

TRƯỜNG ĐẠI HỌC QUẢNG BÌNH

KHOA CÔNG NGHỆ THÔNG TIN

PHẠM XUÂN HÒA

NGHIÊN CỨU VÀ XÂY DỰNG ỨNG DỤNG SỔ TAY
DU LỊCH QUẢNG BÌNH TRÊN ĐIỆN THOẠI DI ĐỘNG

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

QUẢNG BÌNH, 2024

TRƯỜNG ĐẠI HỌC QUẢNG BÌNH
KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

**NGHIÊN CỨU VÀ XÂY DỰNG ỨNG DỤNG SỔ TAY
DU LỊCH QUẢNG BÌNH TRÊN ĐIỆN THOẠI DI ĐỘNG**

Họ tên sinh viên: Phạm Xuân Hòa

Mã số sinh viên: 20D116005

Chuyên ngành: Công nghệ thông tin

Giảng viên hướng dẫn: TS. Hoàng Văn Thành

QUẢNG BÌNH, 2024

LỜI CAM ĐOAN

Tôi xin cam đoan đồ án tốt nghiệp với đề tài là công trình thuộc quyền sở hữu duy nhất của tôi. Những tài liệu tham khảo được sử dụng trong đồ án đã được trích dẫn và nêu rõ trong mục Tài liệu tham khảo. Bên cạnh đó, những kết quả nghiên cứu hoàn toàn mang tính chất trung thực, không sao chép, đạo nhái từ bất kỳ công trình nào trước đây.

Nếu những lời cam đoan trên của tôi không chính xác, tôi xin chịu hoàn toàn trách nhiệm và chịu mọi hình thức kỷ luật từ khoa và nhà trường.

Sinh viên

Phạm Xuân Hòa

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the entire width of the page. There are no margins, text, or other markings present.

(ký và ghi rõ họ tên)



Lời Cảm Ơn

Trong quá trình thực hiện báo cáo Khóa luận tốt nghiệp này em đã nhận được sự quan tâm giúp đỡ tận tình của nhiều tập thể, cá nhân trong và ngoài trường.

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc tới giảng viên TS. Hoàng Văn Thành – Người đã trực tiếp hướng dẫn em trong suốt quá trình nghiên cứu và thực hiện đề tài.

Em xin chân thành cảm ơn các thầy giáo, cô giáo trong khoa Kỹ Công nghệ thông tin Trường Đại học Quảng Bình đã tạo mọi điều kiện thuận lợi giúp đỡ em trong suốt quá trình học tập và rèn luyện tại trường.

Em xin bày tỏ lòng biết ơn vô hạn tới gia đình, người thân và bạn bè đã luôn sát cánh bên em trong suốt quá trình học tập và nghiên cứu đề tài.

Mặc dù trong quá trình nghiên cứu đề tài, bản thân em có những cố gắng nhất định, song do trình độ và thời gian có hạn nên đề tài không tránh khỏi những thiếu sót. Vậy kính mong các thầy cô giáo và các bạn đóng góp ý kiến để đề tài của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!



MỤC LỤC

PHẦN 1: MỞ ĐẦU.....	1
1. Lý do chọn đề tài	1
2. Mục đích nghiên cứu.....	2
3. Nội dung nghiên cứu	2
4. Đối tượng nghiên cứu	2
5. Thời gian nghiên cứu	2
6. Phương pháp nghiên cứu	3
PHẦN 2: NỘI DUNG.....	4
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	4
1.1. Du lịch tỉnh Quảng Bình.....	4
1.2. Tìm hiểu về Flutter	8
1.3. Các thành phần quan trọng trong Flutter	13
1.4. Giới thiệu về ngôn ngữ lập trình Dart	19
1.5. Tìm hiểu về Firebase	21
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	26
2.1. Thực trạng và tính cấp thiết của đề tài.....	26
2.2. Mục tiêu và phạm vi nghiên cứu	26
2.3. Mô hình sử dụng.....	26
2.4. Phân tích chức năng của hệ thống	29
2.5. Sơ đồ mô tả chức năng chính của hệ thống.....	30
2.6. Use case hệ thống	31
2.7. Mô tả chức năng của từng use case	34
2.8. Sơ đồ tuần tự các chức năng	39
2.9. Lược đồ cơ sở dữ liệu	43
CHƯƠNG 3: CÀI ĐẶT VÀ GIAO DIỆN CỦA HỆ THỐNG.....	46
3.1. Cài đặt hệ thống.....	46
3.2. Giao diện đăng ký và đăng nhập hệ thống	48
3.3. Giao diện trang chủ hệ thống.....	49

3.4. Giao diện xem địa điểm du lịch.....	50
3.5. Giao diện tìm kiếm địa điểm du lịch	52
3.6. Giao diện quản lý thông tin cá nhân.....	53
3.7. Giao diện quản lý lịch trình	54
3.8. Giao diện gửi thông báo về email.....	60
3.9. Giao diện cài đặt hệ thống	61
PHẦN 3: KẾT LUẬN VÀ KIẾN NGHỊ.....	66
1. Kết luận	66
2. Kiến nghị.....	66
TÀI LIỆU THAM KHẢO	67

DANH MỤC CÁC BẢNG

Bảng 2-1: Bảng use case đăng ký.....	35
Bảng 2-2: Bảng use case đăng nhập	35
Bảng 2-3: Bảng use case quản lý địa điểm du lịch.....	36
Bảng 2-4: Bảng use case xem thông tin địa điểm du lịch	36
Bảng 2-5: Bảng use case tạo lịch trình du lịch	37
Bảng 2-6: Bảng use case xem lại lịch sử cảnh báo	37
Bảng 2-7: Bảng use case xem bản đồ chỉ đường.....	38
Bảng 2-8: Bảng use case xem thông tin cá nhân.....	38
Bảng 2-9: Bảng use case quản lý người dùng	39
Bảng 2-10: Bảng dữ liệu User	44
Bảng 2-11: Bảng dữ liệu Travel	44
Bảng 2-12: Bảng dữ liệu Trip.....	44
Bảng 2-13: Bảng dữ liệu Activity.....	45

DANH MỤC HÌNH ẢNH

Hình 1-1: Thành phố Đồng Hới hôm nay	4
Hình 1-2: Logo Flutter	9
Hình 1-3: Các tính năng của Flutter	10
Hình 1-4: Mô tả cây Widget	13
Hình 1-5: Một số thuộc tính của Text.....	14
Hình 1-6: Cài đặt path thư mục lưu nội dung trong pubspec.yaml	15
Hình 1-7: Hình ảnh ví dụ cách sử dụng Icon	16
Hình 1-8: Hình ảnh ví dụ về Column	16
Hình 1-9: Hình ảnh ví dụ về Center	17
Hình 1-10: Hình ảnh ví dụ về Padding.....	17
Hình 1-11: Hình ảnh ví dụ về StatelessWidget	18
Hình 1-12: Hình ảnh ví dụ về StatefulWidget.....	18
Hình 1-13: Khái niệm về Firebase	21
Hình 2-1: Mô hình MVVM(Model View ViewModel).....	27
Hình 2-2: Sơ đồ chức năng chính của hệ thống	30
Hình 2-3. Sơ đồ use case hệ thống tổng quát	31
Hình 2-4. Sơ đồ tổng quát phân rã chức năng cho người dùng.....	32
Hình 2-5: Sơ đồ Use case Tạo lịch trình du lịch	32
Hình 2-6: Sơ đồ Use case Xem lại lịch trình đã tạo	33
Hình 2-7: Sơ đồ Use case Xem thông tin cá nhân.....	33
Hình 2-8. Sơ đồ tổng quát phân rã chức năng cho admin	34
Hình 2-9. Sơ đồ tuần tự đăng nhập.....	39
Hình 2-10. Sơ đồ tuần tự đăng ký	40
Hình 2-11. Sơ đồ tuần tự đăng xuất.....	40
Hình 2-12: Sơ đồ tuần tự chỉnh sửa thông tin cá nhân.....	41
Hình 2-13: Sơ đồ tuần tự hiển thị địa điểm	41
Hình 2-14: Sơ đồ tuần tự Tạo lịch trình	42
Hình 2-15: Sơ đồ tuần tự xóa lịch trình.....	42

Hình 2-16: Sơ đồ tuần tự thêm hoạt động	43
Hình 2-17: Lược đồ cơ sở dữ liệu	43
Hình 3-1: Thêm Flutter SDK vào Path.....	46
Hình 3-2: Giao diện đăng nhập	48
Hình 3-3: Giao diện đăng ký	48
Hình 3-4: Giao diện trang chủ.....	49
Hình 3-5: Giao diện trang chủ.....	49
Hình 3-6: Giao diện xem chi tiết một địa điểm (tab giới thiệu).....	50
Hình 3-7: Giao diện xem chi tiết một địa điểm (tab bản đồ)	50
Hình 3-8: Giao diện xem chi tiết một địa điểm (tab video)	50
Hình 3-9: Giao diện xem chi tiết một địa điểm (tab video)	50
Hình 3-10: Giao diện trang chi tiết bản đồ.....	51
Hình 3-11: Giao diện trang chi tiết bản đồ	51
Hình 3-12: Giao diện chi tiết trang du lịch ảo	51
Hình 3-13: Giao diện chi tiết trang du lịch ảo đòi cát quang phú	51
Hình 3-14: Giao diện trang tìm kiếm	52
Hình 3-15: Giao diện đang tìm kiếm một địa điểm theo tên.....	52
Hình 3-16: Giao diện xem trang thông tin cá nhân	53
Hình 3-17: Giao diện chỉnh sửa tên người dùng	53
Hình 3-18: Giao diện chỉnh sửa số điện thoại	53
Hình 3-19: Giao diện chỉnh sửa ảnh đại diện.....	53
Hình 3-20: Giao diện tạo lịch trình	54
Hình 3-21: Giao diện thông báo tạo lịch trình thành công.....	54
Hình 3-22: Giao diện thêm hoạt động cho lịch trình.....	55
Hình 3-23: Giao diện thêm hoạt động cho lịch trình.....	55
Hình 3-24: Giao diện thêm hoạt động cho lịch trình.....	55
Hình 3-25: Giao diện thêm hoạt động cho lịch trình.....	55
Hình 3-26: Giao diện xem lại lịch trình đã tạo.....	56
Hình 3-27: Giao diện xem lại lịch trình chi tiết	56

Hình 3-28: Giao diện xem lại lịch trình chi tiết (tt)	57
Hình 3-29: Giao diện xóa lịch trình đã tạo	57
Hình 3-30: Giao diện chỉnh sửa lịch trình chi tiết.....	58
Hình 3-31: Giao diện chỉnh sửa lịch trình chi tiết (tt)	58
Hình 3-32: Giao diện chỉnh sửa hoạt động của lịch trình	59
Hình 3-33: Giao diện xóa hoạt động của lịch trình	59
Hình 3-34: Giao diện gửi thông báo về email	60
Hình 3-35: Giao diện trang cài đặt	61
Hình 3-36: Giao diện khôi phục mật khẩu	62
Hình 3-37: Giao diện gửi mật khẩu về mail.....	62
Hình 3-38: Giao diện chọn ngôn ngữ ứng dụng.....	63
Hình 3-39: Giao diện trang cài đặt tiếng anh	63
Hình 3-40: Giao diện trang cài đặt tiếng thái	63
Hình 3-41: Giao diện trang cài đặt tiếng trung.....	63
Hình 3-42: Giao diện trang giới thiệu sản phẩm	64
Hình 3-43: Giao diện trang hỏi đáp thông minh	65
Hình 3-44: Giao diện trang hỏi đáp thông minh (tt)	65
Hình 3-45: Giao diện trang hỏi đáp thông minh (tt)	65
Hình 3-46: Giao diện trang hỏi đáp thông minh (tt)	65

PHẦN 1: MỞ ĐẦU

1. Lý do chọn đề tài

Nghiên cứu và xây dựng ứng dụng sổ tay du lịch Quảng Bình đang nổi lên như một hướng đi đầy tiềm năng và ý nghĩa trong việc cung cấp cho du khách một công cụ hiện đại và tiện lợi trong việc ghi chép và quản lý trải nghiệm du lịch của họ tại tỉnh Quảng Bình. Lựa chọn đề tài này không chỉ đặt trọng tâm vào việc tạo ra một sản phẩm ứng dụng hiện đại, mà còn đem lại nhiều giá trị đáng kể cho cả ngành du lịch và lĩnh vực công nghệ thông tin.

Một trong những lý do nổi bật cho sự lựa chọn này là tính cấp thiết và ý nghĩa của việc phát triển một ứng dụng sổ tay du lịch tiện ích. Sổ tay du lịch truyền thống có thể gặp phải nhiều hạn chế về tính tiện lợi và khả năng tương tác. Việc nghiên cứu và xây dựng một ứng dụng sổ tay du lịch sẽ cung cấp cho người dùng một công cụ linh hoạt, dễ sử dụng và đa chức năng để ghi chép thông tin về các địa điểm du lịch, lên lịch trình, ghi chú cá nhân và chia sẻ trải nghiệm.

Đồng thời, việc phát triển ứng dụng này cũng góp phần quảng bá và nâng cao hình ảnh du lịch của tỉnh Quảng Bình. Với các tính năng tiện ích và khả năng tương tác cao, ứng dụng sẽ giúp du khách trải nghiệm hành trình du lịch một cách thông minh và hiệu quả hơn, đồng thời thúc đẩy sự phát triển của ngành du lịch địa phương.

Cùng với việc đáp ứng nhu cầu thực tiễn của du khách, đề tài này cũng mở ra cơ hội thách thức trong việc áp dụng các công nghệ mới và phát triển ứng dụng di động. Việc tìm hiểu, nghiên cứu và áp dụng những công nghệ tiên tiến nhất sẽ giúp đội ngũ nghiên cứu phát triển sản phẩm mang tính cạnh tranh cao và đáp ứng được mong đợi của người dùng.

Ngoài ra, thành công trong việc xây dựng ứng dụng sổ tay du lịch Quảng Bình không chỉ đem lại lợi ích kinh tế mà còn tạo ra giá trị xã hội, từ việc tạo ra một công cụ hữu ích cho du khách đến việc thúc đẩy sự phát triển bền vững của ngành du lịch địa phương và cả ngành công nghệ thông tin.

Với sự gia tăng đáng kinh ngạc của sự phát triển ứng dụng di động trong thập kỷ qua, iOS và Android đã trở thành những hệ điều hành hàng đầu. Điều này dẫn đến sự trỗi dậy của toàn bộ ngành công nghiệp với một số framework cho phép các nhà phát triển xây ứng dụng di động. Hầu hết các framework phát triển di động này được xây dựng để hỗ trợ phát triển ứng dụng kết hợp tương thích với cả iOS và Android. Google đã đưa ra một bộ công cụ phát triển phần mềm giao diện người dùng mã nguồn mở Flutter, được công bố lần đầu tiên vào năm 2015 và chính thức ra mắt vào năm 2017.

Để tạo ra một ứng dụng di động là một công việc rất phức tạp và đầy thử thách. Có rất nhiều framework có sẵn, cung cấp các tính năng tuyệt vời để phát triển các ứng dụng di động. Để phát triển các ứng dụng dành cho thiết bị di động, Android cung cấp một framework gốc dựa trên ngôn ngữ Java và Kotlin, trong khi iOS cung cấp một framework dựa trên ngôn ngữ Objective-C / Swift. Vì vậy, chúng ta cần hai ngôn ngữ và framework khác nhau để phát triển ứng dụng cho cả hai hệ điều hành. Ngày nay, để khắc phục sự phức tạp này, có một số framework đã được giới thiệu hỗ trợ cả hệ điều hành cùng với các ứng dụng dành cho máy tính để bàn. Những loại framework này được gọi là công cụ phát triển đa nền tảng .

Với các lý do trên, đề tài “Nghiên cứu và xây dựng ứng dụng sổ tay du lịch Quảng Bình trên điện thoại di động” không chỉ đáp ứng nhu cầu thực tế mà còn mở ra nhiều cơ hội phát triển và đóng góp ý nghĩa cho cả cộng đồng du lịch và ngành công nghệ.

2. Mục đích nghiên cứu

Mục đích chung : Nghiên cứu và xây dựng thành công ứng dụng sổ tay du lịch Quảng Bình trên điện thoại di động

Mục đích cụ thể:

- Xây dựng thành công ứng dụng với giao diện thân thiện và dễ sử dụng
- Tìm kiếm và tra cứu thông tin về các điểm đến du lịch trong Quảng Bình.
- Tạo khả năng lên lịch trình du lịch và quản lý thông tin các địa điểm du lịch..

3. Nội dung nghiên cứu

- Nghiên cứu các công nghệ và công cụ phát triển ứng dụng di động.
- Thiết kế và xây dựng giao diện ứng dụng sổ tay du lịch.
- Phát triển các tính năng quản lý lịch trình, lưu trữ thông tin địa điểm, và ghi chú cá nhân.
- Cài đặt hệ thống và hoàn thiện sản phẩm.

4. Đối tượng nghiên cứu

- Đối tượng chính của nghiên cứu sẽ là du khách, những người có nhu cầu sử dụng ứng dụng sổ tay du lịch khi tham quan và khám phá các địa điểm du lịch tại tỉnh Quảng Bình.

- Các nhà phát triển ứng dụng sẽ là đối tượng thực hiện nghiên cứu để xây dựng và phát triển ứng dụng sổ tay du lịch.

5. Thời gian nghiên cứu

- Thời gian nghiên cứu dự kiến từ tháng 10/2023 đến tháng 5/2024.
- Quá trình nghiên cứu sẽ bao gồm các giai đoạn: nghiên cứu công nghệ và công cụ phát triển ứng dụng di động, thiết kế giao diện, phát triển tính năng, cài đặt hệ thống và hoàn thiện sản phẩm.
- Phạm vi của đề tài sẽ tập trung vào việc xây dựng và phát triển ứng dụng sổ tay du lịch Quảng Bình.
- Nghiên cứu sẽ tập trung vào việc lập kế hoạch du lịch, quản lý thông tin các địa điểm du lịch, ghi chú cá nhân và khả năng chia sẻ trải nghiệm.

6. Phương pháp nghiên cứu

Để thực hiện nội dung nghiên cứu đã đề ra, đề tài sử dụng các phương pháp nghiên cứu chính sau:

- Nghiên cứu sẽ sử dụng phương pháp tổng hợp thông tin từ các nguồn tài liệu về công nghệ phát triển ứng dụng di động.
- Áp dụng phương pháp thiết kế giao diện người dùng tối ưu để tạo ra một ứng dụng dễ sử dụng và thân thiện với người dùng.
- Tiến hành phát triển và thử nghiệm tính năng của ứng dụng dựa trên các phản hồi từ người dùng thực tế để cải thiện và hoàn thiện sản phẩm.
- Phương pháp nghiên cứu tài liệu lý thuyết liên quan tới vấn đề
- Nghiên cứu các công trình đã được công bố liên quan tới vấn đề
- Phương pháp lập trình, thử nghiệm
- Phương pháp phân tích, đánh giá.

PHẦN 2: NỘI DUNG

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Du lịch tỉnh Quảng Bình

1.1.1. Tổng quan về tỉnh Quảng Bình

Tỉnh Quảng Bình nằm ở Bắc Trung Bộ, Việt Nam, với diện tích tự nhiên 8.065 km², Dân số Quảng Bình năm 2021 có 910.655 người.



Hình 1-1: Thành phố Đồng Hới hôm nay

Vị trí địa lý: Toạ độ địa lý ở phần đất liền là:

- Điểm cực Bắc: 18005' 12" vĩ độ Bắc
- Điểm cực Nam: 17005' 02" vĩ độ Bắc
- Điểm cực Đông: 106059' 37" kinh độ Đông
- Điểm cực Tây: 105036' 55" kinh độ Đông

Quảng Bình có 08 đơn vị hành chính cấp huyện, gồm 05 huyện/thị xã ven biển (các huyện Quảng Trạch, Bố Trạch, Quảng Ninh, Lệ Thủy và thị xã Ba Đồn), 02 huyện vùng núi (Tuyên Hóa, Minh Hóa) và thành phố Đồng Hới là trung tâm của tỉnh; 151 đơn vị hành chính cấp xã, bao gồm 15 phường, 08 thị trấn và 128 xã.

Quảng Bình là phần hẹp nhất của Việt Nam, vì vậy đây là hành lang quan trọng đối với cả hàng hóa lẫn hành khách với các tuyến đường bộ như Quốc lộ 1A, đường Hồ Chí Minh chạy dọc tuyến Bắc - Nam, tuyến đường sắt Bắc - Nam. Các tuyến đường này kết nối các khu vực trọng điểm trong tỉnh, đặc biệt là kết nối các cảng Gianh, cảng Nhật Lệ và cảng Hòn La với Khu Kinh tế Hòn La và thành phố Đồng Hới.

Tỉnh đóng vai trò quan trọng trong giao thương hàng hóa và hành khách tuyến Đông - Tây nhờ có Quốc lộ 12A kết nối Quảng Bình với Lào và từ Lào sang miền Bắc Thái Lan và Myanmar.

Đặc điểm khí hậu: Quảng Bình nằm trong vùng nhiệt đới gió mùa chuyển tiếp giữa miền Bắc và Nam Việt Nam, do đó khí hậu phân thành hai mùa rõ rệt là mùa khô và mùa mưa. Với việc có mùa Hè nắng nóng, trời quang mây và chế độ gió khá ổn định tạo thuận lợi cho Quảng Bình phát triển các loại hình năng lượng sạch như điện gió, điện mặt trời và du lịch biển. Tuy vậy, đặc điểm khí hậu của Quảng Bình nhìn chung gây nhiều trở ngại cho đời sống và sản xuất, đặc biệt là sản xuất nông nghiệp.

Đặc điểm địa hình: Tỉnh Quảng Bình nằm ở sườn Đông dãy Trường Sơn, với địa hình đồi núi cao hiểm trở, hẹp bề ngang và dốc, nghiêng từ Tây sang Đông và chia cắt ở các khu vực phía Tây của tỉnh. Các khu vực phía Đông nhìn chung có độ dốc lớn về phía biển, vùng đồng bằng bị thu hẹp lại ở phía Nam do sự xâm lấn của dãy Trường Sơn.

Phần lớn diện tích tỉnh là đồi núi (chiếm trên 85% diện tích tự nhiên), các dạng địa hình khác là trung du, đồng bằng duyên hải và bãi cát ven biển chỉ chiếm gần 15%. Do vậy, diện tích đất canh tác lúa và nông lâm nghiệp tương đối hạn chế và thường xuyên gánh chịu lũ lụt bất thường vào mùa mưa do hệ thống sông suối ngắn, dốc, chảy từ Tây sang Đông.

Địa hình tỉnh có sự khác nhau giữa các tiểu vùng: (1) Vùng núi cao nằm dọc theo sườn Đông dãy Trường Sơn; (2) Vùng gò đồi và trung du tập trung chủ yếu tại các huyện Bố Trạch, Quảng Trạch, Quảng Ninh, Lệ Thủy và thành phố Đồng Hới; (3) Vùng đồng bằng và vùng cát ven biển ở các huyện Quảng Trạch, Bố Trạch, Quảng Ninh, Lệ Thủy, thị xã Ba Đồn và thành phố Đồng Hới.

Điều kiện văn hóa, xã hội: Quảng Bình vừa là địa bàn phân bố văn hóa Hòa Bình, văn hóa Đông Sơn, văn hóa Bầu Tró ở phía Bắc, vừa chứa đựng, bảo lưu, giao thoa với văn hóa Sa Huỳnh, Chăm Pa ở phía Nam. Quá trình giao thoa, tiếp biến, hòa nhập trong suốt cả chiều dài lịch sử dân tộc ngày càng bồi đắp, làm phong phú hơn truyền thống văn hóa của Nhân dân vùng đất này; đồng thời cũng làm cho sắc thái văn hóa Quảng Bình có những đặc trưng riêng, mở ra tiềm năng to lớn cho việc phát triển các loại hình văn hóa - du lịch.

1.1.2. Lợi thế du lịch của tỉnh Quảng Bình

Những tài nguyên đa dạng của Quảng Bình tạo nên một bức tranh phức tạp và hấp dẫn, thu hút du khách muốn khám phá và trải nghiệm văn hóa, lịch sử, cũng như cảnh quan thiên nhiên độc đáo của vùng đất này.

Tài nguyên đất: Toàn tỉnh có 10 nhóm đất trong đó nhóm đất xám có diện tích lớn nhất tỉnh Quảng Bình (khoảng 515.781,0 ha, chiếm 63,95% diện tích tự nhiên), phân bố khắp các huyện, thị xã, thành phố trong tỉnh. Nhóm đất tầng mỏng có diện

tích lớn thứ hai (khoảng 24.274,0 ha), phân bố tập trung ở gò đồi của huyện Lệ Thủy, Bố Trạch và Quảng Trạch, thực vật tự nhiên chủ yếu là sim, mua. Nhóm đất cát có diện tích lớn thứ ba (khoảng 37.243 ha) phân bố ở ven biển; Nhóm đất phù sa có diện tích khoảng 34.791 ha, phân bố tập trung ở vùng đồng bằng ven biển là sản phẩm phù sa của các con sông suối trên địa bàn của tỉnh.

Tài nguyên khoáng sản: Tài nguyên khoáng sản Quảng Bình tương đối phong phú và đa dạng, tiêu biểu như vàng, đá vôi và kaolin chất lượng cao, trữ lượng lớn, nhiều loại vật liệu xây dựng, nhiều điểm nước khoáng nước nóng nổi tiếng... Tuy nhiên, một số loại khoáng sản phân bố rải rác, trữ lượng nhỏ nên hiệu quả khai thác không cao.

Tài nguyên nước: Quảng Bình có nguồn nước mặt khá phong phú nhờ hệ thống sông suối, hồ đập khá dày đặc. Mật độ sông suối đạt 0,70 - 1,10 km/km², phân bố không đều, có xu hướng giảm dần từ Tây sang Đông, từ vùng núi ra biển. Sông ngòi Quảng Bình đều ngắn và dốc, có 05 con sông chính đổ ra biển Đông là sông Roòn, Gianh, Lý Hòa, Dinh và Nhật Lệ. Diện tích lưu vực 05 con sông là 7.778,0 km², chiều dài 343,0 km. Lượng dòng chảy khá phong phú và thuộc vào lớn nhất ở Việt Nam. Modul dòng chảy bình quân là 55l/s/km², tương đương 04 tỷ m³ nước/năm.

Nguồn nước ngầm của tỉnh khá phong phú, tuy nhiên phân bố không đều, mức độ nông sâu thay đổi phụ thuộc vào địa hình và lượng mưa trong mùa. Thường vùng đồng bằng ven biển có mực nước ngầm nông và dồi dào, đối với vùng trung du nước ngầm sâu và dễ bị cạn kiệt vào mùa khô.

Tài nguyên rừng: Hiện nay, toàn tỉnh có 615.287,91 ha đất lâm nghiệp, chiếm 76,91% diện tích đất tự nhiên. Trong đó, diện tích đất có rừng 543.048,85 ha, gồm 469.767,95 ha rừng tự nhiên và 73.280,9 ha rừng trồng đã thành rừng. Sản lượng gỗ ước tính có khoảng 52.522 nghìn m³ (chủ yếu là trữ lượng gỗ rừng tự nhiên với 49.868.270 m³, chiếm 94,95%). Thảm thực vật rừng Quảng Bình rất đa dạng và phong phú, trong đó có nhiều loại gỗ quý như lim, gụ, sến, táu... và nhiều loại thú quý hiếm như voọc, trĩ sao, gà lôi, các loại bò sát, móng guốc khác. Đặc sản dưới tán rừng khá phong phú, đa dạng và có giá trị lớn như song mây, dược liệu quý, phong lan... Đặc biệt ở khu bảo tồn Phong Nha - Kẻ Bàng có nhiều loại động vật và thực vật quý hiếm.

Thủy hải sản: Vùng biển của tỉnh Quảng Bình được đánh giá có trữ lượng hải sản lớn với tính đa dạng cao. Theo số liệu của Viện Nghiên cứu biển, trữ lượng hải sản ước khoảng 10 vạn tấn với 1.650 loài hải sản các loại, trong đó nhiều loài có giá trị kinh tế cao, như tôm hùm, tôm sú, mực nang, mực ống... Tính theo độ sâu, trữ lượng hải sản khoảng 23.000 tấn ở độ sâu từ 0 - 50 m; từ 51 - 90 m: 76.000 tấn. Phân theo loài, tôm biển: 1.600 - 2.000 tấn, ruốc: 5.000 - 7.000 tấn và cá các loại: 60.000 -

70.000 tấn. Về nuôi trồng hải sản, với 05 cửa sông, Quảng Bình có vùng mặt nước có khả năng nuôi trồng thủy sản khá lớn. Tổng diện tích 15.000 ha. Độ mặn ở vùng mặt nước từ cửa sông vào sâu khoảng 10 - 15km dao động từ 08 - 30‰ và độ pH từ 6,5 - 8 rất thuận lợi cho nuôi tôm, cua xuất khẩu. Chế độ bán nhật triều vùng ven biển thuận lợi cho việc cấp thoát nước cho các ao nuôi tôm, cua.

1.1.3. Tiềm năng du lịch quảng bình

Tiềm năng phát triển du lịch biển: Với chiều dài bờ biển trên 116,04 km, Quảng Bình sở hữu những bãi tắm đẹp còn giữ được nhiều nét hoang sơ, như: Đá Nhảy, Nhật Lệ, Hải Ninh, Vũng Chùa - Đảo Yến... cùng với đó là hệ thống di tích lịch sử, điểm đến văn hóa ở vùng ven biển.

Tiềm năng phát triển cảng biển và dịch vụ cảng: Quảng Bình là tỉnh có bờ biển dài, vùng biển rộng với 05 cửa sông và 02 khu vực hàng hải quan trọng, gồm: Khu vực hàng hải cửa Gianh và khu vực hàng hải Hòn La. Hiện nay, Quảng Bình có 04 bến cảng gồm: Cảng Gianh, cảng Hòn La, cảng Xăng dầu sông Gianh, cảng Thắng Lợi, với gần 600m cầu cảng tiếp nhận tàu trọng tải từ 1.000-20.000 tấn; 04 khu chuyển tải tiếp nhận tàu trọng tải đến 50.000 tấn... Ngoài ra, vịnh Hòn La có diện tích mặt nước 04 km², độ sâu trên 15m và xung quanh có các đảo che chắn, như: Hòn La, Hòn Nồm, Hòn Gió, Hòn Cỏ, Hòn Chùa là điều kiện thuận lợi cho các loại tàu thuyền neo đậu, phát triển dịch vụ hậu cần nghề cá và vận tải biển.

Phát triển các nguồn năng lượng tái tạo: Nằm trong vành đai nhiệt đới Bắc bán cầu, với nền nhiệt độ cao và phân bố khá đồng đều quanh năm, bức xạ mặt trời khá cao, Quảng Bình giàu tiềm năng để phát triển năng lượng tái tạo, đặc biệt là điện gió và điện mặt trời. Số giờ nắng ở vùng đồng bằng ven biển từ 1.650 - 1.820 giờ/năm, lượng bức xạ trong năm ở tỉnh Quảng Bình dao động trong khoảng 1.256 - 1.418 kWh/m²/năm, tập trung vào các tháng mùa hè từ tháng 3 đến tháng 9 thuận lợi cho phát triển điện mặt trời. Vận tốc gió bình quân từ 06 - 6,75m/s (ở độ cao 120m) ở trên biển và đất liền, thích hợp để phát triển điện gió.

Tài nguyên du lịch: Bờ biển dài 116,04 km của tỉnh Quảng Bình có nhiều bãi biển đẹp và các điểm nghỉ ngơi giải trí như Nhật Lệ, Cảng Gianh, Vũng Chùa, đảo Yến, vịnh Hòn La, bãi tắm Đá Nhảy, bãi tắm Hải Ninh. Đặc biệt, Quảng Bình có Di sản thiên nhiên thế giới Vườn Quốc gia Phong Nha - Kẻ Bàng được UNESCO công nhận là di sản thế giới vào năm 2003 và 2015. Các điểm du lịch khác bao gồm suối khoáng nóng Bang, suối khoáng nóng nhất ở Việt Nam với nhiệt độ vượt 1050C; hang Rục Mòn (Hóa Sơn); hang La Vân, hang Sương Mù (Hóa Tiến); hồ Yên Phú (Trung Hóa); Thác Mơ (Hóa Hợp) hay những địa điểm tiềm năng như Thác Bụt - Giếng Tiên (Yên Hóa); Cổng trời - Cha Lo - Nước Rụng (Dân Hóa)... Với tài nguyên nhân văn, di sản văn hóa vật thể và phi vật thể độc đáo, giá trị cùng các cảnh quan thiên nhiên

kì vĩ, nổi bật sẽ là chìa khóa mở ra tiềm năng to lớn của ngành du lịch và sẽ là nền tảng cho chiến lược của ngành để tạo khác biệt của tỉnh Quảng Bình so với các điểm đến du lịch khác.

1.2. Tìm hiểu về Flutter

1.2.1. Flutter là gì?

Flutter là một bộ SDK đa nền tảng (cross- platform) có thể hoạt động trên iOS và Android do Google phát triển dựa trên ngôn ngữ lập trình Dart được sử dụng để tạo ra các ứng dụng dành cho di động (native app). Flutter hoạt động với những code sẵn có được sử dụng bởi các lập trình viên, các tổ chức.

Flutter hoàn toàn miễn phí và cũng là mã nguồn mở.

Flutter gồm 2 thành phần quan trọng:

- Một SDK (Software Development Kit): Một bộ sưu tập các công cụ sẽ giúp bạn phát triển các ứng dụng của mình.
- Một Framework (UI Library based on widgets): Một tập hợp các thành phần giao diện người dùng (UI) có thể tái sử dụng (button, text inputs, slider, v.v.) giúp bạn có thể cá nhân hóa tùy theo nhu cầu của riêng mình.

Nói chung, tạo một ứng dụng di động là một công việc rất phức tạp và đầy thử thách. Có rất nhiều framework có sẵn, cung cấp các tính năng tuyệt vời để phát triển các ứng dụng di động. Để phát triển các ứng dụng dành cho thiết bị di động, Android cung cấp một framework gốc dựa trên ngôn ngữ Java và Kotlin, trong khi iOS cung cấp một framework dựa trên ngôn ngữ Objective-C/Swift.

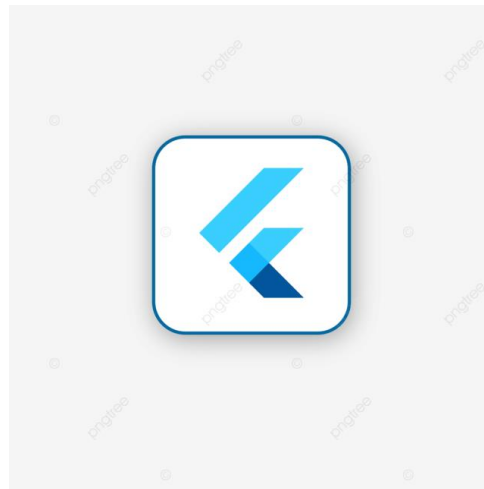
Vì vậy, chúng ta cần hai ngôn ngữ và framework khác nhau để phát triển ứng dụng cho cả hai hệ điều hành. Ngày nay, để khắc phục sự phức tạp này, có một số framework đã được giới thiệu hỗ trợ cả hệ điều hành cùng với các ứng dụng dành cho máy tính để bàn. Những loại framework này được gọi là công cụ phát triển đa nền tảng .

Framework phát triển đa nền tảng có khả năng viết một code và có thể triển khai trên nhiều nền tảng khác nhau (Android, iOS và Máy tính để bàn). Nó tiết kiệm rất nhiều thời gian và nỗ lực phát triển của các nhà phát triển.

Có một số công cụ có sẵn để phát triển đa nền tảng, bao gồm các công cụ dựa trên web. Mỗi framework này có mức độ thành công khác nhau trong ngành công nghiệp di động. Gần đây, một framework công tác mới đã được giới thiệu trong họ phát triển đa nền tảng có tên là Flutter được phát triển từ Google.

Flutter là một bộ công cụ giao diện người dùng để tạo các ứng dụng nhanh, đẹp, được biên dịch nguyên bản cho thiết bị di động, web và máy tính để bàn với một ngôn ngữ lập trình và cơ sở code duy nhất. Nó là miễn phí và code nguồn mở. Ban đầu nó được phát triển từ Google và bây giờ được quản lý theo tiêu chuẩn ECMA. Ứng dụng Flutter sử dụng ngôn ngữ lập trình Dart để tạo ứng dụng.

Flutter chủ yếu được tối ưu hóa cho các ứng dụng di động 2D có thể chạy trên cả nền tảng Android và iOS. Chúng ta cũng có thể sử dụng nó để xây dựng các ứng dụng đầy đủ tính năng, bao gồm máy ảnh, bộ nhớ, vị trí địa lý, mạng, SDK của bên thứ ba [1].



Hình 1-2: Logo Flutter

1.2.2. Quá trình phát triển của Flutter

Vào năm 2015, Google đã công bố Flutter, một SDK mới dựa trên ngôn ngữ Dart, làm nền tảng tiếp theo để phát triển Android và vào năm 2017, phiên bản alpha của nó (0.0.6) đã được phát hành cho công chúng lần đầu tiên.

Tại I/O 2017, Google đã trình diễn việc sử dụng Flutter và các khả năng đa nền tảng của nó, đồng thời tiếp tục quảng bá nó tại I/O 2018. Kể từ đó, Google đã đầu tư vào Flutter và khuyến nghị đây là cách mà mọi người nên phát triển ứng dụng di động.

Vào ngày 4 tháng 12 năm 2018, Flutter 1.0 đã được phát hành tại sự kiện Flutter Live và có sẵn để các nhà phát triển có thể bắt đầu sử dụng SDK để tạo ứng dụng dễ dàng hơn. Đây được đánh dấu là phiên bản “stable” đầu tiên.

Tại Google I/O 2019, việc hỗ trợ Flutter cho desktop và web platforms đã được công bố công khai, các công cụ để phát triển ứng dụng Flutter cho Windows, macOS, Linux và web đã được phát hành. Cùng với đó là sự ra đời của Flutter.

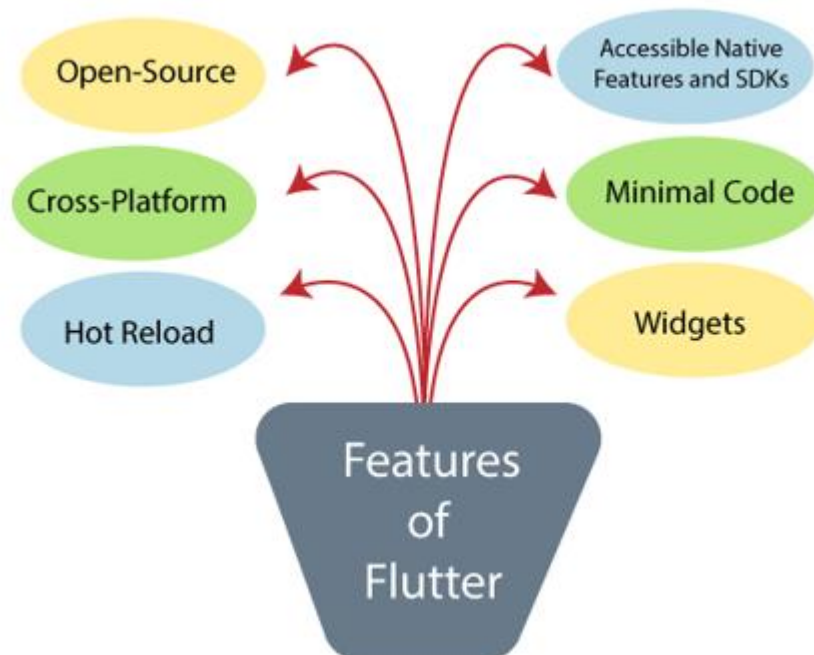
Tính đến ngày 30 tháng 4 năm 2023, 3.7.12 chính là phiên bản cập nhật mới nhất của Flutter [1].

1.2.3. Điều làm Flutter trở lên độc đáo

Flutter khác với các framework khác vì nó không sử dụng WebView cũng như các widget OEM (Original Equipment Manufacturer) đi kèm với thiết bị. Thay vào đó, nó sử dụng công cụ kết xuất hiệu suất cao của riêng mình để vẽ các widget. Nó cũng triển khai hầu hết các hệ thống của nó như hoạt ảnh, cử chỉ và widget bằng ngôn ngữ lập trình Dart cho phép các nhà phát triển đọc, thay đổi, thay thế hoặc loại bỏ mọi thứ một cách dễ dàng. Nó cung cấp khả năng kiểm soát tuyệt vời cho các nhà phát triển đối với hệ thống.

1.2.4. Các tính năng của Flutter

Flutter cung cấp các phương pháp dễ dàng và đơn giản để bắt đầu xây dựng các ứng dụng dành cho thiết bị di động và máy tính để bàn đẹp mắt với một bộ thiết kế material design và widget phong phú. Ở đây, chúng ta sẽ thảo luận về các tính năng chính của nó để phát triển framework di động.



Hình 1-3: Các tính năng của Flutter

Code nguồn mở(Open-Source): Flutter là một framework code nguồn mở và miễn phí để phát triển các ứng dụng di động.

Đa nền tảng(Cross-platform): Tính năng này cho phép Flutter viết code một lần, duy trì và có thể chạy trên các nền tảng khác nhau. Nó tiết kiệm thời gian, công sức và tiền bạc của các nhà phát triển.

Tải lại nóng(Hot Reload): Bất cứ khi nào nhà phát triển thực hiện thay đổi trong code, thì những thay đổi này có thể được nhìn thấy ngay lập tức với Tải lại nóng. Nó có nghĩa là những thay đổi hiển thị ngay lập tức trong chính ứng dụng. Đây là một tính năng rất tiện dụng, cho phép nhà phát triển sửa các lỗi ngay lập tức.

Các tính năng và SDK gốc có thể truy cập (Accessible Native Features and SDKs): Tính năng này cho phép quá trình phát triển ứng dụng dễ dàng và thú vị thông qua code gốc của Flutter, tích hợp bên thứ ba và các API nền tảng. Do đó, chúng tôi có thể dễ dàng truy cập SDK trên cả hai nền tảng.

Code tối thiểu (Minimal code): Ứng dụng Flutter được phát triển bởi ngôn ngữ lập trình Dart, sử dụng biên dịch JIT và AOT để cải thiện thời gian khởi động tổng thể, hoạt động và tăng tốc hiệu suất. JIT nâng cao hệ thống phát triển và làm mới giao diện người dùng mà không cần nỗ lực thêm vào việc xây dựng hệ thống mới.

Widget: framework công tác Flutter cung cấp các widget có khả năng phát triển các thiết kế cụ thể có thể tùy chỉnh. Quan trọng nhất, Flutter có hai bộ widget: Material Design và các widget Cupertino giúp mang lại trải nghiệm không có trục trặc trên tất cả các nền tảng [2].

1.2.5. Ưu điểm của Flutter

Sự ra đời của Flutter đã phần nào giảm bớt những khó khăn thường gặp trong mobile là Fast Development và Native Performance. Trong khi React Native chỉ đảm bảo được Fast Development và Code Native thuần chỉ đảm bảo được Native Performance thì Flutter làm được cả 2 điều trên.

- Hot Reload – JIT(Just-in-time):

+ Là tính năng là tính năng cho phép các nhà phát triển Flutter thực hiện các thay đổi trên mã nguồn và hiển thị kết quả ngay lập tức trên ứng dụng đang chạy. Tính năng này giúp tiết kiệm thời gian và tăng tính tương tác của các nhà phát triển

- Expressive and Flexible UI:

+ Trước đó việc xây dựng các ứng dụng có UI đẹp, animation mượt mà rất phức tạp và gian nan với hầu hết các nhà phát triển ứng dụng di động.

+ Giao diện người dùng được xây dựng với các widget, các khối xây dựng UI nhỏ được lắp ráp bằng một kỹ thuật gọi là Composition. Bản thân Flutter cũng có sẵn rất nhiều Widget dựng sẵn, các developer có thể tận dụng dễ dàng hơn thay vì phải xây dựng chúng từ đầu.

- + Toàn bộ quá trình này tương tự như các thành phần React

- + Có sẵn 2 bộ UI thường được sử dụng trong Flutter:

- Material Design (tương thích với các nguyên tắc thiết kế của Google) dành cho Android.
- Cupertino (tương thích với Apple's Human Interface Guidelines) dành cho iOS. Các developer có thể dùng một trong hai hoặc phối hợp chúng lại với nhau.

- **Công cụ (tools)** : Có 2 con đường để code Flutter

- + Một là sử dụng Web Editor (<https://dartpad.dev/flutter>).

- + Sử dụng các IDE phổ biến như Android Studio và Visual Studio Code.

- **Code đa nền tảng :**

- + Flutter thực sự là một single-code base đa nền tảng. Điều này có nghĩa là bạn chỉ cần viết code một lần và nó sẽ hoạt động được trên IOS, Android và thậm chí cả web.

- + Khác với những nền tảng trước đó, như React Native, code Web và Mobile App dù có chung các khái niệm nhưng cơ bản là bạn cần thay đổi source code để đảm bảo chúng hoạt động tốt. Với Flutter bạn không còn phải làm điều này nữa, thực sự là source code được tái sử dụng với cả Web và Mobile.

- + Một quan niệm sai lầm lớn về việc phát triển trong Flutter là bạn phải nhắm mục tiêu giao diện cho iOS và Android riêng biệt. Vì điều này đánh mất lợi thế mạnh nhất của Flutter. Thay vào đó, bạn hãy thống nhất UI/UX trên cả 2 nền tảng iOS và Android, điều mà các nền tảng framework khác không thể làm được [2].

1.2.6. Nhược điểm của Flutter

- Hầu hết bộ render UI đã được viết lại, không liên quan đến UI sẵn có trong UI Framework native nên memory sẽ dùng khá nhiều. Bên cạnh đó, UI không đi cùng với OS mà được phát triển riêng.

- Thường xuyên bổ sung các kiến thức về ngôn ngữ Dart: Có rất ít lập trình viên hiểu biết rõ về Dart. Có nhiều trường hợp sau khi học xong sẽ dính liền luôn với Dart trong mảng phát triển các ứng dụng trên mobile, chứ không thể linh động như Python, JS có thể qua lại giữa AI, back,..

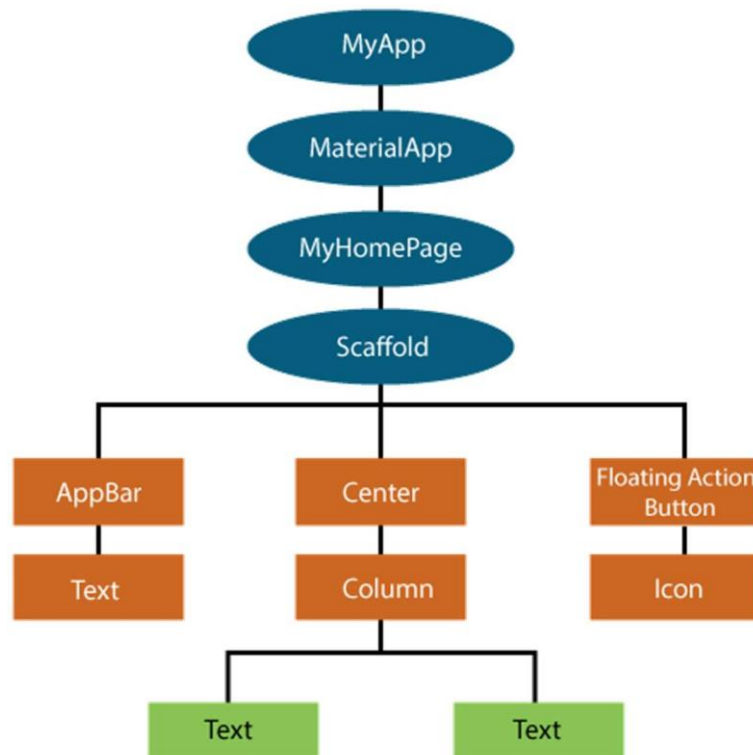
- Mô hình dữ liệu rất mới: Nếu đã quen với Redux thì phải sẽ mất kha khá thời gian để học hỏi thêm về mô hình dữ liệu của Flutter mặc dù nó không quá khó [2].

1.3. Các thành phần quan trọng trong Flutter

1.3.1. Widget Flutter

Bất cứ khi nào bạn định viết mã để xây dựng bất cứ thứ gì trong Flutter, nó sẽ nằm trong một widget. Mục đích chính là xây dựng ứng dụng từ các widget. Nó mô tả chế độ xem ứng dụng của bạn trông như thế nào với cấu hình và trạng thái hiện tại của chúng. Khi bạn thực hiện bất kỳ thay đổi nào trong code, widget con sẽ xây dựng lại mô tả của nó bằng cách tính toán sự khác biệt của widget con hiện tại và trước đó để xác định những thay đổi tối thiểu đối với việc hiển thị trong giao diện người dùng của ứng dụng.

Các widget được lồng vào nhau để xây dựng ứng dụng. Nó có nghĩa là thư mục gốc của ứng dụng của bạn tự nó là một widget, và tất cả các cách nhìn xuống cũng là một widget. Ví dụ: một widget có thể hiển thị một thứ gì đó, có thể xác định thiết kế, có thể xử lý tương tác [3].



Hình 1-4: Mô tả cây Widget

Chúng ta có thể chia widget thành 2 loại: *Widget hiển thị* và *Widget ẩn*

1.3.1.1. Widget hiển thị

Các widget hiển thị có liên quan đến dữ liệu đầu vào và đầu ra của người dùng. Một số loại quan trọng của widget con này là:

- **Text:** Widget này giúp hiển thị văn bản trên màn hình, chúng ta có thể tự động căn chỉnh dựa vào các thuộc tính của nó.



```
const Text(  
  String this.data, {  
    super.key,  
    this.style,  
    this.strutStyle,  
    this.textAlign,  
    this.textDirection,  
    this.locale,  
    this.softWrap,  
    this.overflow,  
    this.textScaleFactor,  
    this.maxLines,  
    this.semanticsLabel,  
    this.textWidthBasis,  
    this.textHeightBehavior,  
    this.selectionColor,  
  })
```

Hình 1-5: Một số thuộc tính của Text

- **Button:** Widget này cho phép bạn thực hiện một số hành động khi nhấp chuột và có rất nhiều loại button khác nhau

- + Nút phẳng(Flat Button)
- + Nút nâng(Raised Button)
- + Nút nổi(Floating Button)
- + Nút thả xuống(Drop Down Button)
- + Nút biểu tượng(Icon Button)
- + Nút Inkwell
- + Nút PopupMenu
- + Nút phác thảo(Outline Button)

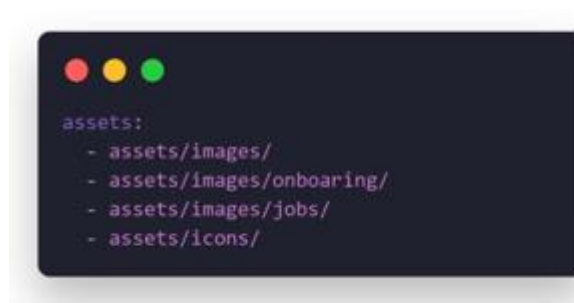
Đi kèm các button chính là các Gesture (cử chỉ) là cách mà người dùng tương tác với các thiết bị di động Tap

- + onTapDown
- + onTapUp
- + onTap

- + onTapCancel
- + Double tap
- + onDoubleTap
- + Long press
- + onLongPress
- + Vertical drag
- ...

- **Image:** Widget con này giữ hình ảnh có thể tìm nạp hình ảnh từ nhiều nguồn như từ thư mục nội dung hoặc trực tiếp từ URL. Nó cung cấp nhiều hàm tạo để tải hình ảnh, được đưa ra dưới đây:

- + Hình ảnh(Image): Đây là một trình tải hình ảnh chung, được sử dụng bởi ImageProvider .
- + Tài sản(asset): Nó tải hình ảnh từ thư mục tài sản dự án của bạn.
- + Tập(file): Nó tải hình ảnh từ thư mục hệ thống.
- + Bộ Nhớ(memory): Nó tải hình ảnh từ bộ nhớ.
- + Mạng(network): Nó tải hình ảnh từ mạng.



Hình 1-6: Cài đặt path thư mục lưu nội dung trong pubspec.yaml

- **Icon:** Widget này hoạt động như một thùng chứa để lưu trữ Biểu tượng trong Flutter. Đoạn mã sau đây giải thích rõ ràng hơn.

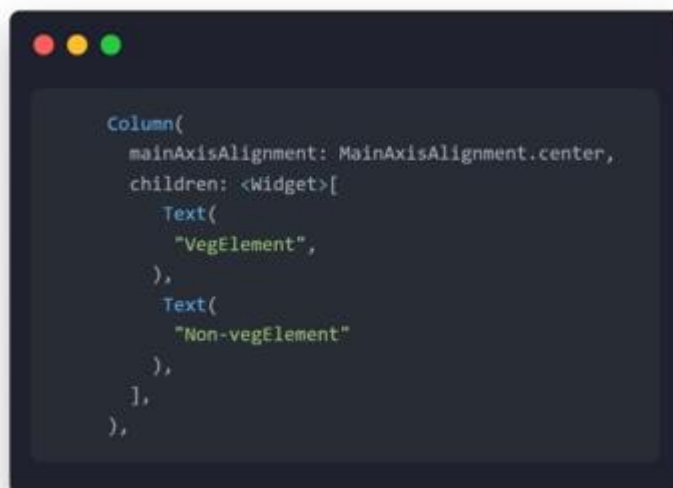


Hình 1-7: Hình ảnh ví dụ cách sử dụng Icon

1.3.1.2. Widget ẩn

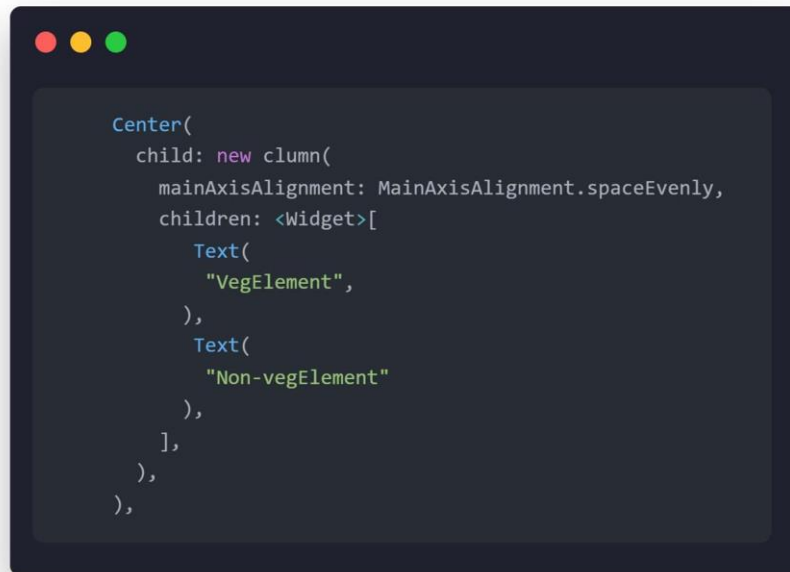
Các widget vô hình có liên quan đến cách bố trí và kiểm soát các widget. Nó cung cấp việc kiểm soát cách các widget thực sự hoạt động và cách chúng sẽ hiển thị trên màn hình. Một số loại widget quan trọng là:

- **Column:** Widget dạng cột là một loại widget sắp xếp tất cả các widget con của nó theo hàng dọc bằng `mainAxisAlignment` và `crossAxisAlignment`. Trong các thuộc tính này, trục chính là trục tung, và trục chéo là trục hoành.



Hình 1-8: Hình ảnh ví dụ về Column

- **Row:** Tương tự ta có widget Row sắp xếp các widget con theo hàng ngang.
- **Center:** Widget con này được sử dụng để căn giữa widget con, nằm bên trong nó.



Hình 1-9: Hình ảnh ví dụ về Center

- **Padding:** Widget này bao bọc các widget khác để cung cấp cho chúng phần đệm theo các hướng được chỉ định. Bạn cũng có thể cung cấp đệm theo mọi hướng.



Hình 1-10: Hình ảnh ví dụ về Padding

- **Scaffold:** Widget này cung cấp một framework cho phép bạn thêm các yếu tố thiết kế material design phổ biến như AppBar, Floating Action Buttons, Drawers.

- **Stack:** Nó là một widget thiết yếu, chủ yếu được sử dụng để chồng lên một widget, chẳng hạn như một nút trên nền gradient.

1.3.2. Quản lý Widget

1.3.2.1. StatelessWidget

StatelessWidget không có bất kỳ thông tin trạng thái nào. Nó vẫn tĩnh trong suốt vòng đời của nó. Các ví dụ về StatelessWidget là Văn bản, Hàng, Cột, Vùng chứa, ...

A screenshot of a code editor with a dark theme. It shows a Dart class named `MyStatelessCarWidget` that extends `StatelessWidget`. The class has a constructor `const MyStatelessCarWidget({Key key}) : super(key: key);` and an overridden `build` method that returns a `Container` widget with a fixed light blue color.

```
class MyStatelessCarWidget extends StatelessWidget {
  const MyStatelessCarWidget ({ Key key }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(color: const Color(0xFFEEFE));
  }
}
```

Hình 1-11: Hình ảnh ví dụ về StatelessWidget

1.3.2.2. StatefulWidget

StatefulWidget có thông tin state. Nó chủ yếu chứa hai lớp state object và widget. Nó là động vì nó có thể thay đổi dữ liệu bên trong trong suốt thời gian tồn tại của widget. widget con này không có phương thức `build()`. Nó có phương thức `createState()`, trả về một lớp mở rộng Lớp state của Flutter.

A screenshot of a code editor with a dark theme. It shows a Dart class named `Car` that extends `StatefulWidget`. The class has a constructor `const Car({Key key, String title}) : super(key: key);`. It overrides `createState` to return a `_CarState` object. `_CarState` extends `State<Car>` and overrides `build` to return a `Container` widget with a light blue color and a child widget.

```
class Car extends StatefulWidget {
  const Car({ Key key, this.title }) : super(key: key);

  @override
  _CarState createState() => _CarState();
}

class _CarState extends State<Car> {
  @override
  Widget build(BuildContext context) {
    return Container(
      color: const Color(0xFFEEFE),
      child: Container(
        child: Container( //child: Container() )
      )
    );
  }
}
```

Hình 1-12: Hình ảnh ví dụ về StatefulWidget

1.4. Giới thiệu về ngôn ngữ lập trình Dart

1.4.1. Dart là gì?

Dart là một ngôn ngữ lập trình hướng đối tượng mã nguồn mở, có mục đích chung với cú pháp kiểu C do Google phát triển vào năm 2011. Mục đích của lập trình Dart là tạo giao diện người dùng frontend cho web và ứng dụng dành cho thiết bị di động. Nó đang được phát triển tích cực, được biên dịch sang mã máy gốc để xây dựng ứng dụng di động, lấy cảm hứng từ các ngôn ngữ lập trình khác như Java, JavaScript, C# và Typed mạnh. Vì Dart là một ngôn ngữ biên dịch nên bạn không thể thực thi code của mình trực tiếp; thay vào đó, trình biên dịch phân tích cú pháp nó và chuyển nó thành code máy.

Nó hỗ trợ hầu hết các khái niệm chung của ngôn ngữ lập trình như lớp, giao diện, hàm, không giống như các ngôn ngữ lập trình khác. Ngôn ngữ Dart không hỗ trợ mảng trực tiếp. Nó hỗ trợ tập hợp, được sử dụng để sao chép cấu trúc dữ liệu như mảng, generic và kiểu tùy chọn.

1.4.2. Đặc điểm và tính chất của Dart

Kiểu dữ liệu: Dart hỗ trợ kiểu dữ liệu tĩnh và động. Nó là một ngôn ngữ kiểu mạnh, cho phép khai báo kiểu dữ liệu tường minh hoặc không tường minh. Điều này giúp trong việc kiểm tra lỗi và tối ưu hóa hiệu suất của ứng dụng.

Cú pháp: Dart có cú pháp gọn gàng và dễ đọc, giống với các ngôn ngữ lập trình như C++, Java hoặc JavaScript. Điều này làm cho việc học và sử dụng Dart trở nên thuận lợi đối với các nhà phát triển có kinh nghiệm trong các ngôn ngữ lập trình khác.

Hỗ trợ cho OOP (Object-Oriented Programming): Dart là một ngôn ngữ hướng đối tượng với hỗ trợ đầy đủ cho các khái niệm OOP như lớp, đối tượng, kế thừa, đa hình, và giao diện.

Asynchronous Programming: Dart hỗ trợ lập trình bất đồng bộ (asynchronous programming) thông qua Future và Stream, giúp xử lý các tác vụ không đồng bộ mà không chặn luồng chính của ứng dụng.

Phong cách viết code: Dart có một cú pháp linh hoạt, cho phép viết code theo nhiều phong cách khác nhau. Điều này giúp nhà phát triển lựa chọn cách viết code phù hợp với dự án và tùy chỉnh theo ý muốn cá nhân.

Tính di động và đa nền tảng: Một trong những ưu điểm lớn của Dart là khả năng chạy trên nhiều nền tảng khác nhau, từ di động đến web và máy chủ.

Hỗ trợ cộng đồng và tài liệu: Dart có một cộng đồng người dùng rộng lớn và tài liệu phong phú, giúp người mới học và nhà phát triển có thể dễ dàng tìm kiếm thông tin và hỗ trợ khi cần.

1.4.3. Ưu điểm của Dart

Năng suất: Cú pháp Dart rõ ràng và súc tích, công cụ của nó đơn giản nhưng mạnh mẽ. Type-safe giúp bạn xác định sớm các lỗi tinh tế. Dart có các thư viện cốt lõi và một hệ sinh thái gồm hàng ngàn package.

Nhanh: Dart cung cấp tối ưu hóa việc biên dịch trước thời hạn để có được dự đoán hiệu suất cao và khởi động nhanh trên các thiết bị di động và web.

Di động: Dart biên dịch thành mã ARM và x86, để các ứng dụng di động của Dart có thể chạy tự nhiên trên iOS, Android và hơn thế nữa. Đối với các ứng dụng web, chuyển mã từ Dart sang JavaScript.

Dễ gàn: Dart quen thuộc với nhiều nhà phát triển hiện có, nhờ vào cú pháp và định hướng đối tượng không gây ngạc nhiên của nó. Nếu bạn đã biết C++, C# hoặc Java, bạn có thể làm việc hiệu quả với Dart chỉ sau vài ngày.

Reactive: Dart rất phù hợp với lập trình Reactive, với sự hỗ trợ để quản lý các đối tượng tồn tại trong thời gian ngắn, chẳng hạn như các widget UI, thông qua phân bổ đối tượng nhanh và GC. Dart hỗ trợ lập trình không đồng bộ thông qua các tính năng ngôn ngữ và API sử dụng các đối tượng Future và Stream.

1.4.4. Nhược điểm của Dart

Là ngôn ngữ lập trình mới nên hiện tại cộng đồng người sử dụng, hỗ trợ có quy mô nhỏ, chưa có quá nhiều tài nguyên phục vụ cho công việc học tập.

Chỉ bao gồm một lớp đối tượng duy nhất, không hỗ trợ quá trình lặp lại mã.

Không hỗ trợ đổi tên hàm mà không thực hiện viết câu lệnh gán mới.

Nhỏ ứng dụng thị trường: So với các ngôn ngữ lập trình khác như JavaScript hoặc Python, Dart chưa đạt được mức độ phổ biến lớn.

Tài nguyên hạn chế: So với một số ngôn ngữ lập trình khác, cộng đồng và tài nguyên học Dart vẫn chưa đủ phong phú.

Chưa được hỗ trợ rộng rãi: Mặc dù Dart có thể sử dụng để phát triển ứng dụng web thông qua framework Angular, nhưng nó chưa được sử dụng rộng rãi như JavaScript.

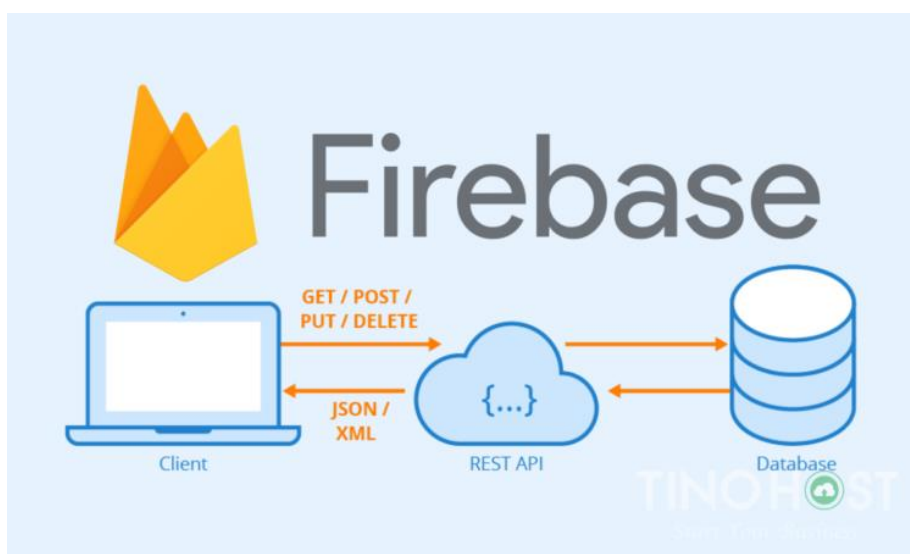
Chưa có nền tảng hỗ trợ mạnh mẽ ngoài Flutter: Ngoài Flutter, Dart chưa có nền tảng hỗ trợ mạnh mẽ cho các loại ứng dụng khác, giới hạn khả năng sử dụng của nó.

1.5. Tìm hiểu về Firebase

1.5.1. Khái niệm về Firebase

Firebase là một nền tảng giúp phát triển các ứng dụng di động trong web. Bên cạnh đó, Firebase còn được hiểu là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây cloud với hệ thống máy chủ mạnh mẽ của Google.

Firebase chứa cơ sở dữ liệu mang đến khả năng code nhanh và thuận tiện hơn. Lập trình viên có thể dễ dàng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu sẵn có.



Hình 1-13: Khái niệm về Firebase

1.5.2. Lịch sử hình thành và phát triển của nền tảng Firebase

Vào năm 2011, James Tamplin và Andrew Lee đã cho ra mắt Evolve. Đây là một nền tảng có cấu trúc khá đơn giản chuyên cung cấp các API cần thiết để tích hợp tính năng trò chuyện vào các trang web. Tuy nhiên, họ nhận ra rằng nền tảng này đang được sử dụng để truyền dữ liệu ứng dụng chứ không đơn giản là chat. Sau đó, họ đã phát triển Envole và tạo nên Firebase.

Đến tháng 4 năm 2012, Firebase đã được công bố trên toàn cầu dưới dạng một công ty riêng biệt. Những năm sau đó, Firebase đã thực hiện nhiều cuộc huy động vốn và ra mắt các sản phẩm mới.

Đến tháng 10 năm 2014, Firebase đã chính thức được Google mua lại và trở thành một ứng dụng đa năng trên nền tảng di động và website.

Ban đầu, Firebase là một sáng chế của James Tamplin và Andrew Lee. Tuy nhiên trải qua nhiều cuộc huy động vốn, Firebase đã chính thức được Google mua lại vào

năm 2014. Vì vậy, tính đến thời điểm hiện tại, nền tảng này là một phần và thuộc quyền sở hữu của Google.

1.5.3. Các tính năng chính của Firebase

1.5.3.1. Realtime Database

Realtime Database là một cơ sở dữ liệu thời gian thực. Ngay sau khi bạn đăng ký tài khoản trên Firebase, bạn sẽ nhận được Realtime Database được lưu trữ dưới dạng JSON và được đồng bộ hóa theo thời gian thực đối với mọi kết nối.

Đối với các ứng dụng được xây dựng trên đa nền tảng như Android, IOS và WebApp, tất cả client sẽ cùng sử dụng một cơ sở dữ liệu. Bên cạnh đó, hệ thống dữ liệu này sẽ tự động cập nhật khi lập trình viên phát triển ứng dụng. Sau đó, tất cả dữ liệu này sẽ được truyền tải thông qua các kết nối SSL có 2048 bit.

1.5.3.2. Authentication

Authentication là tính năng giúp xác thực danh tính của người dùng ứng dụng. Firebase cung cấp các bước xác thực thông qua Email, Facebook, Twitter, GitHub hay Google. Điều này giúp cho các thông tin cá nhân của khách hàng được bảo vệ một cách tốt nhất, hạn chế được tình trạng bị hacker đánh cắp. Đồng thời việc xác thực danh tính qua Firebase sẽ giúp người dùng tiếp cận sản phẩm nhanh chóng và an toàn hơn.

1.5.3.3. Cloud Storage

Cloud Storage là tính năng cho phép lưu trữ và quản lý nội dung đã tạo ra như ảnh, video, nội dung, văn bản,... Firebase Storage cung cấp các API hỗ trợ bạn upload và download các file từ ứng dụng một cách trơn tru mà không cần quan tâm đến chất lượng đường truyền mạng với độ bảo mật cao.

1.5.3.4. Cloud Firestore

Cloud Firestore được phát triển từ tính năng Realtime Database. Trải qua nhiều lần nâng cấp, Cloud Firestore có giao diện trực quan và khả năng mở rộng ưu việt hơn so với Realtime Database. Tính năng này của Firebase giúp đồng bộ mọi dữ liệu trên các ứng dụng thông qua việc đăng ký thời gian thực và cung cấp hỗ trợ ngoại tuyến cho thiết bị di động cũng như website.

1.5.3.5. Hosting

Hosting được phân phối thông qua tiêu chuẩn công nghệ bảo mật SSL từ hệ thống mạng CDN. CDN là một mạng lưới máy chủ giúp lưu trữ các bản sao của các

nội dung tĩnh trên website. Thông qua CDN, người dùng có thể truy cập và sử dụng các dịch vụ trên web khi cài Firebase Hosting một cách nhanh chóng và ổn định hơn.

1.5.3.6. Cloud Functions

Cloud Functions giúp chạy code backend tự động nhằm phản hồi các sự kiện được kích hoạt bởi tính năng của Firebase và HTTPS request. Cloud Functions có tính bảo mật cao và luôn thực hiện chính xác những gì mà bạn muốn.

1.5.3.7. Analytics

Analytics giúp bạn có thể phân tích hành vi của người sử dụng ứng dụng của bạn. Qua đó, bạn sẽ biết được khách hàng thường xuyên truy cập tính năng nào và các thông tin về hiệu quả quảng cáo, tình trạng trả phí,... để có thể đưa ra được chiến lược phát triển phù hợp. Để thực hiện tính năng Analytics của Firebase, bạn cần cài đặt Software Development Kit (SDK).

1.5.4. Chức năng của Firebase

1.5.4.1. Phát triển ứng dụng di động

Với cơ sở dữ liệu không lồ, Firebase giúp lập trình viên phát triển ứng dụng di động một cách nhanh chóng và giúp giảm thiểu chi phí. Song song đó, Firebase cho phép phát triển ứng dụng trên cả nền tảng iOS và Android nên có thể đáp ứng mọi yêu cầu về hệ điều hành.

1.5.4.2. Phát triển ứng dụng web

Ngoài việc hỗ trợ phát triển ứng dụng di động, Firebase còn giúp xây dựng ứng dụng website. Các API đơn giản và hữu dụng giúp bạn có thể xây dựng ứng dụng mà không cần backend và server.

1.5.4.3. Lưu trữ và đồng bộ dữ liệu

Firebase thường xuyên sao lưu dữ liệu nhằm đảm bảo tính sẵn có của dữ liệu. Bên cạnh đó, chức năng này của Firebase còn giúp cho mọi thông tin và dữ liệu của người dùng sẽ được bảo mật một cách tốt nhất nếu có trường hợp xấu xảy ra. Đặc biệt, khi sử dụng gói dịch vụ mất phí, bạn có thể dễ dàng sử dụng cấu hình cơ sở dữ liệu thời gian thực của Firebase để thực hiện các bản sao lưu tự động.

1.5.4.4. Quản lý đăng nhập

Firebase Authentication giúp nhà phát triển quản lý người dùng ứng dụng một cách đơn giản và an toàn. Cụ thể, Firebase cung cấp nhiều phương pháp quản lý đăng

nhập thông qua email với mật khẩu Google và Facebook. Việc đăng nhập này vừa đảm bảo xác thực được người dùng, vừa không để lộ thông tin cá nhân.

1.5.4.5. Phân tích dữ liệu

Chức năng phân tích dữ liệu của Firebase giúp lập trình viên có một cái nhìn tổng quan về ứng dụng thông qua các số liệu thống kê, biểu đồ, bảng biểu,... Cụ thể, bạn sẽ biết được số liệu về số khách hàng đã truy cập vào app và web, khách hàng đang sử dụng trong 30 phút vừa qua,... Qua đó, bạn có thể xây dựng được những chiến lược phát triển ứng dụng sao cho phù hợp với nhu cầu của thị trường.

1.5.4.6. Hosting

Chức năng Hosting giúp lập trình viên đơn giản hóa các thao tác với cơ sở dữ liệu khi thiết lập cấu trúc website. Qua đó, bạn sẽ tiết kiệm được thời gian và công sức khi xây dựng backend cho trang web của mình.

1.5.5. Ưu điểm của Firebase

Hiện nay, Firebase được rất nhiều lập trình viên lựa chọn vì đem lại nhiều lợi ích với các ưu điểm hơn hẳn nhiều nền tảng khác. Những ưu điểm của Firebase là:

- Sử dụng miễn phí và thuận tiện:

+ Bạn có thể sử dụng những tính năng cơ bản của Firebase miễn phí thông qua gói dịch vụ Spark. Với những tính năng cao hơn, bạn cần phải trả phí khá cao. Bên cạnh đó, cách đăng ký tài khoản Firebase cũng rất đơn giản. Bạn có thể đăng ký tài khoản Firebase thông qua tài khoản Google.

- Dễ sử dụng và tích hợp:

+ Firebase có giao diện trực quan và thân thiện. Sau khi đăng ký tài khoản, bạn chỉ cần đăng nhập vào là đã có thể sử dụng các tính năng của Firebase. Ngoài ra, Firebase có một kho backend với giao diện đa dạng, phong phú giúp bạn thoải mái lựa chọn.

- Đáp ứng nhu cầu của người dùng:

+ Firebase chính là một phần của Google. Vì vậy, Google đã không ngừng khai thác và phát triển triết để những điểm mạnh và các tính năng của Firebase nhằm tạo ra được những dịch vụ đáp ứng nhu cầu của người dùng.

- Cập nhật liên tục và đa nền tảng:

+ Firebase liên tục cập nhật và có tốc độ phát triển rất nhanh. Nhờ đó mà lập trình viên tiết kiệm được thời gian phát triển và tiếp thị ứng dụng. Bên cạnh đó, Firebase là một nền tảng đa dịch vụ. Điều này được thể hiện thông qua việc Firebase

cung cấp đa dạng các dịch vụ với mục đích phát triển website. Qua đó, nhà phát triển có thể dễ dàng lựa chọn được database Firestore hoặc Realtime theo mong muốn.

1.5.6. Nhược điểm của Firebase

Bên cạnh những ưu điểm trên, Firebase vẫn còn tồn tại những nhược điểm nhất định khiến các nhà phát triển khó có thể xây dựng ứng dụng một cách hoàn thiện nhất.

- Giới hạn về quy mô ứng dụng:

+ Firebase là một phần Subdomain của Google. Nền tảng này hiện đang bị chặn ở một số quốc gia, trong đó có Trung Quốc. Bên cạnh đó, tất cả cơ sở hạ tầng của nền tảng đều hoạt động trên Google Cloud. Điều này đồng nghĩa với việc người dùng không thể chạy ứng dụng Firebase trên những đơn vị cung cấp đám mây khác.

- Khả năng tùy chỉnh hạn chế:

+ Firebase chỉ có cơ sở dữ liệu được lưu dưới dạng JSON và không có SQL. Điều này sẽ gây khó khăn trong việc di chuyển và xử lý dữ liệu. Hơn nữa, Firebase không phải là một mã nguồn mở. Do đó, lập trình viên không thể thay đổi được mã nguồn của Firebase, từ đó gây trở ngại cho việc cải tiến sản phẩm.

+ Bên cạnh đó, nhà phát triển không có quyền truy cập mã nguồn. Vì vậy, đối với những ứng dụng lớn, việc chuyển đổi sang các nhà cung cấp khác không phải là một nhiệm vụ dễ dàng. Để thực hiện được việc này thì toàn bộ backend sẽ cần phải xây dựng lại từ đầu.

- Bảo mật còn yếu:

+ Nhược điểm bảo mật còn yếu của Firebase có thể được thể hiện thông qua việc tất cả cơ sở dữ liệu được lưu trữ dưới dạng JSON. JSON là một định dạng nhẹ, không phụ thuộc vào ngôn ngữ và dễ sử dụng. Tuy nhiên, trong một số trường hợp, JSON được đánh giá là độc hại và sẽ xảy ra các vấn đề liên quan đến bảo mật.

- Giá các dịch vụ của Firebase:

+ Hiện nay, Firebase cung cấp cả gói dịch vụ mất phí và miễn phí. Trong đó, Spark đang là gói dịch vụ miễn phí với dung lượng lưu trữ 10GB, SSL cùng với nhiều trang web và các tên miền tùy chỉnh,...

+ Còn gói dịch vụ mất phí của Firebase chính là Blaze. Gói này có tất cả các tính năng của gói Spark và có thêm một số tính năng nâng cao nữa nhằm mang đến cho người dùng những trải nghiệm tuyệt vời hơn. Hiện nay, gói Blaze có giá khoảng 0,026 USD/GB cho việc lưu trữ và tương ứng cho từng GB dung lượng.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Thực trạng và tính cấp thiết của đề tài

Trong bối cảnh nền công nghệ thông tin phát triển mạnh mẽ và sự gia tăng của số lượng người sử dụng điện thoại di động, nhu cầu trong việc tìm kiếm thông tin du lịch và quản lý kế hoạch du lịch trở nên ngày càng cần thiết. Quảng Bình, với cảnh quan thiên nhiên tươi đẹp và lịch sử văn hóa phong phú, là một điểm đến hấp dẫn cho du khách. Tuy nhiên, việc tìm kiếm thông tin và lập kế hoạch cho chuyến đi có thể gặp khó khăn do sự phân tán thông tin và thiếu sự hỗ trợ từ các ứng dụng di động.

Ứng dụng sổ tay du lịch Quảng Bình sẽ giải quyết những thách thức này bằng cách cung cấp một nền tảng để người dùng có thể dễ dàng tìm kiếm thông tin du lịch, lập kế hoạch cho chuyến đi và chia sẻ kinh nghiệm du lịch của họ. Điều này không chỉ tạo điều kiện thuận lợi cho du khách mà còn giúp quảng bá hình ảnh du lịch của Quảng Bình, thúc đẩy phát triển kinh tế và du lịch của địa phương. Vì vậy, đây chính là động lực và mục tiêu của đề tài: *“Nghiên cứu và xây dựng ứng dụng sổ tay du lịch Quảng Bình trên điện thoại di động”*.

2.2. Mục tiêu và phạm vi nghiên cứu

2.2.1. Mục tiêu

Mục tiêu của đề tài là nghiên cứu và xây dựng thành công ứng dụng sổ tay du lịch Quảng Bình trên điện thoại di động, giúp người dùng có thể:

- Tìm kiếm và xem thông tin chi tiết về các điểm đến du lịch trong Quảng Bình.
- Lập kế hoạch du lịch và quản lý lịch trình du lịch cá nhân.
- Quản lý thông tin cá nhân.

2.2.2. Phạm vi nghiên cứu

Phạm vi nghiên cứu sẽ tập trung vào việc phát triển các tính năng cơ bản của ứng dụng, bao gồm:

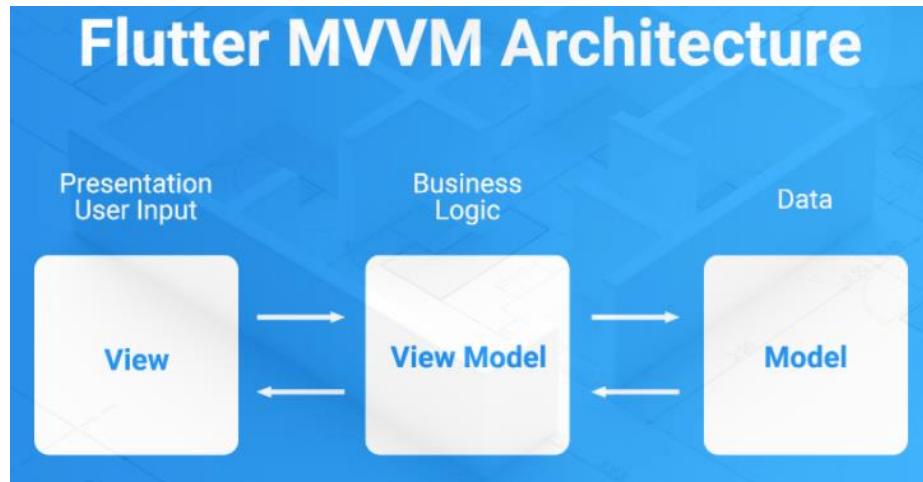
- Xem và tìm kiếm thông tin chi tiết về các địa điểm du lịch.
- Lập kế hoạch lịch trình du lịch cá nhân.
- Quản lý thông tin cá nhân.

2.3. Mô hình sử dụng

2.3.1. Khái niệm mô hình MVVM

MVVM viết tắt là Model View ViewModel – được phát triển bởi Microsoft và sử dụng cho Windows Presentation Foundation (WPF) và Silverlight. Đến 2005, được

John Grossman công bố bằng một bài blog về Avalon (mật mã của Windows Presentation Foundation). Mô hình MVVM phát triển dựa trên cả MVC và MVP, MVVM được phát triển để tách biệt giao diện UI khỏi logic nghiệp vụ và hành vi trong một ứng dụng [5].



Hình 2-1: Mô hình MVVM(Model View ViewModel)

2.3.2. Các thành phần trong mô hình

2.3.2.1. Model

Vai trò chính của Model trong MVVM là thực hiện logic nghiệp vụ trong mẫu thiết kế mà chỉ hoạt động dựa trên nguồn dữ liệu cần thiết trong một hoạt động. Hiểu theo cách khác, phần tử model của mẫu kiến trúc MVVM được sử dụng để đại diện cho dữ liệu thời gian thực cuối cùng sẽ được sử dụng trong phát triển ứng dụng.

Điều tốt nhất về model là hiển thị sự tương tác giữa tất cả các thành phần hoàn toàn bằng cách tìm nạp dữ liệu từ cơ sở dữ liệu phòng. Nếu chúng ta phải xác định model trong một câu duy nhất, chúng ta có thể nói – nó là một phần tử lưu trữ dữ liệu và logic liên quan của kiến trúc

2.3.2.2. View

View là viết tắt của các thành phần UI như HTML, CSS. Trong MVVM, View chịu trách nhiệm về lớp trình bày trong mẫu thiết kế và là một điểm vào ứng dụng. Khi chúng ta nói về lợi ích của kiến trúc MVVM, các thuộc tính của view đã được ca ngợi. Có một tập hợp phân cấp được duy trì giữa mỗi mô hình MVVM trong khi thể hiện sự tương tác. Ví dụ: View không bao giờ tương tác trực tiếp với Model mà thông qua ViewModel. Tất cả dữ liệu được thu thập bởi Model giúp tạo dữ liệu bản trình bày cho View. View thực thi logic thiết kế UI-UX và yêu cầu mô hình gửi lại đầu ra cho người dùng.

2.3.2.3. ViewModel

Ngoài việc làm trung gian giữa các thành phần Model và View, ViewModel triển khai dữ liệu và các lệnh được kết nối với View trong kiến trúc MVVM để thông báo cho phần tử sau về các thay đổi trạng thái. Có một ví dụ ViewModel Android để thực hiện các công việc khác nhau như tạo lớp con hoặc thêm phần phụ thuộc, v.v. ViewModels có thể được liên kết với một hoặc nhiều mô hình.

Về cơ bản, vai trò của ViewModel trong kiến trúc MVVM là hỗ trợ trạng thái của View và hoạt động như một lớp logic của toàn bộ cộng đồng kiến trúc MVVM. Sẽ không sai khi trích dẫn ViewModel là phần tích hợp của tất cả những gì làm cho Mô hình duy trì dữ liệu thực tế và phần View đóng gói dữ liệu cập nhật giữ bộ điều khiển như một công giữa chúng [5].

2.3.3. Lợi ích của kiến trúc MVVM dành cho ứng dụng Android và iOS

2.3.3.1. Khả năng tái sử dụng

Khi hiểu được toàn bộ khái niệm về việc giới thiệu kiến trúc MVVM trong quá trình thiết kế ứng dụng, có thể xác định đây là mô hình giúp các nhà phát triển phân biệt rõ ràng giữa các ứng dụng được kết hợp lỏng lẻo và kết hợp chặt chẽ. Khớp nối là yếu tố phụ thuộc của thiết kế trong ứng dụng và có những thiết kế kiến trúc có khớp nối chặt chẽ, điều này cuối cùng làm tăng chi phí bảo trì phát triển trang web và giảm khả năng tái sử dụng của thành phần. Kiến trúc MVVM đi kèm với khớp nối lỏng lẻo để đạt được sự phân tách trách nhiệm và nâng cao mức độ khớp nối lỏng lẻo. Mặt khác, yếu tố khả năng tái sử dụng cho phép các mã được sử dụng để xây dựng các mẫu thiết kế trong các ứng dụng khác.

2.3.3.2. Thúc đẩy phát triển mã độc lập

MVVM là một mô hình kiến trúc độc lập cung cấp dữ liệu dưới dạng đầu ra thông qua View. Toàn bộ cơ sở mã của mẫu MVVM được tạo ra theo cách khắc phục các trục trặc phổ biến mà hầu hết các mẫu kiến trúc trở thành nạn nhân của nó. Với ba thành phần chính của nó, kiến trúc MVVM truy xuất và lưu giữ các đối tượng thông qua một dịch vụ. Dù nói đến thiết kế mô hình kiến trúc MVVM Android Architecture hoặc iOS , mô hình MVVM thúc đẩy sự phát triển độc lập của ứng dụng. Vì kiến trúc mã và đơn giản hóa mẫu thiết kế là mục tiêu cốt lõi của các nhà phát triển, phương pháp tiếp cận mẫu thiết kế MVVM sẽ giải quyết một số vấn đề tồn tại trong khuôn khổ thiết kế của một ứng dụng và xem xét tất cả các khía cạnh quan trọng mang lại kết quả tuyệt vời ở phần cuối của dự án.

2.3.3.3. Nâng cao khả năng kiểm tra

Mẫu thiết kế MVVM Android hoặc iOS dường như đang thu hút sự chú ý trên thị trường do các nhà phát triển có kỹ năng điều hành. Phần lớn các nhà phát triển tin rằng MVVM có thể đọc được, có thể mở rộng và có thể kiểm tra được so với các mô hình thiết kế kiến trúc khác. Trong số tất cả các thành phần, ViewModel có trách nhiệm rất lớn là đại diện cho dữ liệu và khởi tạo trường hợp thử nghiệm trong mô hình kiến trúc MVVM. Chính ViewModel làm cho thành phần View có thể tái sử dụng và có thể kiểm tra được, giúp cho việc khen ngợi logic nghiệp vụ kiểm thử đơn vị trở nên dễ dàng hơn. Do sự phụ thuộc của phần tử này, việc bắt đầu các trường hợp thử nghiệm trở nên ít phức tạp hơn một chút.

2.3.3.4. Khả năng bảo trì

Mặc dù một số nhà phát triển có thể thấy các chỉ số hiệu suất ứng dụng dành cho thiết bị di động và tính năng chia sẻ của kiến trúc MVVM hơi khó chịu và phức tạp, nhưng khả năng bảo trì của kiến trúc MVVM mang lại lợi ích cho mô hình là mạch lạc và đơn giản hóa, lưu ý các yêu cầu bổ sung. Khả năng bảo trì lồng vào nhau cung cấp khả năng mở rộng tối đa với sự phân tách ranh giới rõ ràng.

Khả năng thay thế hoặc thêm các khối mã mới trong quá trình ứng dụng là một điều lý tưởng cần xem xét để đặt các mã trong cơ sở hạ tầng ứng dụng một cách phù hợp. Ngoài ra, tính năng ánh xạ trong kiến trúc MVVM là chưa từng có. Mô hình MVVM sử dụng ánh xạ một đến nhiều giữa View và ViewModel.

“Vì những đặc điểm và lợi ích trên nên em chọn mô hình MVVM để sử dụng cho hệ thống được xây dựng.”

2.4. Phân tích chức năng của hệ thống

Các tác nhân của hệ thống gồm có:

❖ Người dùng:

- Đăng ký và đăng nhập tài khoản:

+ Cho phép người dùng tạo tài khoản mới bằng cách điền tên tài khoản, số điện thoại, email và mật khẩu.

+ Người dùng đăng nhập bằng email và mật khẩu.

+ Người dùng có thể chỉnh sửa thông tin cá nhân của mình.

+ Đăng xuất.

- Tạo lịch trình du lịch

+ Người dùng có thể lựa chọn các địa điểm, ngày, giờ để tạo kế hoạch du lịch cho bản thân.

- Xem lại lịch trình đã tạo

+ Người dùng có thể xem lại, chỉnh sửa, xóa lịch trình mà mình đã tạo.

- Xem bản đồ

+ Người dùng có thể xem bản đồ chỉ đường đến các địa điểm đã chọn.

- Chọn các ngôn ngữ

+ Người dùng có thể thay đổi ngôn ngữ qua lại giữa tiếng việt và tiếng anh.

❖ Quản trị viên:

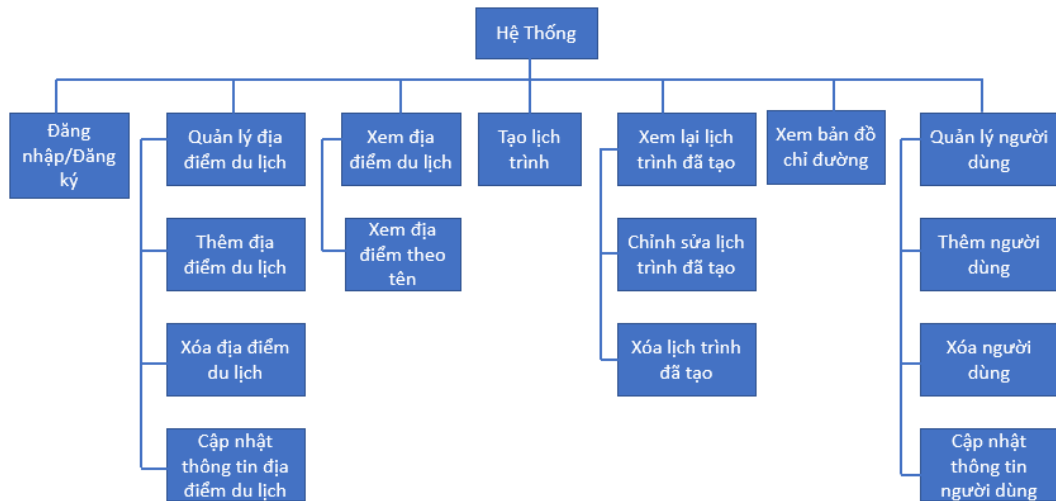
- Quản lý tài khoản người dùng:

+ Quản trị viên có thêm xem, chỉnh sửa, xóa, và tạo tài khoản cho người dùng.

- Quản lý các địa điểm du lịch:

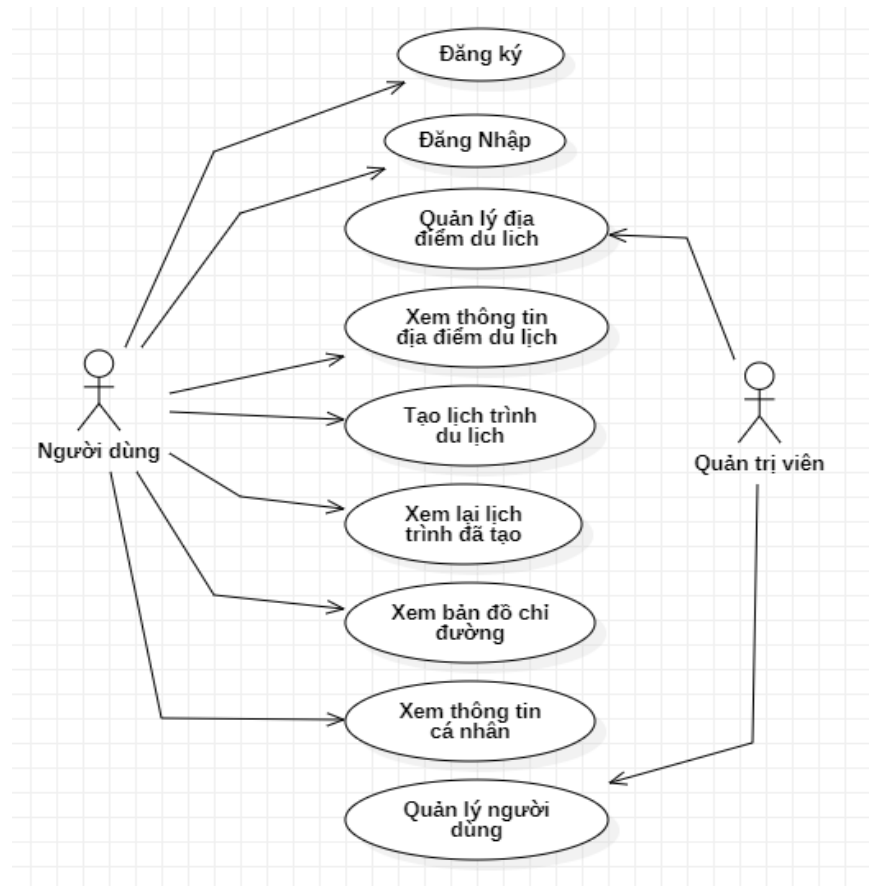
+ Quản trị viên có thêm chỉnh sửa, cập nhập, xóa thông tin các địa điểm du lịch.

2.5. Sơ đồ mô tả chức năng chính của hệ thống



Hình 2-2: Sơ đồ chức năng chính của hệ thống

2.6. Use case hệ thống



Hình 2-3. Sơ đồ use case hệ thống tổng quát

Hệ thống có 2 tác nhân chính:

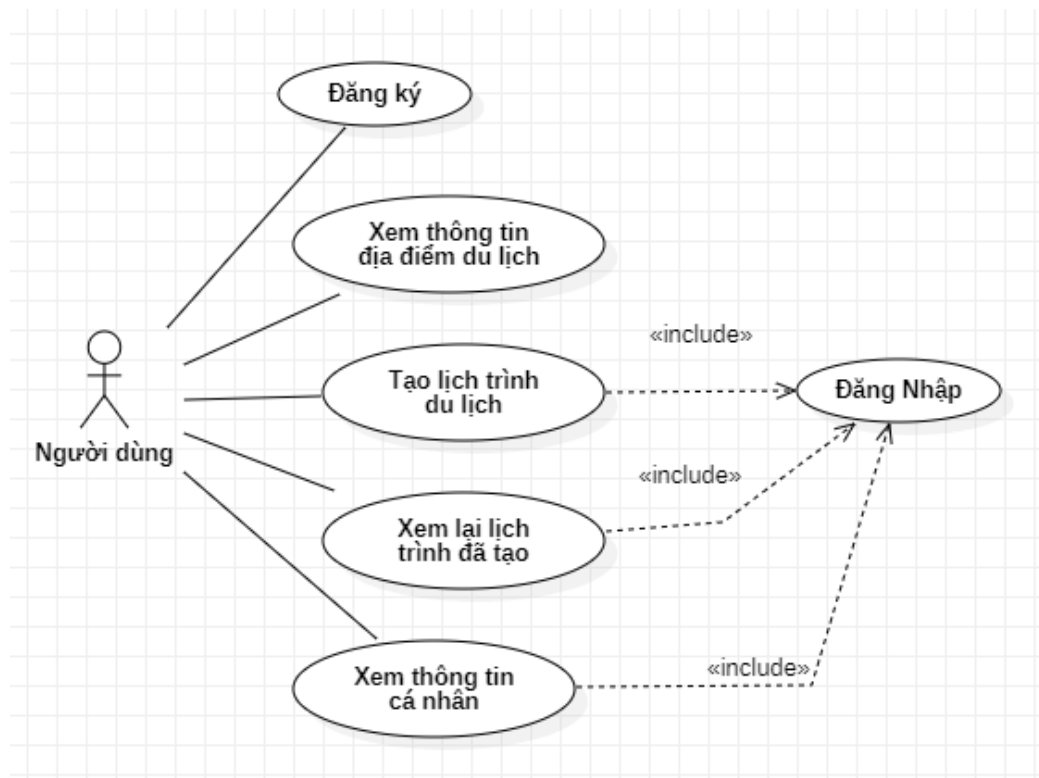
- Quản trị viên
- Người dùng

Các tác nhân này có thể thực hiện các use case khác nhau để quản lý hệ thống.

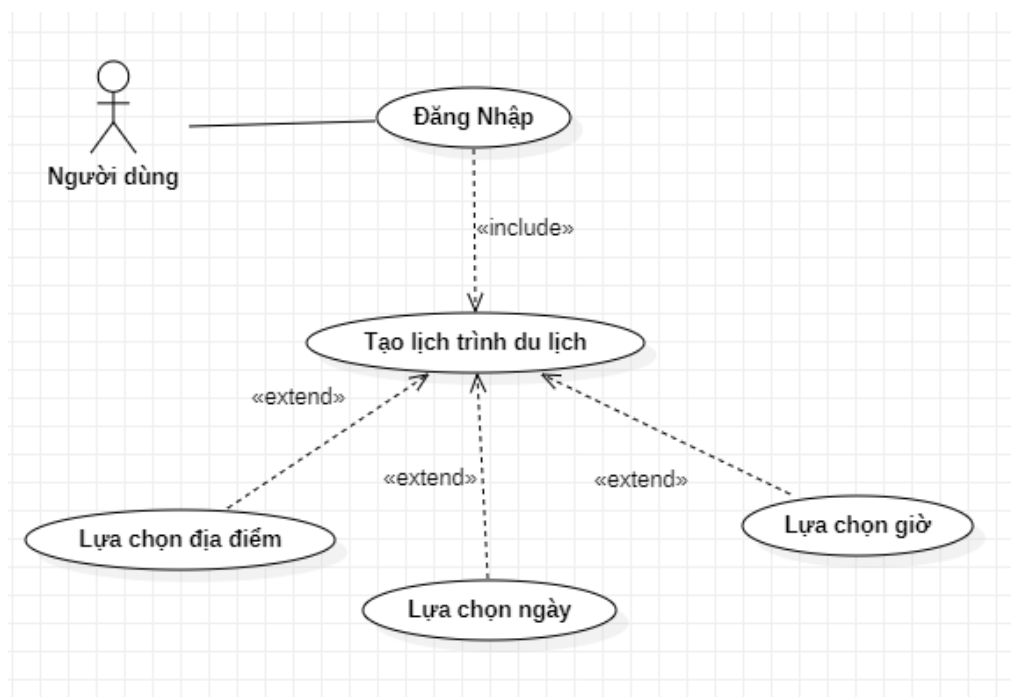
Use case bao gồm các hoạt động chính của hệ thống, bao gồm:

- Đăng nhập
- Đăng ký
- Quản lý địa điểm du lịch
- Xem thông tin địa điểm du lịch
- Tạo lịch trình
- Xem lại lịch trình đã tạo
- Xem bản đồ chỉ đường
- Quản lý người dùng

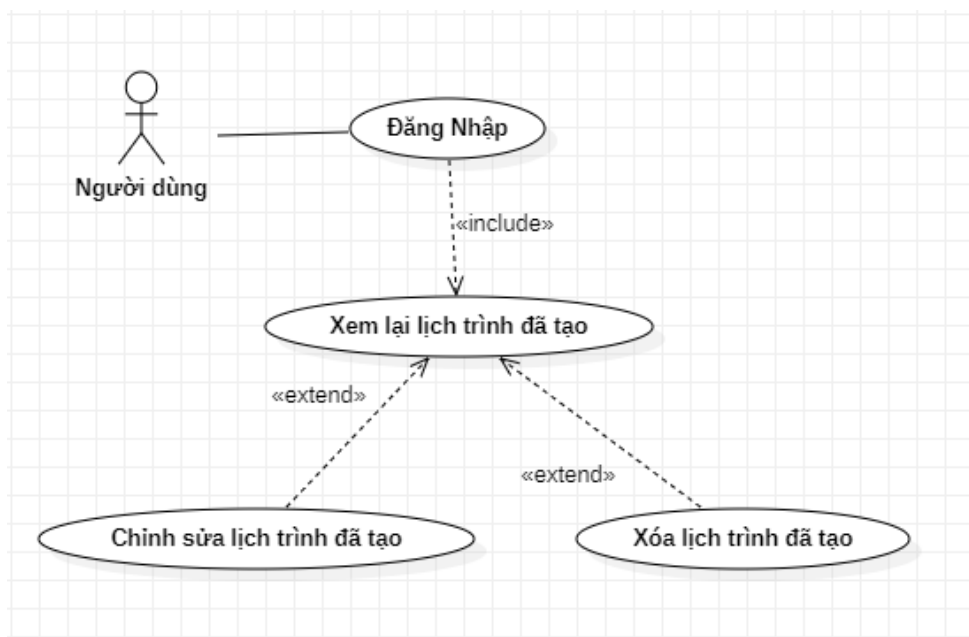
2.6.1. Sơ đồ phân rã chức năng cho người dùng



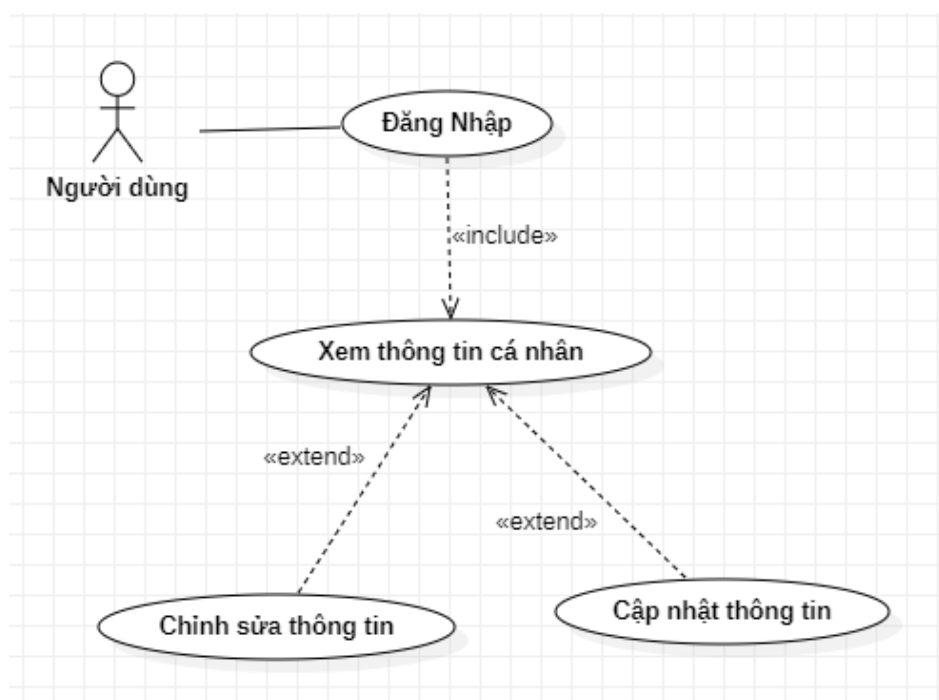
Hình 2-4. Sơ đồ tổng quát phân rã chức năng cho người dùng



Hình 2-5: Sơ đồ Use case Tạo lịch trình du lịch

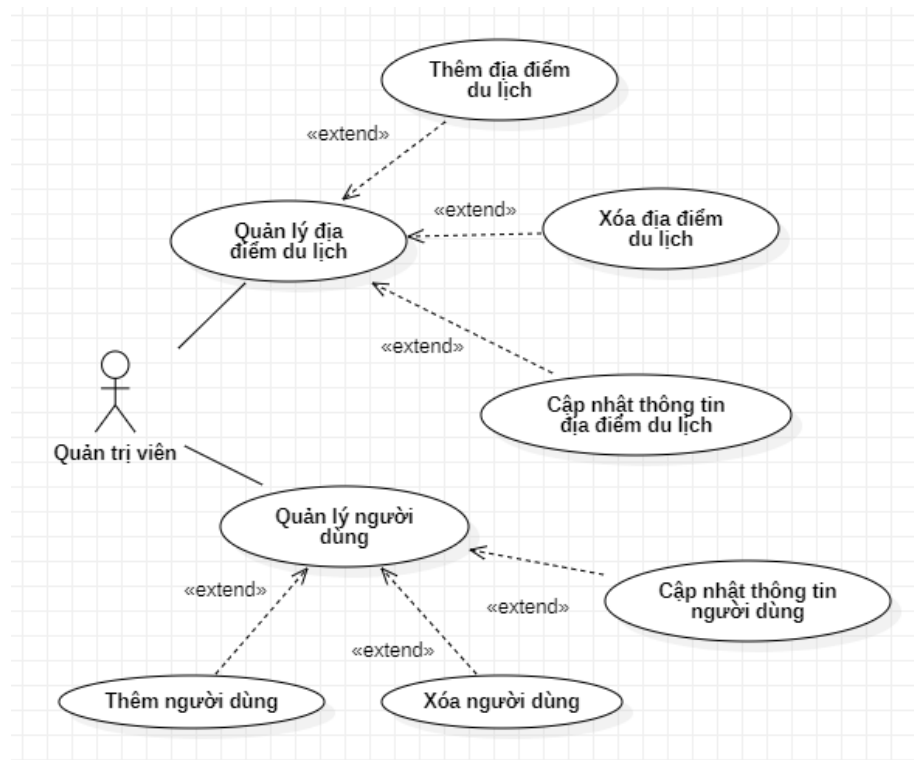


Hình 2-6: Sơ đồ Use case Xem lại lịch trình đã tạo



Hình 2-7: Sơ đồ Use case Xem thông tin cá nhân

2.6.2. Sơ đồ phân rã chức năng cho admin



Hình 2-8. Sơ đồ tổng quát phân rã chức năng cho admin

2.7. Mô tả chức năng của từng use case

Use case Đăng ký:

Use case 1	Đăng Ký	
Mô tả	Cho phép người dùng đăng ký tài khoản vào hệ thống bằng cách đăng ký tên người dùng, số điện thoại, email và mật khẩu. Nếu thông tin đăng ký bị trùng email thì hệ thống sẽ thông báo lỗi.	
Actor	Người dùng	
Luồng chính	Bước	Hành động
	1	Người dùng truy cập vào hệ thống
	2	Hệ thống hiển thị form đăng ký.
	3	Người dùng đăng ký tên, số điện thoại, email và mật khẩu.

	4	Hệ thống hiển thị đăng ký thành công.
	5	Nếu thông tin email đã tồn tại thì hệ thống báo lỗi và phải nhập lại email đăng ký.

Bảng 2-1: Bảng use case đăng ký

Use case Đăng nhập:

Use case 2	Đăng Nhập	
Mô tả	Cho phép người dùng đăng nhập vào hệ thống bằng cách cung cấp email và mật khẩu hợp lệ. Nếu thông tin đăng nhập không chính xác, hệ thống sẽ hiển thị thông báo lỗi.	
Actor	Người dùng	
Luồng chính	Bước	Hành động
	1	Người dùng truy cập vào hệ thống
	2	Hệ thống hiển thị form đăng nhập.
	3	Người dùng nhập email và mật khẩu.
	4	Hệ thống hiển thị đăng nhập thành công.
	5	Nếu sai thông tin email hoặc mật khẩu thì hệ thống báo lỗi và bắt nhập lại thông tin.

Bảng 2-2: Bảng use case đăng nhập

Use case Quản lý địa điểm du lịch:

Use case 3	Quản lý địa điểm du lịch
Mô tả	Cho phép quản trị viên thêm, xóa và cập nhật thông tin về các điểm du lịch trong hệ thống. Các thông tin về điểm du lịch bao gồm tên địa điểm, địa chỉ, mô tả địa điểm, ảnh, tác giả, nguồn trích dẫn, vị trí địa lý.
Actor	Quản trị viên

Luồng chính	Bước	Hành động
	1	Quản trị viên truy cập vào hệ thống.
	2	Tiến hành thêm, xóa và cập nhật thông tin về các địa điểm trong hệ thống.
	3	Hệ thống hiển thị thêm, xóa và cập nhật thông tin địa điểm du lịch thành công

Bảng 2-3: Bảng use case quản lý địa điểm du lịch

Use case Xem thông tin địa điểm du lịch:

Use case 4	Xem thông tin địa điểm du lịch	
Mô tả	Cho phép người dùng xem các thông tin chi tiết của địa điểm (tên, địa chỉ, hình ảnh, mô tả, vị trí địa lý) có trong hệ thống. Người dùng có thể tìm kiếm các địa điểm du lịch theo tên.	
Actor	Người dùng	
Luồng chính	Bước	Hành động
	1	Người dùng truy cập vào hệ thống.
	2	Tiến hành tìm kiếm địa điểm muốn xem.
	3	Hệ thống hiển thị thông tin chi tiết của địa điểm du lịch.

Bảng 2-4: Bảng use case xem thông tin địa điểm du lịch

Use case Tạo lịch trình du lịch:

Use case 5	Tạo lịch trình du lịch
Mô tả	Cho phép người dùng truy cập vào hệ thống sau đó lên lịch trình đi du lịch cho bản thân. Lựa chọn các địa điểm, lựa chọn ngày giờ theo mong muốn của bản thân.
Actor	Người dùng

Luồng chính	Bước	Hành động
	1	Người dùng đăng nhập thành công vào hệ thống để thực hiện chức năng này.
	2	Người dùng tạo lịch trình bằng cách chọn các địa điểm, ngày, giờ theo mong muốn.
	3	Hệ thống thông báo đã tạo lịch trình thành công.

Bảng 2-5: Bảng use case tạo lịch trình du lịch

Use case xem lại lịch trình đã tạo:

Use case 6	Xem lại lịch trình đã tạo	
Mô tả	Cho phép người dùng xem lại các lịch trình đã tạo để có thể chỉnh sửa hoặc xóa nếu có thay đổi đột xuất.	
Actor	Người dùng	
Luồng chính	Bước	Hành động
	1	Người dùng đăng nhập thành công vào hệ thống.
	2	Tiến hành xem lại lịch trình đã tạo.
	3	Hệ thống hiển thị danh sách các lịch trình đã được tạo trước đó
	4	Người dùng tiến hành chỉnh sửa hoặc xóa lịch trình nếu có thay đổi đột xuất
	5	Hệ thống hiển thị thông báo thay đổi hoặc xóa lịch trình thành công

Bảng 2-6: Bảng use case xem lại lịch sử cảnh báo

Use case xem bản đồ chỉ đường:

Use case 7	Xem bản đồ chỉ đường	
Mô tả	Cho phép người dùng xem bản đồ chỉ đường đến các địa điểm đã chọn.	
Actor	Người dùng	
Luồng chính	Bước	Hành động
	1	Người dùng chọn địa điểm du lịch muốn đến.
	2	Hệ thống hiển thị đường đi chỉ dẫn đến địa điểm mà người dùng đã chọn.

Bảng 2-7: Bảng use case xem bản đồ chỉ đường

Use case xem thông tin cá nhân:

Use case 8	Xem thông tin cá nhân	
Mô tả	Cho phép người dùng xem thông tin cá nhân của mình. Các thông tin cá nhân gồm ảnh, tên, số điện thoại, và email.	
Actor	Người dùng	
Luồng chính	Bước	Hành động
	1	Người dùng đăng nhập thành công vào hệ thống.
	2	Tiến hành chỉnh sửa hoặc cập nhật thông tin cá nhân lại nếu muốn. Các thông tin có thể chỉnh sửa là tên, số điện thoại và hình ảnh.
	3	Hệ thống hiển thị chỉnh sửa hoặc cập nhật thông tin cá nhân thành công.

Bảng 2-8: Bảng use case xem thông tin cá nhân

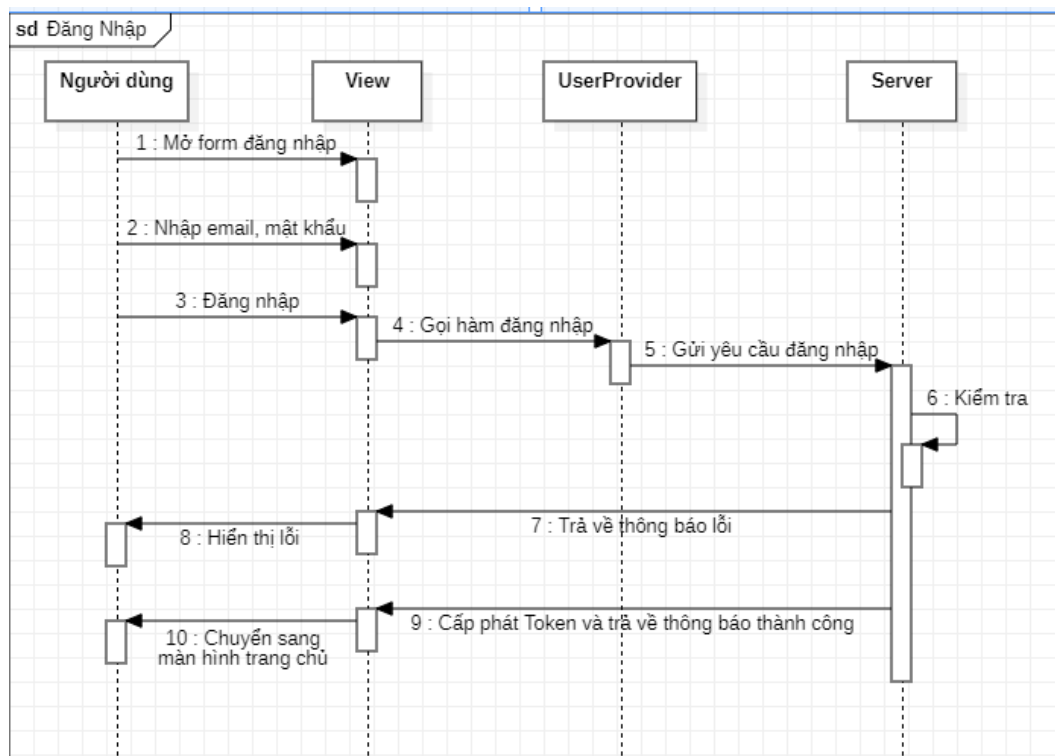
Use case quản lý người dùng:

Use case 9	Quản lý người dùng	
Mô tả	Cho phép quản trị viên thêm, xóa và cập nhật tài khoản người dùng trong hệ thống.	
Actor	Quản trị viên	
Luồng chính	Bước	Hành động
	1	Quản trị viên truy cập vào hệ thống.
	2	Tiến hành thêm, xóa và cập nhật tài khoản người dùng.
	3	Hệ thống hiển thị thêm, xóa và cập nhật tài khoản người dùng thành công.

Bảng 2-9: Bảng use case quản lý người dùng

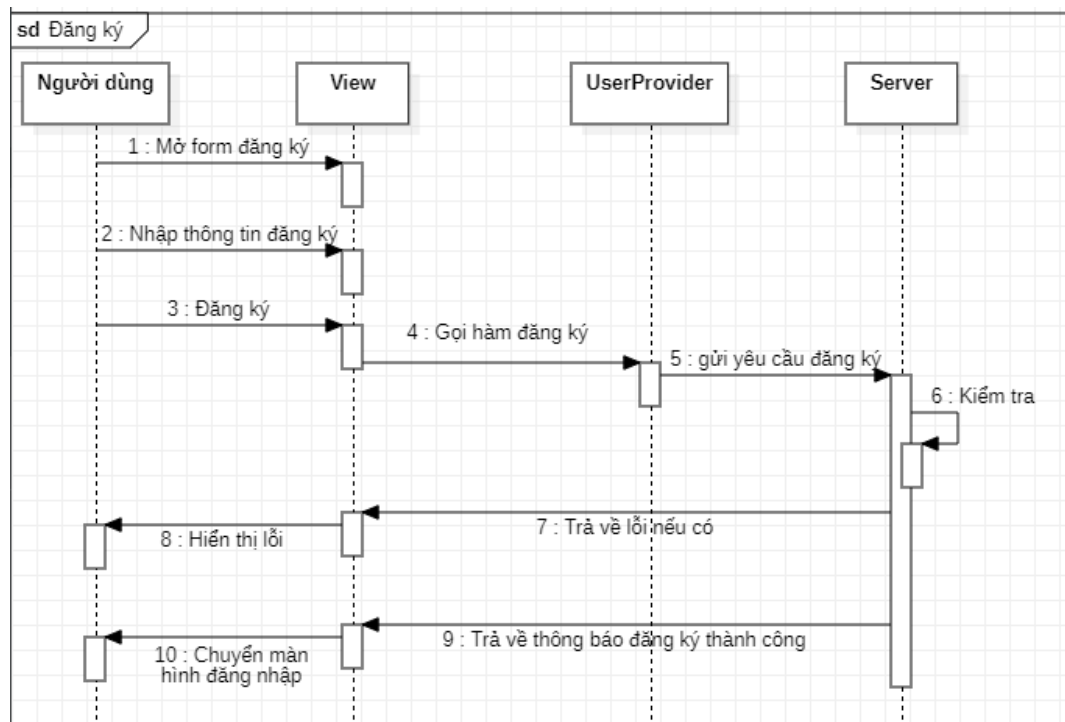
2.8. Sơ đồ tuần tự các chức năng

- Đăng nhập



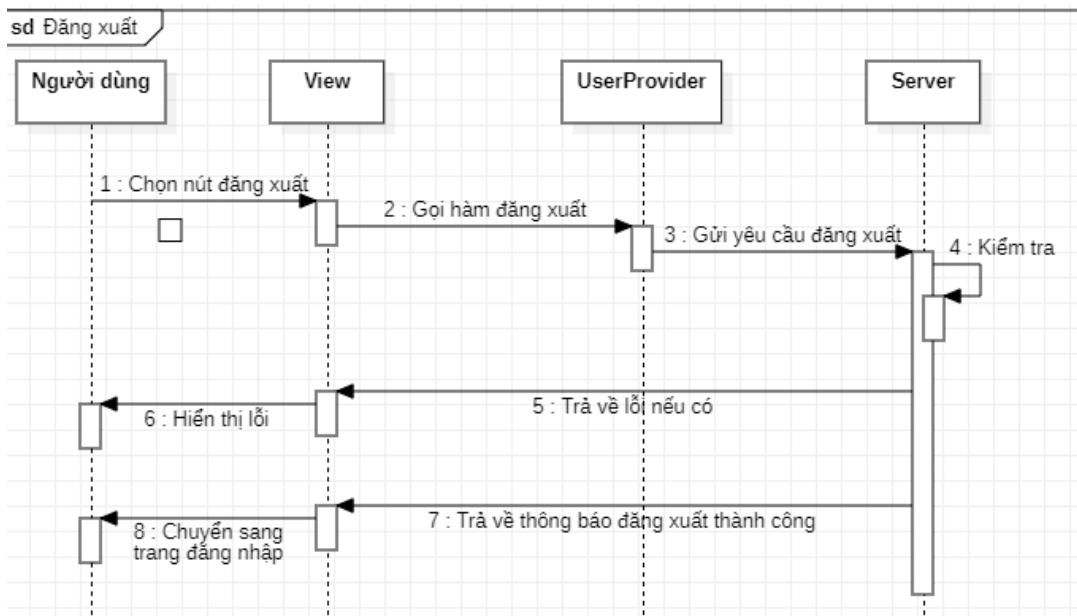
Hình 2-9. Sơ đồ tuần tự đăng nhập

- Đăng ký



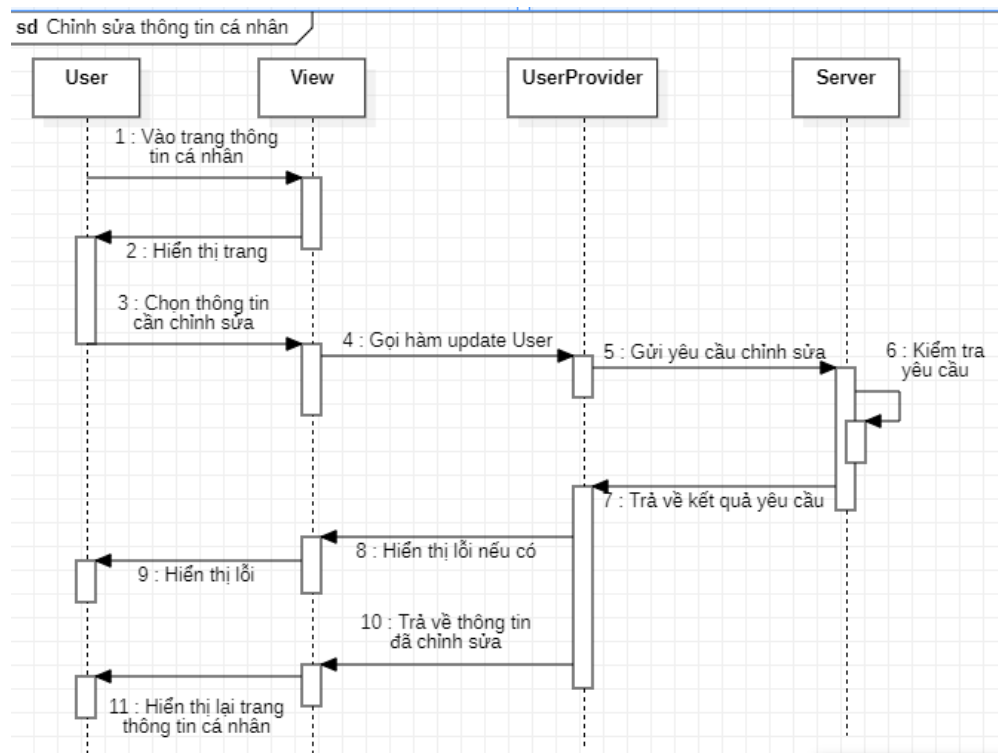
Hình 2-10. Sơ đồ tuần tự đăng ký

- Đăng xuất



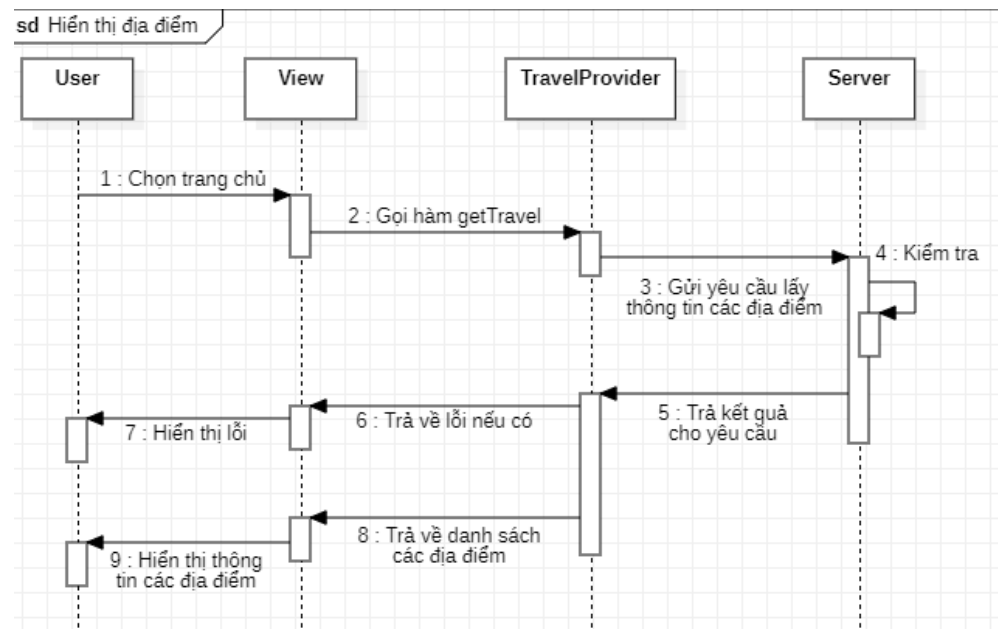
Hình 2-11. Sơ đồ tuần tự đăng xuất

- Chỉnh sửa thông tin cá nhân



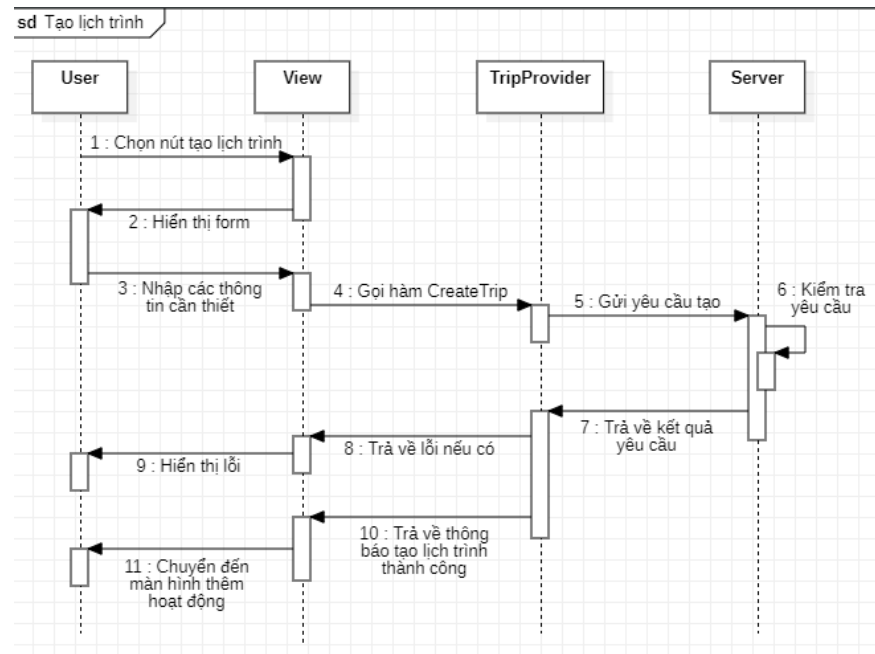
Hình 2-12: Sơ đồ tuần tự chỉnh sửa thông tin cá nhân

- Hiển thị địa điểm



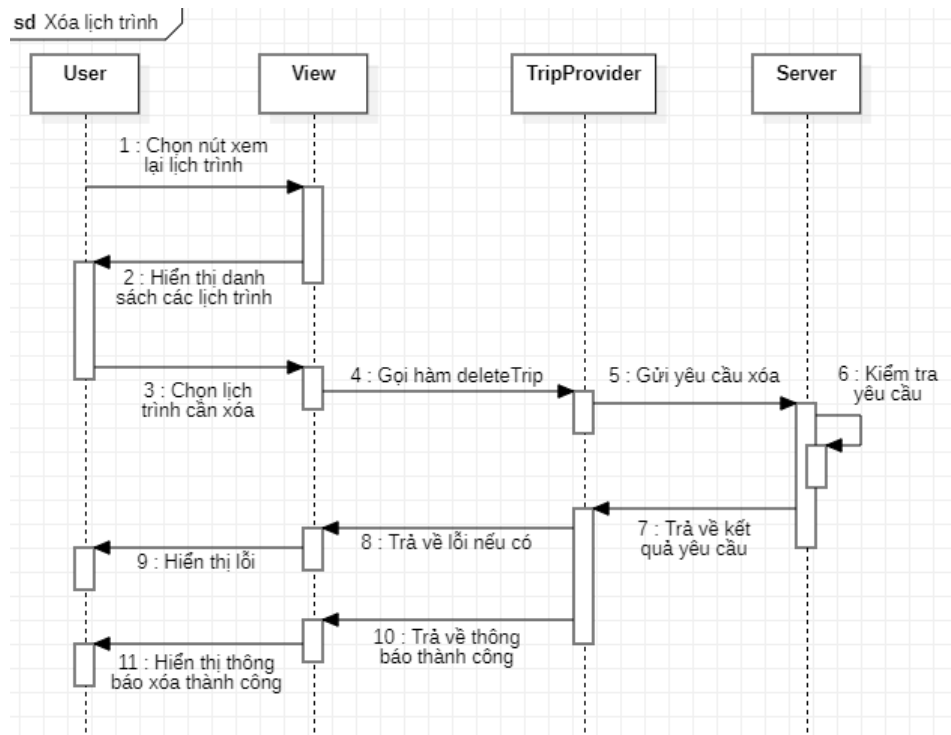
Hình 2-13: Sơ đồ tuần tự hiển thị địa điểm

- Tạo lịch trình



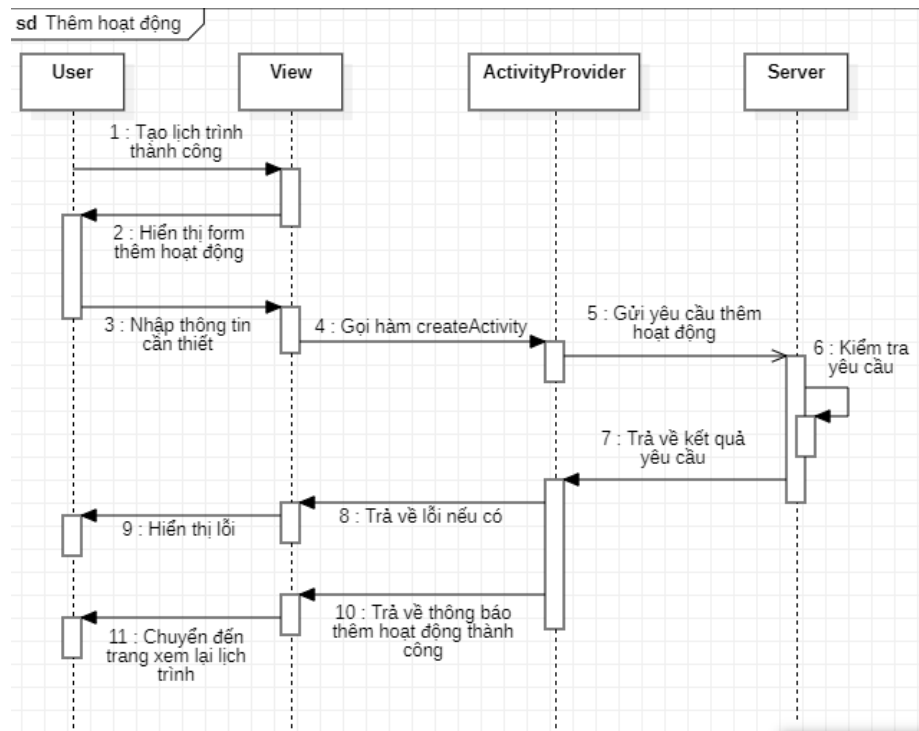
Hình 2-14: Sơ đồ tuần tự Tạo lịch trình

- Xóa lịch trình



Hình 2-15: Sơ đồ tuần tự xóa lịch trình

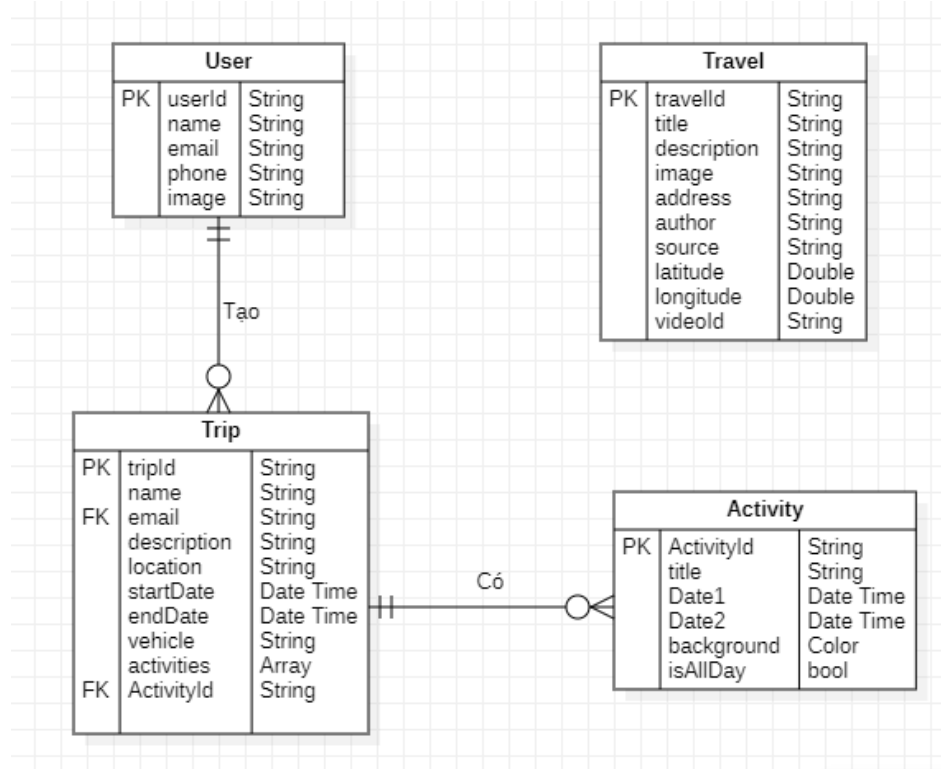
- Thêm hoạt động



Hình 2-16: Sơ đồ tuần tự thêm hoạt động

2.9. Lược đồ cơ sở dữ liệu

- Lược đồ:



Hình 2-17: Lược đồ cơ sở dữ liệu

- Bảng User

STT	Trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	userId	String	Not null, primary key	ID của từng tài khoản
2	name	String	Not null	Tên tài khoản
3	email	String	Not null	Địa chỉ email
4	phone	String	Not null	Số điện thoại
5	image	String	Not null	Ảnh đại diện

Bảng 2-10: Bảng dữ liệu User

- Bảng Travel

STT	Trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	travelId	String	Not null, primary key	ID của từng địa điểm
2	title	String	Not null	Tên địa điểm
3	description	String	Not null	Mô tả địa điểm
4	image	String	Not null	Ảnh địa điểm
5	address	String	Not null	Địa chỉ địa điểm
6	author	String	Not null	Tác giả
7	source	String	Not null	Nguồn
8	latitude	double	Not null	Kinh độ
9	longitude	double	Not null	Vĩ độ
10	videoId	String	Not null	Id của video

Bảng 2-11: Bảng dữ liệu Travel

- Bảng Trip

STT	Trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	tripId	String	Not null, primary key	ID của từng lịch trình
2	name	String	Not null	Tên lịch trình
3	email	String	Not null	Người tạo
4	description	String	Not null	Mô tả lịch trình
5	location	String	Not null	Địa điểm đi
6	startDate	Date Time	Not null	Ngày đi
7	endDate	Date Time	Not null	Ngày về
8	vehicle	String	Not null	Phương tiện
9	activities	Array	Allow null	Các hoạt động

Bảng 2-12: Bảng dữ liệu Trip

- Bảng Activity

STT	Trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	ActivityId	String	Not null, primary key	ID của từng hoạt động
2	title	String	Not null	Tên hoạt động
3	Date1	Date Time	Not null	Ngày giờ đi
4	Date2	Date Time	Not null	Ngày giờ về
5	background	Color	Not null	Màu sắc
6	isAllDay	bool	Not null	Hoạt động cả ngày

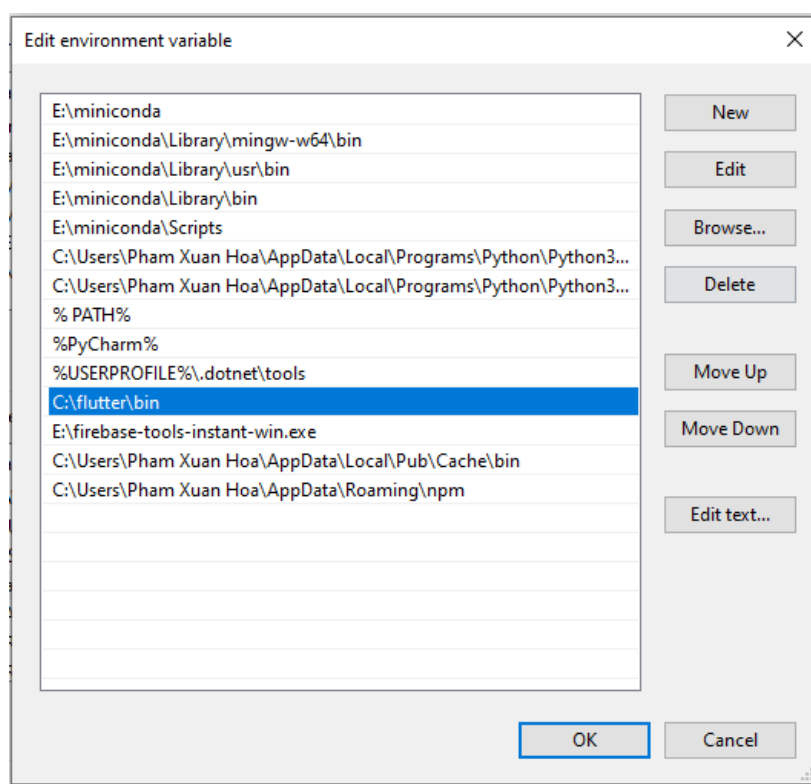
Bảng 2-13: Bảng dữ liệu Activity

CHƯƠNG 3: CÀI ĐẶT VÀ GIAO DIỆN CỦA HỆ THỐNG

3.1. Cài đặt hệ thống

Hệ thống cài đặt bằng ngôn ngữ lập trình Dart và sử dụng framework Flutter, bộ giao diện người thiết kế bởi Google là Material, bộ giao diện người dùng được thiết kế bởi Apple là Cupertino, và sử dụng cơ sở dữ liệu Firebase cho phép bạn truy cập và thao tác với cơ sở dữ liệu đám mây và tích hợp sẵn các tính năng thời gian thực (real-time) và xử lý sự kiện.

- Cài đặt Flutter: Đầu tiên, các bạn tải Flutter SDK từ trang official download. Sau khi tải xong, bạn cần thêm Flutter vào PATH để có truy cập Flutter từ bất kể đâu trong command line.



Hình 3-1: Thêm Flutter SDK vào Path

- Cài đặt ngôn ngữ Dart: Sau khi cài đặt xong Flutter SDK, bạn có thể cài đặt thêm Dart. Thông thường thì trong bộ Flutter SDK đã có sẵn một số thư viện Dart cần thiết. Nhưng nếu bạn cần cài đặt thêm các công cụ dành cho nhà phát triển Dart thì bạn nên cài đặt Dart.

- Cài đặt Install Android SDK Studio: muốn xây dựng app cho Android thì bước này cần phải thực hiện. Khi bạn tải và cài đặt Android Studio từ trang chủ thì về cơ bản là bạn đã có đầy đủ những thứ cần thiết cho flutter.

Flutter yêu cầu 3 công cụ gồm có:

- + Android SDK
- + Android SDK Platform-Tools
- + Android SDK Build-Tools

- Sử dụng Pub Package Manager để quản lý các gói trong dự án Flutter. Trong dự án Flutter sẽ có file cấu hình: pubspec.yaml, file sẽ chứa tất cả các Dart dependencies của dự án. Bạn sử dụng lệnh bên dưới để PPM(Pub Package Manager) cài đặt tất cả dart dependencies trong pubspec.yaml

```
$ flutter packages get
```

Còn bạn muốn upgrade các dart dependencies thì có thể dùng lệnh:

```
$ flutter package upgrade
```

- Tạo một project Flutter: Có nhiều cách để tạo mới một dự án với flutter, bạn có thể sử dụng dòng lệnh (CLI), hoặc Android Studio hoặc IntelliJ... Flutter có thể tạo một dự án mẫu bằng lệnh sau:

```
$ flutter create name_of_your_app
```

Ngay khi lệnh trên hoàn thành, bạn có thể chạy ngay ứng dụng để xem hình thù nó thế nào.

```
$ cd name_of_your_app
```

```
$ flutter run
```

- Cấu hình cơ sở dữ liệu: Sử dụng cơ sở dữ liệu Firebase kết nối với dự án vừa tạo.

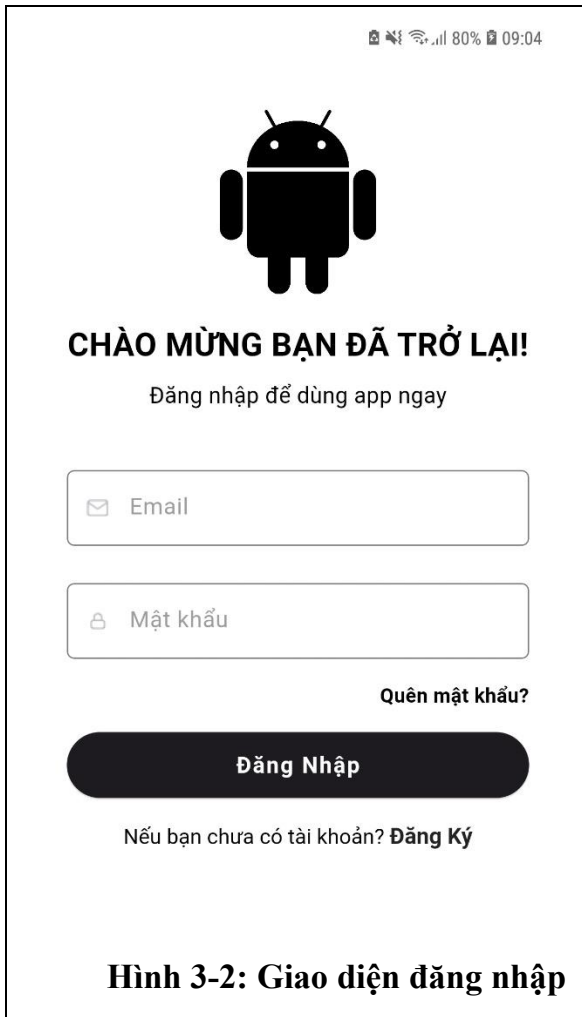
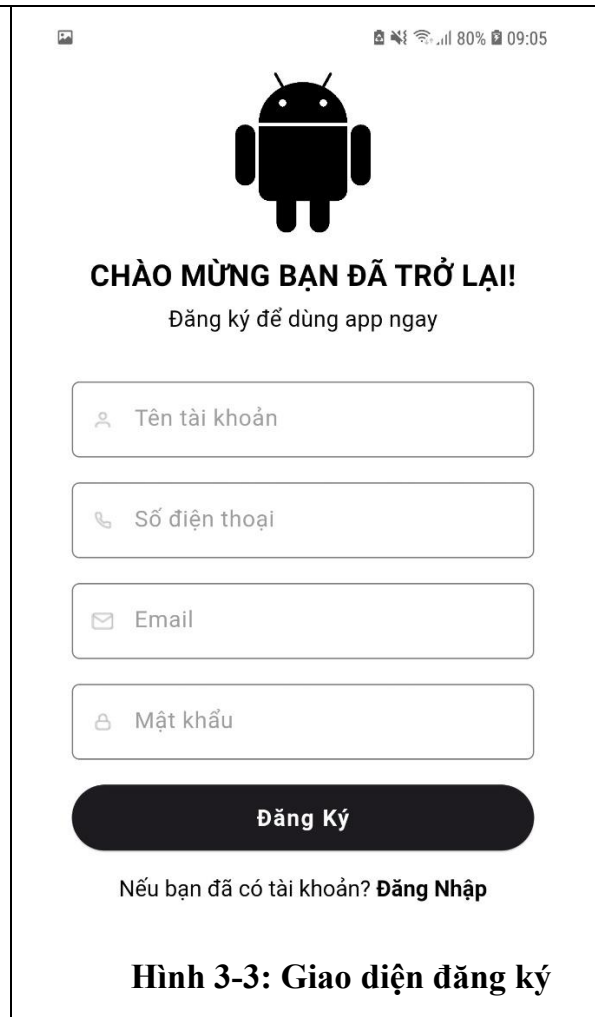
- Tạo các model cho dự án

- Tạo các view_model cho dự án

- Tạo các view cho dự án

Cuối cùng chạy lại dự án bằng lệnh: \$ flutter run để sử dụng ứng dụng.

3.2. Giao diện đăng ký và đăng nhập hệ thống

 <p>CHÀO MỪNG BẠN ĐÃ TRỞ LẠI! Đăng nhập để dùng app ngay</p> <p>Email</p> <p>Mật khẩu</p> <p>Quên mật khẩu?</p> <p>Đăng Nhập</p> <p>Nếu bạn chưa có tài khoản? Đăng Ký</p> <p>Hình 3-2: Giao diện đăng nhập</p>	 <p>CHÀO MỪNG BẠN ĐÃ TRỞ LẠI! Đăng ký để dùng app ngay</p> <p>Tên tài khoản</p> <p>Số điện thoại</p> <p>Email</p> <p>Mật khẩu</p> <p>Đăng Ký</p> <p>Nếu bạn đã có tài khoản? Đăng Nhập</p> <p>Hình 3-3: Giao diện đăng ký</p>
--	---

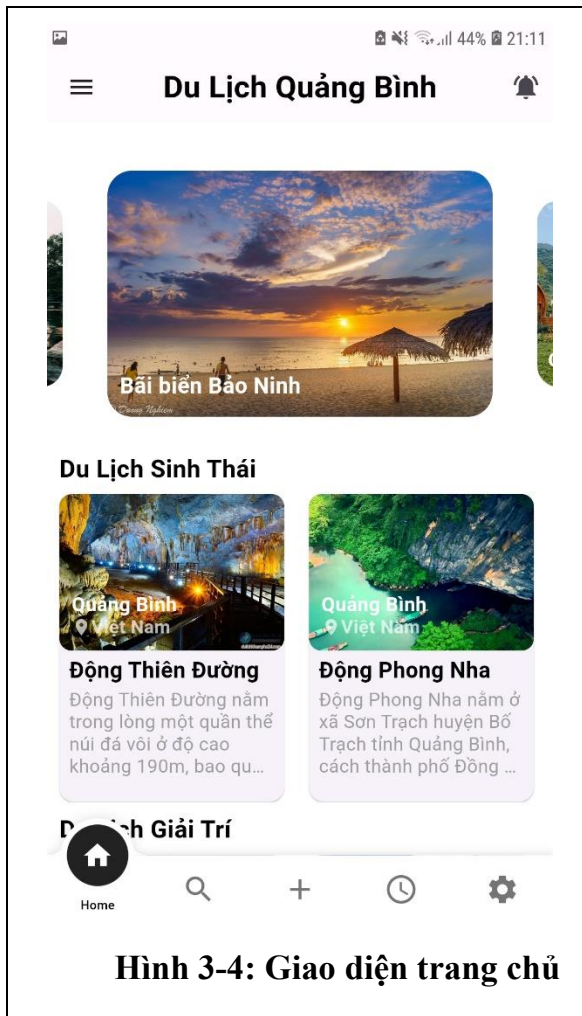
Giao diện đăng ký: cho phép người dùng đăng ký tài khoản để sử dụng ứng dụng, người dùng tạo tài khoản bằng cách nhập các thông tin tên tài khoản, số điện thoại, email và mật khẩu. Khi người dùng điền đầy đủ thông tin và bấm nút đăng ký thì hệ thống báo đăng ký thành công.

Khi đăng ký người dùng phải nhập đầy đủ thông tin không được để trống mục nào và email không thể trùng với email đã đăng ký trước đó. Nếu không hệ thống báo lỗi và bắt nhập lại.

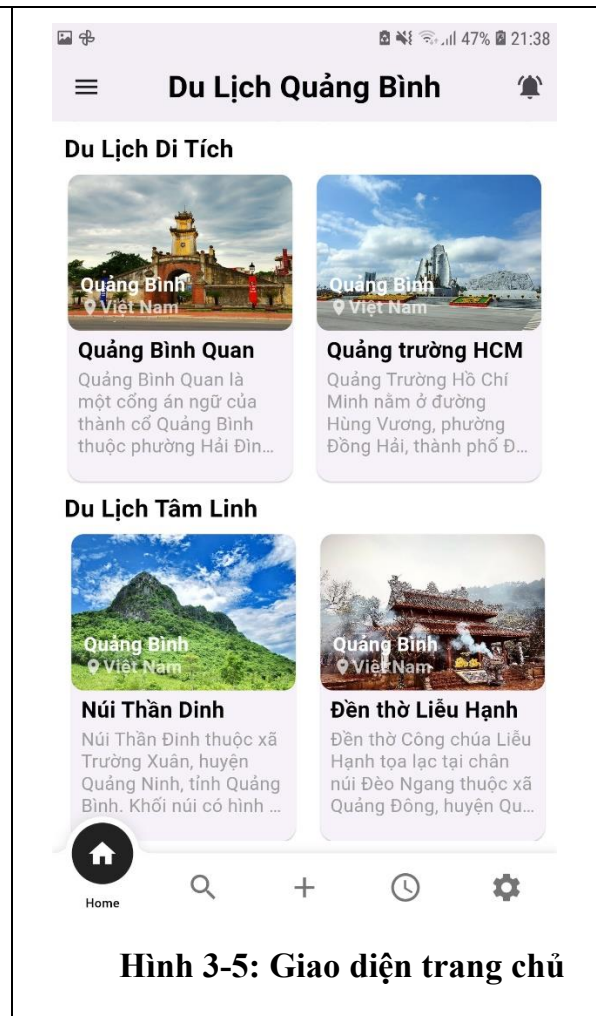
Giao diện đăng nhập: cho phép người dùng sử dụng email và mật khẩu đã đăng ký thành công để đăng nhập vào hệ thống. Khi người dùng nhập đúng thông tin email và mật khẩu đã đăng ký thì hệ thống nhảy vào giao diện trang chủ.

Nếu người dùng nhập sai thông tin email hoặc mật khẩu đã đăng ký thì hệ thống báo lỗi và bắt người dùng nhập lại.

3.3. Giao diện trang chủ hệ thống



Hình 3-4: Giao diện trang chủ

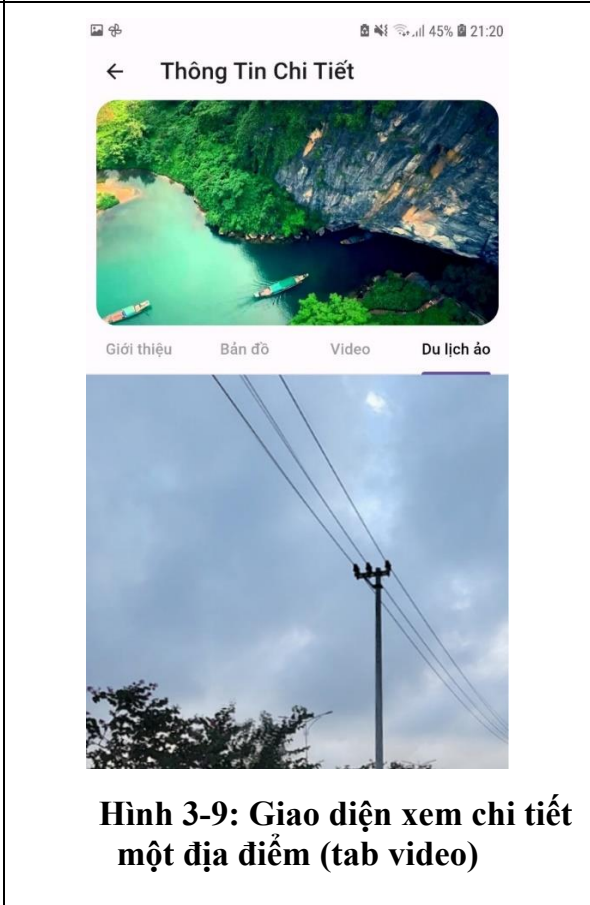
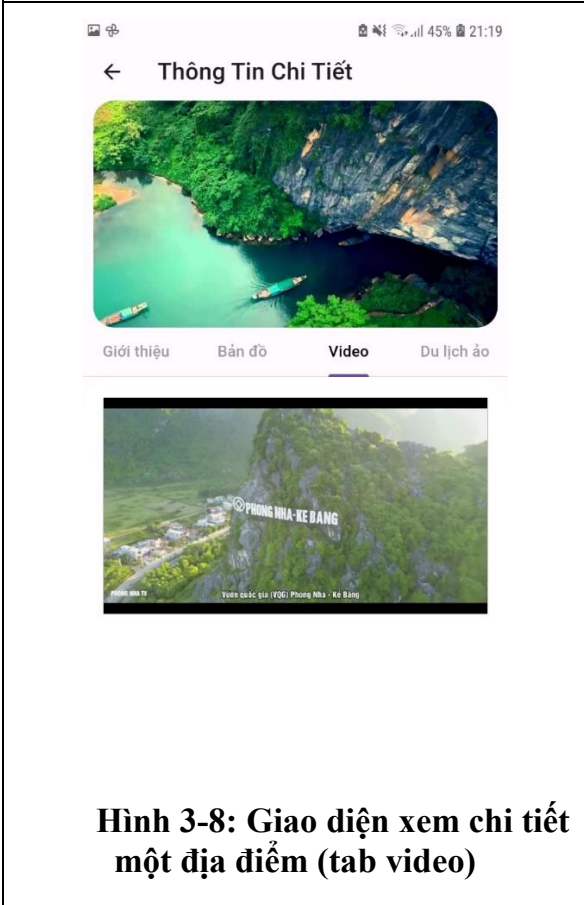
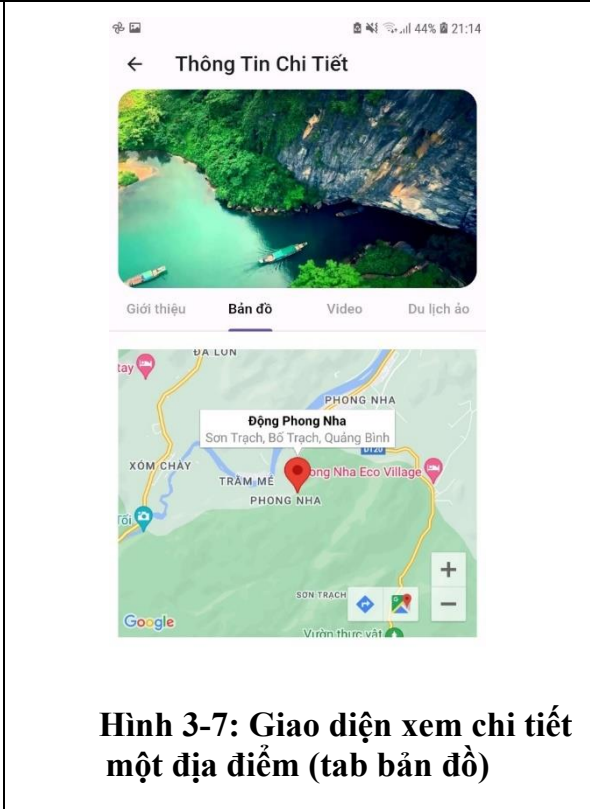
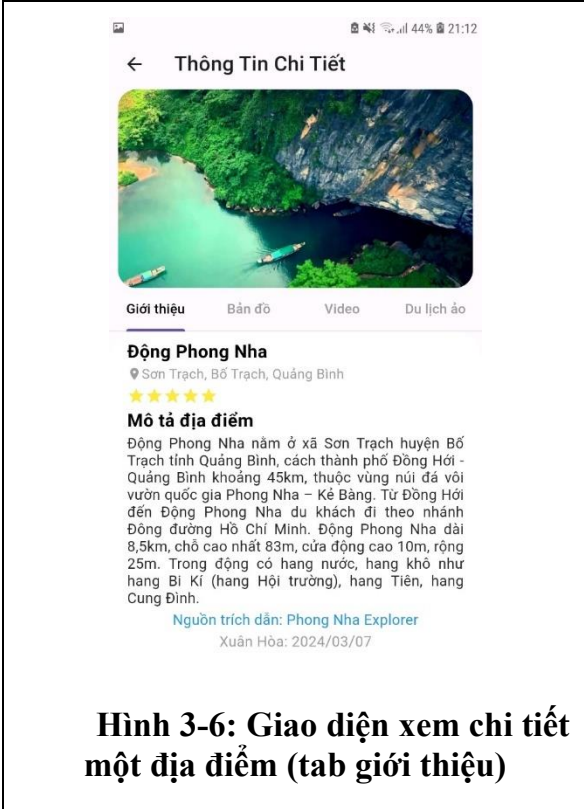


Hình 3-5: Giao diện trang chủ

Giao diện trang chủ cho phép người dùng xem được tổng quan các địa điểm du lịch quảng bình theo các danh mục: du lịch sinh thái, du lịch giải trí, du lịch di tích, và du lịch lịch sử.

Ở từng danh mục sẽ có các địa điểm cụ thể được thiết kế ngắn gọn bắt mắt để kích thích người dùng bấm vào xem.

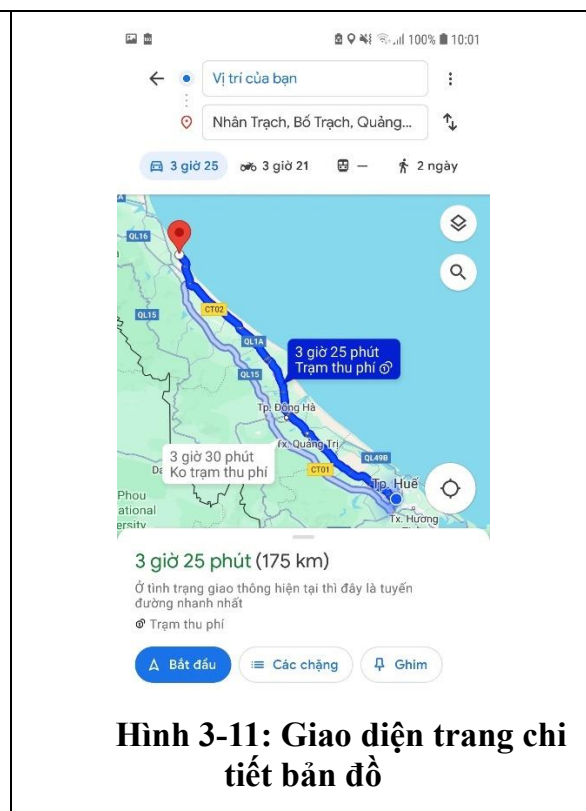
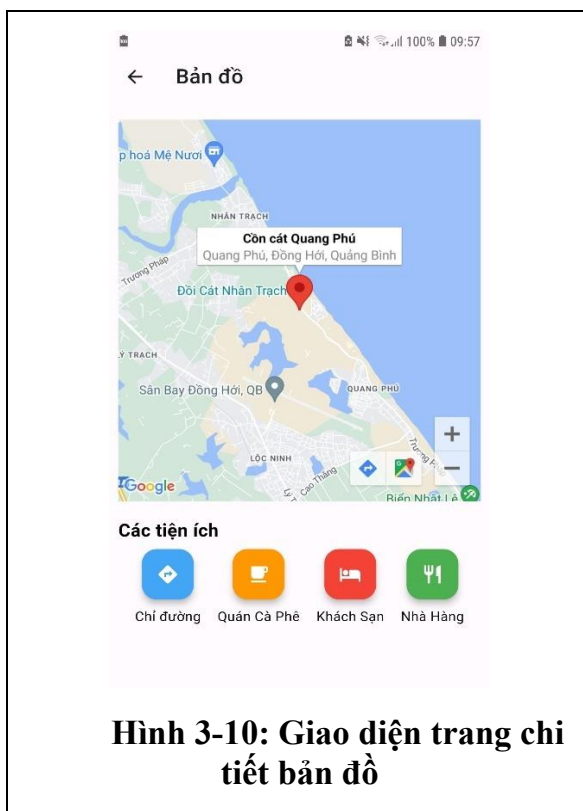
3.4. Giao diện xem địa điểm du lịch



Giao diện xem chi tiết một địa điểm cho phép người dùng xem được chi tiết về một địa điểm du lịch mà người dùng đã chọn xem ở trang chủ hoặc ở tìm kiếm địa điểm.

Giao diện này cho phép người dùng hiểu rõ một cách chi tiết của địa điểm du lịch đã chọn thông qua hình ảnh của địa điểm, các thông tin chi tiết như tên, địa điểm, mô tả về địa điểm, bên cạnh đó còn có bản đồ vị trí và khi người dùng bấm vào sẽ có chỉ đường từ vị trí bản thân đến vị trí địa điểm đó, có video giới thiệu và tour du lịch ảo của địa điểm

3.4.1. Giao diện xem chi tiết bản đồ và du lịch ảo

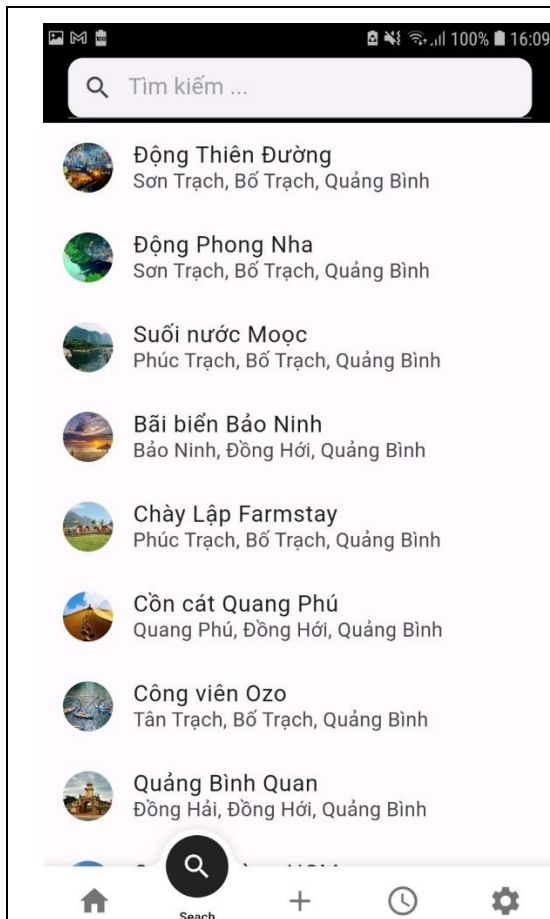


Giao diện trang chi tiết bản đồ này cho phép người dùng lựa chọn các tiện ích ở trang bản đồ chi tiết như chỉ đường, các quán cà phê, khách sạn, nhà hàng lân cận địa

điểm du lịch đang xem. Sau khi chọn chỉ đường thì sẽ hiển thị giao diện chỉ đường đi đến điểm du lịch đó.

Giao diện chi tiết trang du lịch ảo này cho phép người dùng xem được địa điểm du lịch một cách trực quan và thực tế. Giao diện có các nút di chuyển để giúp người dùng thuận tiện trong việc xem địa điểm dưới các góc nhìn khác nhau.

3.5. Giao diện tìm kiếm địa điểm du lịch



Hình 3-14: Giao diện trang tìm kiếm

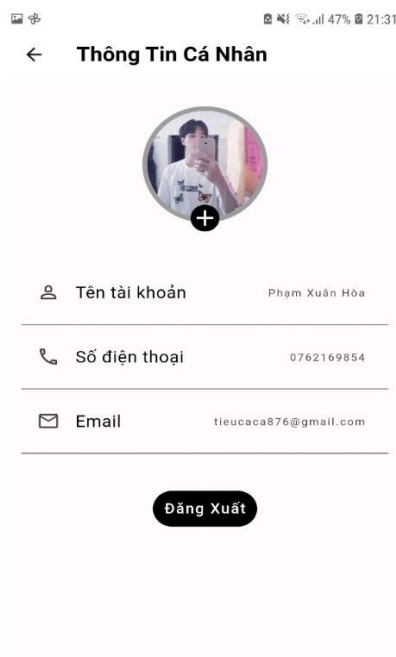


Hình 3-15: Giao diện đang tìm kiếm một địa điểm theo tên

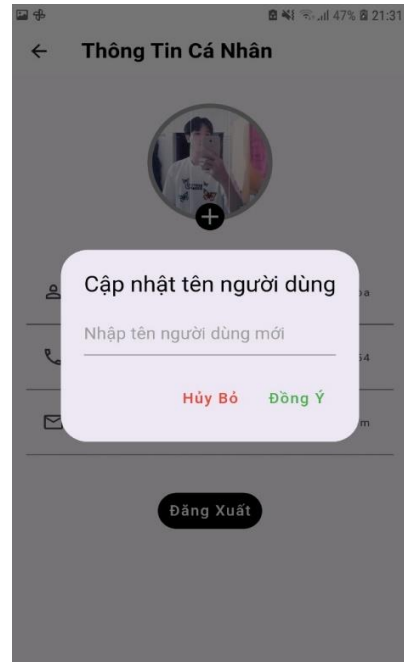
Giao diện tìm kiếm cho phép người dùng tìm kiếm các địa điểm đã có trong hệ thống. Người dùng chỉ cần nhập tên các địa điểm là hệ thống sẽ hiển thị ra các địa điểm mà người dùng đang tìm kiếm.

Khi người dùng tìm kiếm xong và chọn vào địa điểm đó thì sẽ chuyển sang trang thông tin chi tiết của địa điểm để cho người dùng xem và hiểu rõ địa điểm.

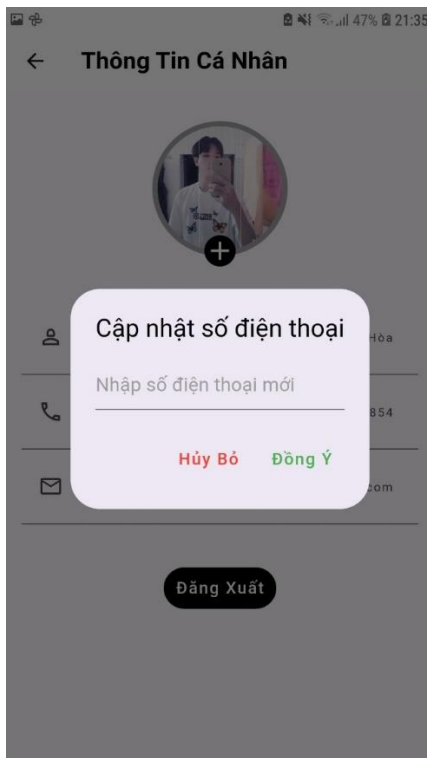
3.6. Giao diện quản lý thông tin cá nhân



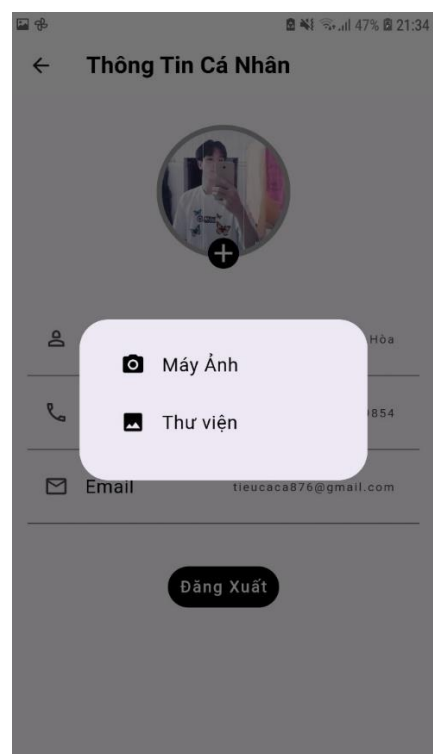
Hình 3-16: Giao diện xem trang thông tin cá nhân



Hình 3-17: Giao diện chỉnh sửa tên người dùng



Hình 3-18: Giao diện chỉnh sửa số điện thoại



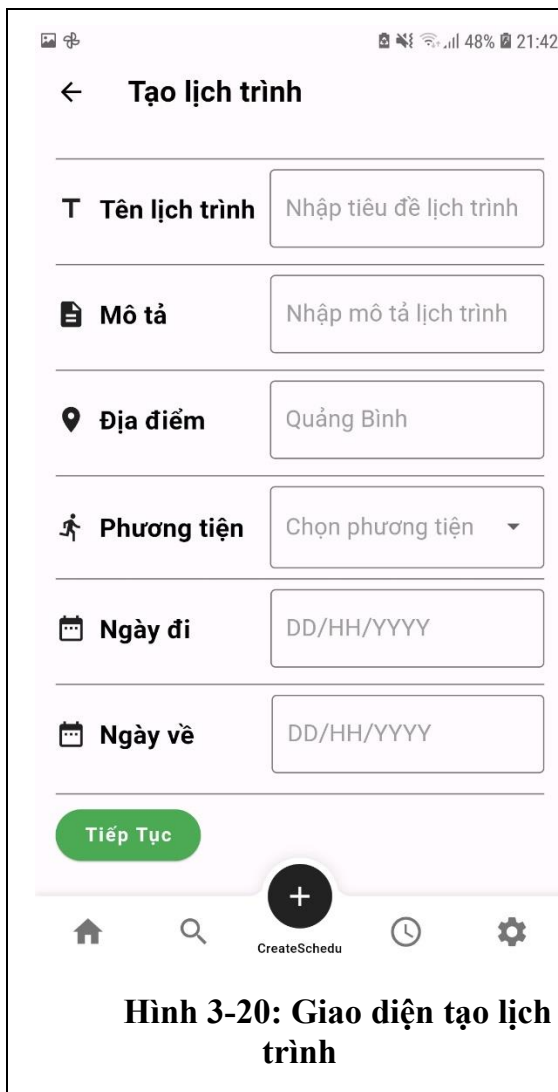
Hình 3-19: Giao diện chỉnh sửa ảnh đại diện

Giao diện này cho phép người dùng xem thông tin cá nhân của bản thân đã đăng ký(tên, số điện thoại, email, hình ảnh đại diện) và cho phép người dùng có thể chỉnh sửa các thông tin tên và số điện thoại nếu chưa hài lòng.

Bên cạnh đó người dùng có thể cập nhật ảnh đại diện bằng cách chọn ảnh có ở thư viện trong máy điện thoại hoặc chụp ảnh trực tiếp bằng camera để thay đổi ảnh đại diện theo ý thích của mình.

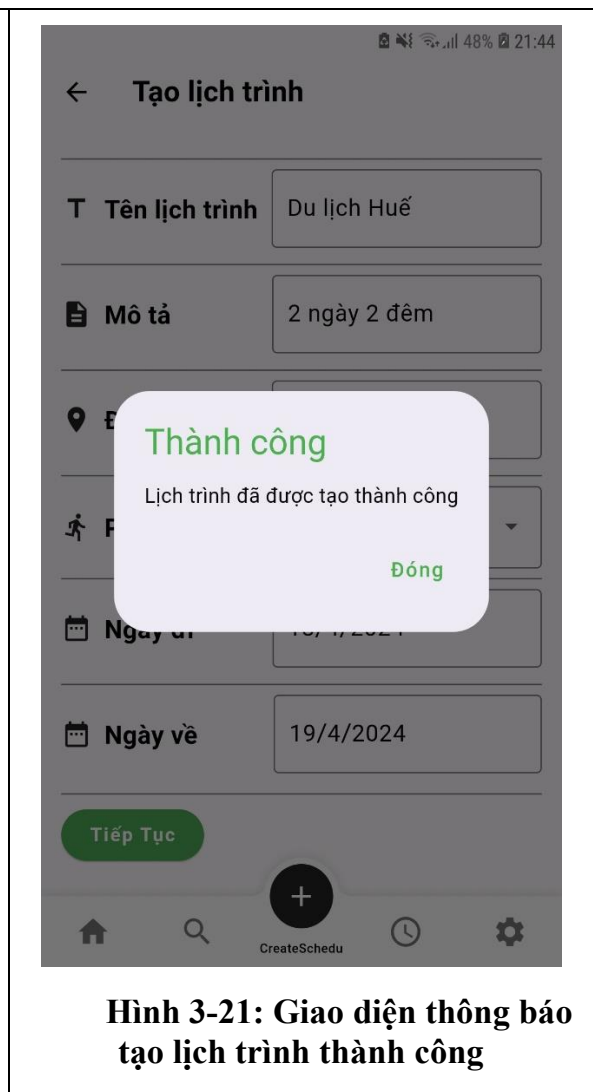
3.7. Giao diện quản lý lịch trình

3.7.1. Giao diện tạo lịch trình



The screenshot shows the 'Tạo lịch trình' (Create Schedule) screen. It features a back arrow at the top left. Below the title, there are several input fields: 'Tên lịch trình' (Schedule Name) with placeholder text 'Nhập tiêu đề lịch trình', 'Mô tả' (Description) with placeholder text 'Nhập mô tả lịch trình', 'Địa điểm' (Location) with the value 'Quảng Bình', 'Phương tiện' (Transportation) with a dropdown menu 'Chọn phương tiện', 'Ngày đi' (Departure Date) with placeholder text 'DD/MM/YYYY', and 'Ngày về' (Return Date) with placeholder text 'DD/MM/YYYY'. At the bottom, there is a green 'Tiếp Tục' (Continue) button and a bottom navigation bar with icons for home, search, a central '+' button labeled 'CreateSchedu', and settings.

Hình 3-20: Giao diện tạo lịch trình



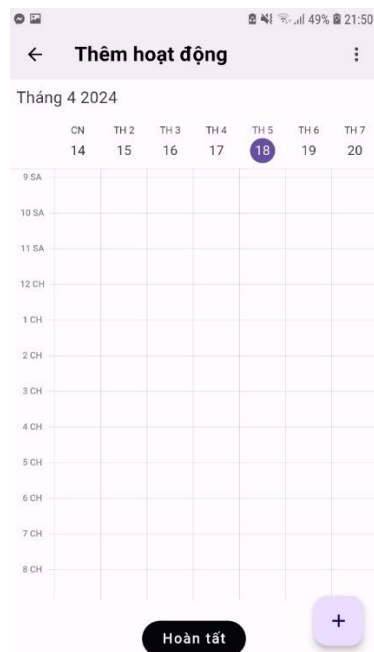
The screenshot shows the same 'Tạo lịch trình' screen as Figure 3-20, but with a success message overlay. The message box is white with green text: 'Thành công' (Success) and 'Lịch trình đã được tạo thành công' (Schedule has been created successfully). There is a green 'Đóng' (Close) button at the bottom right of the message box. The background is dimmed. The bottom navigation bar is also visible.

Hình 3-21: Giao diện thông báo tạo lịch trình thành công

Giao diện này cho phép người dùng tạo lịch trình bằng cách nhập các thông tin cần thiết để tạo lịch trình (tên lịch trình, mô tả, địa điểm, phương tiện, ngày đi, ngày về). Khi tạo lịch trình thành công thì sẽ có thông báo lịch trình tạo thành công.

Các thông tin cần được nhập và không được để trống, nếu để trống sẽ hiển thị thông báo cần nhập đầy đủ các thông tin và khi tạo lịch trình thành công thì sẽ có gửi thông báo về email.

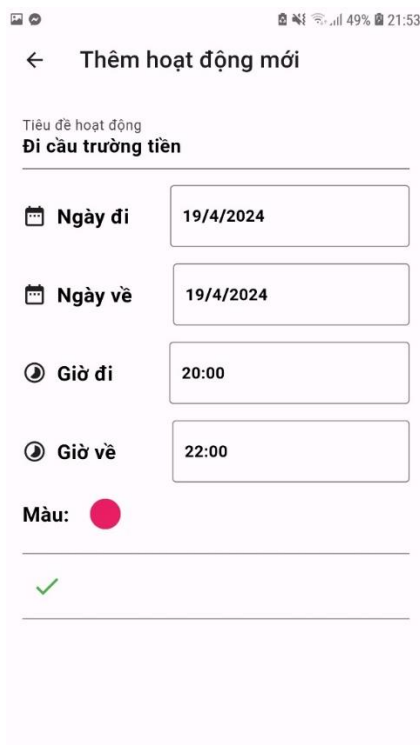
3.7.2. Giao diện thêm hoạt động cho lịch trình



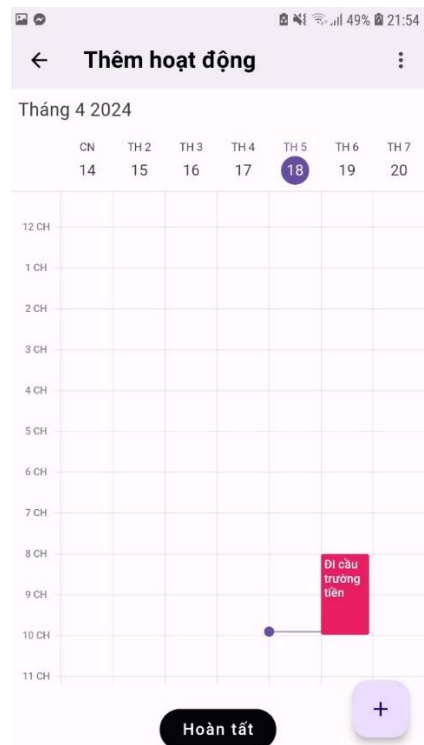
Hình 3-22: Giao diện thêm hoạt động cho lịch trình



Hình 3-23: Giao diện thêm hoạt động cho lịch trình



Hình 3-24: Giao diện thêm hoạt động cho lịch trình

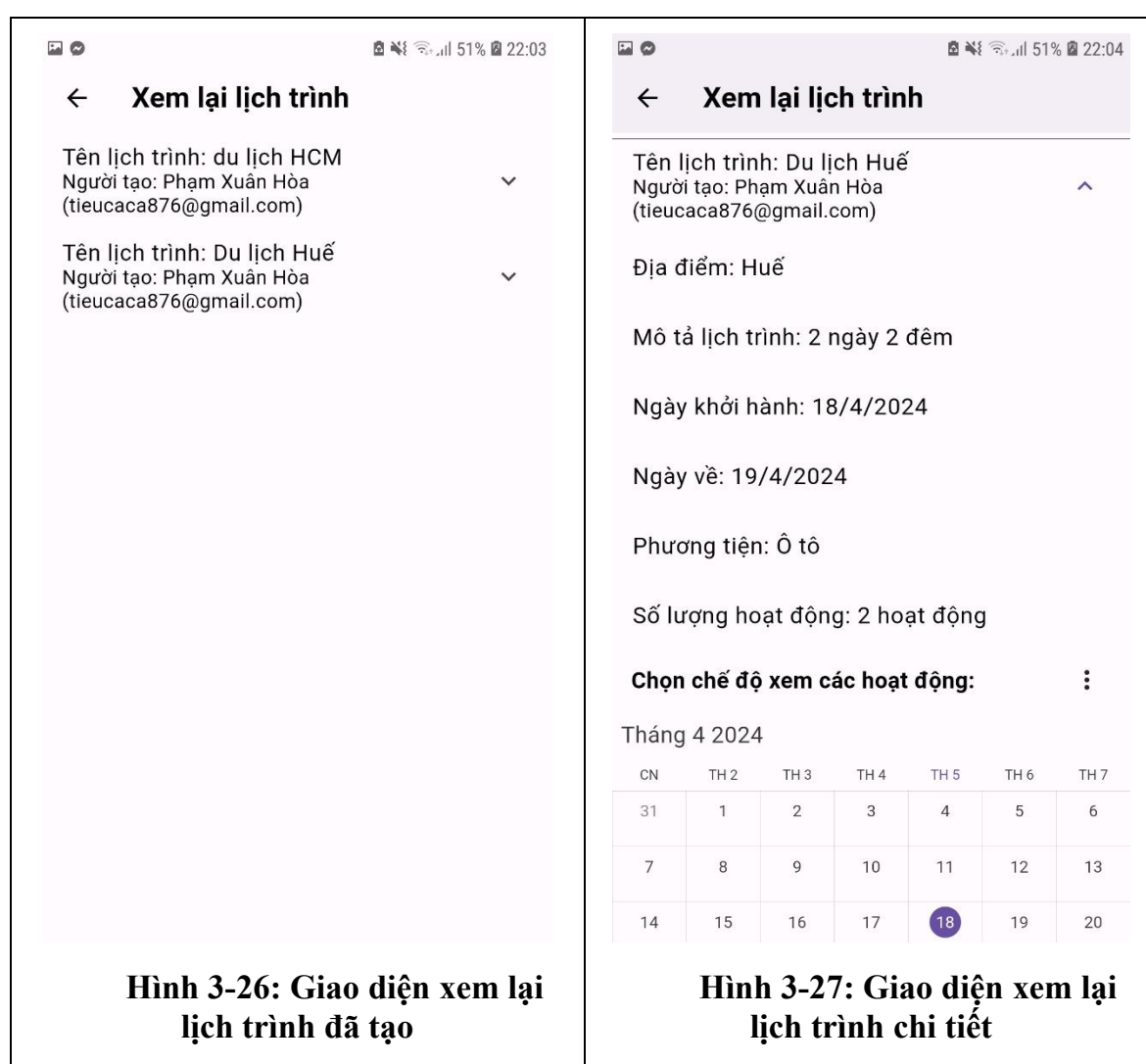


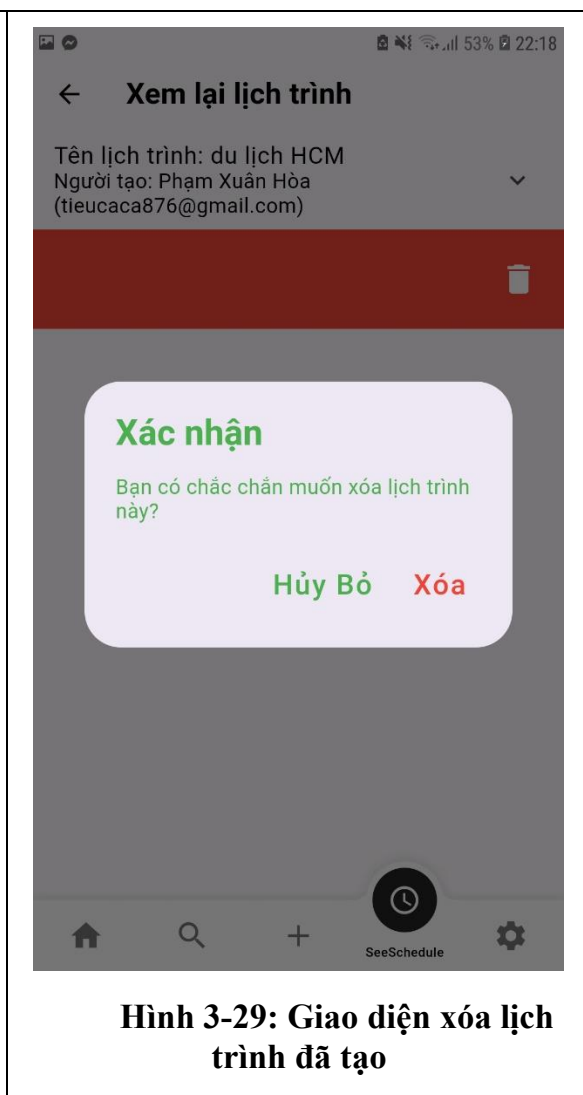
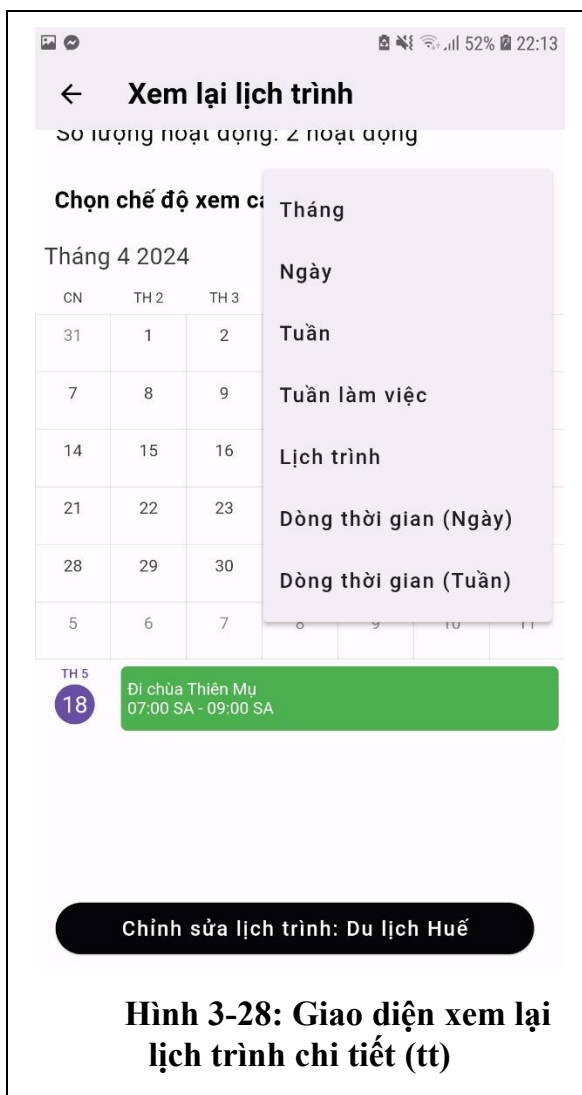
Hình 3-25: Giao diện thêm hoạt động cho lịch trình

Giao diện này cho phép người dùng thêm các hoạt động cho lịch trình đã tạo bằng cách nhấn vào nút dấu cộng góc phải dưới màn hình. Sau đó nhập các thông tin cần thiết để thêm các hoạt động cho lịch trình (tên hoạt động, ngày giờ đi và về, màu sắc của hoạt động). Sau khi nhập xong các thông tin thì bấm dấu tích để thêm hoạt động, hoạt động được thêm thành công thì sẽ hiển thị ở giao diện cho người dùng xem như hình 3-23.

Sau khi thêm các hoạt động cho lịch trình hoàn tất thì bấm nút hoàn tất để kết thúc việc thêm hoạt động cho lịch trình. Và sau khi chọn nút này thì màn hình sẽ chuyển đến trang xem lại các lịch trình đã tạo.

3.7.3. Giao diện xem lại lịch trình và xóa lịch trình đã tạo





Giao diện này cho phép người dùng xem lại các lịch trình đã tạo bao gồm các thông tin chi tiết của lịch trình và các hoạt động đã được tạo trong lịch trình. Người dùng cũng có thể chọn chế độ xem các hoạt động ở các chế độ ngày, tháng, tuần, tuần làm việc, lịch trình, dòng thời gian (ngày), dòng thời gian (tuần).

Người dùng cũng có thể bấm nút chỉnh sửa lịch trình để chỉnh sửa lại các thông tin theo ý muốn của mình. Người dùng cũng có thể xóa lịch trình bằng cách kéo lịch trình qua hết phía trái và màn hình sẽ hiển thị thông báo xác nhận người dùng có chắc chắn xóa lịch trình này không như hình 3-27.

3.7.4. Giao diện chỉnh sửa lịch trình đã tạo

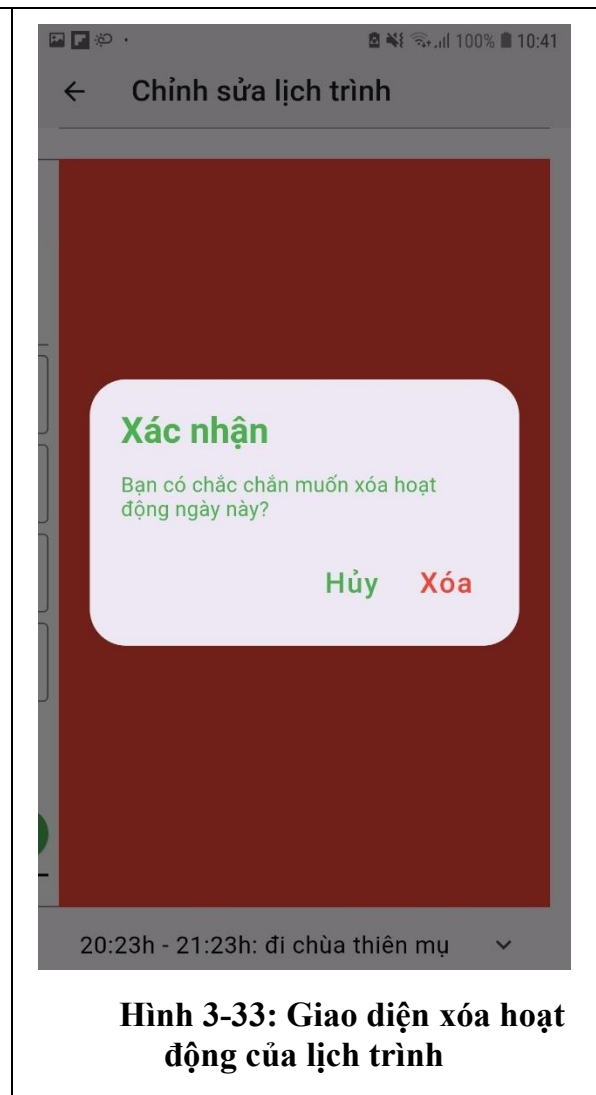
Hình 3-30: Giao diện chỉnh sửa lịch trình chi tiết

Hình 3-31: Giao diện chỉnh sửa lịch trình chi tiết (tt)

Giao diện này cho phép người dùng chỉnh sửa lại các thông tin đã tạo lịch trình trước đó. Sau khi chỉnh sửa lại các thông tin trong lịch trình theo ý muốn của người dùng thì bấm nút cập nhật lịch trình. Sau đó xác nhận cập nhật thì các thông tin trong lịch trình sẽ được cập nhật theo cái người dùng đã sửa.

Người dùng cũng có thể thêm tiếp các hoạt động trong lịch trình bằng cách bấm vào nút dấu cộng Hình 3-33 để thêm các hoạt động mới. Sau khi bấm vào nút dấu cộng thì sẽ hiển thị ra giao diện như Hình 3-25, và người dùng chỉ việc nhập các thông tin của hoạt động và bấm dấu tích để tạo hoạt động mới. Sau đó bấm cập nhật lịch trình để cập nhật.

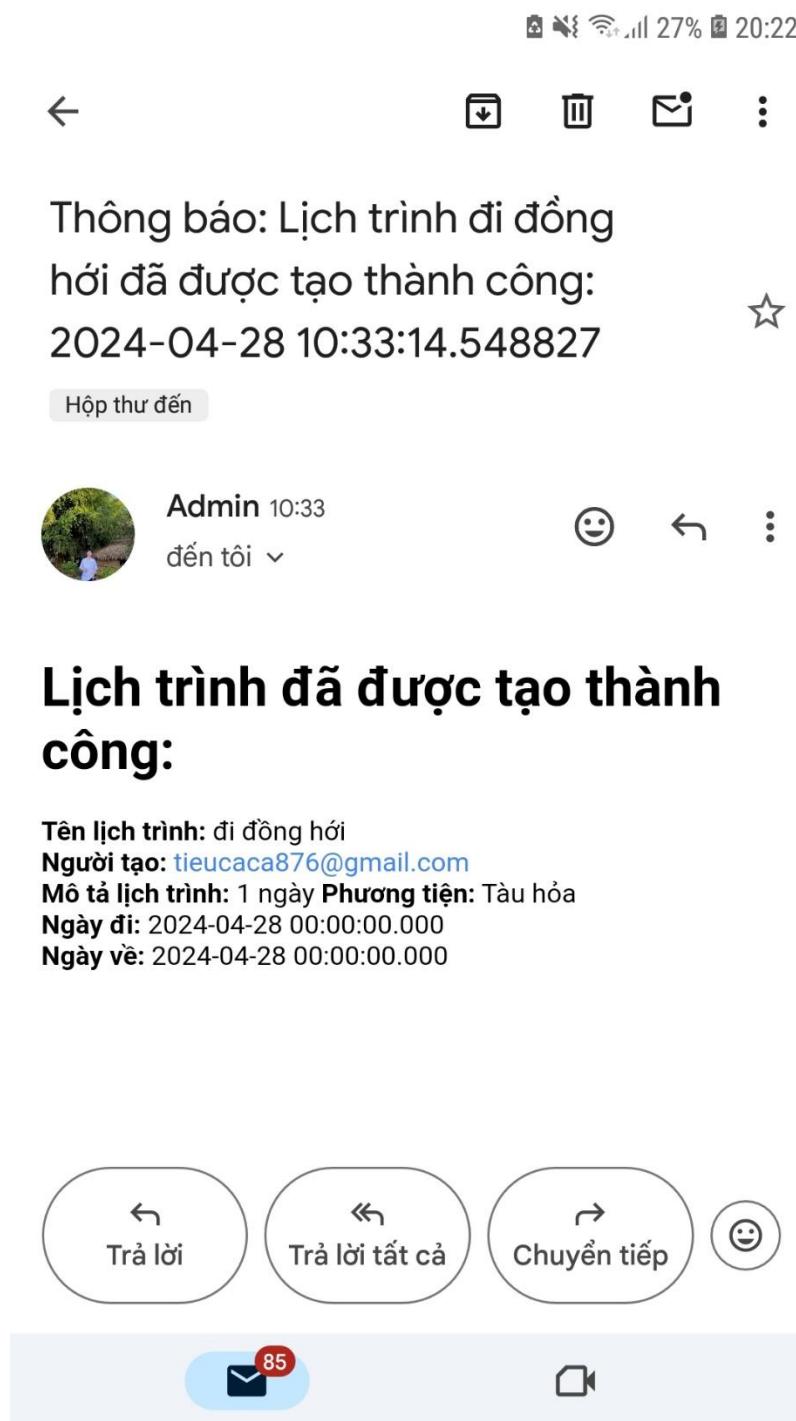
3.7.5. Giao diện chỉnh sửa hoạt động và xóa hoạt động của lịch trình



Giao diện chỉnh sửa hoạt động này cho phép người dùng chỉnh sửa lại các thông tin hoạt động trong lịch trình đã tạo. Sau khi người dùng chỉnh sửa lại các thông tin của hoạt động hoàn tất thì bấm nút cập nhật hoạt động để cập nhật.

Khi người dùng muốn xóa một hoạt động nào đó trong lịch trình thì người dùng kéo hoạt động đó sang bên trái, sau đó sẽ hiển thị thông báo có chắc chắn muốn xóa hoạt động này không. Người dùng chọn xóa để xóa hoạt động.

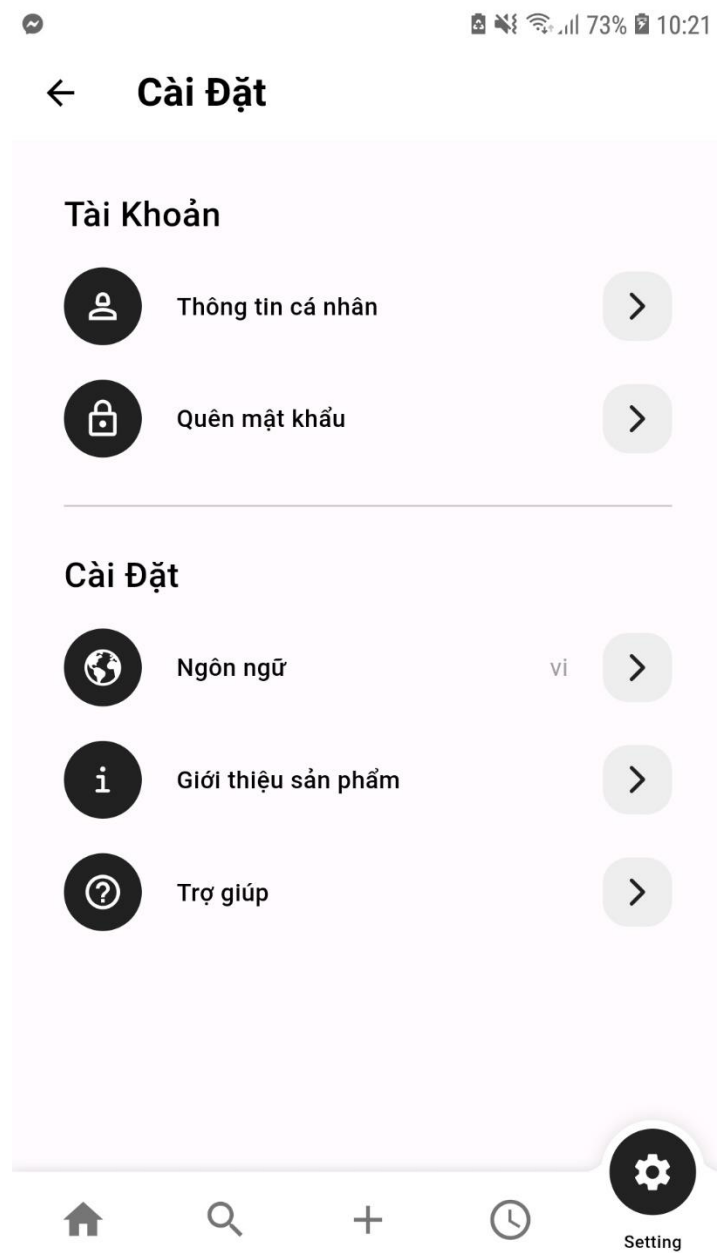
3.8. Giao diện gửi thông báo về email



Hình 3-34: Giao diện gửi thông báo về email

Khi người dùng tạo lịch trình thành công thì sẽ có thông báo về email cho người dùng biết để theo dõi.

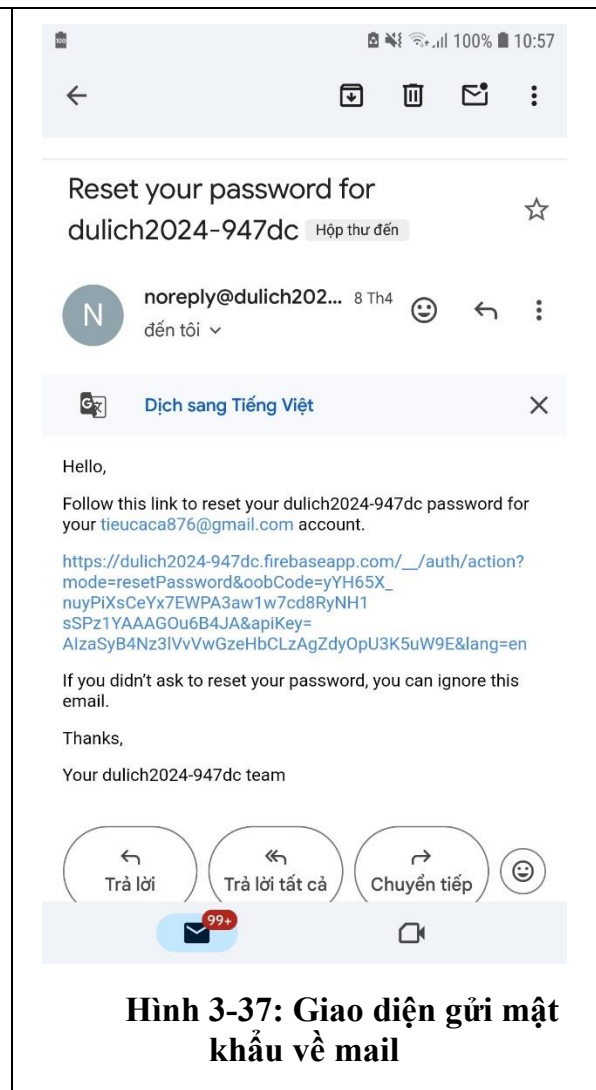
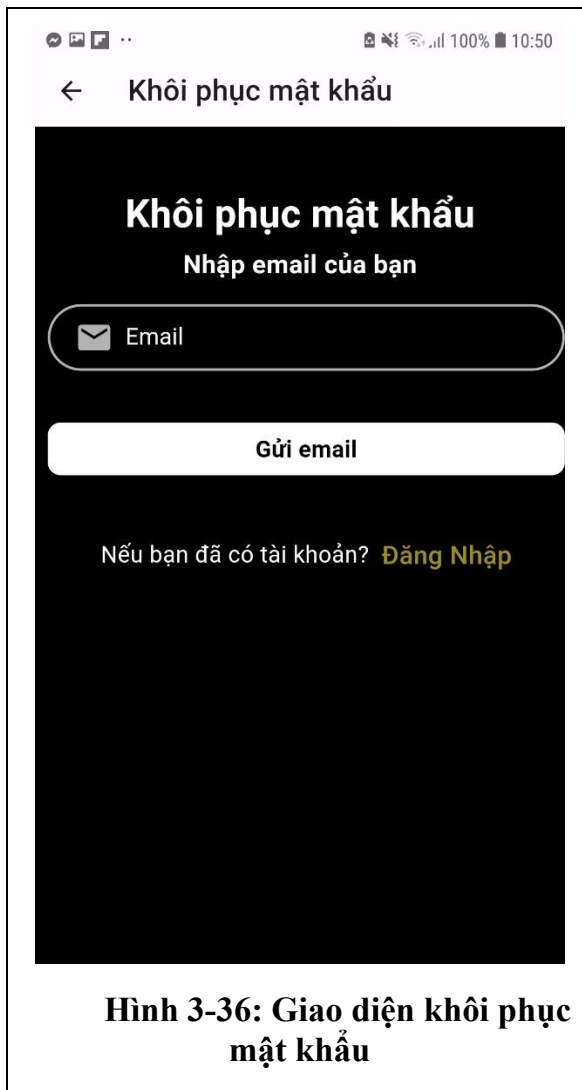
3.9. Giao diện cài đặt hệ thống



Hình 3-35: Giao diện trang cài đặt

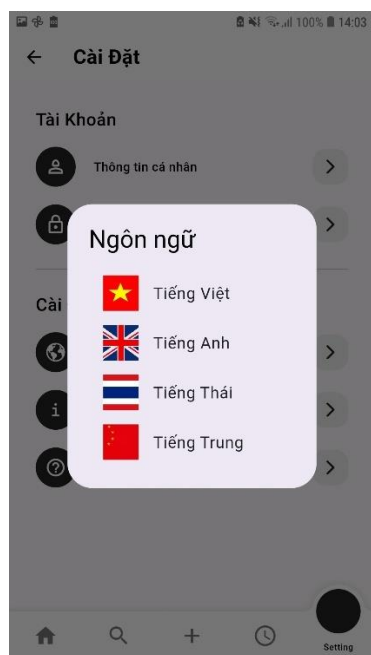
Giao diện này cho phép người dùng cài đặt các tính năng như: chỉnh sửa thông tin cá nhân, quên mật khẩu, lựa chọn ngôn ngữ, giới thiệu sản phẩm, và trợ giúp.

3.9.1. Giao diện tính năng quên mật khẩu

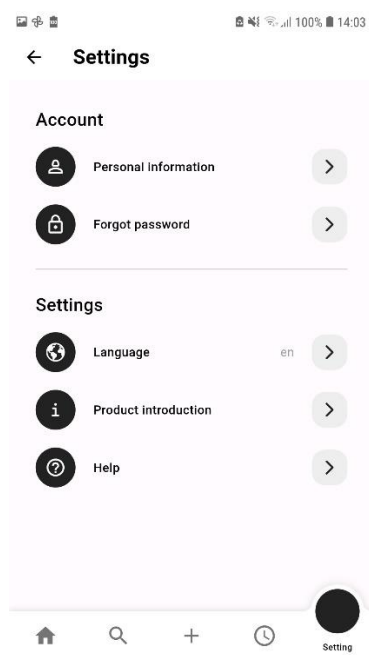


Giao diện này cho phép người dùng lấy lại mật khẩu khi người dùng quên mật khẩu của tài khoản. Người dùng chỉ cần nhập email sau đó sẽ có thư gửi về mail của người dùng. Sau đó người dùng chỉ cần nhập lại mật khẩu mới và nhấn chọn khôi phục mật khẩu.

3.9.2. Giao diện chọn ngôn ngữ của ứng dụng



Hình 3-38: Giao diện chọn ngôn ngữ ứng dụng



Hình 3-39: Giao diện trang cài đặt tiếng anh



Hình 3-40: Giao diện trang cài đặt tiếng thái



Hình 3-41: Giao diện trang cài đặt tiếng trung

Giao diện này cho phép người dùng chọn ngôn ngữ cho ứng dụng. Người dùng có thể lựa chọn các ngôn ngữ như trung quốc, tiếng anh, tiếng việt, tiếng thái.

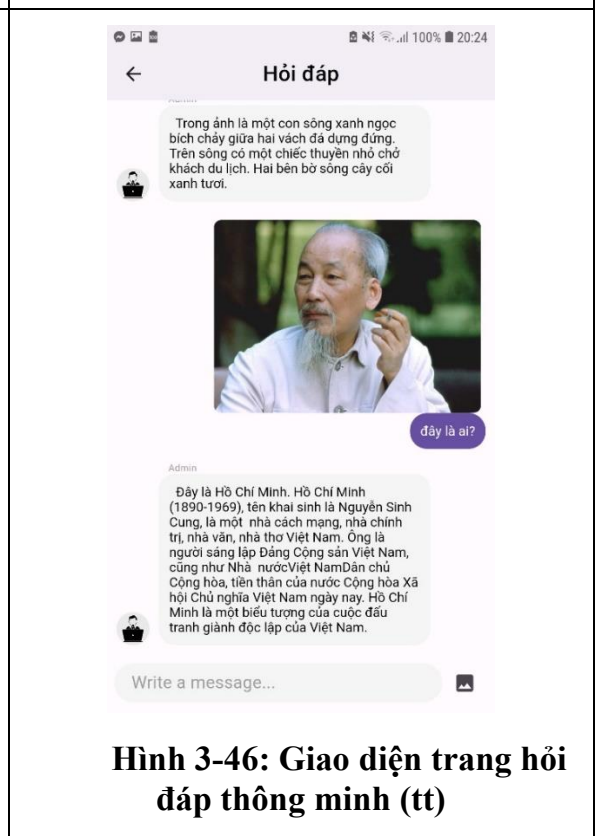
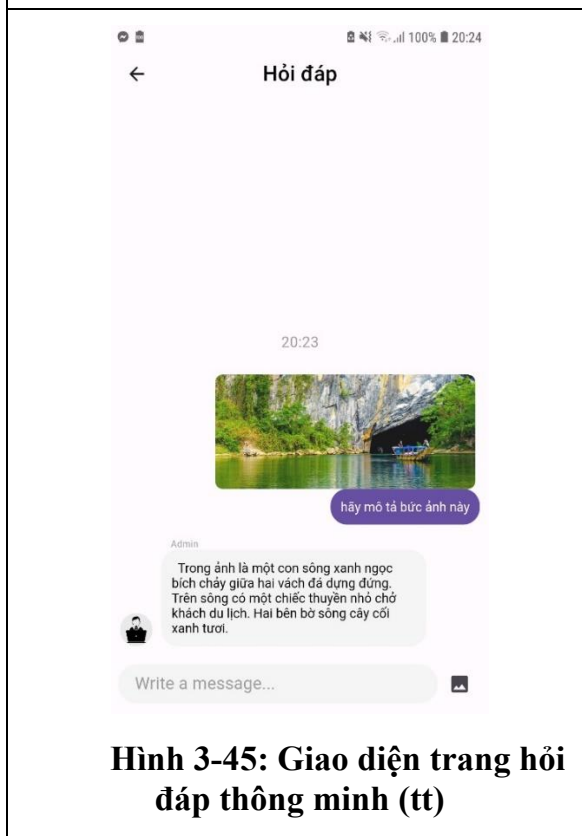
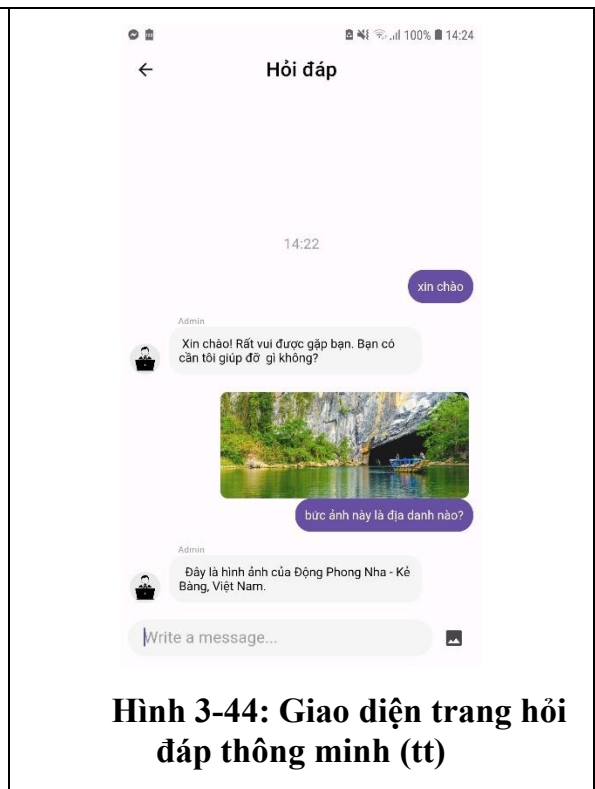
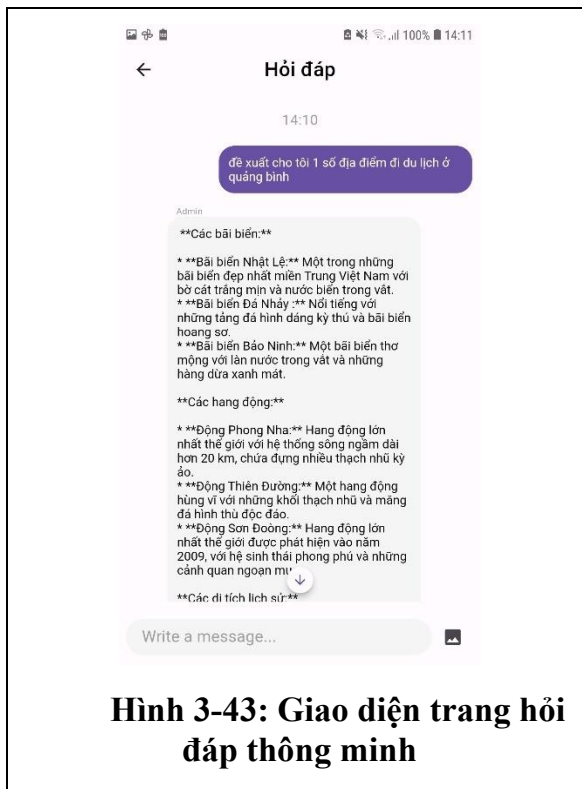
3.9.3. Giao diện tính năng giới thiệu sản phẩm



Hình 3-42: Giao diện trang giới thiệu sản phẩm

Giao diện này cho người dùng biết các thông tin cơ bản của ứng dụng: phiên bản ứng dụng và nhóm tác giả đã làm ra ứng dụng.

3.9.4. Giao diện tính năng trợ giúp



Giao diện này cho người dùng hỏi các câu hỏi để chatbot trả lời cho người dùng biết.

PHẦN 3: KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Qua thời gian được học tập và tìm hiểu các công nghệ mới, sau thời gian thực tập, em đã xây dựng thành công ứng dụng sổ tay du lịch quảng bình trên điện thoại di động với các chức năng giới thiệu chi tiết các địa điểm du lịch tại quảng bình, cho phép người dùng tìm kiếm các địa điểm và có bản đồ chỉ dẫn đến địa điểm. Tạo được lịch trình du lịch, quản lý thông tin các địa điểm, lịch trình và thông tin cá nhân người dùng. Hệ thống được xây dựng dựa trên môi trường phát triển Flutter và ngôn ngữ lập trình Dart. Hệ thống đạt được những kết quả nhất định:

- Thiết kế giao diện sử dụng Flutter đã được triển khai một cách chuyên nghiệp, đảm bảo tính thẩm mỹ và trải nghiệm người dùng tốt nhất.

- Các chức năng chính của ứng dụng như lập kế hoạch du lịch, tìm kiếm thông tin về địa điểm, và ghi chú cá nhân đã được phát triển một cách tỉ mỉ và linh hoạt.

- Sự chú trọng vào việc tối ưu mã nguồn đã giúp cải thiện hiệu suất và độ ổn định của ứng dụng.

- Việc sử dụng Flutter đã mang lại sự linh hoạt và khả năng mở rộng trong quá trình phát triển và bảo trì.

Bên cạnh đó em đã học hỏi thêm cách xây dựng một ứng dụng di động. Hiểu rõ cách vận hành và cách hoạt động nó. Biết thêm nhiều kiến thức mới về công nghệ và kiến trúc của Flutter áp dụng vào thực tiễn.

2. Kiến nghị

Với thời gian và kỹ năng có hạn, đồng thời chưa có nhiều thời gian để kiểm thử hệ thống và chưa rộng về lượng người dùng nên hệ thống còn thiếu sót rất nhiều tính năng nên hướng phát triển và nghiên cứu thêm là:

- Mở rộng tính năng để cung cấp thông tin đa dạng và phong phú hơn về các điểm du lịch và hoạt động tại Quảng Bình.

- Tăng cường bảo mật thông tin cá nhân của người dùng và đảm bảo tính an toàn trong giao dịch trực tuyến.

- Liên tục cập nhật và đổi mới nội dung và tính năng để đáp ứng nhu cầu của người dùng và xu hướng du lịch mới.

- Đưa ứng dụng lên các nền tảng phân phối ứng dụng như Google Play Store và App Store.

- Tiếp tục cải thiện nếu có giải pháp tốt hơn.

TÀI LIỆU THAM KHẢO

- [1]. Giới thiệu về Flutter, <https://viblo.asia/p/gioi-thieu-ve-flutter-bWrZnNxrZxw>, xem 12/12/2023.
- [2]. Flutter là gì? Các đặc tính vượt trội của Flutter, <https://viblo.asia/p/flutter-la-gi-cac-dac-tinh-vuot-troi-cua-flutter-2021-RQqKL60617z/>, xem 12/12/2023.
- [3]. Tự học Flutter | giới thiệu chi tiết về widgets và một số widget cần phải biết trong Flutter, <https://cafedev.vn/tu-hoc-flutter-gioi-thieu-chi-tiet-ve-widgets-va-mot-so-widgets-can-phai-biet-trong-flutter/>, xem 12/12/2023.
- [4]. Tự học Flutter | Kiến trúc Flutter, <https://cafedev.vn/tu-hoc-flutter-kien-truc-cua-flutter/>, xem 12/12/2023.
- [5]. Kiến trúc mô hình MVVM, <https://magz.techover.io/2022/04/06/flutter-kien-truc-mvvm/>, xem 12/12/2023.