moblie.
- Phát triển ứng dụng “Quản lý các dịch vụ cung cấp cho sinh viên của CTUT trên nền tảng di động” dựa trên nền tảng di động.
- Nhằm hỗ trợ những người quản lý dịch vụ trong công tác quản lý các dịch vụ cho sinh viên trong trường. Sinh viên có thể thanh toán trực tiếp trên ứng dụng

qua ví và có thể xem lại các lịch sử thanh toán cho các dịch vụ, góp phần giúp sinh viên có thể quản lý chi tiêu trong nhà trường.

3. Bố cục đề tài

Đề tài gồm có 4 chương và kết luận:

Chương 1 – TỔNG QUAN, giới thiệu sơ lược về đề tài, những phương pháp nghiên cứu cho các chủ thể trong đề tài. Chương này cũng nói về phạm vi nghiên cứu và ý nghĩa của đề tài.

Chương 2 – CƠ SỞ LÝ THUYẾT, chương này nói về các lý thuyết được sử dụng trong đề tài như: phân tích thiết kế hệ thống thông tin, các loại cơ sở dữ liệu, khái niệm về mobile app. Đồng thời giới thiệu sơ lược về các ngôn ngữ lập trình được sử dụng trong đề tài.

Chương 3 – PHÂN TÍCH THIẾT KẾ, phân tích hệ thống Quản lý các dịch vụ cung cấp cho sinh viên của CTUT trên nền tảng di động, phân tích và thiết kế cơ sở dữ liệu, trình bày các chức năng có trong hệ thống và các quy trình hoạt động của ứng dụng.

Chương 4 – CÀI ĐẶT VÀ HƯỚNG DẪN SỬ DỤNG, hướng dẫn cài đặt và sử dụng ứng dụng.

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu đề tài

Ngày nay, cùng với sự phát triển không ngừng của khoa học công nghệ, sự ra đời của công nghệ mạng không dây là một bước tiến vượt bậc trong công nghệ máy tính. Các thiết bị cầm tay ra đời ứng dụng công nghệ mạng không dây cũng có một ý nghĩa lớn đối với đời sống con người. Với các thiết bị cầm tay làm tăng tính tiện dụng trong nhiều hoạt động thực tiễn như giao dịch thương mại, trao đổi học tập, quản lý tài liệu, ... Việc nghiên cứu và phát triển các ứng dụng trên các thiết bị cầm tay là một hướng mới, và có nhiều cách tiếp cận khác nhau.

Những ứng dụng trên thiết bị cầm tay hay mobile trong lĩnh vực quản lý là những ứng dụng vô cùng quan trọng. Nó không những giải phóng công sức cho những người quản lý mà còn đem lại sự nhanh nhạy và chính xác. Đặc biệt, trong lĩnh vực giáo dục của các trường đại học thì việc quản lý một lượng thông tin khổng lồ của toàn bộ sinh viên là khó tránh khỏi, bên cạnh đó còn những thông tin về các dịch vụ dành cho sinh viên trong trường. Với lượng thông tin khổng lồ trên nếu quản lý bằng thủ công sẽ rất khó khăn, tốn thời gian và thiếu chính xác. Vì vậy xây dựng một ứng dụng quản lý các dịch vụ dành cho sinh viên sẽ giải quyết được những vấn đề đã kể trên.

Với ứng dụng quản lý dịch vụ dành cho sinh viên sẽ giúp các nhà quản lý dễ dàng quản lý thông tin dịch vụ của mình một cách dễ dàng và nhanh gọn. Thông qua ứng dụng, nhà quản lý có thể thêm sản phẩm vào kho để theo dõi số lượng sản phẩm, xem báo cáo thu chi trong khoảng thời gian chỉ định và xem được số lượng đơn hàng đã bán. Đồng thời ứng dụng cũng giúp sinh viên có thể thanh toán trực tiếp trên ứng dụng khi quét mã QR của từng sản phẩm mà không cần tiền mặt, có thể xem chi tiết các lịch sử giao dịch từ đó góp phần quản lý chi tiêu của mình. Về phía quản trị viên có thể quản lý số lượng người đăng ký dịch vụ trong trường, xem thông tin nhân viên quản lý và sinh viên.

Trong đề tài “Xây dựng ứng dụng quản lý các dịch vụ cung cấp cho sinh viên của CTUT trên nền tảng di động” này sẽ giải quyết những chức năng đã kể trên nhằm giúp nhà trường, nhân viên quản lý và sinh viên có thể quản lý thông tin của mình.

1.2 Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Các dịch vụ cung cấp cho sinh viên tại trường Đại học Kỹ Thuật – Công Nghệ Cần Thơ.
- Cách tổ chức và lưu trữ dữ liệu trên Cloud Firestore.
- Một số công cụ hỗ trợ xây dựng ứng dụng di động.

Phạm vi nghiên cứu:

- Mô phỏng dữ liệu cho thông tin sinh viên và các dịch vụ dành cho sinh viên tại trường Đại học Công Nghệ - Kỹ Thuật Cần Thơ.
- Nghiên cứu về cách xây dựng ứng dụng dựa trên các yêu cầu đã đề ra.
- Công cụ hỗ trợ xây dựng ứng dụng di động là Expo và Visual Studio Code (VS Code).

1.3 Phương pháp nghiên cứu

Nghiên cứu lý thuyết: Tổng hợp các vấn đề lý thuyết liên quan từ các nguồn tài liệu như Giáo trình, Bài giảng, Internet, chuyên đề, luận văn và luận án.

Nghiên cứu thực nghiệm:

- Nghiên cứu cài đặt, thực hành công cụ hỗ trợ xây dựng ứng dụng moblie.
- Mô tả kết quả thu được có ý nghĩa thực tiễn phù hợp với đề tài.

1.4 Ý nghĩa thực tiễn và khoa học của đề tài

Cùng với sự phát triển mạnh mẽ của công nghệ thông tin và sự xâm nhập nhanh chóng của tin học vào mọi lĩnh vực của đời sống xã hội thì việc áp dụng công nghệ thông tin vào trong công tác quản lý đã trở thành một nhu cầu tất yếu, nó là một trong

những yếu tố không thể thiếu nhằm nâng cao chất lượng và hiệu quả trong công tác quản lý.

Trong lĩnh vực quản lý các dịch vụ dành cho sinh viên việc điều chỉnh và bổ sung thông tin thực hiện rất khó khăn và không rõ ràng, việc tìm kiếm thông tin mất thời gian, độ chính xác kém.

Do đó xây dựng phần mềm quản lý dịch vụ cho sinh viên rất cần thiết. Phần mềm này sẽ giúp cho người quản lý thoát khỏi lao động thủ công, nâng cao hiệu quả của công việc, tiết kiệm được rất nhiều thời gian.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Các lý thuyết được sử dụng

2.1.1 Lý thuyết về phân tích và thiết kế hệ thống thông tin

Khái niệm hệ thống thông tin:

- Hệ thống thông tin là tập hợp người, thủ tục và các nguồn lực để thu thập, xử lý và phát thông tin trong một tổ chức.
- Hệ thống thông tin có thể là thủ công nếu dựa vào giấy, bút.
- Hệ thống thông tin hiện đại là hệ thống tự động hóa dựa vào máy tính (phần cứng và phần mềm) và các công nghệ thông tin khác.

Phân tích hệ thống:

- Là một quá trình thu thập và diễn giải các dữ kiện, xác định các vấn đề và phân rã một hệ thống thành các thành phần của nó.
- Phân tích hệ thống được thực hiện với mục đích nghiên cứu một hệ thống hoặc các bộ phận của nó để xác định các mục tiêu của nó. Đây là một kỹ thuật giải quyết vấn đề nhằm cải thiện hệ thống và đảm bảo rằng tất cả các thành phần của hệ thống hoạt động hiệu quả để hoàn thành mục đích của chúng.

Thiết kế hệ thống:

- Nó là một quá trình lập kế hoạch một hệ thống mới hoặc thay thế một hệ thống hiện có bằng cách xác định các thành phần hoặc mô-đun của nó để thỏa mãn các yêu cầu cụ thể.

2.1.2 Lý thuyết về cơ sở dữ liệu

Khái niệm cơ sở dữ liệu: Một cơ sở dữ liệu (CSDL) là một tập hợp các dữ liệu có liên quan với nhau, được lưu trữ trên máy tính, có nhiều người sử dụng và được tổ chức theo một mô hình. Hay nói cách khác, CSDL là một bộ các dữ liệu tác nghiệp được lưu trữ lại và được các hệ ứng dụng của một đơn vị cụ thể nào đó sử dụng.

Cơ sở dữ liệu quan hệ: là cơ sở dữ liệu dựa trên mô hình dữ liệu quan hệ, mô hình này lưu trữ dữ liệu dưới dạng hàng và cột, và cùng nhau tạo thành một bảng (quan hệ). Cơ sở dữ liệu quan hệ sử dụng SQL để lưu trữ, thao tác cũng như duy trì dữ liệu. Edgar F. Codd đã phát minh ra cơ sở dữ liệu vào năm 1970. Mỗi bảng trong cơ sở dữ liệu mang một khóa làm cho dữ liệu trở thành duy nhất so với những bảng khác.

Cơ sở dữ liệu phi quan hệ: là một loại cơ sở dữ liệu được sử dụng để lưu trữ một loạt các tập dữ liệu. Nó không phải là cơ sở dữ liệu quan hệ vì nó lưu trữ dữ liệu không chỉ ở dạng bảng mà còn theo nhiều cách khác nhau. Nó ra đời khi nhu cầu xây dựng các ứng dụng hiện đại ngày càng tăng.

2.1.3 Mobile app

Mobile app là một chương trình ứng dụng dành riêng cho thiết bị di động như tablet hay smartphone cho phép người dùng có thể sử dụng để truy cập vào nội dung mà người dùng mong muốn ngay trên thiết bị điện thoại di động đó. Những nhà lập trình mobile app sẽ biến nó trở thành một không gian lớn được thu nhỏ để người dùng có thể giải trí, xem tin tức, mua sắm,...

Phân loại mobile app:

- Native Mobile App: Đây là loại ứng dụng mà chúng ta sử dụng để tải những nội dung mong muốn xuống thiết bị điện thoại di động và sẽ sử dụng chúng bằng cách truy cập.
- Hybrid Mobile App: Hybrid Mobile App là một ứng dụng lai kết hợp được những điểm đặc trưng nổi bật giữa web app và native app. Web ứng dụng này sử dụng một số công nghệ web như CSS3, Javascript, hay HTML5 và được bao bọc bởi một lớp vỏ container ở bên ngoài để có thể tương tự như native mobile app và được tải về trên kho ứng dụng.

2.1.4 Cơ sở dữ liệu Firebase

Firebase là một cơ sở dữ liệu sở hữu bởi Google giúp chúng ta phát triển các ứng dụng di động và web. Họ cung cấp rất nhiều công cụ và dịch vụ tiện ích để phát

triển ứng dụng nên một ứng dụng chất lượng. Điều đó rút ngắn thời gian phát triển và giúp ứng dụng sớm ra mắt với người dùng.

Firebase cung cấp cho người dùng các dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây với hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính của Firebase là giúp người dùng lập trình ứng dụng, phần mềm trên các nền tảng web, di động bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Với Firebase, chúng ta có thể tạo ra những ứng dụng real-time như app chat, cùng nhiều tính năng như xác thực người dùng, Cloud Messaging, ... Người dùng có thể dùng Firebase giống như phần backend của app.

Các dịch vụ của Firebase hoàn toàn miễn phí, tuy nhiên người dùng cần phải trả thêm tiền nếu muốn nâng cấp lên. Điều này nên cân nhắc nếu muốn xây dựng một ứng dụng lớn sử dụng phần backend là Firebase, vì cái giá khi muốn nâng cấp còn khá đắt so với việc xây dựng backend truyền thống.

- Ưu điểm của Firebase:
 - + Tạo tài khoản và sử dụng dễ dàng: Firebase cho phép người dùng đăng nhập bằng tài khoản Google đơn giản. Gói Spark của Firebase miễn phí và cung cấp nhiều tính năng để giúp các nhà phát triển sử dụng.
 - + Tốc độ phát triển nhanh: Firebase là một tùy chọn phát triển ứng dụng phù hợp có thể giúp các nhà phát triển tiết kiệm thời gian và giảm thời gian tiếp thị ứng dụng
 - + Có nhiều dịch vụ trong một nền tảng: Firebase cũng cung cấp danh sách đầy đủ các sản phẩm để hỗ trợ các Developer trong quá trình phát triển với hai tùy chọn cơ sở dữ liệu là Firestore và Realtime Database.
 - + Được cung cấp bởi Google: Firebase được cung cấp bởi Google, một trong những tên tuổi nổi bật và đáng tin cậy nhất trong thế giới công nghệ.
 - + Tập trung vào phát triển giao diện người dùng: Một ưu điểm khác khiến Firebase được yêu thích đó là nó cho phép Developer tập trung vào việc tạo mã

Frontend cho các ứng dụng di động. Nền tảng này giúp cho việc phát triển ứng dụng trở nên thuận tiện và giúp giảm chi phí đáng kể.

- + **Firestore không có máy chủ:** Firestore cung cấp một môi trường hoàn toàn không có máy chủ. Firestore đi kèm với kiến trúc không máy chủ (Serverless Architecture). Do đó, sẽ không cần lo lắng về cơ sở hạ tầng máy chủ.

- + **Học máy (Machine Learning):** Firestore đi kèm với bộ ML với các API sẵn có cho các tính năng khác nhau của nền tảng di động như nhận dạng văn bản, nhận diện khuôn mặt, ghi nhãn hình ảnh, quét mã vạch, ...

- + **Tạo lưu lượng truy cập:** Firestore hỗ trợ lập chỉ mục ứng dụng để cho phép người dùng thu hút lại những khách tìm kiếm trên Google. Cụ thể, nó cung cấp liên kết ứng dụng trên mục tìm kiếm của Google.

- + **Theo dõi lỗi:** Firestore có thể theo dõi cả lỗi không nghiêm trọng và lỗi nghiêm trọng. Đồng thời, báo cáo được tạo dựa trên mức độ ảnh hưởng của lỗi đến trải nghiệm của người dùng.

- + **Sao lưu:** Firestore đảm bảo tính bảo mật tối ưu và tính sẵn có của dữ liệu nhờ các bản sao lưu thường xuyên. Các ứng dụng được bảo vệ khỏi mọi khả năng mất dữ liệu bằng cách dựa vào tính năng sao lưu tự động của nền tảng này.

- Hạn chế của Firestore:

- + **Không phải mã nguồn mở:** Firestore không phải là một tùy chọn mã nguồn mở để phát triển ứng dụng di động. Điều này khiến nó trở thành một lựa chọn không tối ưu cho nhiều nhà phát triển.

- + **Người dùng không có quyền truy cập mã nguồn:** Đây là một hạn chế lớn khác của Firestore cho người dùng. Việc không có quyền truy cập vào mã nguồn có thể khá khó khăn đối với một số nhà phát triển.

- + **Firestore không hoạt động ở nhiều quốc gia:** Firestore là một Subdomain của Google. trang Web chính thức của nó là <https://firebase.google.com> và bị chặn ở nhiều quốc gia.

+ **Chỉ hoạt động với cơ sở dữ liệu NoSQL:** Cả Firestore và Firebase Realtime Database đều cung cấp cấu trúc NoSQL. Chúng không có tùy chọn nào để sử dụng cơ sở dữ liệu quan hệ.

+ **Truy vấn chậm:** Không phải tất cả cơ sở dữ liệu đều lý tưởng cho mọi trường hợp sử dụng và Cloud Firestore cũng không ngoại lệ. Kích thước yêu cầu API tối đa 10 MiB, không có truy vấn tổng hợp gốc, giới hạn kích thước tài liệu là 1 MiB, kết nối đồng thời tối đa 1M, truy vấn hiệu suất chậm.

+ **Không phải tất cả các dịch vụ Firebase đều miễn phí:** API Cloud Vision trong tính năng học máy cũng không khả dụng trong gói Spark miễn phí của nền tảng này. Dịch vụ này chỉ có sẵn cho người dùng của gói Blaze và tính phí 1,5 USD/ k API.

2.2 NoSQL

2.2.1 Giới thiệu NoSQL

NoSQL là cơ sở dữ liệu được xây dựng dành riêng cho mô hình dữ liệu và có sơ đồ linh hoạt để xây dựng các ứng dụng hiện đại, dữ liệu lớn và ứng dụng nền web thời gian thực.

Cơ sở dữ liệu NoSQL sử dụng nhiều mô hình dữ liệu để truy cập và quản lý dữ liệu. Các loại cơ sở dữ liệu này được tối ưu hóa dành riêng cho các ứng dụng yêu cầu mô hình dữ liệu linh hoạt có lượng dữ liệu lớn và độ trễ thấp, có thể đạt được bằng cách giảm bớt một số hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác.

Ưu điểm của NoSQL:

- Khả năng mở rộng cao: Cơ sở dữ liệu NoSQL được thiết kế để mở rộng quy mô bằng cách thêm máy chủ khi nhu cầu cho khối lượng công việc gia tăng (hay được gọi là mở rộng quy mô kiểu scaling out).
- Hiệu năng cao: Cơ sở dữ liệu NoSQL được tối ưu hóa theo các mô hình dữ liệu cụ thể và các mẫu truy cập giúp tăng hiệu năng cao hơn so với việc cố gắng đạt được mức độ chức năng tương tự bằng cơ sở dữ liệu quan hệ.

2.2.2 Các loại cơ sở dữ liệu NoSQL

Có bốn loại cơ sở dữ liệu NoSQL thường được sử dụng:

- Cơ sở dữ liệu Key-value: là cơ sở dữ liệu NoSQL đơn giản nhất. Mỗi mục trong cơ sở dữ liệu được lưu trữ dưới dạng tên thuộc tính (hoặc 'khóa'), cùng với giá trị của nó.
- Cơ sở dữ liệu Document: Cơ sở dữ liệu tài liệu lưu trữ các giá trị dưới dạng tài liệu (*document*). Dữ liệu sẽ được lưu trữ dưới dạng BSON/JSON/XML dưới database.
- Cơ sở dữ liệu Column family: Dữ liệu được lưu trong database dưới dạng các cột, thay vì các hàng như SQL. Mỗi hàng sẽ có một key/id riêng. Điểm đặc biệt là các hàng trong một bảng sẽ có số lượng cột khác nhau.
- Cơ sở dữ liệu Graph: Dữ liệu trong graph database được lưu dưới dạng các node. Mỗi node sẽ có 1 label, 1 số properties như một row trong SQL. Các node này được kết nối với nhau bằng các relationship. Graph database tập trung nhiều vào relationship giữa các node, áp dụng nhiều thuật toán duyệt node để tăng tốc độ.

2.3 Giới thiệu sơ lược về Cloud Firestore

2.3.1 Firestore là gì

Firestore là cơ sở dữ liệu có thể mở rộng NoSQL và là một phần của Firebase, được phát triển cho các ứng dụng di động. Nó là với mô hình dữ liệu mới và trực quan hơn từ sự kế thừa của Realtime Database. Trong đó, Cloud Firestore phong phú, nhanh và có khả năng mở rộng vượt trội hơn so với Realtime Database.

2.3.2 Các tính năng của Firestore

Tính linh hoạt: Cloud Firestore hỗ trợ các cấu trúc dữ liệu linh hoạt, phân cấp dữ liệu. Lưu trữ dữ liệu của trong các document, được tổ chức thành các collection. Các document có thể chứa các đối tượng phức tạp.

Truy vấn tương trưng: có thể sử dụng các truy vấn để truy xuất các document riêng lẻ hoặc để truy xuất tất cả các document trong collection khớp với các tham số truy vấn của người dùng. Các truy vấn có thể bao gồm nhiều bộ lọc, kết hợp giữa bộ lọc và sắp xếp.

Cập nhật thời gian thực: Cloud Firestore sử dụng đồng bộ hóa dữ liệu để cập nhật dữ liệu trên mọi thiết bị được kết nối. Nó cũng được thiết kế để thực hiện các truy vấn tìm nạp một lần.

Hỗ trợ offline: Cloud Firestore lưu trữ dữ liệu tại local, vì vậy ứng dụng có thể viết, đọc, nghe và truy vấn dữ liệu ngay cả khi thiết bị ngoại tuyến. Khi thiết bị trở lại trực tuyến, Cloud Firestore sẽ đồng bộ hóa mọi thay đổi cục bộ lên Cloud Firestore.

Khả năng mở rộng: Cloud Firestore mang đến cơ sở hạ tầng mạnh mẽ tốt nhất của Google Cloud: sao chép dữ liệu đa vùng tự động, đảm bảo tính nhất quán mạnh mẽ, hoạt động hàng loạt và hỗ trợ giao dịch thực.

2.4 Các công cụ hỗ trợ

2.4.1 Visual Studio Code (VS Code)

Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

Một số tính năng của Visual Studio Code:

- Hỗ trợ nhiều ngôn ngữ lập trình: Visual Studio Code hỗ trợ nhiều ngôn ngữ lập trình như C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript, ... Vì vậy, nó dễ dàng phát hiện và đưa ra thông báo nếu chương trình có lỗi.
- Hỗ trợ đa nền tảng: Các trình viết code thông thường chỉ được sử dụng hoặc cho Windows hoặc Linux hoặc Mac Systems. Nhưng Visual Studio Code có thể hoạt động tốt trên cả ba nền tảng trên.
- Cung cấp kho tiện ích mở rộng: Trong trường hợp lập trình viên muốn sử dụng một ngôn ngữ lập trình không nằm trong số các ngôn ngữ Visual Studio hỗ trợ,

họ có thể tải xuống tiện ích mở rộng. Điều này vẫn sẽ không làm giảm hiệu năng của phần mềm, bởi vì phần mở rộng này hoạt động như một chương trình độc lập.

- Kho lưu trữ an toàn: Đi kèm với sự phát triển của lập trình là nhu cầu về lưu trữ an toàn. Với Visual Studio Code, người dùng có thể hoàn toàn yên tâm vì nó dễ dàng kết nối với Git hoặc bất kỳ kho lưu trữ hiện có nào.
- Hỗ trợ web: Visual Studio Code hỗ trợ nhiều ứng dụng web. Ngoài ra, nó cũng có một trình soạn thảo và thiết kế website.
- Lưu trữ dữ liệu dạng phân cấp: Phần lớn tệp lưu trữ đoạn mã đều được đặt trong các thư mục tương tự nhau. Ngoài ra, Visual Studio Code còn cung cấp các thư mục cho một số tệp đặc biệt quan trọng.
- Hỗ trợ thiết bị đầu cuối: Visual Studio Code có tích hợp thiết bị đầu cuối, giúp người dùng khỏi phải chuyển đổi giữa hai màn hình hoặc trở về thư mục gốc khi thực hiện các thao tác.
- Màn hình đa nhiệm: Người dùng Visual Studio Code có thể mở cùng lúc nhiều tệp tin và thư mục – mặc dù chúng không hề liên quan với nhau.
- Hỗ trợ git: Visual Studio Code hỗ trợ kéo hoặc sao chép mã trực tiếp từ GitHub. Mã này sau đó có thể được thay đổi và lưu lại trên phần mềm.

2.4.2 Expo

Expo là một framework dùng để phát triển nhanh các ứng dụng React Native. Expo cung cấp một lớp nằm trên cùng của React Native API để giúp chúng dễ sử dụng và quản lý. Nó còn cung cấp các công cụ giúp người dùng dễ dàng khởi tạo và kiểm thử các ứng dụng React Native. Sau cùng, nó cung cấp các thành phần UI và các dịch vụ thường chỉ có sẵn khi người dùng cài đặt một thành phần React Native của bên thứ ba. Tất cả đều được cung cấp thông qua Expo SDK.

2.4.3 NodeJs

Node.js là một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, sử dụng kỹ thuật điều khiển theo sự kiện, nhập/xuất không đồng bộ để tối thiểu tổng chi phí và tối đa khả năng mở rộng. Node.js bao gồm có V8 JavaScript engine của Google, libUV, và vài thư viện khác.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS. NodeJS cung cấp rất nhiều thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

2.5 Các ngôn ngữ lập trình được sử dụng cho đề tài:

2.5.1 React native

React Native là framework được phát triển bởi Facebook, cho phép các Developer (DEV) sử dụng JavaScript để làm mobile apps trên cả Android và iOS với cảm nhận và giao diện native. React Native giải quyết được vấn đề về hiệu suất trên Hybrid và tối ưu được chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.

Ưu điểm của React native:

- Có thể tái sử dụng code: Thông thường các DEV phải viết và tạo ra các ứng dụng riêng biệt với các nền tảng khác nhau, nhưng bây giờ, khi sử dụng React Native thì lập trình viên có thể tái sử dụng lên đến 90% các đoạn code. Điều này cho phép DEV tiết kiệm tối đa thời gian trong khi phát triển các ứng dụng đa nền tảng.
- Hiệu năng ổn định: React Native là công cụ xây dựng ứng dụng ít native code nhất cho nhiều hệ điều hành khác nhau. Thực tế nhiều ứng dụng sử dụng lập trình bằng React Native đã chứng minh framework này có tính ổn định cao hơn hẳn so với các công cụ khác hiện có.
- Cộng đồng người dùng lớn: Một cộng đồng lớn cho phép người dùng thoải mái chia sẻ, học hỏi kinh nghiệm và giao lưu kỹ năng để tạo ra những sản phẩm

hoàn chỉnh hơn. React Native có một cộng đồng người dùng cực lớn trên toàn thế giới, họ sẵn sàng hỗ trợ lập trình viên trong bất kỳ trường hợp nào.

- Tối ưu thời gian và chi phí: React Native được phát triển bởi Facebook. Và cũng như tất cả sản phẩm khác của công ty này, React Native thực sự hoạt động tối ưu trên nhiều nền tảng và công nghệ khác nhau. Điều đó giúp các lập trình viên tiết kiệm được thời gian và chi phí đáng kể. Bên cạnh đó React Native cũng giúp tăng trải nghiệm người dùng trên các ứng dụng di động.
- Giao diện người dùng và chế độ xem tùy chỉnh: React Native sử dụng thư viện React.js, cho phép nó bao gồm nhiều trường hợp. Nó không có một tập hợp đầy đủ các thành phần gốc, nhưng chúng có thể được thay thế bằng các plugin cộng đồng. Một trong những giải pháp để khắc phục các hạn chế về giao diện người dùng là sử dụng Expo, một nền tảng với các thành phần và plugin giao diện người dùng được cài đặt sẵn cho các ứng dụng dựa trên React Native. Nhưng khi sử dụng Expo, điều quan trọng cần nhớ là không thể sử dụng các plugin gốc cùng với các plugin Expo.

Hạn chế của React native:

- Hiệu năng còn chưa sánh bằng với Native app: Do native app được tạo ra bằng ngôn ngữ và các tính năng có sẵn trên hệ điều hành đó. Bởi vậy các ứng dụng có thể vận hành với tốc độ cao nhất mà không cần thông qua bất kì ứng dụng bên thứ ba hay engine nào.
- Không thích hợp với các app cần khả năng tính toán cao.

2.5.2 Javascript

Khái niệm:

- Javascript là ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa. Javascript được sử dụng rộng rãi trong các ứng dụng Website và hỗ trợ hầu hết các trình duyệt như Firefox, Chrome,... trên máy tính lẫn điện thoại di động.

- Javascript có thể cập nhật hoặc thay đổi các thuộc tính CSS và các thẻ HTML một cách dễ dàng. Hay nói cách khác, Javascript là ngôn ngữ lập trình trên trình duyệt ở phía client. Tuy nhiên, hiện nay với sự xuất hiện của NodeJS đã giúp Javascript có thể làm việc ở backend.

JavaScript Framework: JavaScript Framework là một bộ thư viện được xây dựng dựa vào ngôn ngữ lập trình JavaScript. Từ đó, mỗi framework được tạo ra để phục vụ cho từng lĩnh vực khác nhau. Hiện nay, có rất nhiều JavaScript Framework thông dụng như:

- Reactjs: Thư viện dùng cho ứng dụng mobile.
- Node.js: Dùng để xây dựng và phát triển ứng dụng realtime từ phía máy chủ.
- Angular: Dùng để xây dựng ứng dụng Single Page....

Client-side JavaScript: Client-side JavaScript là form được sử dụng phổ biến hiện nay. Một trang web không cần là một HTML tĩnh, nhưng có thể bao gồm các chương trình tương tác được với người dùng hay điều khiển trình duyệt và tạo nội dung HTML động.

2.6 Giới thiệu về ví điện tử

2.6.1 Khái niệm ví điện tử

Ví điện tử hay còn được gọi là Ví tiền Online là một tài khoản thanh toán các giao dịch trực tuyến phổ biến nhất hiện nay như: Thanh toán tiền điện, nước, học phí, nạp tiền điện thoại, mua vé xem phim,....

Chức năng hoạt động của ví điện tử thực hiện bằng cách liên kết tài khoản ngân hàng, nạp tiền vào ví và thanh toán bất kỳ dịch vụ có liên kết một cách đơn giản, tiện lợi. Chỉ cần có điện thoại hay laptop có kết nối internet và một tài khoản đã được đăng ký chúng ta có thể thực hiện giao dịch mọi lúc mọi nơi mà không cần đến tiền mặt.

2.6.2 Lợi ích của ví điện tử

Ví điện tử hiện nay được sử dụng phổ biến trên toàn thế giới với các thương hiệu như Moneybooker, WebMoney,... Còn ở Việt Nam, hình thức thanh toán này cũng đang dần trở nên bùng nổ hơn. Các đơn vị tham gia phát triển và khách hàng đang ngày

càng tin tưởng, tin dùng nhiều hơn. Bởi lẽ ví tiền điện tử mang đến cho người sở hữu nhiều lợi ích vô cùng đặc biệt. Trong đó chẳng hạn như:

- Đảm bảo tiện lợi cho người dùng: Ví điện tử khi mang theo có thể giúp người dùng hạn chế số lượng thẻ và tiền mặt. Tất cả những gì người dùng cần làm là sử dụng ví để thanh toán các mặt hàng bằng cách quét mã hay chạm vào thiết bị thanh toán. Vì vậy quá trình thanh toán được rút ngắn đáng kể và tiết kiệm tối đa thời gian chờ đợi.
- Bảo mật tốt hơn: Ví tiền điện tử giống như một thẻ ghi nợ trong quá trình giao dịch. Ngay khi thực hiện bất kỳ thanh toán nào, người dùng cũng phải cần nhập mã PIN để ủy quyền thanh toán. Một số thiết bị tích hợp sinh trắc học cần có dấu vân tay thì mới thanh toán được.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

3.1. Đặc tả hệ thống:

Ứng dụng quản lý các dịch vụ cho sinh viên của CTUT được xây dựng dành cho sinh viên CTUT, các nhân viên quản lý dịch vụ và một quản trị viên. Ứng dụng tập trung vào giải quyết các chức năng như: quản lý thông cá nhân, quản lý các sản phẩm và danh mục có trong dịch vụ, tìm kiếm và xem chi tiết lịch sử các giao dịch.

Đối với sinh viên, sau khi đăng nhập thành công vào ứng dụng bằng gmail trường có thể sử dụng các chức năng dành cho sinh viên như: quản lý thông tin cá nhân, tạo hóa đơn để mua các sản phẩm tại mỗi điểm dịch vụ bằng cách quét mã QR có trên sản phẩm, thanh toán trực tiếp trên ví mà không cần tiền mặt, số tiền sau khi sinh viên thanh toán sẽ được chuyển tới ví của nhân viên quản lý dịch vụ tương ứng đồng thời trừ đi số lượng sản phẩm còn lại trong kho theo hóa đơn. Sau khi thanh toán thành công ứng dụng sẽ chuyển đến trang chi tiết hóa đơn, tại đây sinh viên và nhân viên quản lý có thể đối chiếu là đã thanh toán thành công hay chưa qua mã hóa đơn trên tranh chi tiết hóa đơn của sinh viên và trang chi tiết hóa đơn của nhân viên quản lý, khi đã xác nhận mã trùng khớp với nhau thì sinh viên được quyền mang những món đồ đã mua về. Sinh viên có thể xem lại danh sách lịch sử giao dịch và chi tiết giao dịch để quản lý chi tiêu tốt hơn tại giao diện ví của ứng dụng, với chức năng quản lý ví sinh viên có thể nạp tiền vào ví một cách nhanh chóng và dễ dàng.

Đối với nhân viên quản lý dịch vụ, sau khi đăng nhập vào hệ thống bằng tài khoản đã được quản trị viên cấp trước đó, có thể sử dụng các chức năng dành cho nhân viên quản lý như: quản lý các sản phẩm của dịch vụ, quản lý danh mục và các sản phẩm có trong danh mục. Nhân viên quản lý có thể tìm kiếm và xem chi tiết hóa đơn bán hàng thông qua ứng dụng, đồng thời có thể kiểm tra số dư trong ví của mình khi sinh viên thanh toán các hóa đơn tại quầy dịch vụ của nhân viên quản lý.

Đối với quản trị viên hệ thống, sau khi đăng nhập bằng tài khoản quản trị, có thể sử dụng các chức năng dành cho quản trị viên như: xem danh sách sinh viên đã đăng ký ứng dụng, quản lý các tài khoản nhân viên quản lý có trong hệ thống và quản lý các dịch vụ dành cho sinh viên tại trường.

3.2 Đặc tả yêu cầu

3.2.1 Yêu cầu chức năng

Ứng dụng thực hiện các chức năng như sau:

- Xem thông tin cá nhân: chức năng này giúp sinh viên có thể xem thông tin cá nhân và cập nhật lại thông tin cá nhân.
- Tạo hóa đơn: chức năng này giúp sinh viên có thể tạo một hóa đơn tạm thời được lưu trên cơ sở dữ liệu và thanh toán hóa đơn bằng số tiền có trong ví.
- Quản lý ví: chức năng này giúp nạp tiền vào tài ví và xem danh sách lịch sử giao dịch.
- Quản lý sản phẩm: chức năng này giúp nhân viên quản lý có thể thêm, sửa, xóa sản phẩm của dịch vụ.
- Quản lý tài khoản: giúp quản trị viên xem được danh sách sinh viên đã đăng ký ứng dụng và thêm, sửa, xóa tài khoản của nhân viên quản lý.
- Quản lý dịch vụ: giúp quản trị viên thêm, sửa hoặc xóa các dịch vụ trong trường.
- Quản lý hóa đơn: chức năng này có thể tìm kiếm hóa đơn theo ngày và xem chi tiết hóa đơn đã chọn.
- Quản lý danh mục: chức năng này có thể thêm, sửa, xóa danh mục, cũng có thể xóa sản phẩm có trong danh mục.




3.2.2 Yêu cầu phi chức năng

Ứng dụng đáp ứng những yêu cầu phi chức năng sau:

- Giao diện: dễ nhìn, thân thiện, dễ sử dụng.
- Phản hồi nhanh chóng, chính xác.
- Thời gian vận hành: Vận hành 24/24.

3.3 Phân tích yêu cầu hệ thống

3.3.1 Các tác nhân của hệ thống

STT	Tác nhân	Ý nghĩa, nhiệm vụ
1	 Sinh Viên	<p>Tác nhân sinh viên là người truy cập ứng dụng, có thể quản lý thông tin cá nhân, khởi tạo và thanh toán hóa đơn.</p> <p>Thực hiện các chức năng: Tạo hóa đơn, xem thông tin cá nhân, quản lý ví.</p>
2	 Nhân viên quản lý	<p>Tác nhân quản lý là người quản lý các sản phẩm mình kinh doanh, theo dõi và quản lý doanh thu.</p> <p>Thực hiện các chức năng:</p> <ul style="list-style-type: none">- Quản lý sản phẩm gồm: thêm sản phẩm, xóa sản phẩm, cập nhật sản phẩm.- Quản lý danh mục gồm: thêm danh mục, xóa danh mục, cập nhật danh mục.- Quản lý hóa đơn: tìm kiếm hóa đơn, xem danh sách hóa đơn và chi tiết hóa đơn.
3	 Quản trị viên	<p>Tác nhân quản trị viên là người điều hành, quản lý và theo dõi mọi hoạt động của ứng dụng.</p> <p>Thực hiện các chức năng: Quản lý tài khoản nhân viên quản lý, quản lý dịch vụ.</p>

Bảng 2. Các tác nhân hệ thống

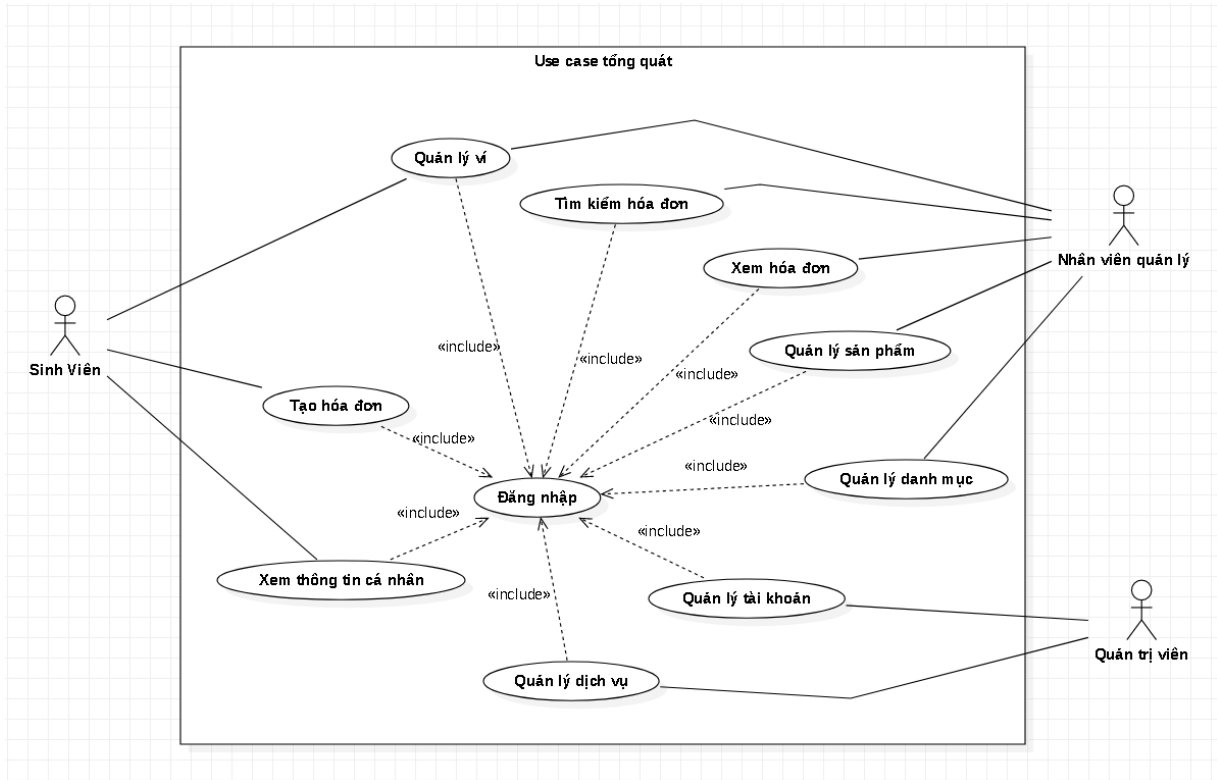
3.3.2 Các chức năng của hệ thống

STT	Chức năng	Mô tả
1	Đăng nhập	Chức năng này cho phép các tác nhân đăng nhập vào ứng dụng bằng tài khoản để thực hiện các chức năng của họ.
2	Quản lý sản phẩm	Chức năng này cho phép quản lý có thể thêm sản phẩm, cập nhật sản phẩm, xóa sản phẩm, xem thông tin sản phẩm.
3	Quản lý danh mục	Chức năng này cho phép nhân viên quản lý thêm danh mục, sửa danh mục, xóa danh mục, xóa sản phẩm trong danh mục.
4	Quản lý thông tin cá nhân	Chức năng này cho phép sinh viên có thể cập nhật, xem thông tin hồ sơ của tài khoản cá nhân.
5	Tạo hóa đơn	Chức năng này cho phép sinh viên có thể khởi tạo và thanh toán hóa đơn.
6	Quản lý ví	Chức năng này cho phép sinh viên nạp tiền vào ví, xem lịch sử giao dịch và chi tiết lịch sửa giao dịch. Cho phép nhân viên quản lý kiểm tra số dư có trong ví của mình.
7	Quản lý tài khoản	Chức năng này cho phép quản trị viên có thể cập nhật, khởi tạo, xóa, hoặc xem thông tin tài khoản.
8	Quản lý dịch vụ	Chức năng này cho phép quản trị viên có thể cập nhật, khởi tạo, xóa, hoặc xem thông tin các dịch vụ trên ứng dụng.

Bảng 3. Các chức năng hệ thống

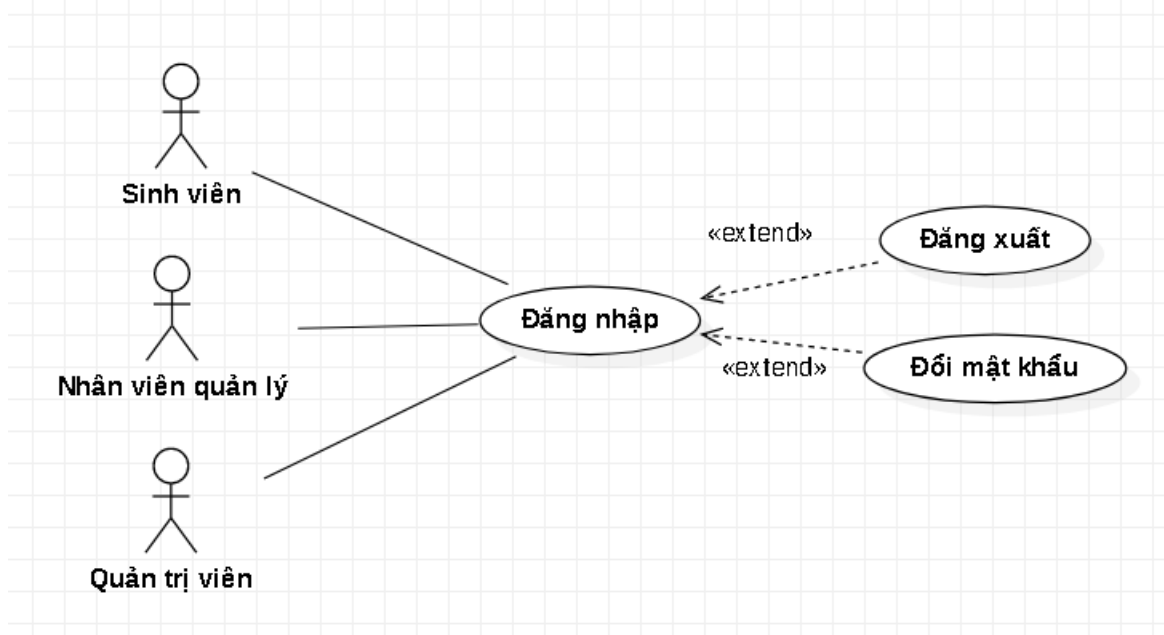
3.3.3 Sơ đồ use case

3.3.3.1 Sơ đồ use case tổng quát



Hình 3.1 Sơ đồ use case tổng quát

3.3.3.3 Use case đăng nhập



Hình 3.2 Use case đăng nhập

- Tác nhân: Người quản lý, sinh viên và quản trị viên
- Mô tả tổng quát: Nhân viên quản lý, sinh viên và quản trị viên đăng nhập vào hệ thống để thực hiện các chức năng được cấp quyền.
- Điều kiện kích hoạt: Use case sử dụng bắt đầu khi người dùng muốn đăng nhập vào.
- Điều kiện thành công:
 - + Ứng dụng yêu cầu người dùng sử dụng tài khoản đã được tạo sẵn.
 - + Thiết bị người dùng đã được kết nối internet khi thực hiện đăng nhập.
- Dòng sự kiện chính:
 1. Người dùng truy cập ứng dụng.
 2. Nếu là sinh viên chọn đăng nhập bằng gmail trường, nếu là nhân viên quản lý hoặc quản trị viên chọn đăng nhập bằng tài khoản thường.
 3. Người dùng nhập tài khoản và mật khẩu (đối với quản trị và nhân viên quản lý) hoặc chọn Gmail sinh viên (đối với sinh viên).

4. Hệ thống xác thực thông tin đăng nhập và cho phép người dùng truy cập ứng dụng.

- Dòng sự kiện phụ:

4a. Tài khoản Google của sinh viên chưa được đăng nhập và bổ sung thông tin trước đó.

4a1. Hệ thống chuyển đến trang bổ sung thông tin.

4a2. Sinh viên nhập thông tin và chọn lệnh tiếp tục.

Use case tiếp tục bước 4.

- Dòng sự kiện ngoại lệ:

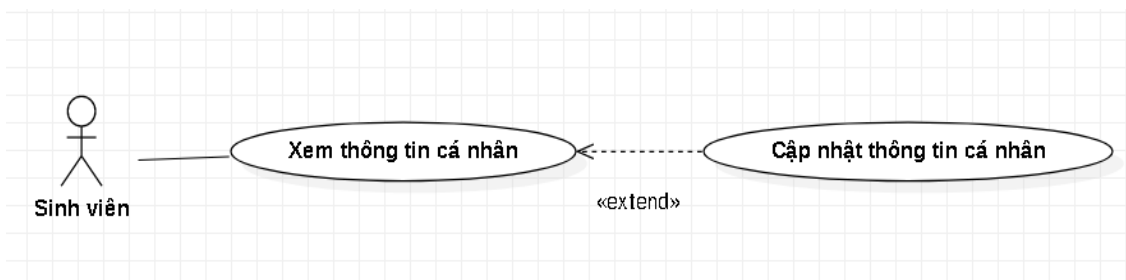
4c. Ứng dụng xác thực thông tin đăng nhập không thành công và hiện thông báo đăng nhập thất bại.

Use case tiếp tục bước 2.

4c1. Người dùng hủy quá trình đăng nhập.

Use case kết thúc.

3.3.3.4 Use case xem thông tin cá nhân

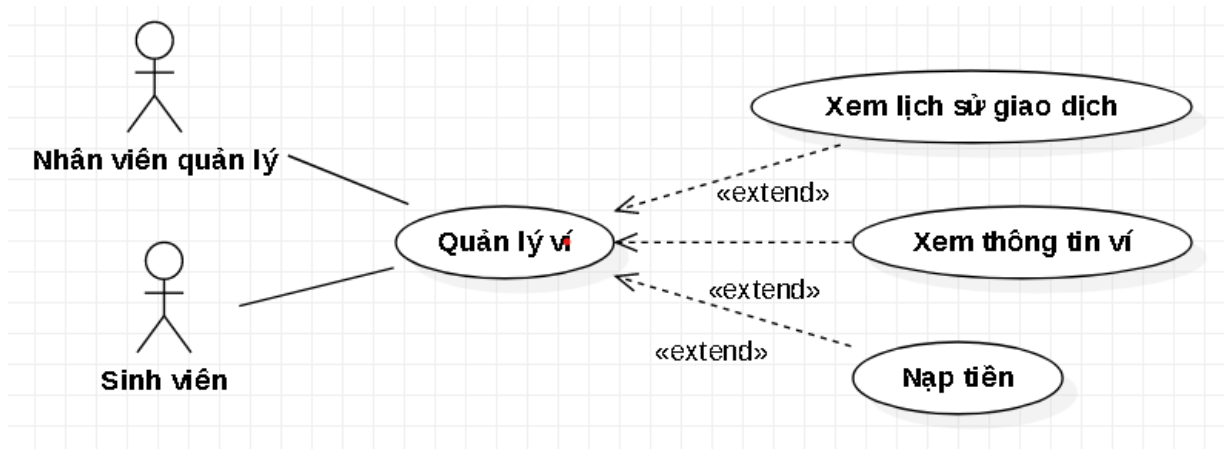


Hình 3.3 Use case xem thông tin cá nhân

- Tác nhân: Sinh viên
- Mô tả tổng quát: Sinh viên sử dụng chức năng này để xem thông tin cá nhân và cập nhật thông tin cá nhân.
- Điều kiện kích hoạt: Use case sử dụng bắt đầu khi sinh viên muốn sử dụng chức năng xem thông tin cá nhân.
- Điều kiện thành công:

- + Sinh viên đã đăng nhập thành công vào ứng dụng.
- + Thiết bị người dùng đã được kết nối internet khi thực chức năng xem thông tin.
- Dòng sự kiện chính:
 - 1. Sinh viên chọn chức năng xem thông tin cá nhân.
 - 2. Hệ thống hiển thị thông tin cá nhân của sinh viên.
- Dòng sự kiện phụ:
 - 2a. Sinh viên chọn chỉnh sửa thông tin cá nhân.
 - 2a1. Sinh viên chọn thông tin cần sửa trừ gmail và chọn lệnh cập nhật.
 - 2a2. Hệ thống kiểm tra thông tin vừa thay đổi.
 - 2a3. Hệ thống cập nhật lại cơ sở dữ liệu và hiển thị các thông tin vừa thay đổi lên màn hình.
- Dòng sự kiện ngoại lệ:
 - 4a. Hệ thống kiểm tra thông tin sinh viên nhập không hợp lệ và hiện thông báo.
Use case quay lại bước 3.
 - 4c1. Sinh viên hủy thao tác sửa.
Use case kết thúc.

3.3.3.5 Use case quản lý ví



Hình 3.4 Use case quản lý ví

- Tác nhân: Sinh viên, nhân viên quản lý
- Mô tả tổng quát:
 - + Sinh viên sử dụng chức năng quản lý ví để xem thông tin ví, nạp tiền đơn giản cho ví và danh sách lịch sử giao dịch.
 - + Nhân viên quản lý có thể xem số dư còn lại trong ví.
- Điều kiện kích hoạt: Use case bắt đầu khi người dùng chọn lệnh Quản lý ví.
- Điều kiện thành công:
 - + Sinh viên, nhân viên quản lý đã đăng nhập thành công vào ứng dụng.
 - + Thiết bị người dùng đã được kết nối internet khi thực chức năng quản lý ví.
- Dòng sự kiện chính:
 1. Người dùng chọn chức năng Ví trên trang chủ.
 2. Hệ thống hiển thị thông tin ví của của người dùng.
 3. Người dùng chọn lệnh nạp tiền.
 4. Hệ thống yêu cầu nhập số tiền cần nạp.
 5. Người dùng chọn lệnh “Nạp tiền” trên của sổ nạp tiền để xác nhận.
 6. Hệ thống kiểm tra và cộng tiền vào số dư của ví đồng thời thêm chi tiết giao dịch vào cơ sở dữ liệu.
 7. Hệ thống hiển thị nạp tiền thành công.

Use case kết thúc.

- Dòng sự kiện phụ:

3a. Sinh viên chọn dòng lịch sửa giao dịch.

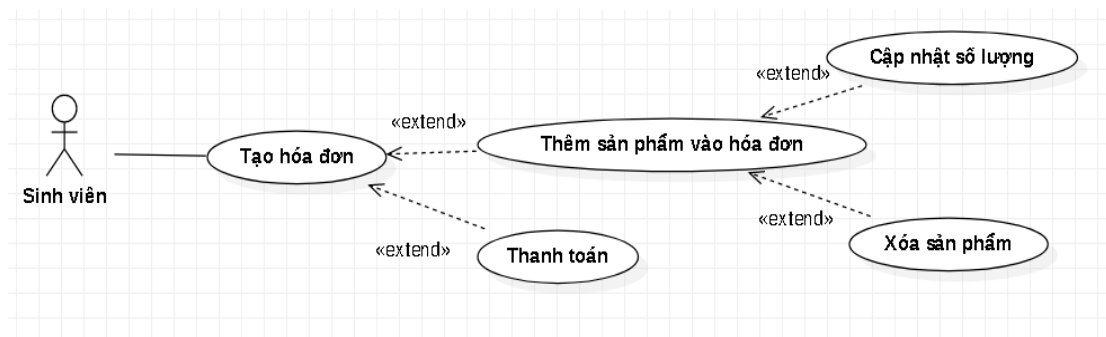
3a1. Hệ thống hiển thị chi tiết giao dịch.

- Dòng sự kiện ngoại lệ:

+ Sai định dạng: Người dùng nhập số tiền sai định dạng hệ thống sẽ hiển thị lỗi và cho người dùng nhập lại.

+ Hủy thao tác: người dùng hủy thao tác nạp tiền hệ thống sẽ trở về bước trước.

3.3.3.6 Use case tạo hóa đơn



Hình 3.5 Use case Tạo hóa đơn

- Tác nhân: Sinh viên

- Mô tả tổng quát: Sinh viên sử dụng chức năng tạo hóa đơn để tạo hóa đơn tạm thời và thêm sản phẩm vào hóa đơn để thanh toán khi mua hàng.

- Điều kiện kích hoạt: Use case bắt đầu khi người dùng chọn lệnh tạo hóa đơn.

- Điều kiện thành công:

+ Sinh viên đã đăng nhập thành công vào ứng dụng.

+ Thiết bị người dùng đã được kết nối internet khi thực chức năng tạo hóa đơn.

+ Số dư trong ví còn đủ để thanh toán hóa đơn.

- Dòng sự kiện chính:

1. Sinh viên chọn chức năng tạo hóa đơn.

2. Hệ thống tạo một hóa đơn trống cho phép sinh viên thêm sản phẩm vào hóa đơn.
3. Sinh viên quét mã QR có trên sản phẩm.
4. Hệ thống tự động kiểm tra thông tin mã QR và thêm sản phẩm tương ứng với số lượng mua mặc định là một sản phẩm.
5. Sinh viên chọn lệnh tiếp tục để chuyển sang trang xác nhận đơn hàng.
6. Tại trang xác nhận đơn hàng sinh viên chọn lệnh thanh toán.
7. Hệ thống kiểm tra thông tin đơn hàng.
8. Hệ thống đồng thời trừ tiền trong ví bên phía sinh viên, đồng thời tiến hành cộng số tiền tương ứng vào ví của nhân viên quản lý.
9. Hệ thống lưu trữ thông tin hóa đơn lên cơ sở dữ liệu và chuyển sang trang thông tin chi tiết hóa đơn.
10. Sinh viên chọn lệnh quay về trang chủ.

Use case kết thúc.

- Dòng sự kiện phụ:

- 5a. Sinh viên ấn vào dòng hiển thị sản phẩm để chỉnh sửa số lượng mua sản phẩm.
- 5a1. Sinh viên chỉnh sửa số lượng sản phẩm cần mua và chọn lệnh cập nhật.
- 5a2. Hệ thống kiểm tra thông tin vừa thay đổi.
- 5a3. Hệ thống cập nhật lại thông tin của sản phẩm trong hóa đơn.

Use case tiếp tục tại bước 5 của luồng chính.

- 6a. Sinh viên chọn lệnh thêm sản phẩm.

6a1. Hệ thống chuyển về trang tạo hóa đơn cho phép sinh viên tiếp tục quét mã QR để thêm sản phẩm mới.

Use case tiếp tục tại bước 5 của luồng chính.

- Dòng sự kiện ngoại lệ:

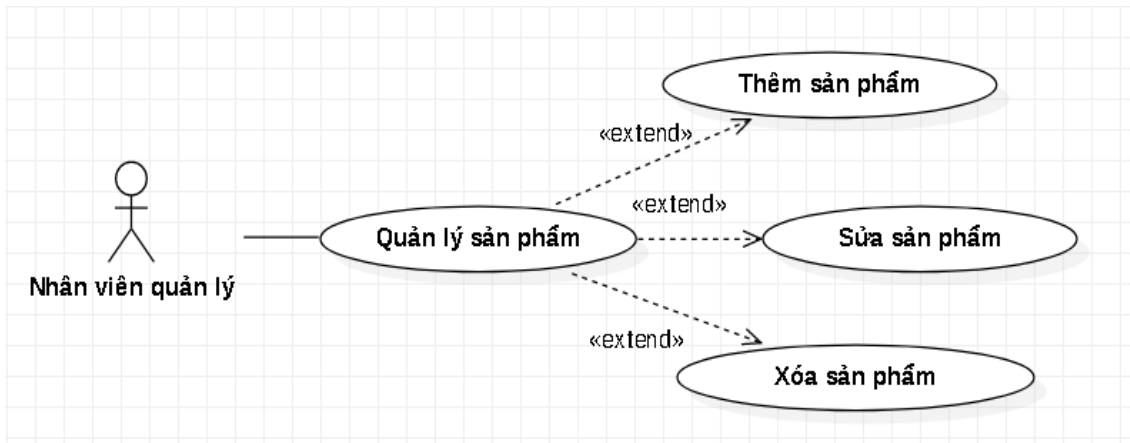
- 7a. Hệ thống kiểm tra thông tin số dư ví không đủ để thanh toán và hiển thị thông báo số dư ví không đủ.

Use case quay lại bước 6 luồng chính.

4c1.Sinh viên hủy thao tác tạo hóa đơn.

Use case kết thúc.

3.3.3.7 Use case quản lý sản phẩm



Hình 3.6 Use case Quản lý sản phẩm

- Tác nhân: Nhân viên quản lý.
- Mô tả tổng quát:
Use case cho phép nhân viên quản lý sản phẩm bao gồm: thêm sản phẩm, cập nhật sản phẩm, xóa sản phẩm.
- Điều kiện kích hoạt: Use case bắt đầu khi người dùng chọn chức năng quản lý sản phẩm.
- Điều kiện thành công:
 - + Nhân viên đã đăng nhập vào hệ thống.
 - + Thiết bị người dùng đã được kết nối internet khi thực chức năng tạo hóa đơn.
- Dòng sự kiện chính:
 1. Nhân viên quản lý chọn chức năng quản lý sản phẩm.
 2. Hệ thống hiển thị giao diện quản lý sản phẩm.
 3. Nhân viên quản lý chọn chức năng cần thi hành.
 - Nếu nhân viên quản lý chọn “Thêm sản phẩm”, luồng “Thêm sản phẩm” sẽ được thực thi.

- Nếu nhân viên quản lý chọn “Sửa sản phẩm”, luồng “Sửa sản phẩm” sẽ được thực thi.
- Nếu nhân viên quản lý chọn “Xóa sản phẩm”, luồng “Xóa sản phẩm” sẽ được thực thi.

+ Thêm sản phẩm:

4a. Nhân viên quản lý chọn lệnh thêm sản phẩm.

4a1. Hệ thống yêu cầu nhân viên nhập thông tin sản phẩm, bao gồm: tên sản phẩm, số lượng sản phẩm, danh mục sản phẩm nếu có, hình ảnh, giá bán, đơn vị.

4a2. Khi nhân viên quản lý cung cấp đầy đủ thông tin, hệ thống sẽ tự động phát sinh mã sản phẩm (mã này là duy nhất). Sản phẩm được lưu vào cơ sở dữ liệu.

+ Sửa sản phẩm:

4b. Nhân viên quản lý chọn lệnh sửa sản phẩm.

4b1. Hệ thống truy xuất và hiển thị thông tin sản phẩm mà nhân viên quản lý đã chọn.

4b2. Nhân viên quản lý chỉnh sửa các thông tin sản phẩm đã đề cập ở “Thêm sản phẩm” (trừ mã sản phẩm).

4b3. Hệ thống hiện thông báo yêu cầu nhân viên quản lý xác nhận sửa sản phẩm.

4b4. Nhân viên quản lý xác nhận sửa sản phẩm.

4b5. Hệ thống tiến hành cập nhật lại thông tin sản phẩm ứng với mã sản phẩm.

4b6. Hệ thống thông báo cập nhật thành công.

+ Xóa sản phẩm:

4c. Nhân viên quản lý chọn sản phẩm cần xóa.

4c1. Nhân viên quản lý chọn lệnh xóa sản phẩm.

4c2. Hệ thống yêu cầu nhân viên quản lý xác nhận xóa sản phẩm.

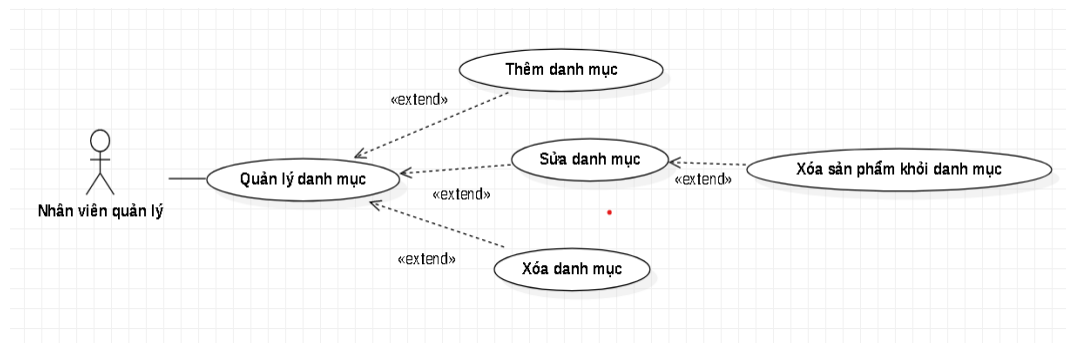
4c3. Nhân viên quản lý xác nhận xóa sản phẩm.

4c4. Hệ thống thực hiện xóa sản phẩm.

4c5. Hệ thống hiển thị thông báo xóa sản phẩm thành công.

- Dòng sự kiện phụ: Không có.
- Dòng sự kiện ngoại lệ:
 - + Từ chối xác nhận: nếu trong các luồng sự kiện chính hoặc phụ, nhân viên quản lý từ chối xác nhận các thao tác tương ứng, hệ thống sẽ trở lại thao tác trước đó của từng luồng sự kiện tương ứng.

3.3.3.8 Use case quản lý danh mục



Hình 3.7 Use case quản lý danh mục

- Tác nhân: Nhân viên quản lý.
- Mô tả tổng quát:

Use case cho phép nhân viên quản lý danh mục bao gồm: thêm danh mục sản phẩm, cập nhật danh mục sản phẩm, xóa danh mục sản phẩm.
- Điều kiện kích hoạt: Use case bắt đầu khi người dùng chọn chức năng quản lý danh mục.
- Điều kiện thành công:
 - + Nhân viên quản lý đã đăng nhập vào hệ thống.
 - + Thiết bị người dùng đã được kết nối internet khi thực chức năng quản lý danh mục.
- Dòng sự kiện chính:
 1. Nhân viên quản lý chọn chức năng quản lý danh mục.

2. Hệ thống hiển thị giao diện quản lý danh mục.

3. Nhân viên quản lý chọn chức năng cần thi hành.

- Nếu nhân viên quản lý chọn “Thêm danh mục”, luồng “Thêm danh mục” sẽ được thực thi.
- Nếu nhân viên quản lý chọn “Sửa danh mục”, luồng “Sửa danh mục” sẽ được thực thi.
- Nếu nhân viên quản lý chọn “Xóa danh mục”, luồng “Xóa danh mục” sẽ được thực thi.

+ Thêm danh mục:

4a. Nhân viên quản lý chọn trang danh mục có trên giao diện quản lý.

4a1. Nhân viên quản lý chọn lệnh thêm danh mục.

4a2. Hệ thống yêu cầu nhập tên danh mục.

4a3. Nhân viên quản lý chọn lệnh tạo mới.

4a4. Hệ thống kiểm tra thông tin và tiến hành tạo mã danh mục sản phẩm (mã danh mục là duy nhất), danh mục sản phẩm được lưu vào cơ sở dữ liệu.

4a5. Hệ thống hiển thị thông báo thêm danh mục thành công.

+ Sửa danh mục:

4b. Nhân viên quản lý ấn chọn danh mục cần sửa và vuốt trang trái.

4b1. Nhân viên quản lý ấn chọn lệnh sửa.

4b2. Hệ thống truy xuất thông tin và cửa sổ để nhân viên quản lý sửa tên danh mục

4b3. Nhân viên quản lý sửa tên danh mục sản phẩm và chọn lệnh cập nhật.

4b4. Hệ thống kiểm tra tên danh mục và tiến hành cập nhật tên danh mục theo mã danh mục lên cơ sở dữ liệu.

4b5. Hệ thống hiển thị thông báo cập nhật thành công danh mục.

+ Xóa danh mục:

4c. Nhân viên quản lý chọn danh mục cần sửa và vuốt trang trái.

4c1. Nhân viên quản lý ấn chọn lệnh xóa.

4c1. Hệ thống hiển thị thông báo yêu cầu nhân viên quản lý xác nhận xóa danh mục.

4c2. Nhân viên quản lý xác nhận xóa danh mục sản phẩm.

4c3. Hệ thống thực hiện cập nhật tất cả sản phẩm có trong danh mục mã danh mục là rỗng, sau đó tiến hành xóa danh mục trên cơ sở dữ liệu.

4c4. Hệ thống hiển thị thông báo đã xóa danh mục sản phẩm thành công.

- Dòng sự kiện phụ:

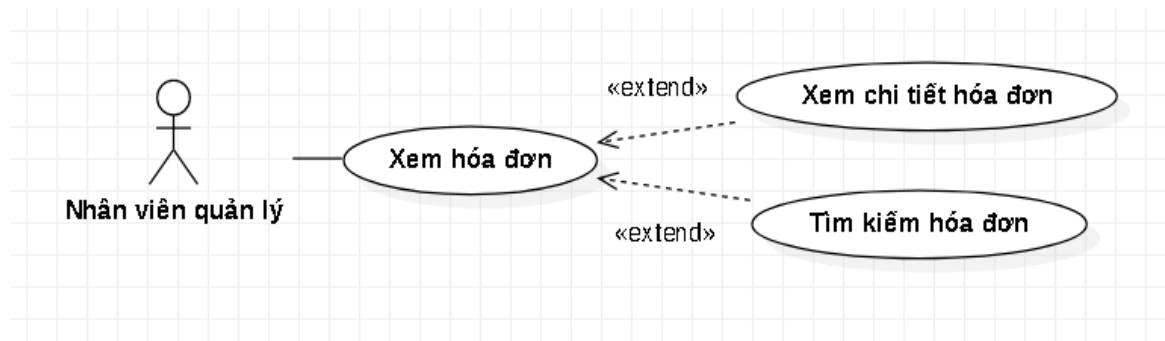
Nếu nhân viên quản lý ấn chọn vào danh mục thì luồng “Xóa sản phẩm ra khỏi danh mục” sẽ được thực thi.

+ Xóa sản phẩm ra khỏi danh mục:

- Nhân viên quản lý ấn chọn vào danh mục.
- Hệ thống chuyển đến trang danh sách sản phẩm có trong danh mục.
- Nhân viên quản lý chọn sản phẩm cần xóa và vuốt sang trái để hiển thị nút xóa.
- Nhân viên quản lý chọn lệnh xóa.
- Hệ thống hiển thị thông báo yêu cầu xác nhận xóa sản phẩm khỏi danh mục.
- Nhân viên quản lý xác nhận xóa sản phẩm khỏi danh mục.
- Hệ thống sẽ tiến hành cập nhật mã danh mục của sản phẩm thành rỗng.
- Hệ thống hiển thị thông báo xóa sản phẩm ra khỏi danh mục thành công.
- Hệ thống hiển thị thông báo yêu cầu xác nhận của nhân viên quản lý.
- Nhân viên quản lý xác nhận.
- Hệ thống tiến hành cập nhật mã danh mục của sản phẩm.
- Hệ thống hiển thị thêm sản phẩm thành công.

- Dòng sự kiện ngoại lệ:
 - + Từ chối xác nhận: nếu trong các luồng sự kiện chính hoặc phụ, nhân viên quản lý từ chối xác nhận các thao tác tương ứng, hệ thống sẽ trở lại thao tác trước đó của từng luồng sự kiện tương ứng.
 - + Tên danh mục đã tồn tại: nếu trong luồng sự kiện “Thêm danh mục”, nếu tên danh mục đã có trong cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo yêu cầu nhân viên quản lý nhập tên khác.

3.3.3.9 Use case xem hóa đơn



Hình 3.8 Use case xem hóa đơn

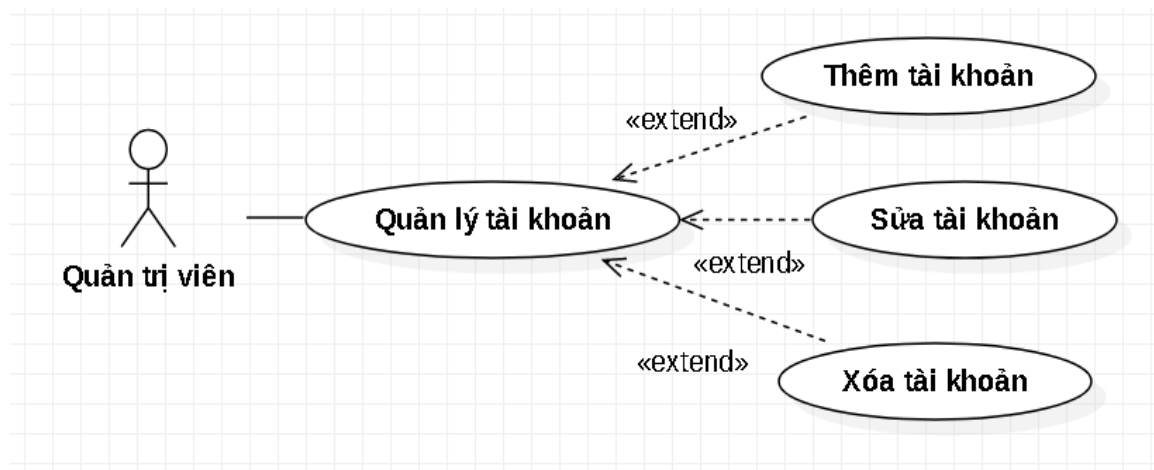
- Tác nhân: Nhân viên quản lý
- Mô tả tổng quát: Nhân viên quản lý sử dụng chức năng xem hóa đơn để xem danh sách hóa đơn và chi tiết hóa đơn.
- Điều kiện kích hoạt: Use case bắt đầu khi người dùng chọn lệnh “Hóa đơn”.
- Điều kiện thành công:
 - + Nhân viên quản lý đã đăng nhập thành công vào ứng dụng.
 - + Thiết bị người dùng đã được kết nối internet khi thực chức năng xem hóa đơn.
- Dòng sự kiện chính:
 1. Nhân viên quản lý chọn chức hóa đơn trên giao diện trang chủ.
 2. Hệ thống hiển thị danh sách hóa đơn.
 3. Nhân viên quản lý tìm kiếm hóa đơn trên thanh tìm kiếm bằng cách nhập thời gian cần tìm.
 4. Nhân viên quản lý chọn hóa đơn cần xem chi tiết.

5. Hệ thống hiển thị thông tin chi tiết hóa đơn nhân viên quản lý đã chọn.

Use case kết thúc.

- Dòng sự kiện phụ: Không có.
- Dòng sự kiện ngoại lệ:
 - Không có hóa đơn: nếu nhân viên quản lý nhập thời gian không có hóa đơn nào tồn tại thì danh sách sẽ hiển thị trống.
 - Nhân viên quản lý thoát khỏi trang hóa đơn.

3.3.3.10 Use case quản lý tài khoản



Hình 3.9 Use case Quản lý tài khoản

- Tác nhân: Quản trị viên
- Mô tả tổng quát:

Use case này cho phép người quản trị quản lý tài khoản của nhân viên quản lý trong hệ thống, bao gồm: thêm tài khoản, cập nhật tài khoản, khóa tài khoản.
- Điều kiện kích hoạt: Use case bắt đầu khi người quản trị chọn chức năng quản lý tài khoản.
- Điều kiện thành công:
 - + Người quản trị đã đăng nhập thành công vào ứng dụng.
 - + Thiết bị người quản trị đã được kết nối internet khi thực hiện chức năng quản lý tài khoản

- Dòng sự kiện chính:
 1. Người quản trị chọn chức năng quản lý tài khoản.
 2. Hệ thống hiển thị giao diện quản lý tài khoản.
 3. Người quản trị chọn chức năng cần thi hành.
 - Nếu người quản trị chọn chức năng “Thêm tài khoản”, luồng sự kiện “Thêm tài khoản” sẽ được thực thi.
 - Nếu người quản trị chọn chức năng “Cập nhật tài khoản”, luồng sự kiện “Cập nhật tài khoản” sẽ được thực thi.
 - Nếu người quản trị chọn chức năng “Xóa tài khoản”, luồng sự kiện “Xóa tài khoản” sẽ được thực thi.

+ Thêm tài khoản:

- 4a. Người quản trị chọn chức năng thêm tài khoản.
 - 4a1. Hệ thống yêu cầu người quản trị nhập thông tin nhân viên quản lý, bao gồm: username, mật khẩu, tên đầy đủ, tên dịch vụ (username là duy nhất).
 - 4a2. Người quản trị chọn lệnh thêm.
 - 4a3. Hệ thống kiểm tra thông tin.
 - 4a4. Hệ thống tạo tài khoản nhân viên quản lý mới lưu vào cơ sở dữ liệu.

+ Cập nhật tài khoản:

- 4b. Người quản trị chọn tài khoản cần cập nhật.
 - 4b1. Hệ thống truy xuất và hiển thị thông tin tài khoản (mật khẩu chỉ là ô trống để người quản trị cập nhật lại) mà người quản trị đã chọn.
 - 4b2. Người quản trị chỉnh sửa các thông tin đã đề cập ở luồng “Thêm tài khoản” trừ username và chọn lệnh cập nhật.
 - 4b3. Hệ thống hiển thị thông báo yêu cầu người quản trị xác nhận.
 - 4b4. Người quản trị xác nhận yêu cầu.
 - 4b5. Hệ thống tiến hành cập nhật thông tin tài khoản theo mã nhân viên.
 - 4b6. Hệ thống hiển thị thông báo cập nhật tài khoản thành công.

+ Xóa tài khoản:

4c. Người quản trị chọn tài khoản cần xóa.

4c1. Người quản trị chọn lệnh khóa tài khoản.

4c2. Hệ thống hiển thị thông báo yêu cầu người quản trị xác nhận xóa tài khoản.

4c3. Người dùng xác nhận xóa tài khoản.

4c4. Hệ thống thực hiện xóa tài khoản.

4c5. Hệ thống hiển thị thông báo xóa tài khoản thành công.

- Dòng sự kiện phụ:

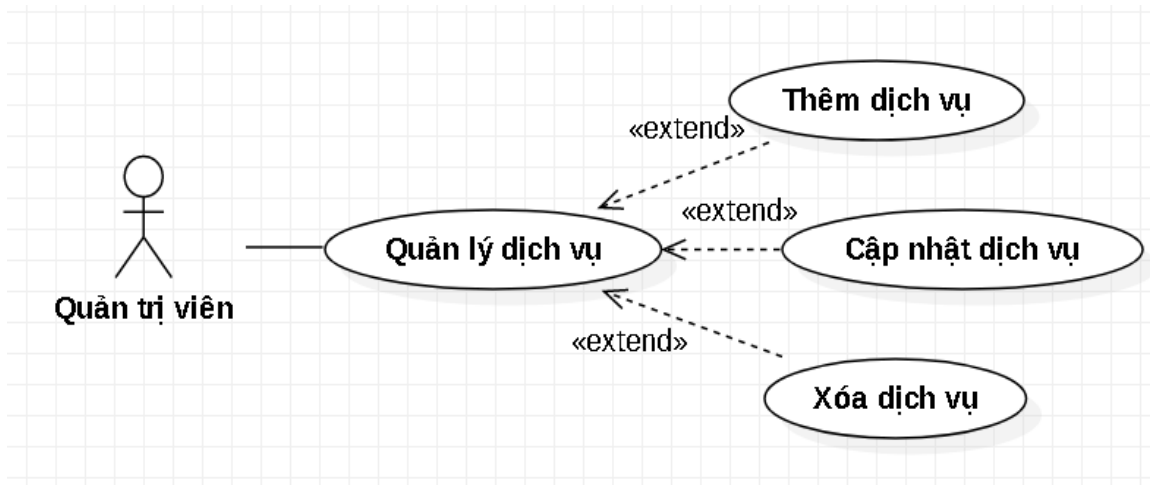
Không có.

- Dòng sự kiện ngoại lệ:

+ Từ chối xác nhận: nếu trong các luồng sự kiện chính hoặc phụ, nhân viên quản lý từ chối xác nhận các thao tác tương ứng, hệ thống sẽ trở lại thao tác trước đó của từng luồng sự kiện tương ứng.

+ Username đã tồn tại: nếu trong luồng sự kiện “Thêm tài khoản”, nếu username đã có trong cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi. Luồng sự kiện được bắt đầu lại từ đầu.

3.3.3.11 Use case quản lý dịch vụ



Hình 3.10 Use case Quản lý dịch vụ

- Tác nhân: Quản trị viên.
- Mô tả tổng quát:
Use case này cho phép người quản trị quản lý dịch vụ trong hệ thống, bao gồm: thêm dịch vụ, cập nhật dịch vụ, xóa dịch vụ.
- Điều kiện kích hoạt:
Use case bắt đầu khi người quản trị chọn chức năng quản lý dịch vụ.
- Điều kiện thành công:
 - + Người quản trị đã đăng nhập thành công vào ứng dụng.
 - + Thiết bị người quản trị đã được kết nối internet khi thực hiện chức năng quản lý dịch vụ.
- Dòng sự kiện chính:
 1. Người quản trị chọn chức năng quản lý dịch vụ.
 2. Hệ thống hiển thị giao diện quản lý dịch vụ.
 3. Người quản trị chọn chức năng cần thi hành.
 - Nếu người quản trị chọn chức năng “Thêm dịch vụ”, luồng sự kiện “Thêm dịch vụ” sẽ được thực thi.

- Nếu người quản trị chọn chức năng “Cập nhật dịch vụ”, luồng sự kiện “Cập nhật dịch vụ” sẽ được thực thi.
- Nếu người quản trị chọn chức năng “Xóa dịch vụ”, luồng sự kiện “Xóa dịch vụ” sẽ được thực thi.

+ Thêm dịch vụ:

4a. Người quản trị chọn chức năng thêm dịch vụ.

4a1. Hệ thống yêu cầu người quản trị nhập tên dịch vụ.

4a2. Người quản trị chọn lệnh thêm.

4a3. Hệ thống kiểm tra thông tin.

4a4. Hệ thống tự tạo mã dịch vụ và thêm dịch vụ mới vào cơ sở dữ liệu.

4a5. Hệ thống hiển thị thông báo thêm dịch vụ thành công

+ Cập nhật dịch vụ:

4b. Người quản trị chọn dịch vụ cần cập nhật.

4b1. Hệ thống truy xuất và hiển thị thông tin dịch vụ mà người quản trị đã chọn.

4b2. Người quản trị chỉnh sửa tên dịch vụ.

4b3. Hệ thống hiển thị thông báo yêu cầu người quản trị xác nhận.

4b4. Người quản trị xác nhận yêu cầu.

4b5. Hệ thống tiến hành cập nhật thông tin dịch vụ theo mã dịch vụ.

4b6. Hệ thống hiển thị thông báo cập nhật dịch vụ thành công.

+ Xóa dịch vụ:

4c. Người quản trị chọn dịch vụ cần khóa.

4c1. Người quản trị chọn lệnh xóa dịch vụ.

4c2. Hệ thống hiển thị thông báo yêu cầu người quản trị xác nhận xóa dịch vụ.

4c3. Người dùng xác nhận xóa dịch vụ.

4c4. Hệ thống thực hiện xóa dịch vụ.

4c5. Hệ thống hiển thị thông báo xóa dịch vụ thành công.

- Dòng sự kiện phụ: Không có.

- Dòng sự kiện ngoại lệ:
- + Từ chối xác nhận: nếu trong các luồng sự kiện chính hoặc phụ, nhân viên quản lý từ chối xác nhận các thao tác tương ứng, hệ thống sẽ trở lại thao tác trước đó của từng luồng sự kiện tương ứng.
- + Tên dịch vụ đã tồn tại: nếu trong luồng sự kiện “Thêm dịch vụ”, nếu tên dịch vụ đã có trong cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi. Luồng sự kiện được bắt đầu lại từ đầu.
- + Dịch vụ đang được sử dụng: nếu trong luồng sự kiện “Xóa dịch vụ”, dịch vụ vẫn còn tài khoản nhân viên quản lý sử dụng, hệ thống sẽ hiển thị thông báo lỗi. Luồng sự kiện bắt đầu lại từ đầu.

3.3.4 Sơ đồ lớp

TaiKhoan	QuanLy	SinhVien
{ “_id”: ObjectID, “username”: String, “password”: String }	{ “_id”: ObjectID, “TaiKhoan_id”: ObjectID, “DichVu_id”: ObjectID, “hovaten”: String, “image”: String, “trangthai”: Number }	{ “_id”: ObjectID, “hovaten”: String, “Lop”: String, “SDT”: String, “image”: String }

DichVu	Vi	GiaoDich
{ “_id”: ObjectID, “TenDV”: String, }	{ “_id”: ObjectID, “TaiKhoan_id”: ObjectID, “SoDu”: Double, }	{ “_id”: ObjectID, “SinhVien_id”: ObjectID, “DichVu_id”: ObjectID, “MoTa”: String, “ThoiGian”: String, “SoTien”: Double }

GioHang	SanPham	HoaDon
{ “_id”: ObjectID, “SinhVien_id”: ObjectID, “SanPham_id”: ObjectID, “Gia”: Double, “SoLuong”: Number, }	{ “_id”: ObjectID, “DichVu_id”: ObjectID, “DanhMuc_id”: ObjectID, “TenSP”: String, “Gia”: Double, }	{ “_id”: ObjectID, “GiaoDich_id”: ObjectID, “DichVu_id”: ObjectID, “SoLuongSP”: Number, “TongTien”: Double, }

}	“image” : String, “TrangThai” : Number, }	“ChiTiet” : [{ “SanPham_id” : ObjectID, “TenSP” : String, “Gia” : Double, “SoLuong” : Number “image” : String }] }
---	---	---

DanhMuc
{ “_id” : ObjectID, “DichVu_id” : ObjectID, “TenDanhMuc” : String, }

Bảng 4: Bảng cơ sở dữ liệu

CHƯƠNG 4 CÀI ĐẶT VÀ HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

4.1 Chức năng đăng nhập

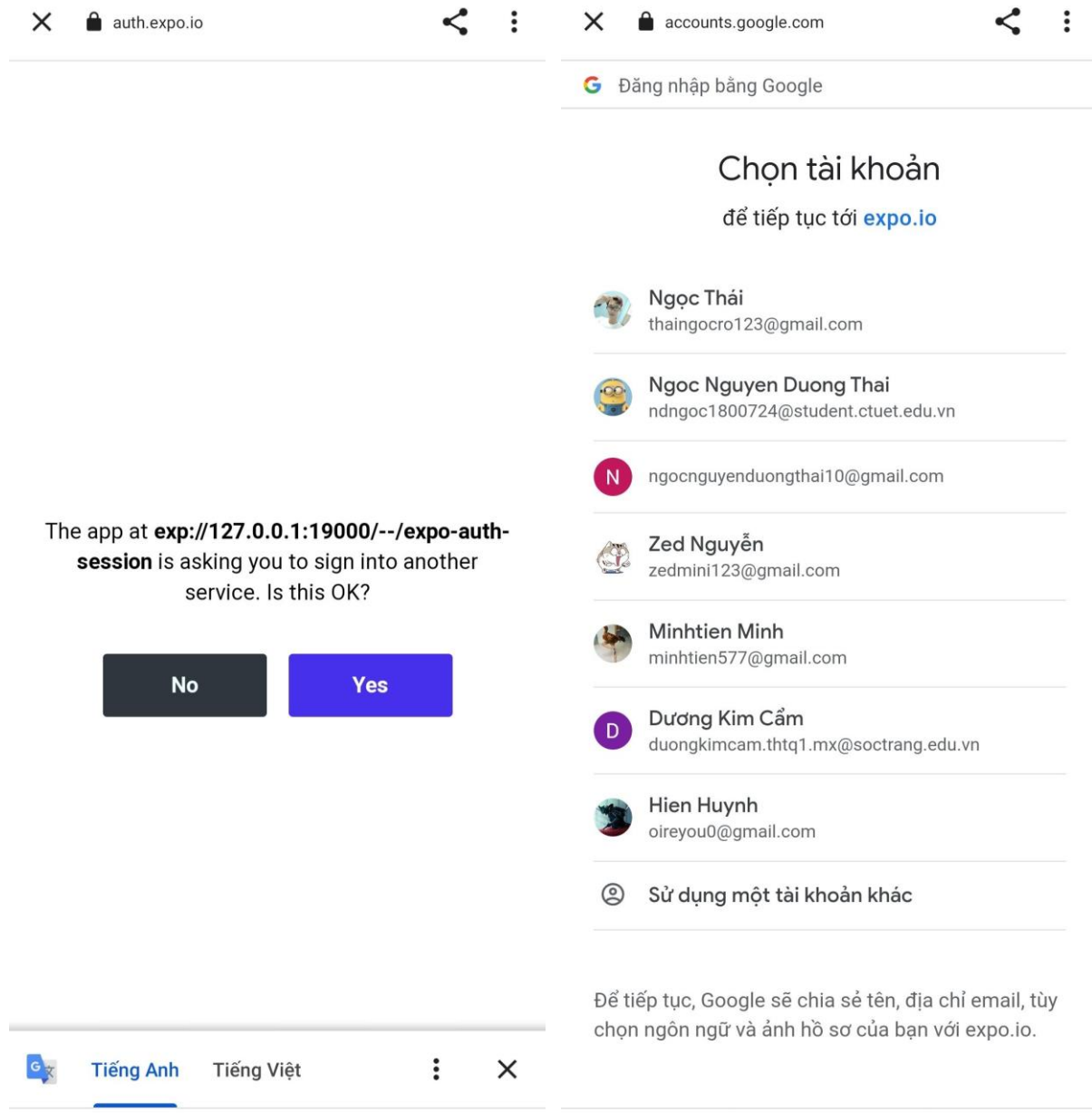
Đầu tiên người dùng mở ứng dụng lên chọn hình thức đăng nhập.

Đối với sinh viên chọn hình thức “Đăng nhập với Student email”, đối với quản trị viên và nhân viên quản lý chọn hình thức “Đăng nhập tài khoản khác”.




Hình 4.1 Giao diện đăng nhập

Khi người dùng chọn hình thức đăng nhập với email sinh viên, hệ thống sẽ yêu cầu quyền truy cập, người dùng cần cho phép để tiếp tục chọn email.



Hình 4.2 Chấp nhận yêu cầu quyền truy cập và chọn tài khoản email

Nếu tài khoản email sinh viên chọn đã đăng ký thông tin, hệ thống chuyển đến giao diện trang chủ, ngược lại hệ thống sẽ chuyển sang trang bổ sung thông tin. Sau khi bổ sung thông tin và chọn lệnh hoàn tất, hệ thống chuyển đến trang chủ.

0:15    

VUI LÒNG ĐIỀN ĐẦY ĐỦ THÔNG TIN

thaingocro123@gmail.com

Lớp

HTTT0118 

Giới tính

Nam 

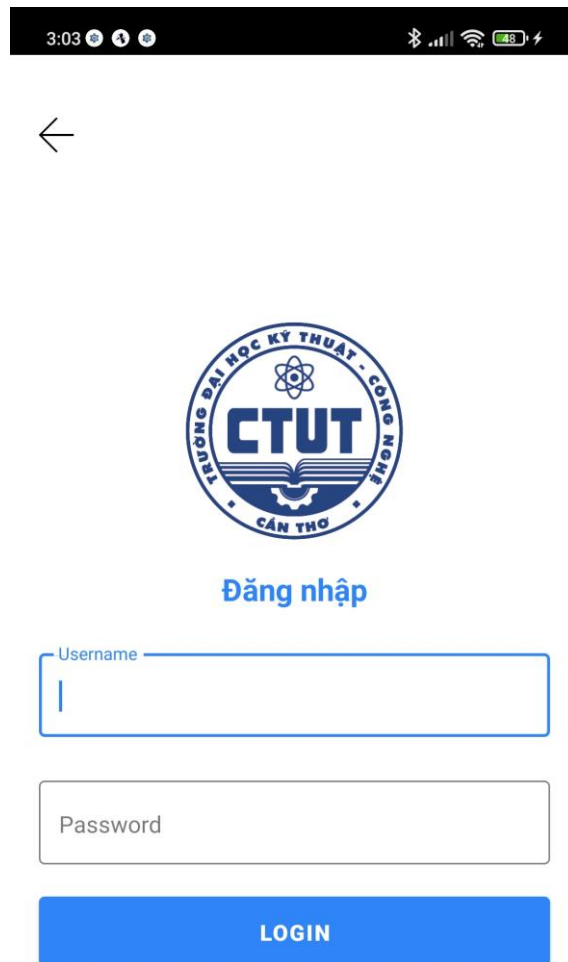
Ngày sinh

2022-06-10 

HOÀN TẤT

Hình 4.3 Giao diện bổ sung thông tin

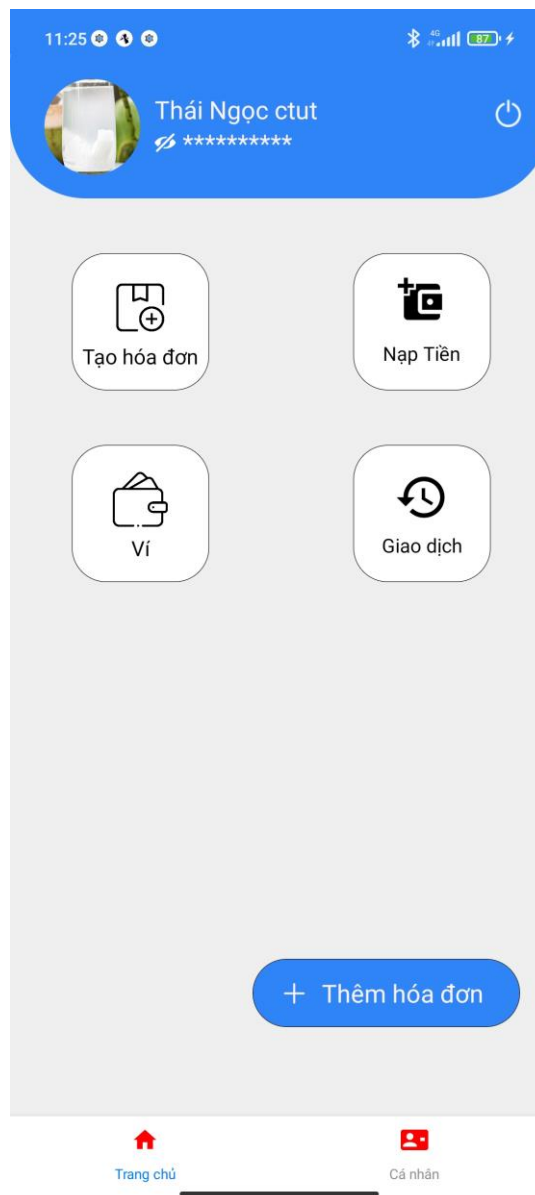
Khi người dùng chọn hình thức đăng nhập bằng tài khoản khác, hệ thống chuyển sang trang đăng nhập bằng username và password.



The image shows a mobile application interface for logging in. At the top is a black status bar with the time 3:03, notification icons, and connectivity symbols. Below this is a white arrow pointing left. The center features the CTUT logo, which is a circular emblem with 'TRƯỜNG ĐẠI HỌC KỸ THUẬT - CÔNG NGHỆ' around the top and 'CẦN THƠ' at the bottom, with 'CTUT' in the middle. Below the logo is the text 'Đăng nhập' in blue. There are two input fields: the first is labeled 'Username' in blue and has a blue border; the second is labeled 'Password' in gray and has a gray border. At the bottom is a blue button with the text 'LOGIN' in white.

Hình 4.4 Giao diện đăng nhập username và password

Sau khi đăng nhập thành công, hệ thống sẽ chuyển người dùng đến trang chủ theo vai trò của tài khoản đã đăng nhập.



Hình 4.5 Trang chủ sinh viên

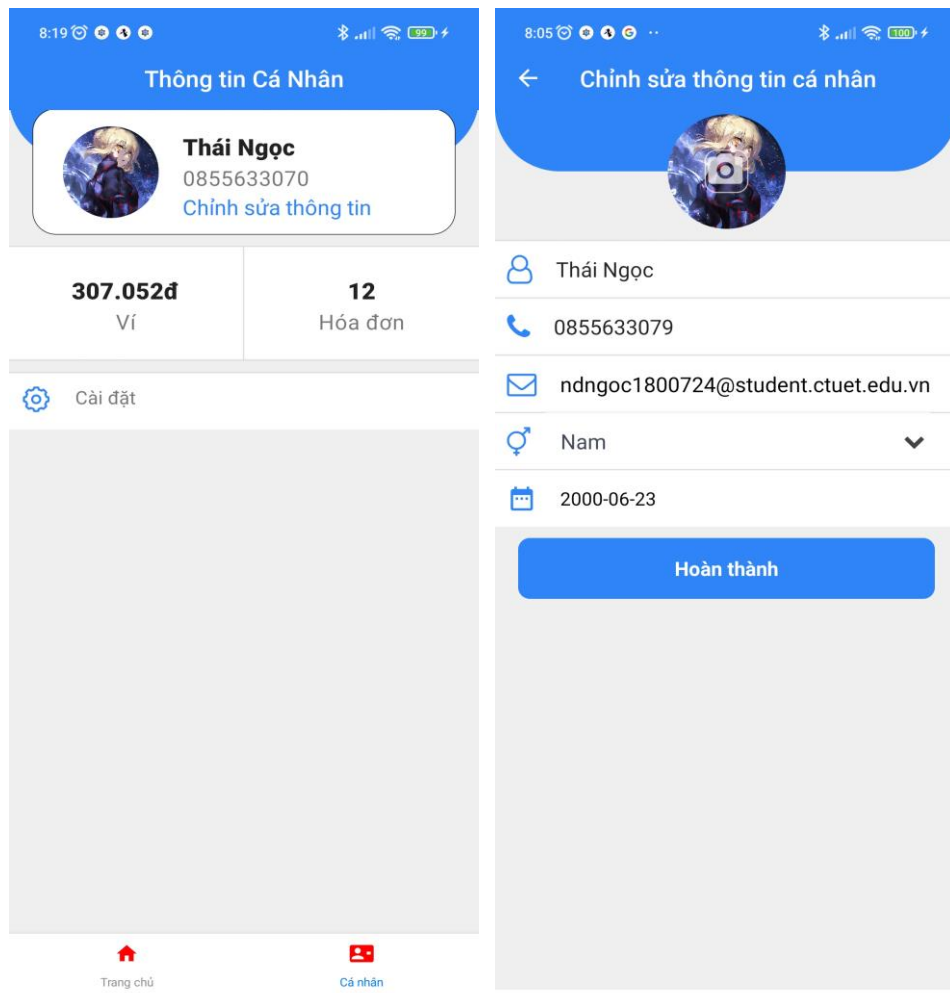
4.2 Thông tin cá nhân

Tại trang chủ, sinh viên có thể xem thông cá nhân khi chọn lệnh cá nhân nằm dưới cùng hoặc ấn vào ảnh đại diện, hệ thống sẽ chuyển đến trang thông tin cá nhân.

Ở trang thông tin cá nhân sẽ hiển thị các thông tin như số dư ví, số hóa đơn đã thanh toán, họ và tên.

Nếu muốn chỉnh sửa thông tin cá nhân, sinh viên chọn lệnh cài đặt hoặc chỉnh sửa thông tin, hệ thống sẽ chuyển đến trang chỉnh sửa thông tin cá nhân.

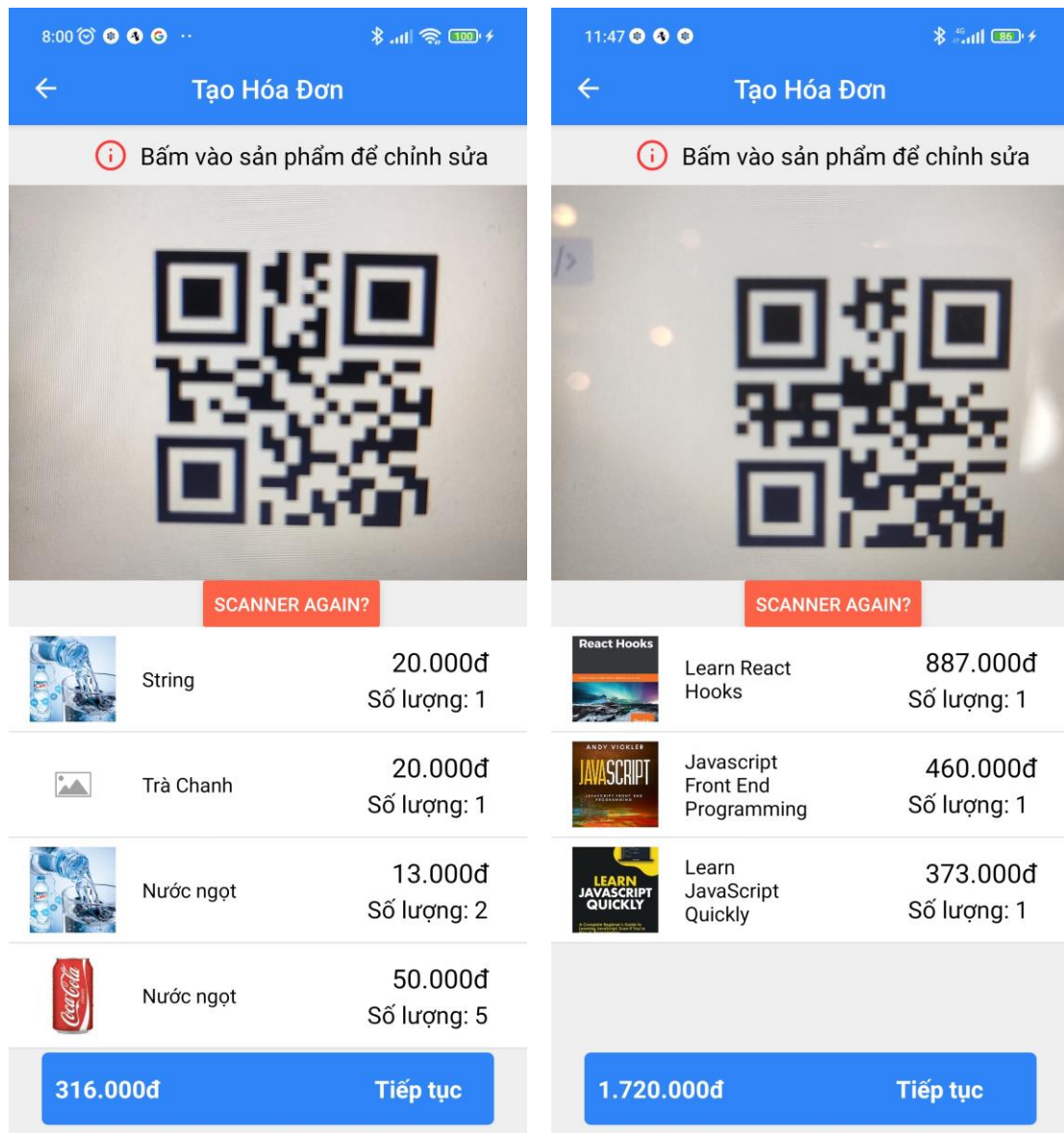
Sinh viên chỉnh sửa thông tin và chọn lệnh hoàn thành để cập nhật.



Hình 4.6 Giao diện thông tin và chỉnh sửa thông tin cá nhân

4.3 Tạo hóa đơn

Tại trang chủ, sinh viên chọn lệnh tạo đơn hoặc tạo hóa đơn để tạo hóa đơn.



Hình 4.7 Giao diện tạo hóa đơn

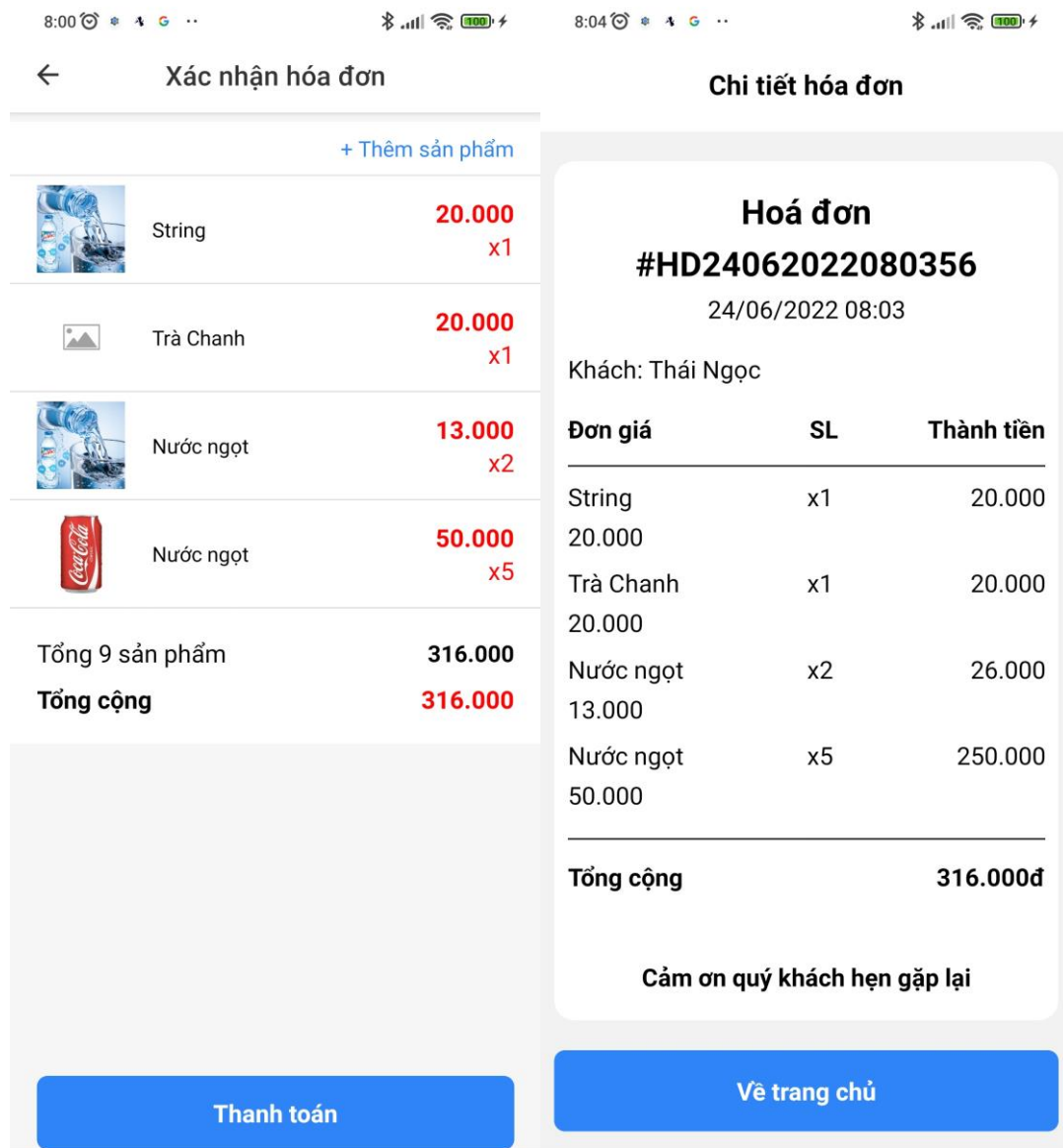
Sau khi tạo hóa đơn, sinh viên quét mã QR có trên sản phẩm để thêm sản phẩm vào hóa đơn.

Khi muốn chỉnh sửa số lượng sản phẩm, sinh viên ấn vào sản phẩm, hệ thống sẽ chuyển đến trang chỉnh sửa sản phẩm.



Hình 4.8 Giao diện sửa sản phẩm hóa đơn

Sau khi hoàn tất quá trình thêm sản phẩm vào hóa đơn, sinh viên ấn tiếp tục để xác nhận đơn hàng, hệ thống sẽ chuyển đến trang xác nhận hóa đơn.



Hình 4.9 Giao diện xác nhận hóa đơn và chi tiết hóa đơn

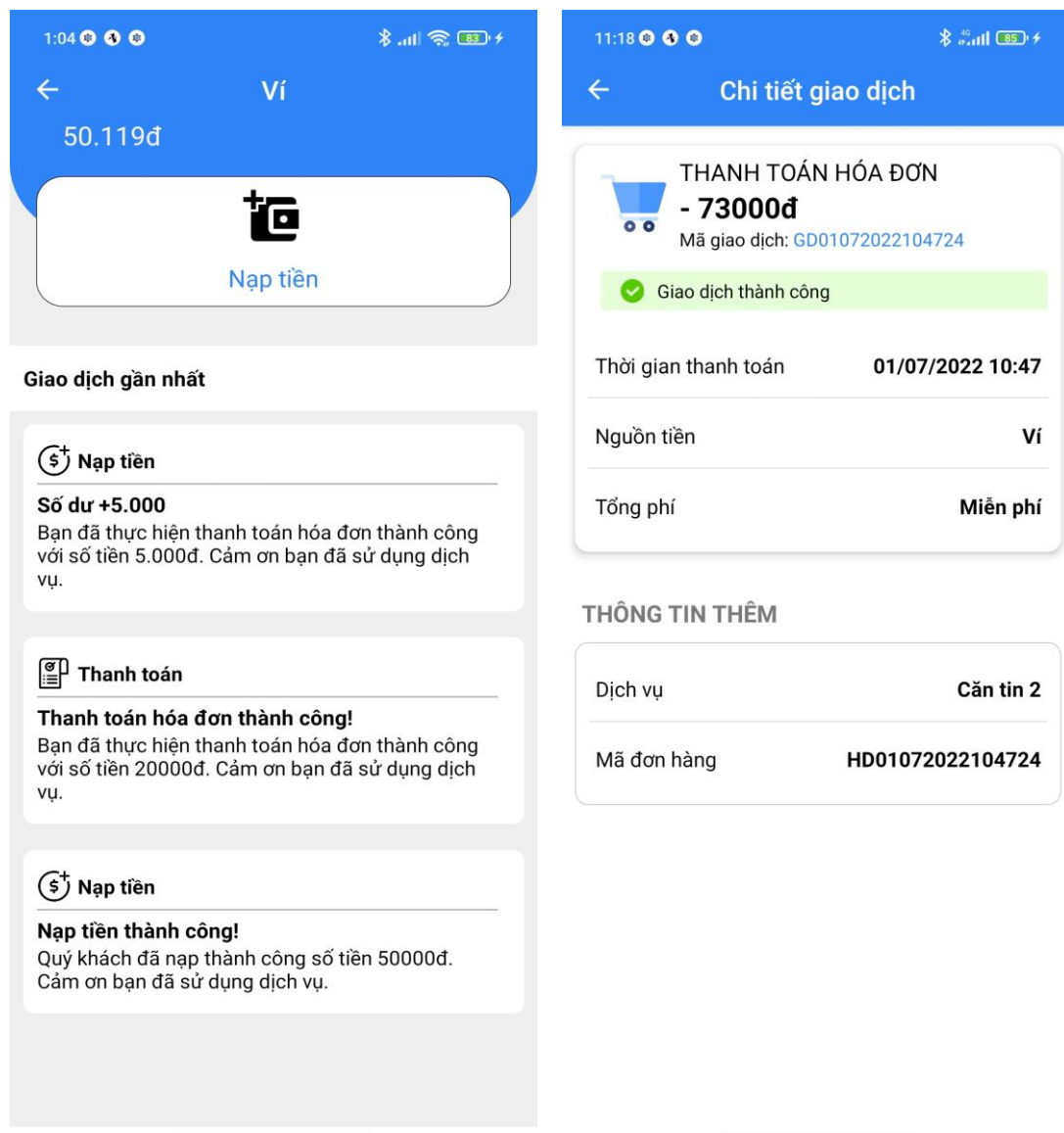
Sau khi đã xác nhận hóa đơn đúng, sinh viên ấn thanh toán, hệ thống tiến hành trừ tiền trong ví và thêm hóa đơn vào cơ sở dữ liệu.

Hệ thống sẽ chuyển sang trang chi tiết hóa đơn.

4.4 Ví

Ở chức năng này sinh viên có thể xem số dư ví, xem lịch sử giao dịch và nạp tiền đơn giản cho ví.

Sinh viên chọn giao dịch cần xem chi tiết và ấn vào dòng hiển thị giao dịch, hệ thống sẽ chuyển sang trang chi tiết giao dịch

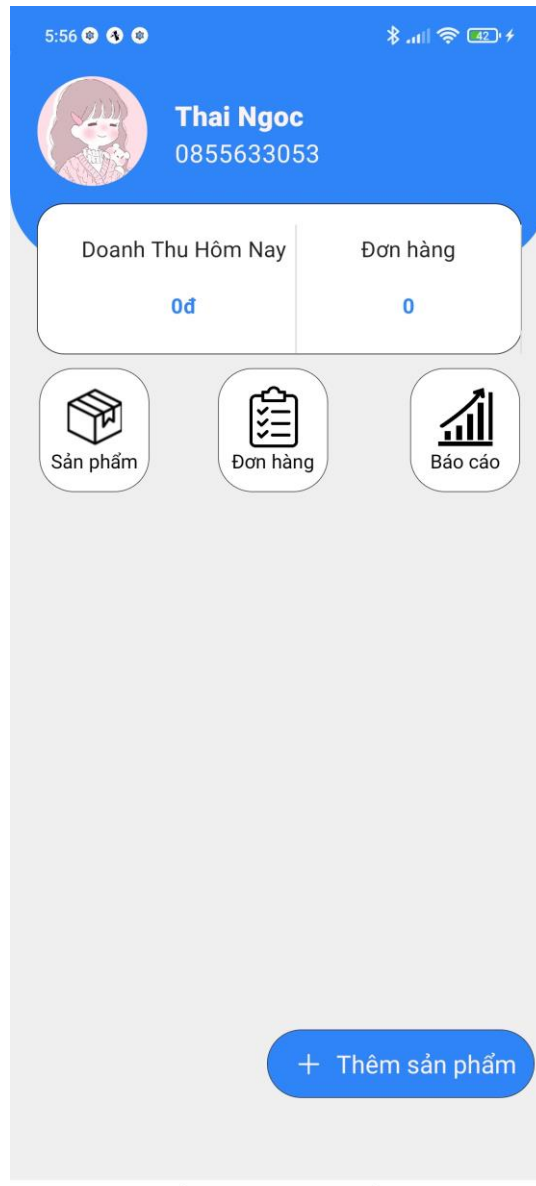


Hình 4.11 Giao diện ví và chi tiết giao dịch

4.5 Quản lý sản phẩm

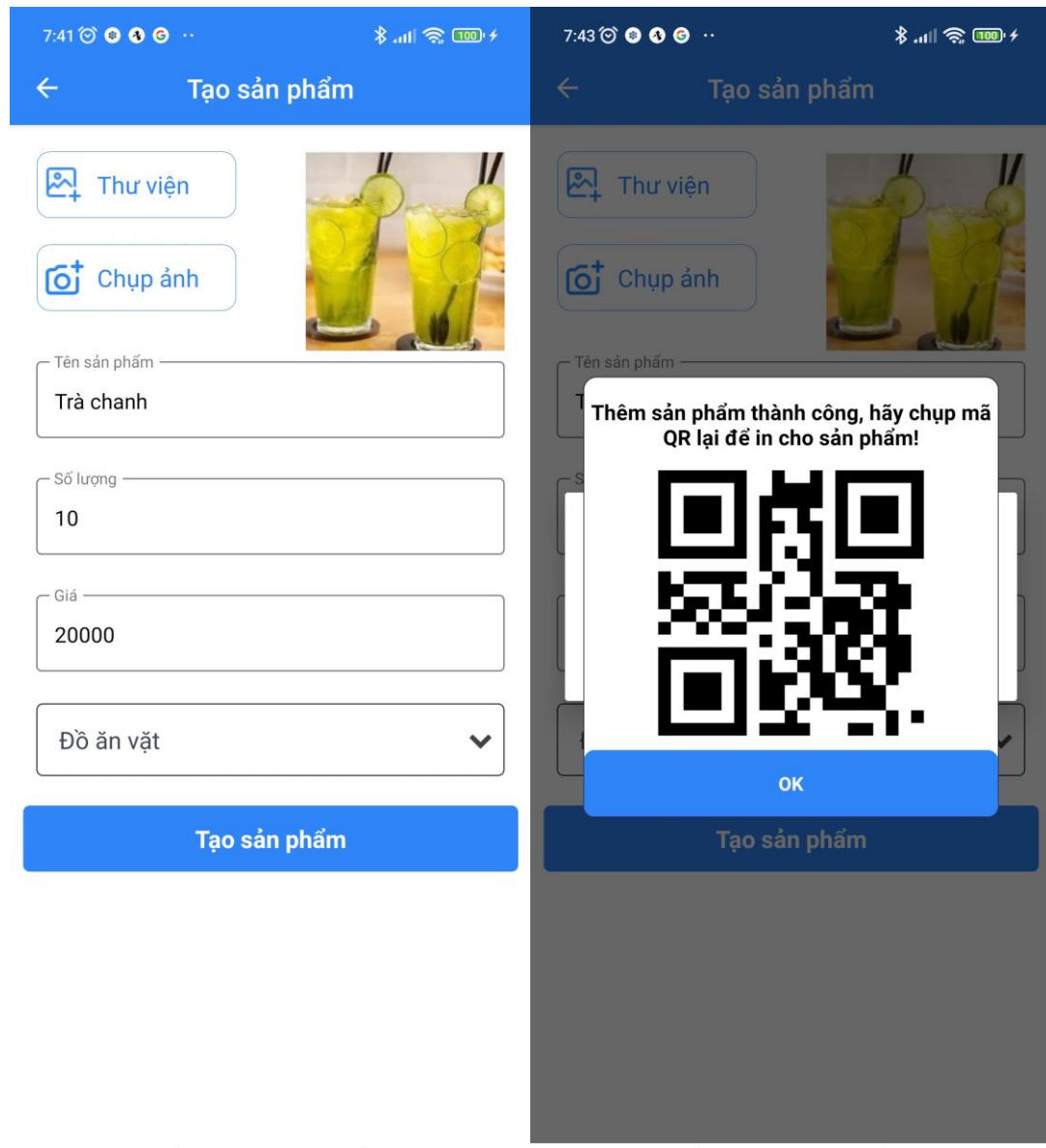
4.5.1 Tạo sản phẩm

Nhân viên quản lý mở ứng dụng, đăng nhập bằng username và password mà admin đã cung cấp để truy cập trang quản lý.



Hình 4.12 Giao diện trang chủ quản lý

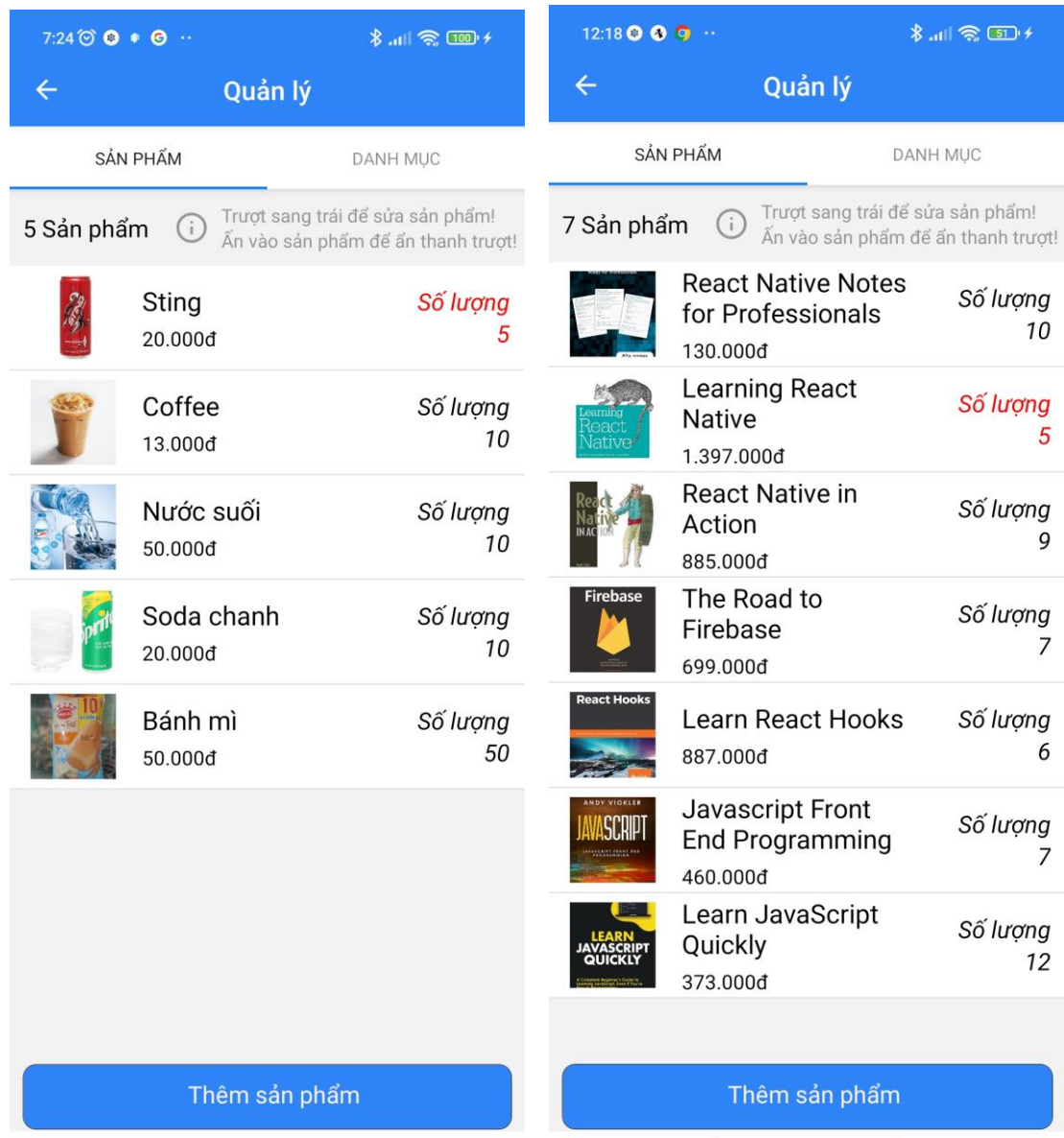
Trên giao diện trang chủ, nhân viên quản lý chọn lệnh thêm sản phẩm để thêm sản phẩm. Hệ thống sẽ chuyển đến trang tạo sản phẩm.



Hình 4.13 Giao diện tạo sản phẩm

Nhân viên điền đầy đủ thông tin sản phẩm, chọn ảnh cho sản phẩm, sau khi hoàn tất chọn lệnh tạo sản phẩm. Sau khi thêm sản phẩm thành công hệ thống sẽ hiển thị mã QR của sản phẩm dựa trên mã sản phẩm. Nhân viên quản lý cần lưu ảnh này lại để in cho sản phẩm.

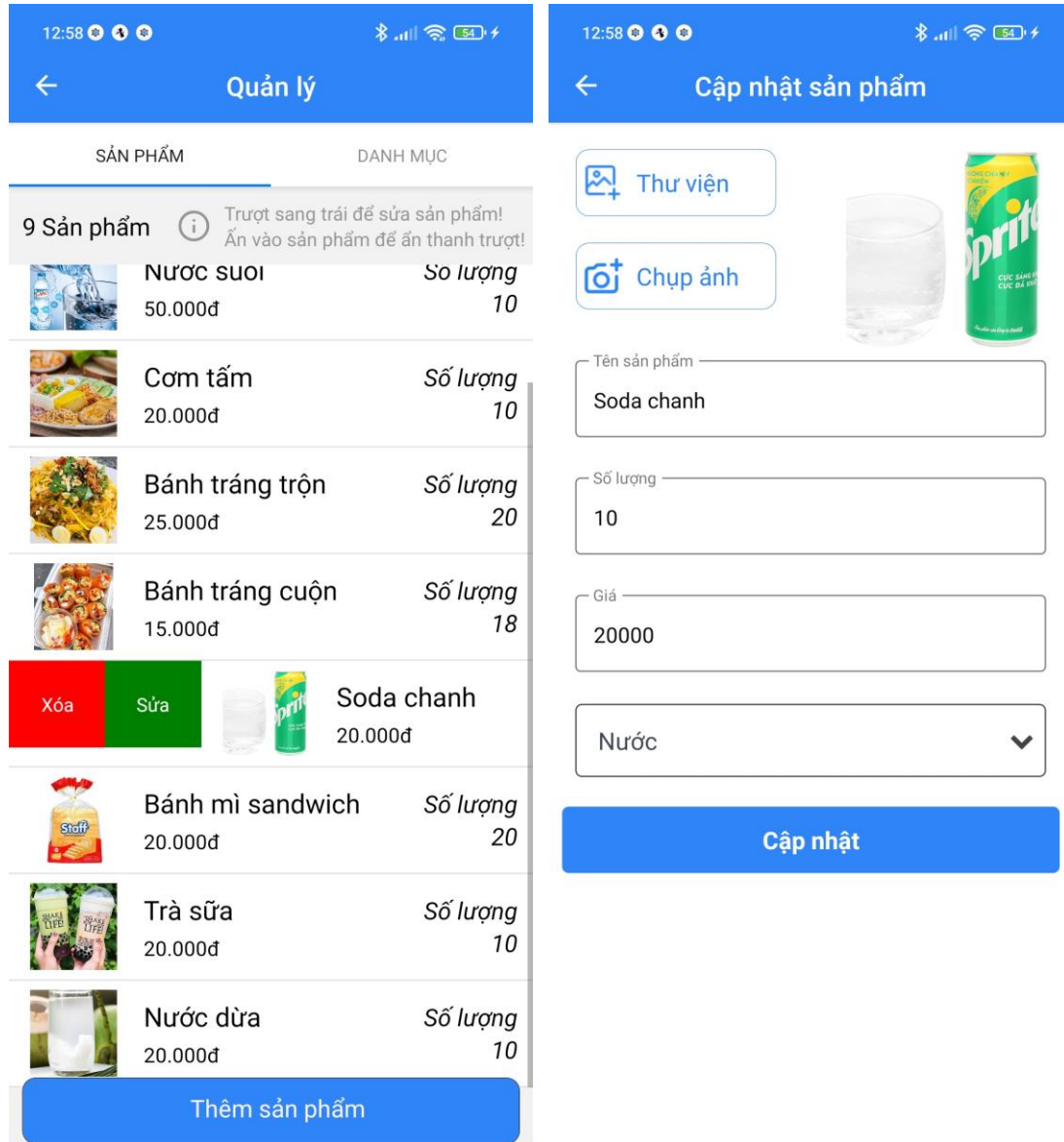
Nhân viên có thể xem danh sách sản phẩm khi chọn lệnh sản phẩm trên giao diện trang chủ.



Hình 4.14 Giao diện quản lý sản phẩm

4.5.2 Cập nhật và xóa sản phẩm

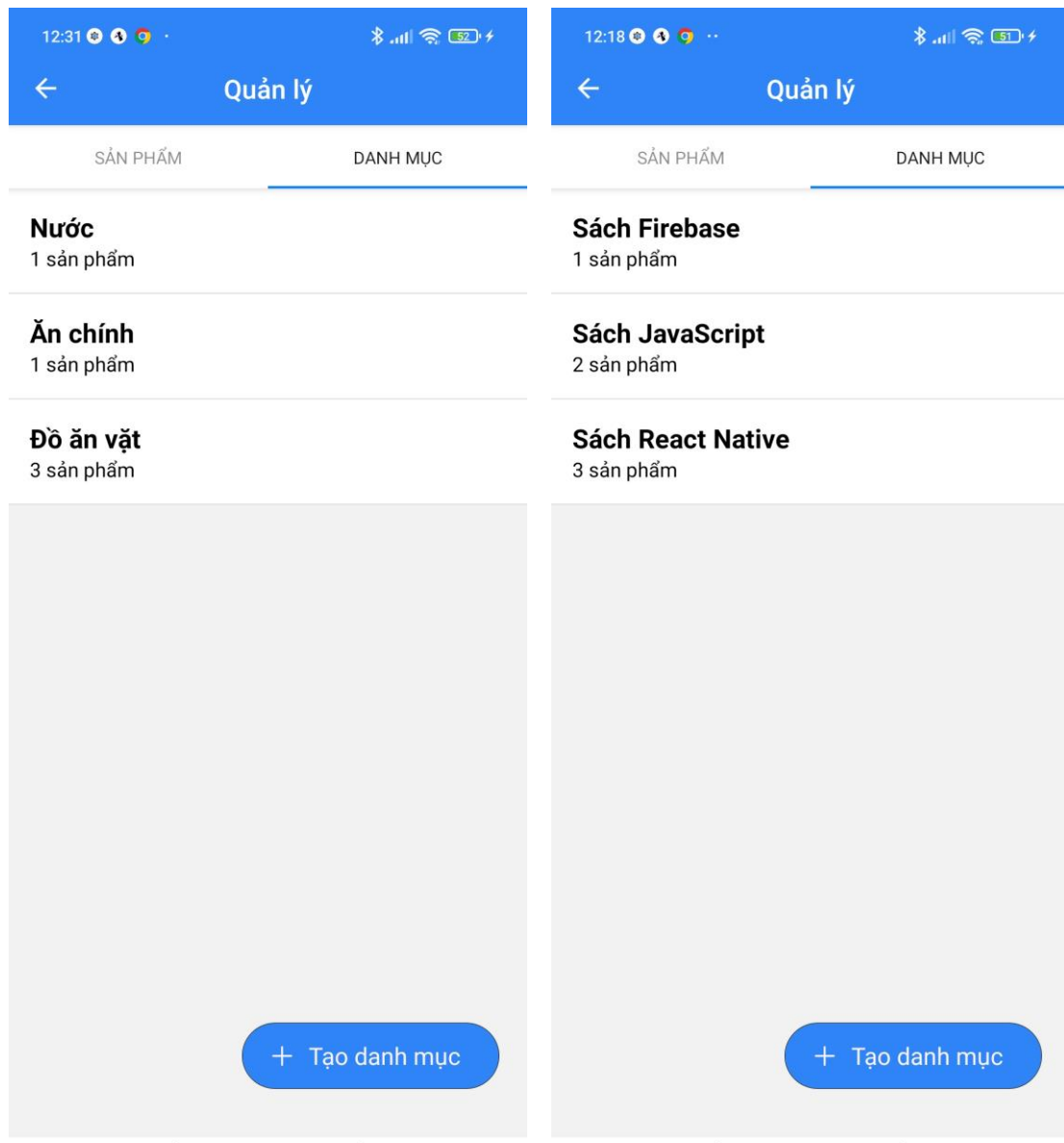
Tại giao diện quản lý sản phẩm, nhân viên quản lý chọn và vuốt sang phải sản phẩm cần cập nhật hoặc xóa. Sau đó nhân viên quản lý chọn lệnh “Sửa” hoặc lệnh “Xóa”, nếu nhân viên chọn lệnh “Sửa” hệ thống sẽ chuyển sang trang cập nhật sản phẩm, nếu chọn lệnh “Xóa” hệ thống hiển thị yêu cầu xác nhận xóa sản phẩm, nhân viên quản lý cần xác nhận để xóa sản phẩm.



Hình 4.15 Giao diện cập nhật hệ thống

4.5.2 Tạo danh mục

Nhân viên quản lý chọn mục “Danh mục” trên giao diện quản lý để xem danh sách danh mục hiện có trong dịch vụ của mình.

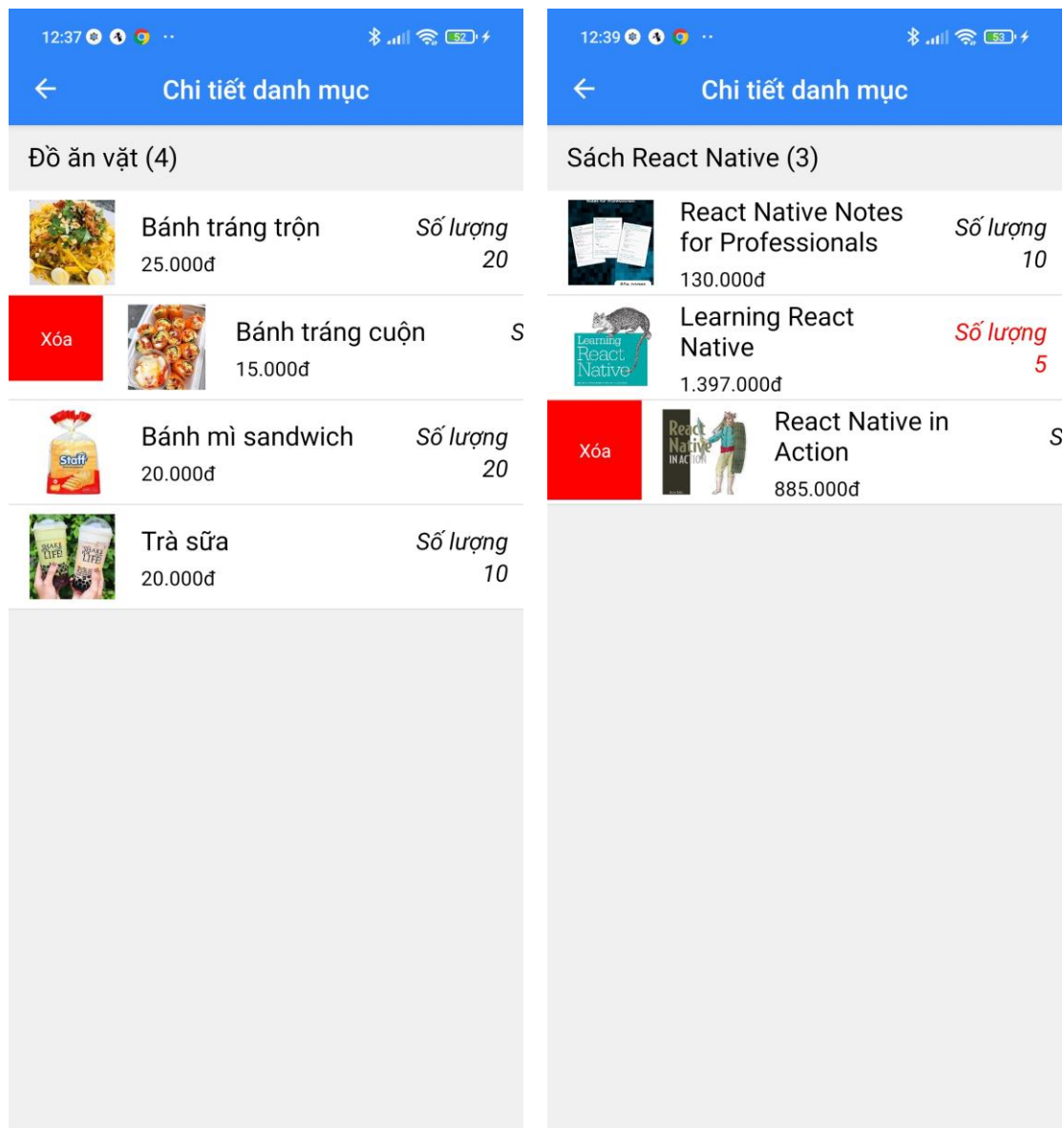


Hình 4.16 Giao diện quản lý danh mục

Để thêm danh mục nhân viên quản lý chọn lệnh tạo danh mục trên giao diện quản lý danh mục, hệ thống sẽ hiển thị cửa sổ yêu cầu nhân viên quản lý nhập tên danh mục. Sau khi nhân viên quản lý đã nhập tên danh mục ấn chọn lệnh tạo danh mục để tạo danh mục mới.

The screenshot displays a mobile application interface. At the top, a blue header bar contains a back arrow and the text 'Quản lý'. Below this, two tabs are visible: 'SẢN PHẨM' and 'DANH MỤC', with the latter being the active tab. The main content area lists two categories: 'Nước' with '1 sản phẩm' and 'Đồ ăn vặt' with '0 sản phẩm'. A modal dialog titled 'Thêm danh mục' (Add category) is open, featuring a text input field labeled 'Tên danh mục' (Category name) containing the text 'Mì gói', and a blue button labeled 'Thêm danh mục' (Add category). At the bottom of the screen, there is a blue button with a plus icon and the text '+ Tạo danh mục' (Create category).

Hình 4.17 Cửa sổ tạo danh mục

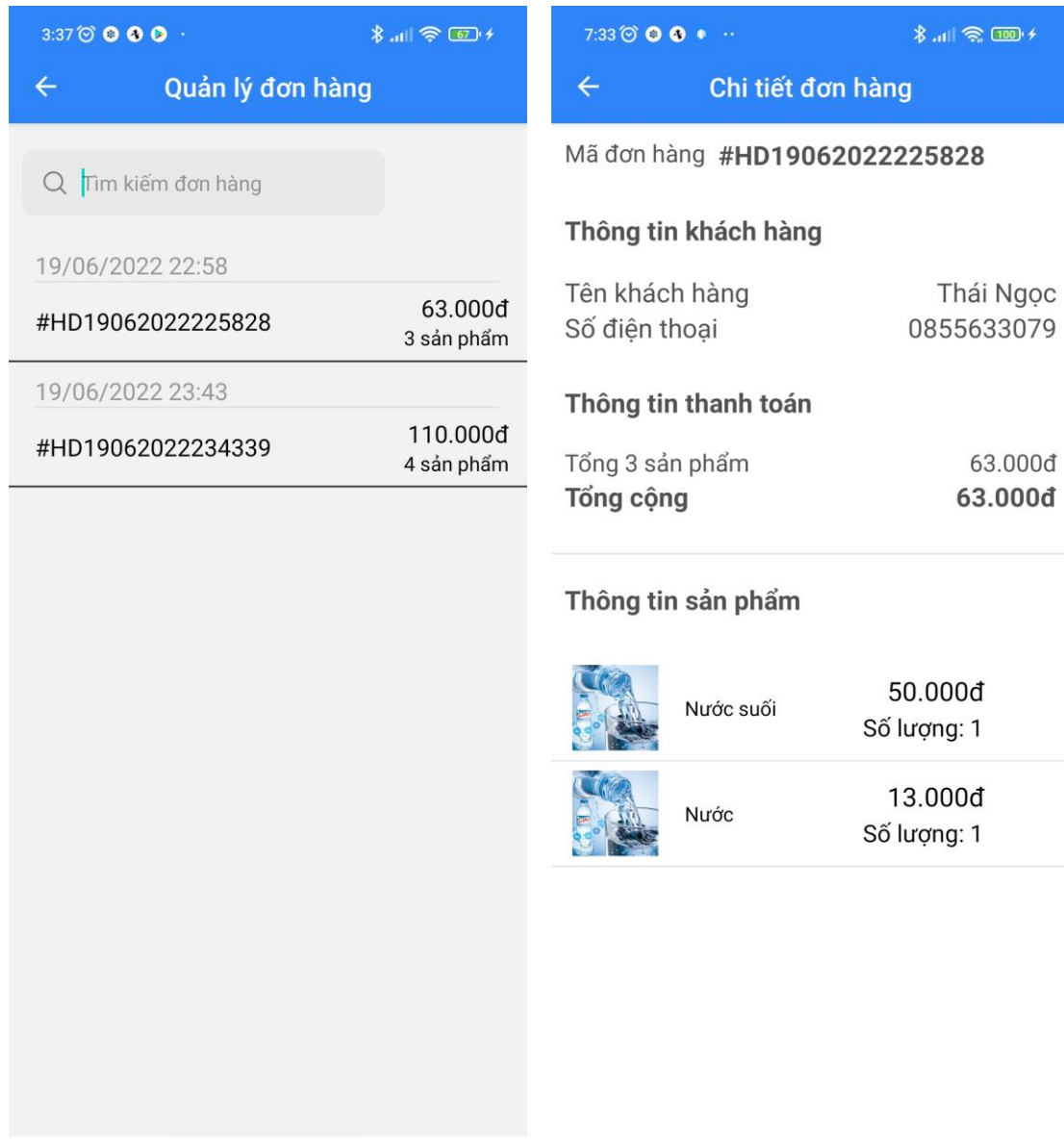


Hình 4.18 Giao diện chi tiết danh mục

Trang chi tiết danh mục sẽ hiển hiện danh sách sản phẩm trang có trong danh mục đã chọn. Nếu muốn xóa sản phẩm trong danh mục, vuốt sang phải sản phẩm cần xóa và chọn lệnh “Xóa”. Hệ thống sẽ hiện thị thông báo yêu cầu xác nhận xóa sản phẩm, nhân viên quản lý cần xác nhận để xóa sản phẩm ra khỏi danh mục.

4.6 Quản lý đơn hàng

Tại trang quản lý đơn hàng, nhân viên quản lý tìm kiếm hóa đơn trên thanh tìm kiếm, sau khi nhân viên quản lý đã tìm được đơn hàng cần tìm ấn vào đơn hàng để xem chi tiết đơn hàng.



Hình 4.19 Giao diện đơn hàng và chi tiết đơn hàng

4.7 Báo cáo thu nhập:

Chọn chức năng “Báo cáo” trên trang chủ quản lý để xem thống kê thu nhập.

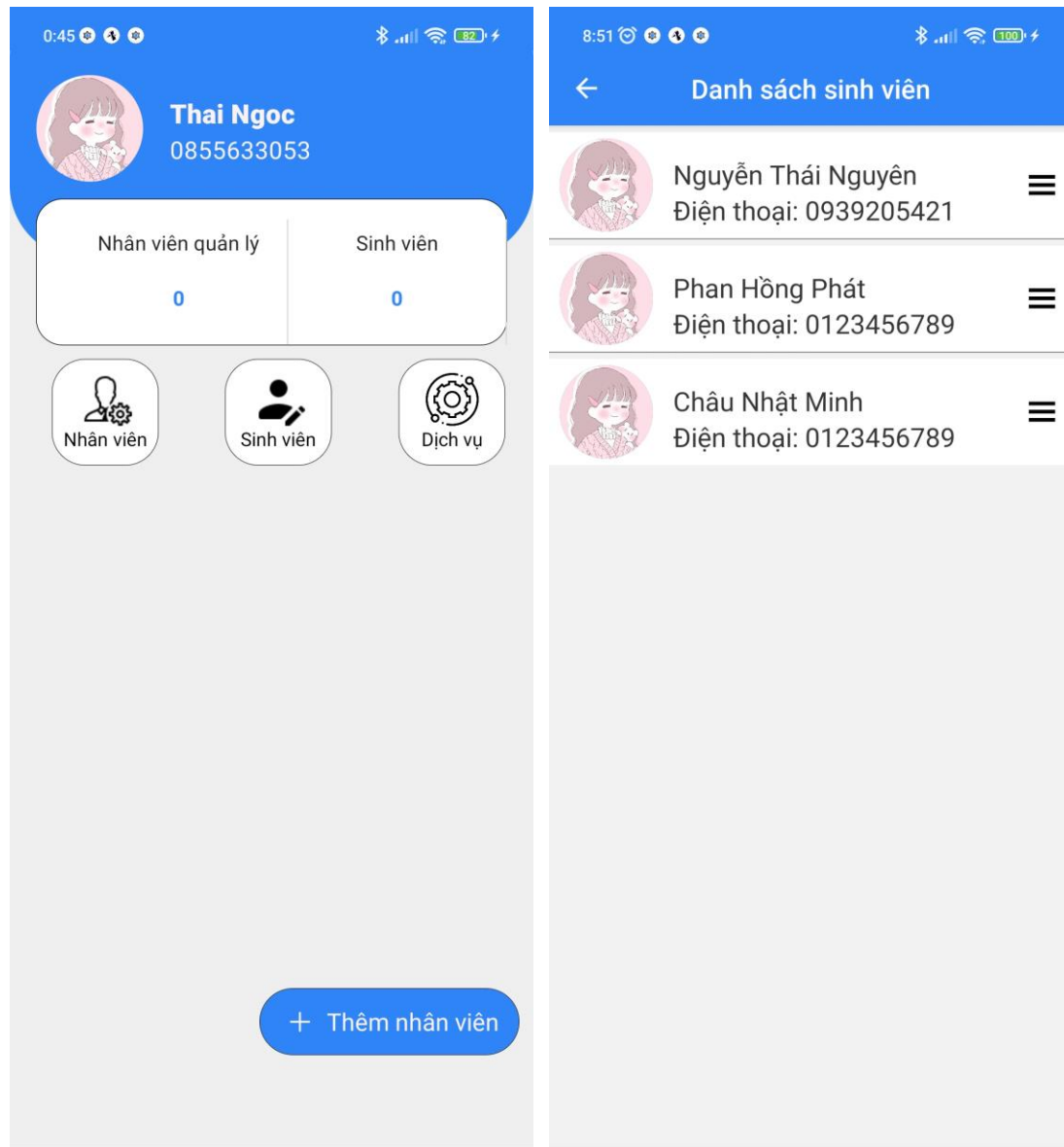
Có thể chọn mốc thời gian để xem thu nhập theo thời gian.



Hình 4.20 Giao diện báo cáo thu nhập

4.8 Quản lý sinh viên:

Chọn chức năng “Sinh viên” trên trang chủ quản trị viên để xem danh sách sinh viên đã đăng ký ứng dụng.

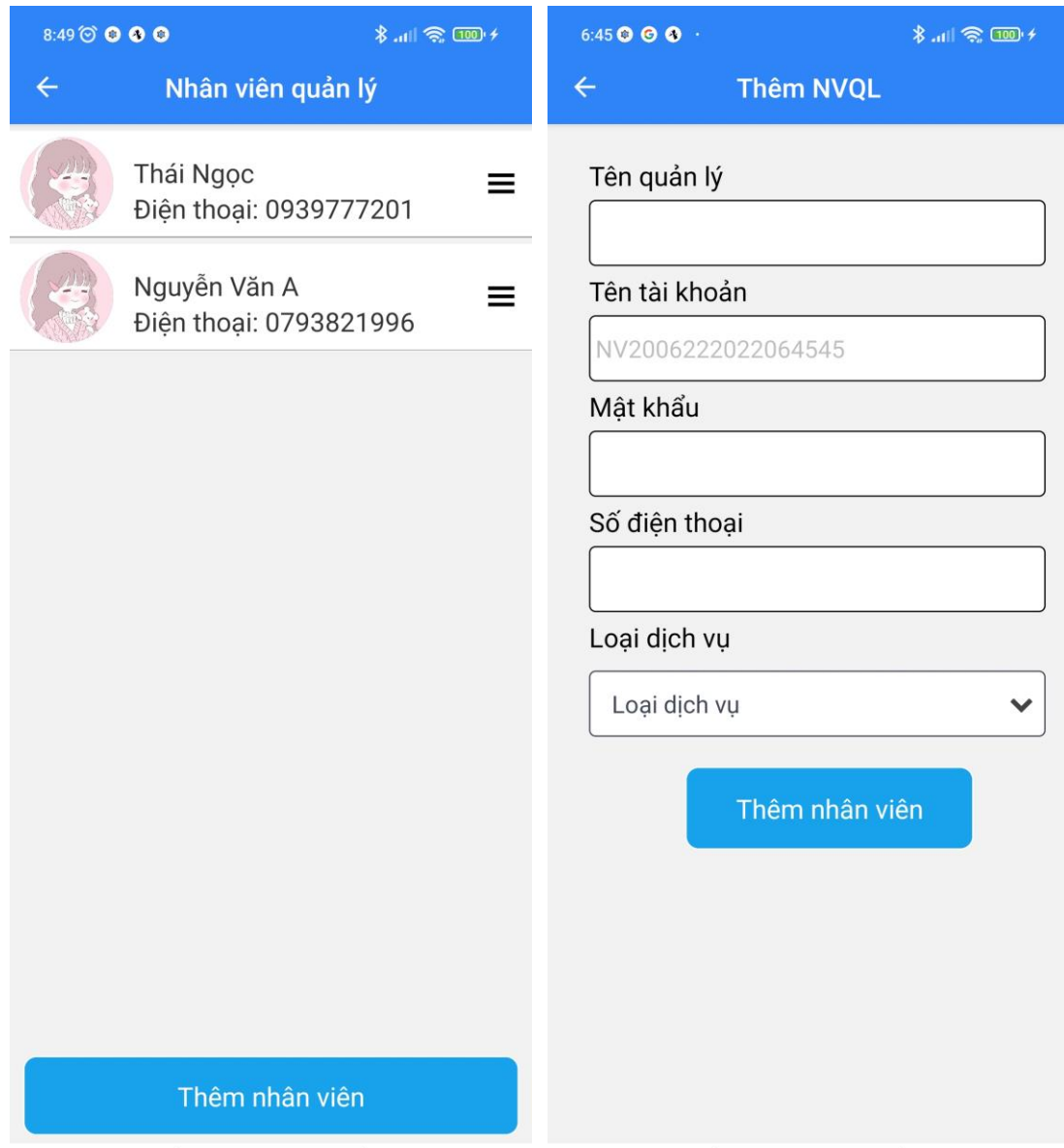


Hình 4.21 Trang chủ quản trị viên và danh sách sinh viên

4.9 Quản lý nhân viên

Chọn chức năng “Nhân viên” trên trang chủ quản trị viên để xem danh sách nhân viên quản lý đã đăng ký ứng dụng.

Tại trang danh sách nhân viên quản lý chọn “Thêm nhân viên quản lý” để thêm nhân viên.

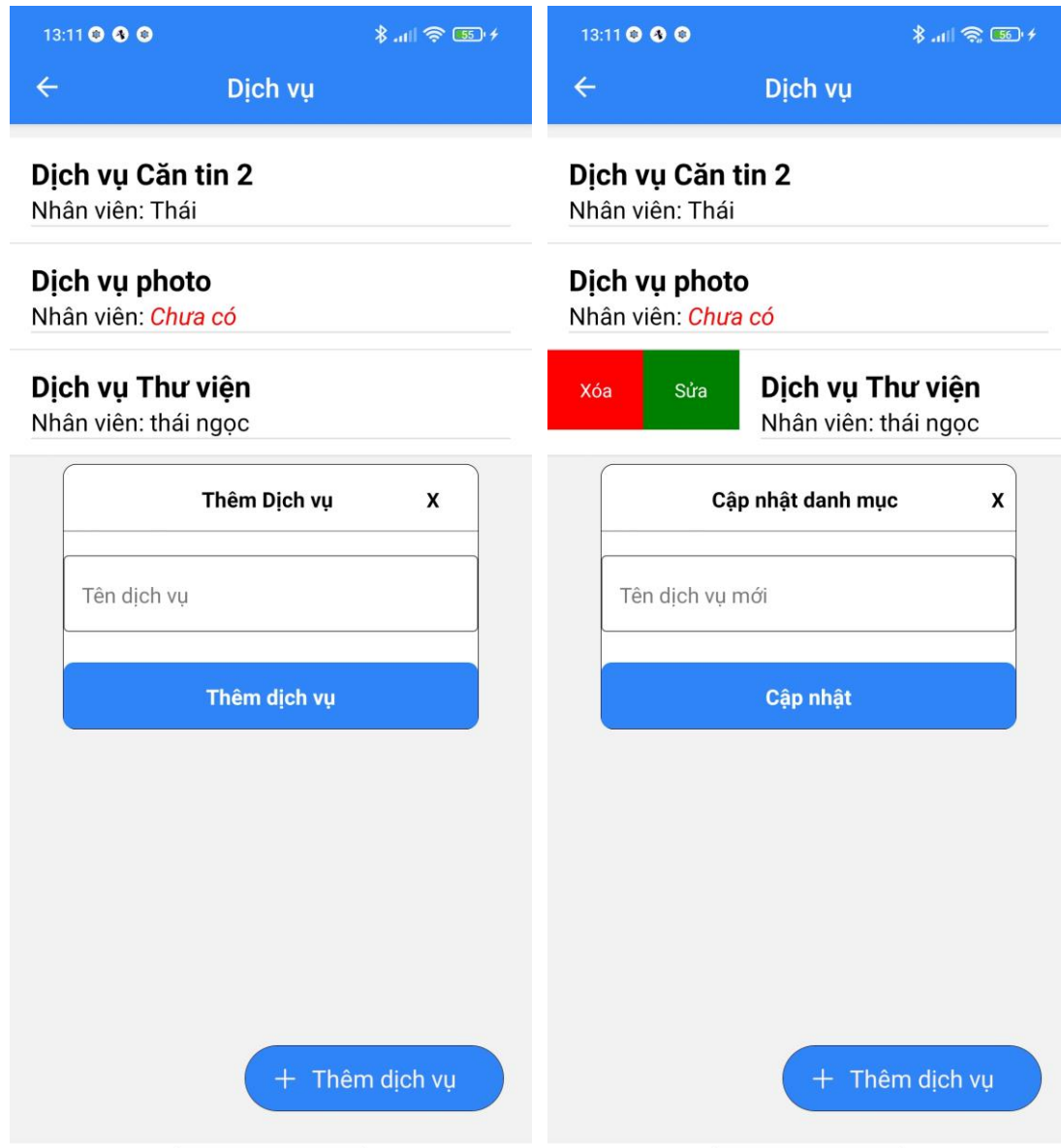


Hình 4.22 Giao diện danh sách nhân viên và thêm tài khoản nhân viên

4.10 Quản lý dịch vụ

Chọn lệnh “Dịch vụ” trên trang chủ quản trị viên để xem danh sách dịch vụ và thêm dịch vụ.

Nếu muốn xóa hoặc sửa dịch vụ trượt sang phải dịch vụ, chọn lệnh “Xóa” để xóa dịch vụ hoặc chọn lệnh “Sửa” để cập nhật dịch vụ.



Hình 4.23 Giao diện quản lý dịch vụ

KẾT LUẬN

1. Kết quả đạt được:

Thông qua bài luận này tôi đã giới thiệu một số vấn đề liên quan tới ứng dụng di động và việc ứng dụng vào việc quản lý dịch vụ trong nhà trường. Đồng thời xây dựng một ứng dụng quản lý dịch vụ cung cấp cho sinh viên của trên nền tảng di động.

Việc xây dựng một ứng dụng giúp hỗ trợ quản lý các dịch vụ thông qua nền tảng di động là một hướng khá hiệu quả để quản lý các dịch vụ. Hiện nay các ứng dụng trên điện thoại di động đang ngày càng phổ biến, nhưng việc xây dựng ra một ứng dụng hoàn chỉnh trong một thời gian ngắn cũng gặp nhiều vấn đề khó khăn.

Kết quả bước đầu đạt được là xây dựng một ứng dụng quản lý dịch vụ và có thể cho phép sinh viên thanh toán bằng ví ảo đơn giản với một số chức năng cơ bản:

- Chức năng quản lý dịch vụ và các sản phẩm có trong dịch vụ.
- Cho phép sinh viên quét mã QR trên sản phẩm để thêm vào hóa đơn và thanh toán trực tiếp trên ví ảo mà không cần tiền mặt.
- Tự quản lý thông tin cá nhân của chủ tài khoản.
- Sinh viên có thể tra cứu được chi tiêu trong quá khứ.

2. Hạn chế và hướng phát triển:

2.1 Hạn chế:

Do điều kiện thời gian cũng như kiến thức còn hạn chế nên bên cạnh các chức năng đã làm được thì ứng dụng cũng còn rất nhiều nhược điểm, chưa đầy đủ chức năng cần thiết. Những nhược điểm như:

- Ứng dụng chỉ mới thực hiện thanh toán bằng ví ảo, chưa sử dụng được ví điện tử thực tế. Chức năng thống kê thu nhập còn nhiều hạn chế
- Hiệu suất hoạt động của ứng dụng còn chưa được tối ưu tốt, xảy ra tình trạng chậm.
- Việc quản lý giữa mã QR code và sản phẩm chưa tốt.
- Chưa kiểm tra được dữ liệu đầu vào là tiếng Việt, gây bất hợp lý cho cơ sở dữ liệu.

2.2 Hướng phát triển:

- Hoàn thiện chức năng thanh toán sử dụng ví điện tử thực tế, thống kê thu nhập.
- Tra cứu được mã QR code của sản phẩm.
- Xây dựng thêm các chức năng quản lý cho admin.

TÀI LIỆU THAM KHẢO

1. Gary B. Shelly & Harry J. Rosenblatt (2011), *Information Systems Analysis and Design*, Nicole Pinard, United States of America.
2. Harry J. Rosenblatt (2013), *Systems Analysis and Design*, Course Technology, United States of America.
3. Dan Ward (January 31, 2019), *React Native Cookbook: Recipes for solving common React Native development problems, 2nd Edition*, Packt Publishing, Birmingham.
4. Houssein Yahiaoui (November 29, 2017), *Firebase Cookbook*, Packt Publishing, Birmingham.
5. Laurence Moroney (November 12, 2017), *The Definitive Guide to Firebase*, Laurence Moroney, Apress, Washinton, USA.
6. React native(6/2022), <https://reactnative.dev>.
7. Firebase(6/2022), <https://firebase.google.com/docs/build>.
8. Expo (6/2022), <https://docs.expo.dev/>.