

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM

KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM



BÁO CÁO MÔN HỌC

CÔNG NGHỆ LẬP TRÌNH ĐA NỀN TẢNG CHO

ỨNG DỤNG DI ĐỘNG

**ĐIỀU HƯỚNG MÀN HÌNH VÀ LẬP TRÌNH BẮT
ĐỒNG BỘ TRONG ỨNG DỤNG ĐẶT ĐỒ ĂN BẰNG
FLUTTER**

Giảng viên hướng dẫn : **ThS. Nguyễn Thanh Truyền**

Nhóm sinh viên thực hiện : **Nhóm 2**

Lớp : **08_DH_CNPM**

Khóa : **2019-2023**

TP. Hồ Chí Minh, tháng 08 năm 2023

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM

KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THĂM



BÁO CÁO MÔN HỌC

CÔNG NGHỆ LẬP TRÌNH ĐA NỀN TẢNG CHO

ỨNG DỤNG DI ĐỘNG

**ĐIỀU HƯỚNG MÀN HÌNH VÀ LẬP TRÌNH BẤT
ĐỒNG BỘ TRONG ỨNG DỤNG ĐẶT ĐỒ ĂN BẰNG
FLUTTER**

Sinh viên thực hiện:

Nguyễn Hoàng Nam - 0750080130

Võ Trương Phát Tài - 0850080044

Lê Chí Bảo - 0850080005

Lê Tuấn Khôi - 0850080026

Nguyễn Minh Châu - 0650080088

Giảng viên: **ThS. Nguyễn Thanh Truyền**

Lớp: **08_ĐH_CNPM**

TP. Hồ Chí Minh, tháng 08 năm 2023

MỞ ĐẦU

Ngày nay chính là kỷ nguyên của Công nghệ thông tin, công nghệ thông tin xuất hiện trong tất cả các ngành nghề, ngành công nghệ thông tin phát triển với tốc độ nhanh nhất và ngày càng thể hiện được vai trò to lớn, hết sức quan trọng đối với toàn bộ đời sống con người. Ta có thể nói rằng hầu hết tất cả các lĩnh vực trong đời sống hiện nay đều ứng dụng công nghệ thông tin. Ta chắc chắn một điều là công nghệ thông tin mang đến một hiệu quả khổng lồ cho cuộc sống. Công nghệ thông tin cung cấp cho ta những gì?

Công nghệ thông tin đã mang lại rất nhiều phần mềm để ta có thể thực hiện các công việc trong cuộc sống một cách dễ dàng hơn. Tuy nhiên cuộc sống thì luôn luôn thay đổi không ngừng nên công nghệ thông tin phải không ngừng phát triển để thỏa mãn và đáp ứng được sự thay đổi không ngừng của cuộc sống này. Việc chúng ta ứng dụng công nghệ thông tin vào tất cả các lĩnh vực được xem là một trong những yếu tố mang tính quyết định để tạo ra những bước phát triển mạnh mẽ.

Việc sử dụng điện thoại di động để thực hiện quản lý từ xa thông qua các ứng dụng ngày càng phổ biến khi mà người dùng không cần phải có máy tính, sổ sách hay phải luôn ở tại cửa hàng để quản lý mọi việc. Cho nên, nhóm em đã tiến hành tìm hiểu về các tài liệu liên quan và xây dựng sản phẩm liên quan đến nội dung tìm hiểu. Đề tài nhóm em thực hiện **“Điều Hướng Màn Hình và Lập Trình Bất Đồng Bộ trong Ứng Dụng Đặt Đồ Ăn bằng Flutter”**

LỜI CẢM ƠN

Trong suốt quá trình chuẩn bị và thực hiện đề tài tiểu luận đã cho em có thêm kiến thức và học hỏi bạn bè xung quanh.

Lời đầu tiên em xin cảm ơn thầy **Nguyễn Thanh Truyền** đã giúp đỡ nhóm em và hỗ trợ nhóm em các kiến thức và tài liệu chuyên môn cần thiết trong suốt quá trình thực hiện. Đây không chỉ là cơ sở để nhóm em hoàn thành bài tiểu luận này mà còn là hành trang tri thức vững chắc để em có thể bước tiếp vào môi trường học tập mới cũng như làm việc sau này.

Tuy nhiên kiến thức chuyên môn còn hạn chế và nhóm còn nhiều thiếu sót, em rất mong được sự góp ý của thầy để hoàn thiện kiến thức cần thiết cho nhóm, để em có thể dùng làm hành trang thực hiện tiếp các đề tài khác trong môi trường học tập mới, cũng như là việc làm sau này.

Em xin cảm ơn ạ!

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

....., ngày.....tháng.....năm.....

NGƯỜI NHẬN XÉT

(ký tên)

BẢNG PHÂN CÔNG CÔNG VIỆC

BẢNG PHÂN CÔNG CÔNG VIỆC VÀ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH

→ Tất cả công việc đều theo quyết định của nhóm trưởng **Nguyễn Hoàng Nam**, nếu có sai sót trong bài mong thầy thông cảm và em xin chịu trách nhiệm.

| SST | Họ tên | MSSV | Công việc |
|-----|--------------------|------------|---|
| 1 | Nguyễn Hoàng Nam | 0750080130 | <ul style="list-style-type: none">- Tìm hiểu Navigator, Router- Tìm hiểu về điều hướng màn hình- Fix lỗi ứng dụng- Tìm hiểu lý thuyết về lập trình bất đồng bộ |
| 2 | Võ Trương Phát Tài | 0850080044 | <ul style="list-style-type: none">- Tìm hiểu Navigator, Router- Lập trình bất đồng bộ với Await- Tìm hiểu về điều hướng màn hình |
| 3 | Lê Tuấn Khôi | 0850080026 | <ul style="list-style-type: none">- Tìm hiểu Navigator, Router- Tìm hiểu về điều hướng màn hình |
| 4 | Lê Chí Bảo | 0850080005 | <ul style="list-style-type: none">- Tìm hiểu Navigator, Routes- Fix lỗi ứng dụng- Lập trình bất đồng bộ với Async, Future- Tìm hiểu về điều hướng màn hình |
| 5 | Nguyễn Minh Châu | 0650080088 | <ul style="list-style-type: none">- Tìm hiểu Navigator, Routes- Tìm hiểu về điều hướng màn hình- Quy trình hoạt động hệ thống |

PHIẾU THEO DÕI TIẾN ĐỘ THỰC HIỆN BÁO CÁO ĐỒ ÁN NHÓM

MÔN: CÔNG NGHỆ LẬP TRÌNH ĐA NỀN TẢNG CHO ỨNG DỤNG DI ĐỘNG

1. Nhóm: 2 Thực hiện đề tài: **Điều Hướng Màn Hình và Lập Trình Bất Đồng Bộ trong Ứng Dụng Đặt Đồ Ăn bằng Flutter**

| Họ và tên | Ngày sinh: | MSSV: | Lớp: | Chuyên ngành | Mức độ đóng góp (%) | Ghi chú |
|-----------------------|------------|------------|-------------|---------------------|---------------------|---------|
| 1. Nguyễn Hoàng Nam | | 0750080130 | 08_DH_CNPM | Công nghệ phần mềm | | |
| 2. Võ Trương Phát Tài | | 0850080044 | 08_DH_CNPM | Công nghệ phần mềm | | |
| 3. Lê Chí Bảo | | 0850080005 | 08_DH_CNPM | Công nghệ phần mềm | | |
| 4. Lê Tuấn Khôi | | 0850080026 | 08_DH_CNPM | Công nghệ phần mềm | | |
| 5. Nguyễn Minh Châu | | 0650080088 | 06_DH_CNTT3 | Công nghệ thông tin | | |

2. Đề tài được giao: Ngày 19 tháng 07 năm 2023

4. Tên đề tài: **Điều Hướng Màn Hình và Lập Trình Bất Đồng Bộ trong Ứng Dụng Đặt Đồ Ăn bằng Flutter**

| STT | Ngày | Kế hoạch thực hiện | Nhận xét của GVHD (Ký tên) |
|---------------------------|-------------------|--|-------------------------------|
| 1 | 19.7.23 → 23.7.23 | <p>Tìm hiểu về đề tài</p> <p>Tìm hiểu về các công cụ liên quan</p> <p>Cài đặt phần mềm kỹ thuật</p> | |
| Kiểm tra ngày: 24/07/2023 | | <p>.....</p> <p>.....</p> <p>Được tiếp tục: <input type="checkbox"/> Không tiếp tục: <input type="checkbox"/></p> | |
| 4 | 24.7.23 → 30.7.23 | <p>Tìm hiểu về Navigator, Router</p> <p>Tìm hiểu về di chuyển dữ liệu giữa các màn hình</p> <p>Tìm hiểu về lập trình bất đồng bộ</p> | |
| 5 | 31.7.23 → 6.8.23 | <p>Tìm hiểu về Navigator, Router</p> <p>Tìm hiểu về di chuyển dữ liệu giữa các màn hình</p> <p>Tìm hiểu về lập trình bất đồng bộ</p> <p>Tiến hành viết báo cáo</p> | |

| STT | Ngày | Kế hoạch thực hiện | Nhận xét của GVHD (Ký tên) |
|---------------------------|------|---|-------------------------------|
| | | Xây dựng App (Di chuyển dữ liệu giữa các màn hình) Xây dựng App (Minh họa lập trình bất đồng bộ) | |
| Kiểm tra ngày: 06/08/2023 | | Hoàn thành báo cáo: Hoàn thành tiến độ | |

Ghi chú: Nhóm trưởng đánh giá chính xác mức độ đóng góp của từng thành viên trong nhóm đối với đề tài.

TP.HCM ngày 07 tháng 08 năm 2023

Nhóm Sinh viên thực hiện

(Ký và ghi rõ họ tên)

MỤC LỤC

| | |
|---|-----------|
| CHƯƠNG I. TỔNG QUAN VỀ ĐỀ TÀI | 1 |
| 1.1. Tổng quan | 1 |
| 1.2. Mục tiêu nghiên cứu..... | 1 |
| 1.3. Đối tượng nghiên cứu..... | 2 |
| 1.4. Phương pháp nghiên cứu..... | 2 |
| 1.5. Giới hạn và phạm vi nghiên cứu..... | 2 |
| 1.6. Mục đích nghiên cứu..... | 2 |
| 1.7. Yêu cầu kỹ thuật..... | 3 |
| 1.8. Kết cấu báo cáo..... | 3 |
| CHƯƠNG II. CƠ SỞ LÝ THUYẾT | 4 |
| 2.1. Tổng quan về lý thuyết | 4 |
| 2.1.1. Giới thiệu các hệ điều hành trong nền tảng di động..... | 4 |
| 2.1.2. Tổng quan về Navigator Flutter – Routes Fluuter | 7 |
| 2.1.3. Ví dụ về điều hướng giữa hai tuyến đường | 8 |
| 2.2. Tổng quan về kỹ thuật | 13 |
| 2.2.1. Ngôn ngữ Dart..... | 13 |
| 2.2.1. Flutter | 14 |
| 2.2.1. Android Studio | 16 |
| 2.2.2. Genymotion | 17 |
| 2.2.3. Visual Studio Code | 18 |
| 2.2.4. Android SDK..... | 19 |
| 2.2.5. Môi trường lập trình trong Android Studio..... | 19 |
| 2.3. Tổng quan về lập trình bất đồng bộ | 20 |
| 2.3.1. Future..... | 20 |
| 2.3.2. Async / Await..... | 21 |

| | |
|---|----|
| CHƯƠNG III. CÀI ĐẶT THỰC NGHIỆM | 22 |
| 3.1. Cài đặt phần mềm kỹ thuật | 22 |
| 3.1.1. Cài đặt Android Studio | 22 |
| 3.1.2. Cài đặt SDK cho Android | 24 |
| 3.1.3. Cài đặt Flutter | 28 |
| 3.2. Giao diện Project | 30 |
| 3.2.1. Giao diện chính | 30 |
| 3.2.2. Giao diện Home (Sản phẩm) | 31 |
| 3.2.3. Giao diện giỏ hàng (Hóa đơn) | 32 |
| 3.2.4. Giao diện sản phẩm yêu thích | 33 |
| 3.3. Điều hướng màn hình | 34 |
| 3.3.1. Home → My Cart | 34 |
| 3.3.2. My Cart → Home | 36 |
| 3.4. Lập trình bất đồng bộ | 37 |
| CHƯƠNG IV. KẾT LUẬN VÀ KIẾN NGHỊ | 41 |
| 4.1. Kết quả đạt được | 41 |
| 4.2. Ưu điểm | 41 |
| 4.3. Nhược điểm | 41 |
| 4.4. Hướng phát triển | 41 |
| 4.5. Kiến nghị | 41 |
| TÀI LIỆU THAM KHẢO | 42 |

DANH MỤC HÌNH

| | |
|--|----|
| Hình 2. 1. Hệ điều hành Android | 4 |
| Hình 2. 2. Hệ Điều Hành IOS..... | 6 |
| Hình 2. 3. Giao diện điện thoại khi dự án | 12 |
| Hình 2. 4. Hình ảnh về Dart | 13 |
| Hình 2. 5. Logo Flutter | 14 |
| Hình 2. 6. Android Studio | 16 |
| Hình 2. 7. Genymotion | 17 |
| Hình 2. 8. Hình ảnh Visual Studio Code | 18 |
| Hình 2. 9. Môi trường lập trình trong Android | 20 |
| Hình 3. 1. Giao diện chính của hệ thống | 31 |
| Hình 3. 2. Giao diện trang Home "Food - Fruit" | 32 |
| Hình 3. 3. Giao diện trang Home "Gracery - Vegetable" | 32 |
| Hình 3. 4. Giao diện giỏ hàng (xem - sửa - xóa) hàng hóa | 33 |
| Hình 3. 5. Giao diện yêu thích (xem - sửa - xóa) hàng hóa..... | 34 |

CHƯƠNG I. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Tổng quan

Trong quá trình phát triển của xã hội công nghệ thông tin đã và đang phát triển mạnh mẽ ngày càng thâm nhập sâu, trở thành một công cụ hỗ trợ đắc lực cho con người trong mọi lĩnh vực. Đặc biệt đó là ứng dụng phần mềm với nhiều chương trình ứng dụng có hiệu quả. Công nghệ thông tin đã mang lại rất nhiều phần mềm để ta có thể thực hiện các công việc trong cuộc sống một cách dễ dàng hơn. Tuy nhiên cuộc sống thì luôn luôn thay đổi không ngừng nên công nghệ thông tin phải không ngừng phát triển để thỏa mãn và đáp ứng được sự thay đổi không ngừng của cuộc sống này. Việc chúng ta ứng dụng công nghệ thông tin vào tất cả các lĩnh vực được xem là một trong những yếu tố mang tính quyết định để tạo ra những bước phát triển mạnh mẽ.

Thay vì chúng ta phải ngồi trên máy vi tính để thực hiện công việc quản lý cho một cửa hàng tại sao ta không thiết kế một ứng dụng (Application, viết tắt là App) chạy trên điện thoại thông minh (Smartphone) để tiện cho người quản lý khi họ có thể thực hiện công việc của mình khi ở bất cứ đâu không còn phải phụ thuộc vào máy tính nữa.

Vì các lý do cũng như sự thuận lợi của các ứng dụng di động. Nhóm em đã sử dụng ngôn ngữ lập trình Android đã học và tìm hiểu về công cụ Android Studio để thực hiện đề tài “Điều Hướng Màn Hình và Lập Trình Bất Đồng Bộ trong Ứng Dụng Đặt Đồ Ăn bằng Flutter”

1.2. Mục tiêu nghiên cứu

Project được xây dựng cho khách hàng là những người đam mê hưởng thức các món ăn mà không có thời gian đi ra các quán ăn hay các cửa hàng. Vì vậy cửa hàng đã áp dụng phát triển ứng dụng đặt đồ ăn để giúp cho các khách hàng ở xa có thể đặt các món ăn mình thích và đặt giao đến tận nơi. Ứng dụng sẽ hỗ trợ người dùng trong các công việc quản lý các đơn hàng, món ăn, danh thu và quản lý về nhân viên cửa hàng... Giúp cho khách hàng dễ dàng tiếp cận với cửa hàng, đồng thời làm tăng doanh thu người dùng.

1.3. Đối tượng nghiên cứu

- Nghiên cứu về ngôn ngữ lập trình Android.
- Công cụ Android Studio.
- Quy trình đặt đồ ăn

1.4. Phương pháp nghiên cứu

- Sử dụng tài liệu hướng dẫn về lập trình android
- Tham khảo một số đồ án về lập trình android
- Sử dụng các video hướng dẫn có liên quan từ Youtube

1.5. Giới hạn và phạm vi nghiên cứu

- 🚦 Đề tài "Điều Hướng Màn Hình và Lập Trình Bất Đồng Bộ trong Ứng Dụng Đặt Đồ Ăn bằng Flutter" tập trung vào việc nghiên cứu và phân tích tài liệu liên quan, tập trung đặc biệt vào mô hình điều hướng (Navigation), ứng dụng các khái niệm liên quan đến Navigator, Routes, cũng như việc điều chỉnh giao diện màn hình và triển khai lập trình bất đồng bộ sử dụng các kỹ thuật Future, Async, và Await.
- 🚦 Phạm vi nghiên cứu sẽ xoay quanh các doanh nghiệp vừa và nhỏ, nhằm đảm bảo rằng các tính năng và chức năng của ứng dụng được thực hiện một cách hiệu quả và linh hoạt, đáp ứng tốt nhu cầu của người dùng và mang lại trải nghiệm tốt nhất.
- 🚦 Qua đó, báo cáo sẽ trình bày kết quả của quá trình nghiên cứu, phân tích và thử nghiệm các phương pháp và kỹ thuật được áp dụng trong việc phát triển ứng dụng đặt đồ ăn bằng Flutter.

1.6. Mục đích nghiên cứu

Ứng dụng được thiết kế nhằm mục đích thực hiện đầy đủ các yêu cầu điều hướng màn hình và lập trình bất đồng bộ và giúp cho người dùng có thể quản lý cửa hàng của mình (nhân viên, hóa đơn, món ăn) và gia tăng các khách hàng ở xa có thể biết đến các món ăn của cửa hàng.

1.7. Yêu cầu kỹ thuật

| Cấu hình | Tối thiểu | Đề nghị |
|----------|---------------------------|-----------------------------------|
| CPU | Bộ xử lí 2GHz | Bộ xử lí 4GHz Dual Core hoặc 3GHz |
| RAM | 4 GB RAM | 8 GB RAM trở lên |
| HDD | 1 GB ổ trống | 2 GB ổ trống |
| VGA | DirectX 9.0c hoặc cao hơn | DirectX 9.0c hoặc cao hơn. |
| Flutter | 3.0.0 | 3.0.6 hoặc cao hơn |
| SDK | 3.0.6 | 3.0.6 hoặc cao hơn |
| DevTool | 2.23.1 | |

1.8. Kết cấu báo cáo

Cấu trúc bài báo cáo bao gồm (4 chương)

- Chương I. Tổng quan về đề tài
- Chương II. Cở sở lý thuyết (Tổng quan về lý thuyết – Kỹ thuật – Lập trình bất đồng bộ)
- Chương III. Cài đặt thực nghiệm
- Chương IV. Kết luận và kiến nghị

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về lý thuyết

2.1.1. Giới thiệu các hệ điều hành trong nền tảng di động

2.1.1.1. Hệ điều hành Android

Android là một nền tảng Platform được những lập trình viên viết, phát hành ở dạng mã nguồn mở. Vì là mã nguồn mở, cùng với việc có giấy phép hoạt động không bị ràng buộc nên những nhà phát triển thiết bị, các nhà mạng di động cùng lập trình viên đã thực hiện điều chỉnh hệ điều hành. Nhờ vào đó mà hiện tại, Android đang được tối ưu và càng ngày càng được bổ sung theo đó nhiều tính năng độc đáo.



Hình 2. 1. Hệ điều hành Android

Những yếu tố trên là cơ sở giúp cho Android đã và đang trở thành một nền tảng của điện thoại thông minh phát triển – phổ biến nhất trên thế giới, được nhiều công ty công nghệ trên thế giới lựa chọn sử dụng nếu cần đến hệ điều hành không quá nặng nề và khả năng tinh chỉnh, giá thành rẻ mà chạy tốt trên thiết bị công nghệ cao.

Dù khi mới ra đời, Android được thiết kế với mục tiêu là chạy tốt trên máy tính bảng và điện thoại. Vậy nhưng, tính đến thời điểm hiện tại Android đã xuất hiện trên máy chơi game, trên Smart TV và nhiều thiết bị điện tử khác.

 **Ưu điểm:**

- **Kho ứng dụng đa dạng:** Với hệ thống cửa hàng ứng dụng **Google Play**, hệ điều hành Android có thể đáp ứng các nhu cầu từ chơi game cho đến làm việc với **hơn 3,3 triệu ứng dụng** để bạn thoải mái lựa chọn.
- **Mẫu mã đa dạng:** Với nhiều nhà sản xuất lớn như **Samsung, OPPO, Xiaomi, Nokia,...** bạn có thể lựa chọn giữa nhiều mẫu mã thiết bị khác nhau, từ **các mẫu giá rẻ** cho đến **các mẫu cao cấp**.
- **Có thể mở rộng bộ nhớ bằng thẻ nhớ:** Với các thiết bị của Apple, bạn chỉ có thể sử dụng bộ nhớ trong có sẵn của máy. Còn với phần lớn các thiết bị Android, bạn sẽ có lựa chọn **mở rộng bộ nhớ có sẵn với các loại thẻ nhớ dung lượng cao**.
- **Tùy biến dễ dàng:** Do bản chất nguồn mở của hệ điều hành Android, ai cũng có thể lấy được mã nguồn của hệ điều hành này.
- **Cộng đồng người dùng lớn:** Android có cộng đồng người dùng và lập trình viên độc lập khá lớn, nên khi bạn gặp vấn đề về thiết bị hay về phiên bản Android của bạn, bạn sẽ được hỗ trợ rất nhiệt tình từ phía cộng đồng.

Nhược điểm:

- **Nhiều ứng dụng chạy ngầm làm chậm máy:** Không giống như iOS, Android tối ưu hóa bộ nhớ RAM khá kém, dẫn đến việc nhiều ứng dụng chạy ngầm **gây chậm máy** hoặc **thậm chí là đơ máy**.
- **Ứng dụng không được tối ưu hóa tốt:** Do có quá nhiều mẫu mã khác nhau, các nhà phát triển không thể tối ưu hóa ứng dụng cho tất cả các mẫu thiết bị Android trên thị trường, nên các ứng dụng có thể gặp các lỗi như **không hiển thị được toàn màn hình** hoặc **không thể tận dụng hết sức mạnh phần cứng của máy**.
- **Chất lượng ứng dụng kém:** Một số ứng dụng trên Google Play có chất lượng khá kém với **hàng loạt các quảng cáo khó chịu** và **không có các chức năng hữu dụng**, gây cản trở cho công việc hay thời gian giải trí của bạn.

- **Để bị virus xâm nhập:** Nếu bạn tải các ứng dụng ở các nguồn khác ngoài Google Play nhưng lại không cài ứng dụng diệt virus, máy của bạn **có khả năng cao sẽ bị nhiễm virus** từ các ứng dụng ngoài này.

2.1.1.2. Hệ điều hành IOS

IOS là một nền tảng Platform được các nhà lập trình viết, sử dụng phổ biến trên các thiết bị di động chạy Apple. Khi mới ra đời, IOS được viết và phát triển để chạy trên điện thoại Iphone, nhưng hiện tại nó lại được mở rộng và chạy trên nhiều thiết bị khác của Apple như là Ipad, Ipod Touch và Apple TV.

Khi mới được viết ra hệ điều hành này cho các thiết bị điện tử của Apple thì IOS chưa có một cái tên riêng nào, nên được gọi đơn giản với cái tên iPhone chạy OS X. Đến ngày 6/3/2008, khi Apple cho bản dùng thử được phát lần đầu tiên thì nó có tên gọi mới cho hệ điều hành là Iphone OS.



Hình 2. 2. Hệ Điều Hành IOS

Tháng 6/2010 Iphone OS được Apple đổi tên thành IOS và nhãn hiệu IOS cũng được Cisco lựa chọn đặt tên hệ điều của mình. Vì không muốn gặp những bất chắc và liên quan tới kiện cáo vì sử dụng nhãn hiệu, Apple đã có văn bản xin phép bên Cisco được sử dụng nhãn hiệu IOS.

Đến thời điểm hiện tại, iOS đã có hơn 14 phiên bản được phát hành, với phiên bản mới nhất là iOS 14.6. Qua nhiều phiên bản với nhiều thay đổi, iOS ngày càng được cập nhật nhiều tính năng mới, bảo mật cao hơn và ổn định hơn.


Ưu điểm:


- iOS là một hệ điều hành với độ ổn định cao, mượt mà qua thời gian dài.
- Là hệ điều hành có độ bảo mật và tin cậy cao nhất hiện nay
- Hệ điều hành được cập nhật đều đặn hàng năm, bạn được quyền lựa chọn update hoặc không.
- Kho ứng dụng phong phú và được kiểm duyệt nghiêm ngặt, không có ứng dụng lừa đảo, ứng dụng rác.
- Đơn giản, dễ sử dụng cho người mới
- Có khả năng đồng bộ tốt giữa các thiết bị iOS với nhau

Nhược điểm:

- iOS hiện chỉ được tích hợp trên các thiết bị của Apple, tuy nhiên giá các sản phẩm Apple thì không hề rẻ.
- Việc trao đổi dữ liệu như chép tập tin, hình ảnh, video, v.v... đều phải sử dụng phần mềm iTunes, khá bất tiện cho người mới.
- Hạn chế khả năng tùy chỉnh.

2.1.2. Tổng quan về Navigator Flutter – Routes Fluuter

 **Navigation (Điều hướng):** Đây là quá trình điều hướng người dùng giữa các màn hình khác nhau trong ứng dụng của bạn. Trong Flutter, bạn có thể sử dụng "Navigator" để thực hiện điều hướng. Navigator cho phép bạn thêm, xóa và thay đổi các màn hình trên một ngăn xếp (stack) màn hình. Khi bạn chuyển đổi giữa các màn hình, bạn thêm các màn hình mới vào ngăn xếp, và màn hình hiện tại thường sẽ được đẩy ra khỏi ngăn xếp.

 **Routes (Đường dẫn):** Trong Flutter, mỗi màn hình hay giao diện được gọi là một "route" (đường dẫn). Mỗi route thường có một tên duy nhất để bạn có thể xác định và tham chiếu đến nó. Khi bạn muốn chuyển đến một màn hình cụ thể, bạn sẽ sử dụng tên của route để thực hiện điều hướng.

➦ Điều hướng đến một tuyến đường từ một tuyến đường khác → `Navigator.push()`

➦ Điều hướng đến tuyến đường đầu tiên → `Navigator.pop()`

2.1.3. Ví dụ về điều hướng giữa hai tuyến đường

➦ Tạo hai Routes

```
class FirstRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Route'),
      ),
      body: Center(
        child: RaisedButton(
          child: Text('Open route'),
          onPressed: () {
            // Navigate to second route when tapped.
          },
        ),
      ),
    );
  }
}

class SecondRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Second Route"),
      ),
    ),
  }
}
```

```
body: Center(  
  child: RaisedButton(  
    onPressed: () {  
      // Navigate back to first route when tapped.  
    },  
    child: Text('Go back!'),  
  ),  
,  
);  
}
```

Điều hướng đến route thứ hai bằng phương thức Navigator.push()

Phương thức Navigator.push() được sử dụng để điều hướng / chuyển sang một tuyến đường / trang / màn hình mới. Ở đây, phương thức push() thêm một trang / tuyến đường trên ngăn xếp và sau đó quản lý nó bằng cách sử dụng Bộ điều hướng. Một lần nữa, chúng ta sử dụng lớp MaterialPageRoute cho phép chuyển đổi giữa các tuyến bằng cách sử dụng hoạt ảnh dành riêng cho nền tảng.

```
// Within the `FirstRoute` widget  
onPressed: () {  
  Navigator.push(  
    context,  
    MaterialPageRoute(builder: (context) => SecondRoute()),  
  );  
}
```

Quay lại route đầu tiên bằng phương thức Navigator.pop()

Bây giờ, chúng ta cần sử dụng phương thức Navigator.pop () để đóng tuyến thứ hai và quay lại tuyến đầu tiên. Phương thức pop() cho phép chúng ta loại bỏ

tuyến đường hiện tại tại khối ngăn xếp, được quản lý bởi Bộ điều hướng. (Xuan, 2021)

Để triển khai quay lại tuyến ban đầu, chúng ta cần cập nhật phương thức gọi lại `onPressed()` trong tiện ích con `SecondRoute` như đoạn mã bên dưới:

```
// Within the SecondRoute widget
onPressed: () {
  Navigator.pop(context);
}
```

Tiến hành chèn đoạn mã vào tệp **main.dart**

```
import 'package:flutter/material.dart';

void main() {
  runApp(MaterialApp(
    title: 'Flutter Navigation',
    theme: ThemeData(
      // This is the theme of your application.
      primarySwatch: Colors.green,
    ),
    home: FirstRoute(),
  ));
}

class FirstRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Screen'),
      ),
      body: Center(
        child: RaisedButton(
```

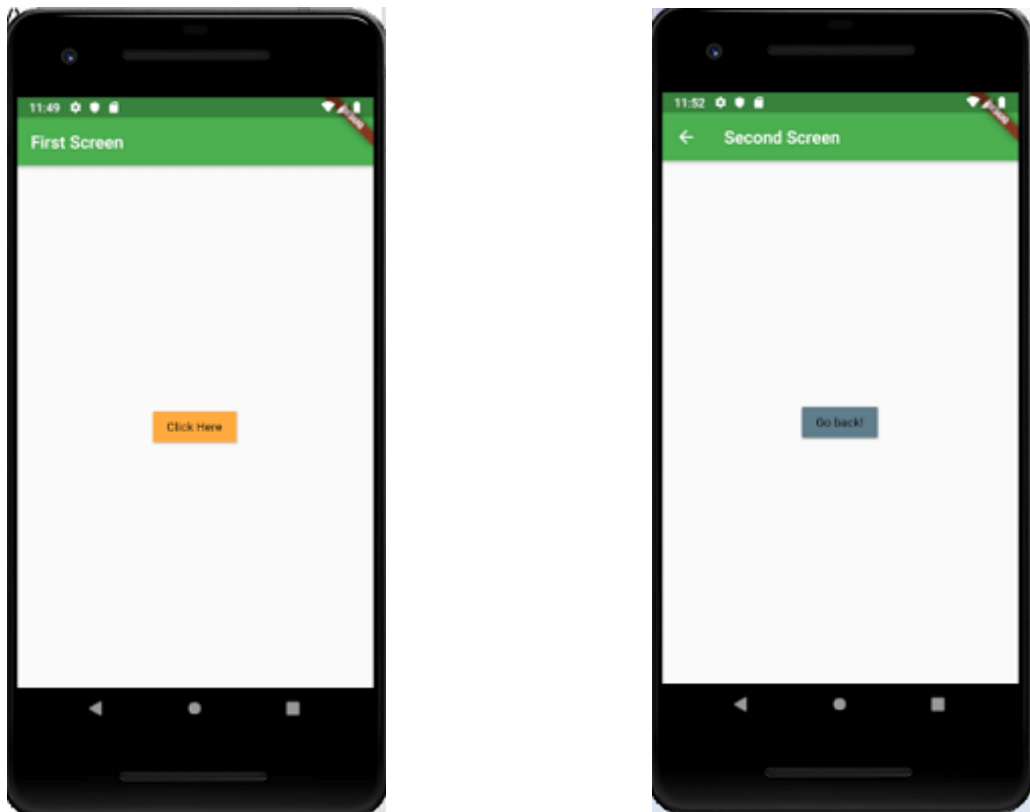
```
        child: Text('Click Here'),
        color: Colors.orangeAccent,
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) =>
SecondRoute()),
          );
        },
      ),
    ),
  );
}
}

class SecondRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Second Screen"),
      ),
      body: Center(
        child: RaisedButton(
          color: Colors.blueGrey,
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Go back'),
        ),
      ),
    );
  }
}
```

```
}  
  
}
```

Khi bạn chạy dự án trong **Android Studio**, bạn sẽ nhận được màn hình sau trong trình mô phỏng của mình. Đây là màn hình đầu tiên chỉ chứa một nút duy nhất.

Nhấp vào nút Click Here, và bạn sẽ điều hướng đến màn hình thứ hai như hình ảnh bên dưới. Tiếp theo, khi bạn nhấp vào nút **Quay lại**, bạn sẽ trở lại trang đầu tiên.



Hình 2. 3. Giao diện điện thoại khi dự án

2.2. Tổng quan về kỹ thuật

2.2.1. Ngôn ngữ Dart



Hình 2. 4. Hình ảnh về Dart

- ✚ Dart là ngôn ngữ lập trình đa mục đích, xây dựng và phát triển bởi Google. Sau đó, nó được Ecma (ECMA-408) phê duyệt làm tiêu chuẩn.
- ✚ Dart thường được dùng trong việc xây dựng những ứng dụng web, server trên máy tính để bàn và nhiều thiết bị di động khác.
- ✚ Dart còn là một ngôn ngữ hướng đối tượng, được xác định theo lớp và hoạt động dưới cơ chế garbage-collected. Nó sử dụng cú pháp kiểu C để dịch mã sang JavaScript đúng như yêu cầu.
- ✚ **Ưu điểm:**
 - Cú pháp mang lại năng suất rõ ràng và súc tích.
 - Bộ công cụ hỗ trợ đơn giản nhưng mạnh mẽ.
 - Có type-safe giúp bạn xác định sớm những lỗi tinh tế.
 - Sở hữu các thư viện cốt lõi và một hệ sinh thái bao gồm hàng ngàn package.
 - Nó 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 cùng với web.
 - Biên dịch thành mã ARM và x86, giúp các ứng dụng di động có thể chạy được một cách tự nhiên trên iOS, Android... hơn thế nữa.
 - Với các ứng dụng web, Dart giúp chúng linh hoạt hơn trong quá trình chuyển mã sang JavaScript.

- Quen thuộc với nhiều nhà phát triển hiện có, nhờ vào cú pháp và định hướng đối tượng không gây ngạc nhiên.
 - Nếu đã quá quen thuộc với C++, C # và Java thì bạn sẽ dễ dàng làm việc hiệu quả với Dart chỉ trong vài ngày.
 - Rất phù hợp với lập trình Reactive, hỗ trợ trong việc quản lý các đối tượng tồn tại ở một khoảng thời gian không dài như widget UI...
 - Hỗ trợ lập trình không đồng bộ thông qua các tính năng ngôn ngữ và API sử dụng đối tượng Future/Stream.
- ✚ Dart cũng nổi tiếng như là một ngôn ngữ mã nguồn mở, hỗ trợ sử dụng miễn phí và được phát triển trên GitHub. Ngôn ngữ Dart hoạt động tương tự với Java và là phiên bản cải tiến từ JavaScript.

2.2.1. Flutter

- ✚ Flutter là một framework mã nguồn mở phát triển bởi Google, được sử dụng để xây dựng ứng dụng đa nền tảng chất lượng cao với giao diện người dùng tương tự trên các nền tảng khác nhau như iOS, Android, web và desktop. Flutter sử dụng ngôn ngữ lập trình Dart, cùng với một tập hợp các thư viện và công cụ mạnh mẽ để giúp nhà phát triển xây dựng ứng dụng nhanh chóng và dễ dàng.



Hình 2. 5. Logo Flutter

✚ Ưu điểm:

- Ngôn ngữ Dart: Dart - OOP, nó khá dài, nhưng những người quen viết bằng Java / C # sẽ hài lòng với Dart.

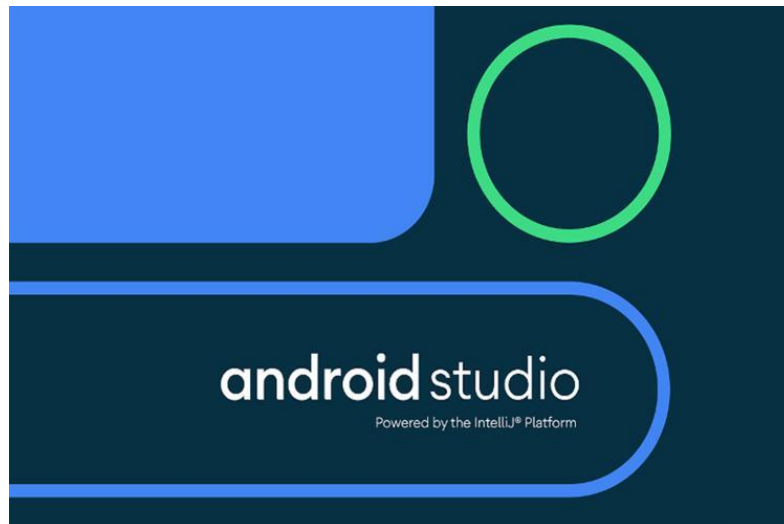
- Ít phải viết các config khi tạo project mới theo cách thủ công, ngược lại với React Native.
- Chỉ cần tải về Flutter từ git, chạy “flutter doctor”, và nó sẽ cho bạn biết những loại vấn đề bạn có thể có trong hệ thống. Sau đó, tạo một project từ Android Studio hoặc bất kỳ IDE nào sau khi cài đặt Flutter Plug-in sơ bộ và start project. Ngoài ra nó còn có Hot-reload giống như trong React-Native.
- Hỗ trợ tuyệt vời cho IDE (Android Studio, IDE IntelliJ, VSCode)
- Trình điều hướng tích hợp sẵn. Sau khi tạo ra một ứng dụng trên Flutter, bạn có thể thấy nó có Navigator được xây dựng sẵn. Bạn có thể tạo một route mới mà không cần kết nối bất kỳ thứ gì (không giống như trong RN, nơi bạn cần kết nối bất kỳ package điều hướng gốc nào vv);
- Giải pháp quản lý dữ liệu được tạo sẵn, ví dụ trong Flutter này là "Mọi thứ đều có một Widget" cho nó. Có một StatefulWidget đặc biệt, với sự giúp đỡ của nó mà chúng ta có thể quản lý trạng thái widget và thay đổi nó một cách năng động.

Nhược điểm:

- Có một rào cản lớn cho những người chỉ học JS, hoặc đến từ RN. Do thực tế là Dart được sử dụng với sự thừa kế, đa hình của nó và tất cả các thứ về OOP.
- Không có JSX mà tất cả chúng ta đã quen khi làm RN. Nhìn vào tệp Dart trong đó không có phân chia thành template, styling và data, nó trở nên khó chịu.
- Styling là một nhược điểm. Thực tế là không có sự tách biệt thành Styles, templates và controller, có một vấn đề khi mô tả một thành phần thì chúng ta cũng cần mô tả các styles cùng một lúc.
- Animation còn khó khăn hơn. Mặc dù nó có một animation tốt, nhưng nó sẽ khó hơn khi tạo hiệu ứng động, không giống như trong RN.
- Tối ưu hóa. Trong Flutter, chỉ có thừa kế widget với các điều kiện có thể thay đổi (trạng thái) và không thể thay đổi (không trạng thái). Trong khi ở React-Native chúng ta có thể quản lý vòng đời. Ngoài ra, một nhược điểm lớn là

thiếu các công cụ để lưu trạng thái của ứng dụng, vấn đề này có thể, tuy nhiên, được giải quyết bằng cách tuần tự hóa trạng thái hiện tại.

2.2.1. Android Studio



Hình 2. 6. Android Studio

Android Studio là IDE chính thức được sử dụng trong phát triển ứng dụng Android dựa trên IntelliJ IDEA. Chức năng chính của Android Studio là cung cấp các giao diện giúp người dùng có thể tạo các ứng dụng và xử lý các công cụ file phức tạp sau hậu trường. Ngôn ngữ lập trình được sử dụng trong Android Studio là Java và nó sẽ được cài đặt sẵn trên thiết bị của bạn. Khi **sử dụng Android Studio** thì bạn chỉ cần viết, chỉnh sửa và lưu trữ chúng trên các dự án của mình và các file nằm trong dự án đó. Đồng thời, Android Studio còn cung cấp quyền truy cập vào Android SDK. Android studio có rất nhiều lợi ích cũng như những tính năng nổi bật mà người dùng có thể khai thác được. Đặc biệt đó là, cài đặt android studio còn có thể giúp các lập trình viên mô phỏng để có thể tiến hành sửa chữa và nâng cấp ngay các sản phẩm của mình khi cần.

Ngoài ra, bạn có thể xem Android Studio là đầu cho code Java cho phép nó chạy trơn tru trên các thiết bị Android rồi tận dụng được lợi thế của phần cứng gốc. Bạn chỉ cần sử dụng ngôn ngữ lập trình Java để có thể viết chương trình, khi đó Android SDK sẽ có nhiệm vụ kết nối các phần này lại với nhau. Khi đó, Android Studio sẽ kích hoạt

để thực hiện chạy code và thông qua trình giả lập hoặc dựa vào bất kỳ phần cứng giúp kết nối với thiết bị. Sau đó, bạn có thể gỡ rối cho chương trình của mình ngay khi nó chạy và nhận phản hồi giúp giải thích các sự cố,..Cho đến hiện nay, Google đã và đang rất nỗ lực để giúp cho Android Studio sẽ trở nên mạnh mẽ và hữu ích hơn. Khi bạn gõ code, nó sẽ giúp bạn cung cấp danh sách gợi ý hoàn thành để giúp người dùng có thể hoàn thiện được dòng code đó. Đây là một trong những chức năng rất hữu ích đề phòng cho trường hợp người dùng không nhớ chính xác cú pháp giúp bạn tiết kiệm thời gian hiệu quả hơn. Mời bạn tham khảo thêm một số việc làm Android tại Itnavi.

2.2.2. Genymotion

Genymotion nói một cách dễ hiểu là *phần mềm giả lập Android*. Tại đây sẽ có các tính năng hoàn chỉnh để bạn trải nghiệm như trên môi trường hệ điều hành Android ảo. Mục đích ra đời phần mềm này chính là mong muốn người dùng có thêm nhiều lựa chọn khi chơi game. Bên cạnh đó, phần mềm này còn giúp người dùng sử dụng được các app trên Android phù hợp với cấu hình máy tính hay laptop.



Hình 2. 7. Genymotion

Máy ảo này sở hữu nhiều tính năng nổi trội, hỗ trợ người chơi một cách tốt nhất. Có nhiều điểm tạo nên sự thành công của Genymotion như:

- Không chỉ hỗ trợ Windows mà còn có thể ứng dụng trên **Mac OS**.
- Giúp người dùng sử dụng Webcam ở laptop có chức năng như camera điện thoại.

- Chương trình khởi động và vận hành **tốc độ cao và mượt mà**.
- Hỗ trợ thiết lập cấu hình ảo phù hợp cấu hình máy.
- Có tính năng chụp hình và quay màn hình trực tiếp tương đối tốt.
- Âm thanh có thể chỉnh dễ dàng.
- Có thể liên kết mạng xã hội khác như Facebook, Zalo
- Người dùng có thể cài đặt nhanh chóng nhờ hệ thống tiện ích mở rộng **Open Gapps**.
- Cho phép tương tác với máy ảo bằng các dòng lệnh.
- Hỗ trợ ADB...

2.2.3. Visual Studio Code

Visual Studio Code được biết đến là một trình biên tập lập trình code miễn phí dành cho **Windows**, **Linux** và **macOS**. Nó được phát triển bởi Microsoft là sự kết hợp hoàn hảo giữa IDE¹ và Code Editor². Trình biên tập này được hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting³, tự hoàn thành mã thông minh, snippets⁴, cùng



đó là cải tiến mã nguồn. Ngoài ra, Visual Studio Code còn cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

Hình 2. 8. Hình ảnh Visual Studio Code

¹ IDE (Integrated Development Environment) là môi trường tích hợp dùng để viết code để phát triển ứng dụng.

² Code Editor: Trình soạn thảo code

³ Syntax highlighting là một công cụ giúp tô màu sắc các đoạn mã nguồn.

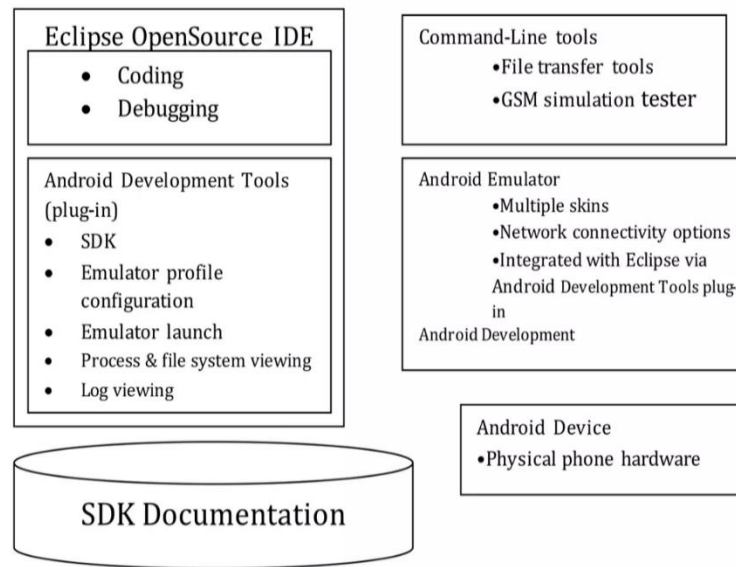
⁴ Snippet là một đoạn code được định nghĩa sẵn và gắn bó vào từ khóa nào đó, khi cần dùng thì ta sẽ gọi từ khóa đó để chèn nội dung mà ta đã định nghĩa.

2.2.4. Android SDK

- ✚ SDK viết tắt của Software Development Kit là các công cụ và phần mềm dùng để phát triển ứng dụng thông qua một nền tảng nhất định.
- ✚ SDK cung cấp các thư viện, tài liệu, mẫu template, sample code, tiện ích gỡ rối (debugging), các ghi chú hỗ trợ (documentation) hoặc các tài liệu bổ sung,... để nhà phát triển có thể tích hợp thêm vào phần mềm/ứng dụng của mình. Đa phần chúng thường là chức năng hiển thị quảng cáo, push notification,...
- ✚ SDK cũng có thể chứa các API dưới dạng thư viện hoặc một hệ thống phân cứng phức tạp nào đó.
- ✚ SDK thường được xây dựng và tùy chỉnh sao cho tương thích với ngôn ngữ lập trình và các đặc điểm tương ứng. Theo đó, chúng ta có thể phân loại SDK thành các loại phổ biến sau:
 - Bộ công cụ SDK Android: sử dụng ngôn ngữ Java, được dùng để lập trình các ứng dụng trên nền tảng Android.
 - Bộ công cụ SDK iOS: sử dụng ngôn ngữ Swift, được dùng để hỗ trợ lập trình ứng dụng trên nền tảng iOS.
 - Bộ công cụ SDK Windows: Windows yêu cầu phải có .NET Framework SDK đi kèm với .NET để lập trình các phần mềm chuyên dụng.
 - Bộ công cụ SDK VMware: được dùng để tích hợp với nền tảng VMware (cho phép ảo hóa trên công nghệ điện toán đám mây).
 - Bộ công cụ SDK Bắc Âu: được dùng để hỗ trợ tạo ra các sản phẩm Bluetooth hoặc không dây.

2.2.5. Môi trường lập trình trong Android Studio

Công cụ lập trình cho Android (Android Development Tools). Chúng ta sẽ dần làm quen với Eclipse và Android Development Tool plug-in, thông qua Android SDK và công cụ của nó, chạy một ứng dụng Android trên Emulator (tạm dịch là trình giả lập giao diện của Android trên PC). Với những kỹ năng đó trong tay, chúng ta sẽ tìm hiểu các gói Java packages được cung cấp trong SDK giúp nâng cao khả năng lập trình trên Android.



Hình 2. 9. Môi trường lập trình trong Android

2.3. Tổng quan về lập trình bất đồng bộ

Lập trình bất đồng bộ (Asynchronous Programming) là một phong cách lập trình cho phép các tác vụ (hoặc hàm) thực hiện mà không cần chờ đợi tác vụ trước hoàn thành. Thay vì blocking (chặn) việc thực hiện các tác vụ, lập trình bất đồng bộ cho phép chương trình tiếp tục thực thi các tác vụ khác trong khi các tác vụ bất đồng bộ đang chạy.

2.3.1. Future

Về cơ bản, cách thức hoạt động của Future của Dart giống với Promise từ ngôn ngữ Javascript. Nó bao gồm hai trạng thái: Uncompleted và Completed.

- **Completed Future** sẽ cho chúng ta giá trị trả về hoặc lỗi xảy ra.
- **Uncompleted Future** là trạng thái chờ đợi hoạt động bất đồng bộ của hàm sẽ hoàn thành hoặc xảy ra lỗi.

Future có một vài hàm khởi tạo cơ bản, dưới đây là một số loại mình thường dùng:

- **Future.delayed()**: Dart sẽ thực hiện việc tính toán và trả ra kết quả sau khoảng thời gian delay trôi qua.
- **Future.error()**: Dart sẽ trả về lỗi sau khi thực thi việc tính toán.

2.3.2. Async / Await

Các phương pháp tiếp cận async và await trong Dart về cơ bản giống với các phương ngữ khác nhau.

Async functions: Hàm bất đồng bộ cho phép bạn thực hiện các tác vụ bất đồng bộ như call API lấy dữ liệu, lấy data hoặc lưu trữ dưới local, v.v bên trong body của hàm. Dưới đây là một ví dụ minh họa:

```
void flutter() async {  
    print('Flutter Devs');  
}
```

Await expressions: Nó giúp bạn thực hiện code bất đồng bộ như thể nó diễn ra theo từng dòng. Dưới đây là ví dụ minh họa:

```
void main() async {  
    await flutter();  
    print('flutter Devs done');  
}
```

CHƯƠNG III. CÀI ĐẶT THỰC NGHIỆM

3.1. Cài đặt phần mềm kỹ thuật

3.1.1. Cài đặt Android Studio

🔗 Bước 1: Vào trang chủ của android studio theo link sau (<https://developer.android.com/studio>) nhấn vào download Android Studio.



Android Studio provides the fastest tools for building apps on every type of Android device.

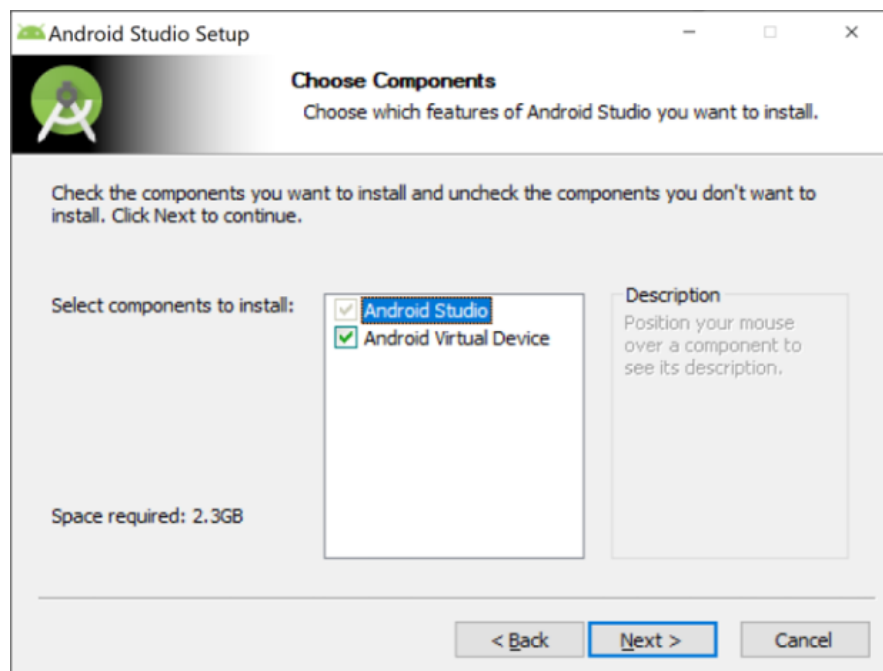
Download Android Studio

4.2.1 for Windows 64-bit (933 MiB)

Download options

Release notes

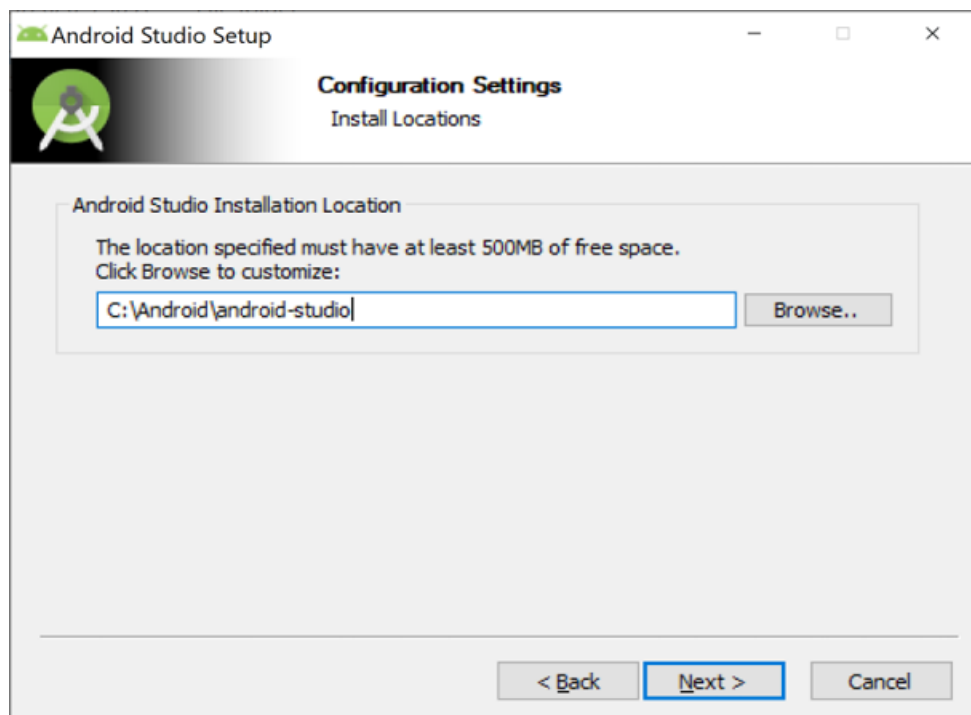
🔗 Bước 2: Sau khi màn hình setup hiện lên, nhấn Next để tiếp tục



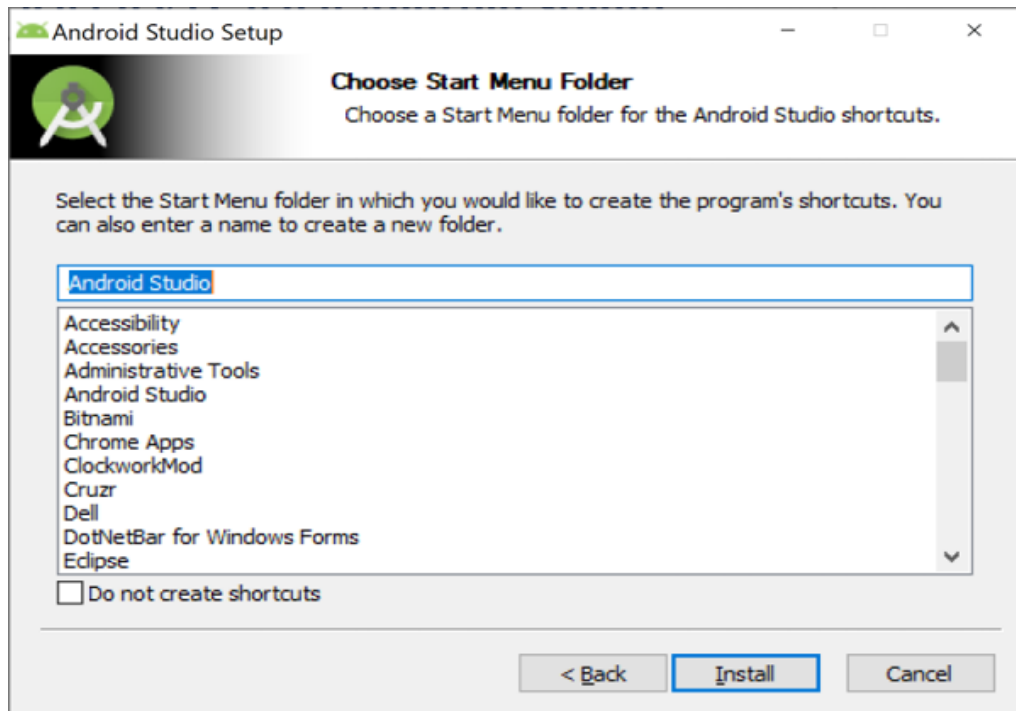
🔗 Bước 3: Để mặc định rồi nhấn Next để tiếp tục



+ Bước 4: Lưu ý cài vào thư mục C:\Android\android-studio sau đó nhấn Next

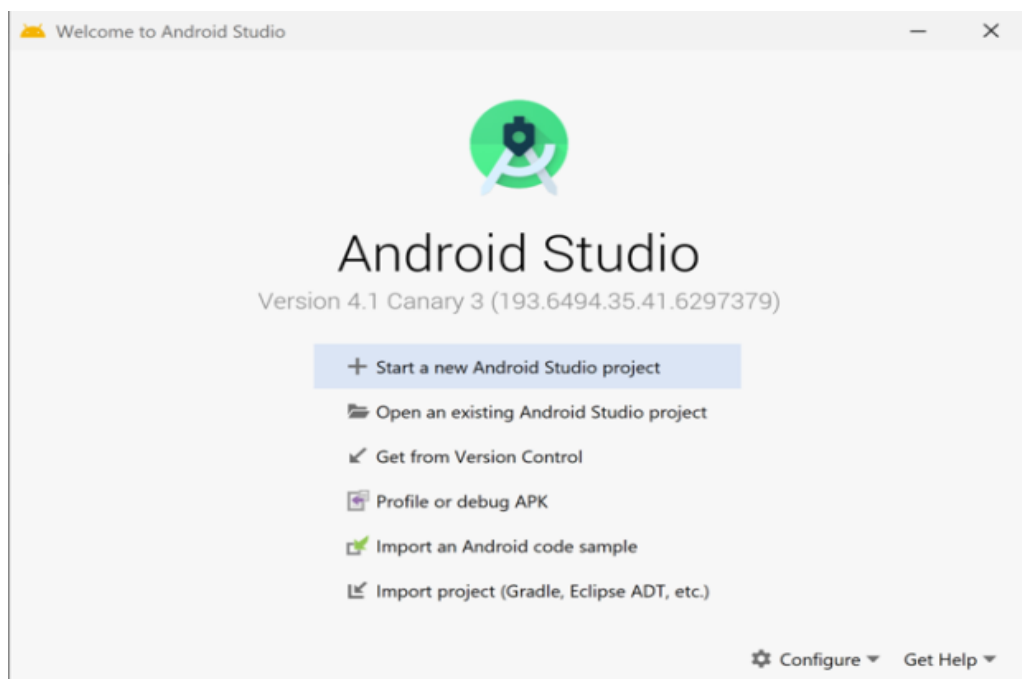


+ Bước 5: Sau đó nhấn install để bắt đầu cài đặt.

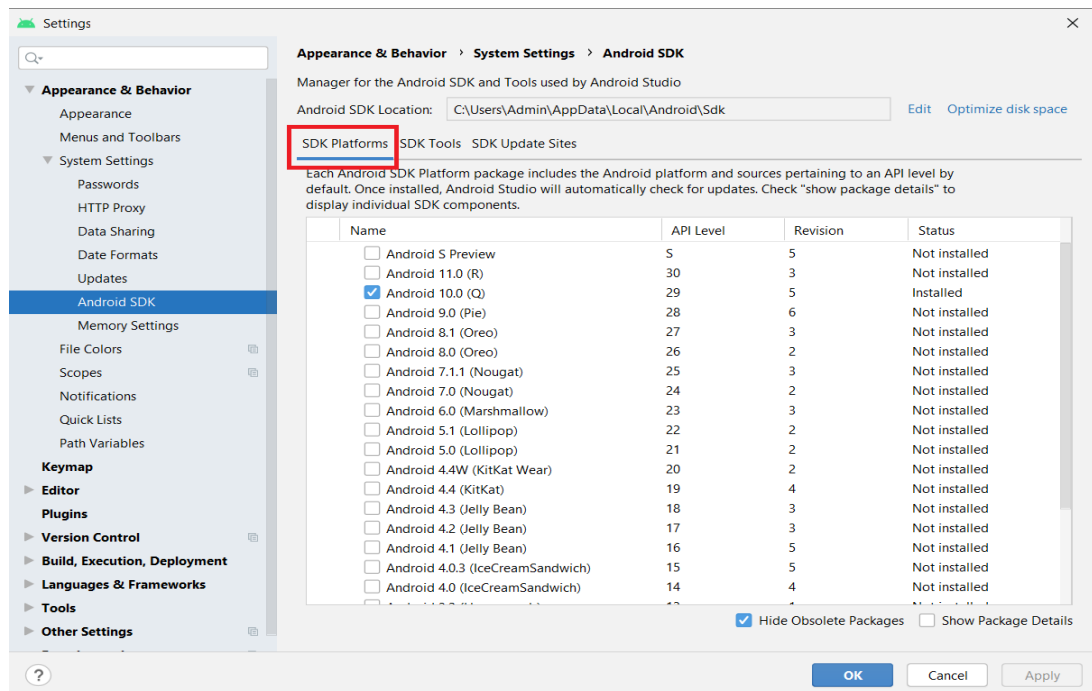


3.1.2. Cài đặt SDK cho Android

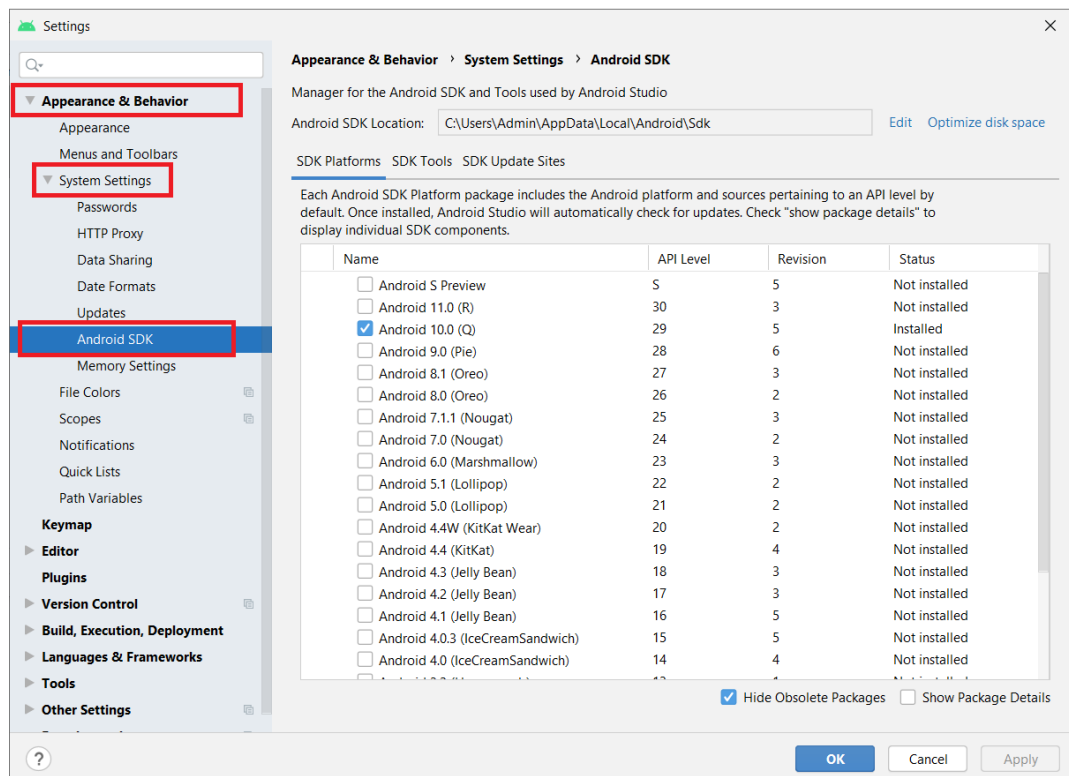
- ✚ Bước 1: Lúc này phần mềm Android Studio sẽ hiện lên như sau, nhấn vào configure để thiết lập android cho máy ảo.



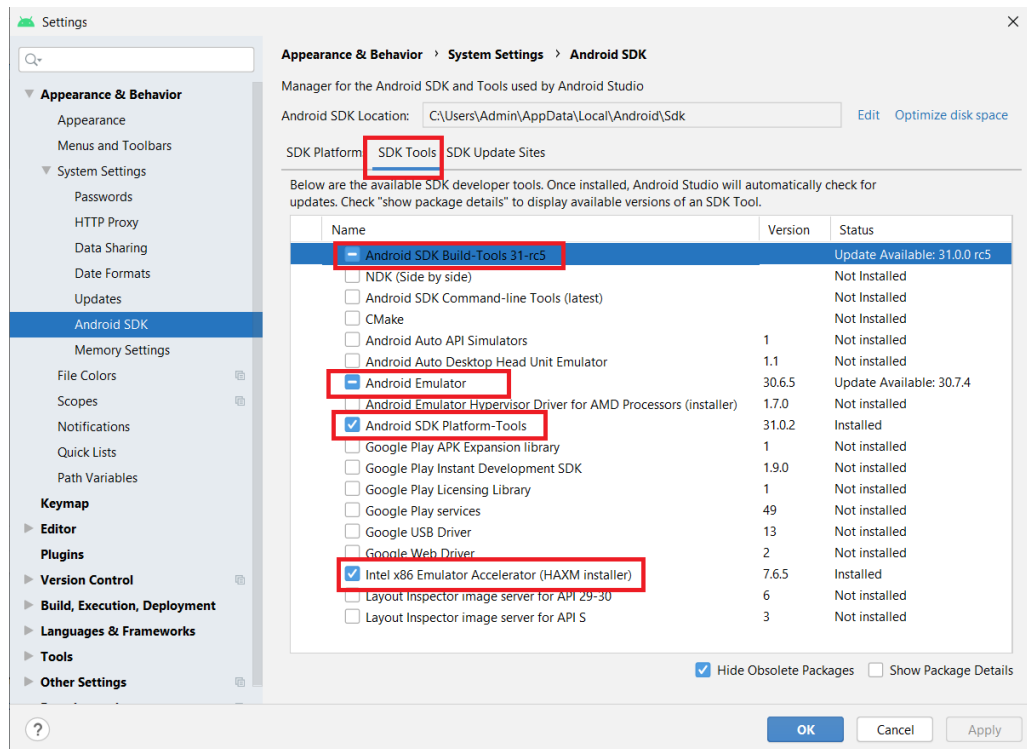
Bước 2: Chọn vào Appearance & Behavior => System Setting => Android SDK



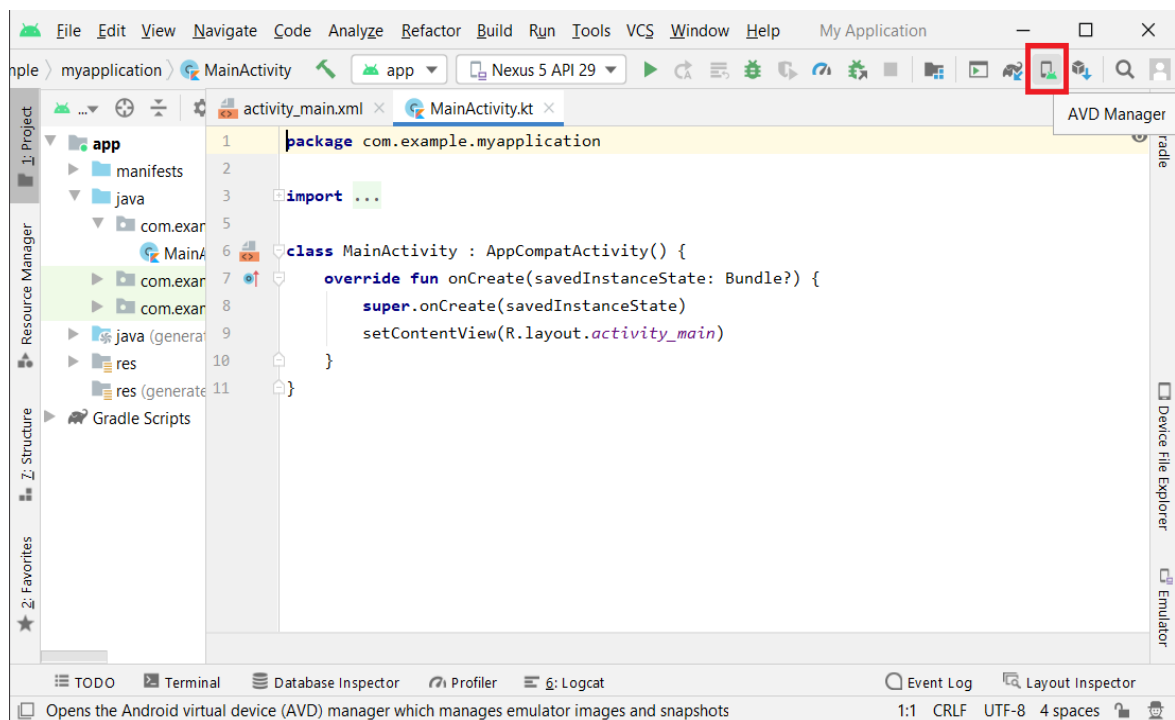
Bước 3: Sau đó vào mục SDK Platforms, chọn hệ điều hành Android mà mình muốn cài đặt.



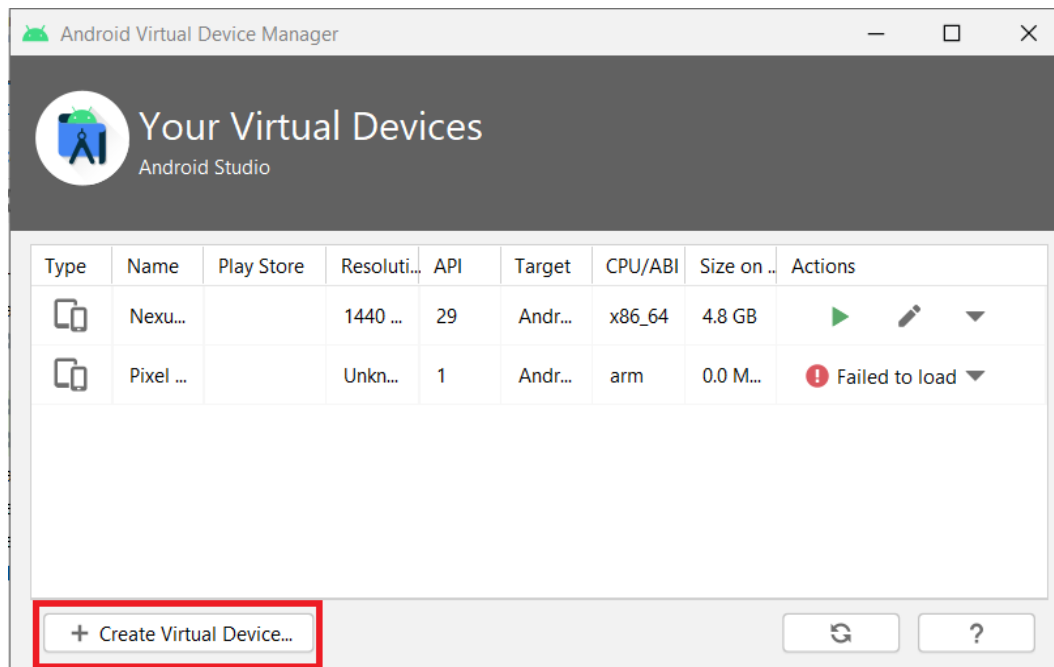
- ✚ Bước 4: Sau đó qua SDK Tools, tích chọn Android SDK Build – Tools, Android SDK Platform – Tools... rồi nhấn Apply và OK.



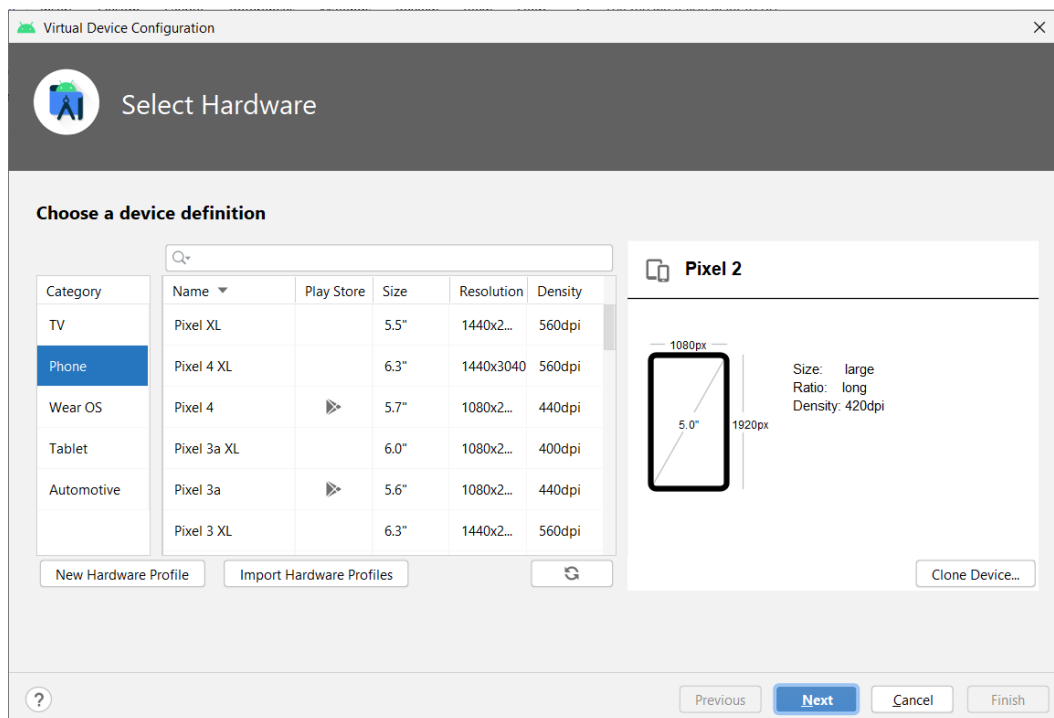
- ✚ Bước 5: Mở Android Studio lên, nhấp vào AVD Manager



- Bước 6: Sau khi màn hình AVD Manager hiện lên, nhấn chọn Virtual Device Configuraion.

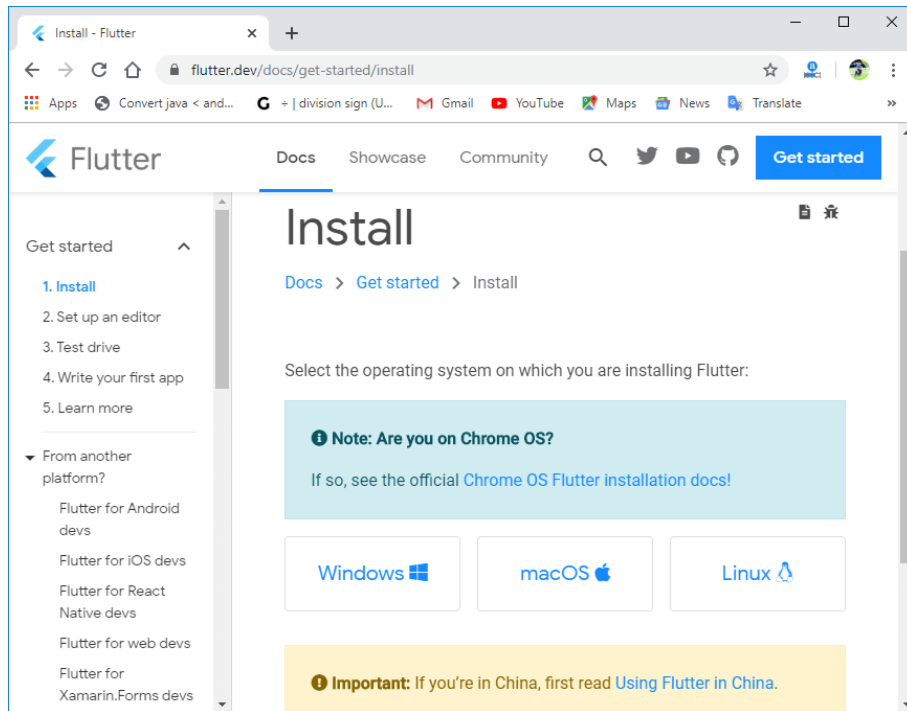


- Bước 7: Chọn kích cỡ điện thoại mong muốn rồi nhấn Next, sau đó nhấn Finish để hoàn tất

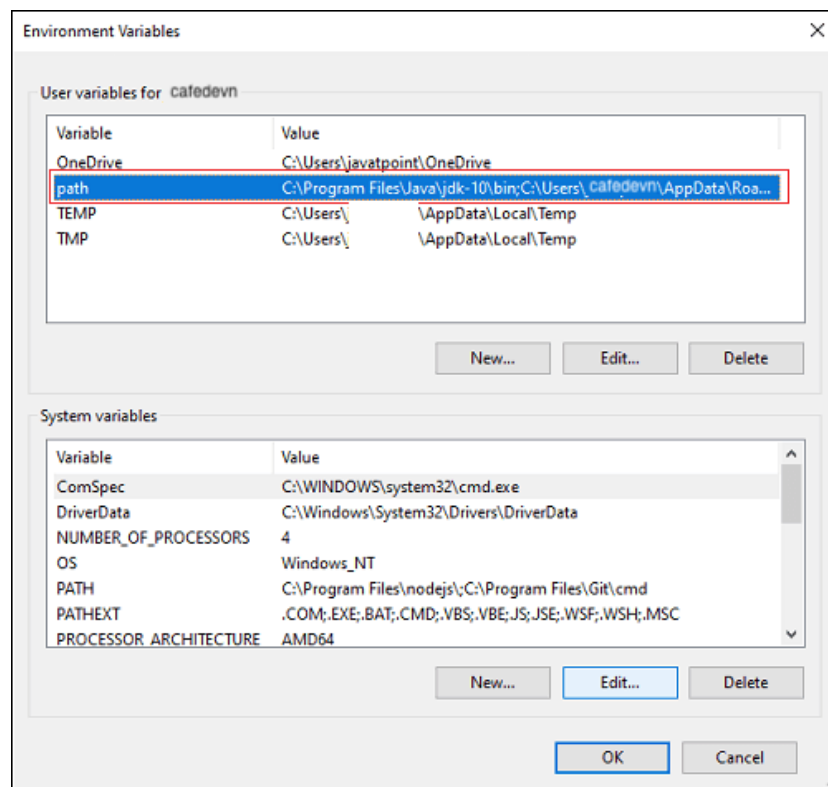


3.1.3. Cài đặt Flutter

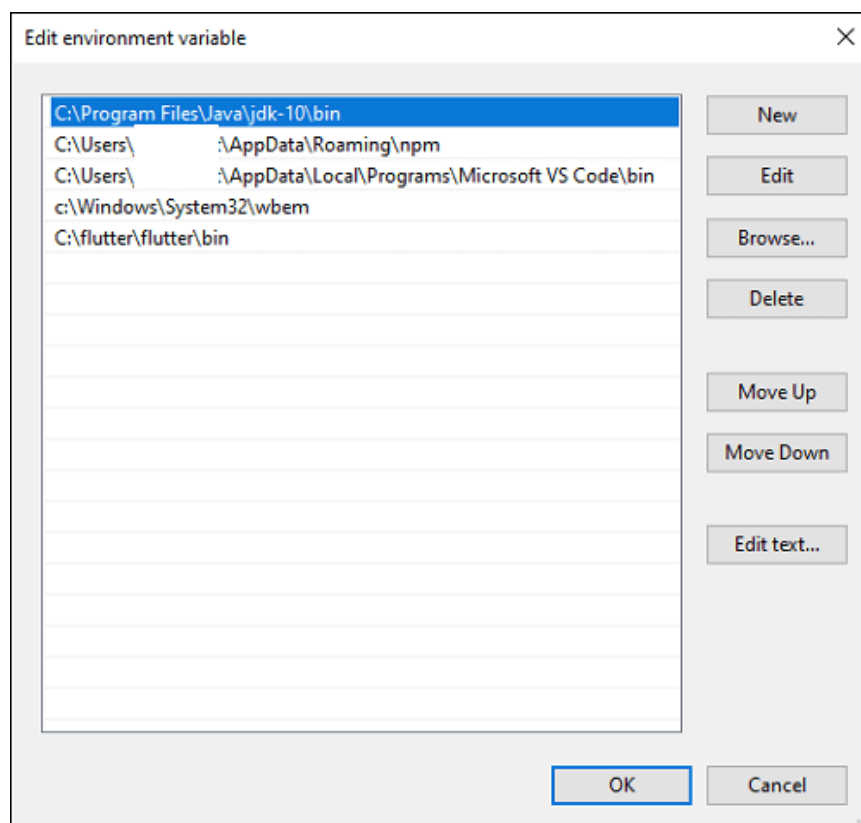
- ✚ Bước 1: Tải xuống gói cài đặt của Bộ phát triển phần mềm Flutter dành cho windows. Để tải xuống Flutter SDK, hãy truy cập trang web chính thức của nó, nhấp vào nút Bắt đầu, bạn sẽ nhận được màn hình sau.



- ✚ Bước 2: Tiếp theo, để tải xuống Flutter SDK mới nhất, hãy nhấp vào biểu tượng Windows. Tại đây, bạn sẽ tìm thấy liên kết tải xuống cho SDK.
 - ✚ Bước 3: Khi quá trình tải xuống của bạn hoàn tất, hãy giải nén tệp zip và đặt nó vào thư mục hoặc vị trí cài đặt mong muốn, ví dụ: D: / Flutter.
- **Lưu ý: Không nên đặt Flutter SDK ở nơi cần có sự cho phép của quản trị viên.**
- ✚ Bước 4: Để chạy lệnh Flutter trong bảng điều khiển windows thông thường, bạn cần cập nhật đường dẫn hệ thống để bao gồm thư mục flutter bin. Các bước sau được yêu cầu để thực hiện việc này:
 - ✚ Bước 4.1: Vào thuộc tính MyComputer -> tab nâng cao -> biến môi trường. Bạn sẽ nhận được màn hình sau.



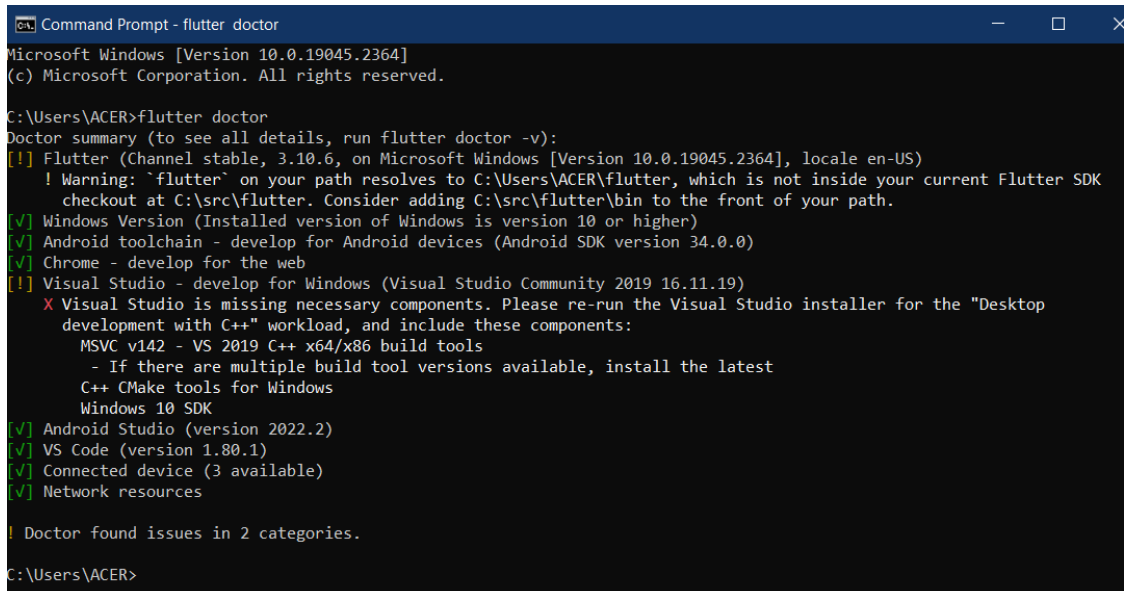
- ✚ Bước 4.2: Bây giờ, chọn đường dẫn -> nhấp vào chỉnh sửa. Màn hình sau xuất hiện.



- ✚ Bước 4.3: Trong cửa sổ trên, nhấp chuột vào New-> ghi đường dẫn của thư mục bin Flutter vào giá trị biến -> ok -> ok -> ok.
- ✚ Bước 5: Bây giờ, chạy lệnh \$ flashing doctor. Lệnh này kiểm tra tất cả các yêu cầu phát triển ứng dụng Flutter và hiển thị báo cáo về trạng thái cài đặt Flutter của bạn.

```
flutter doctor
```

- ✚ Bước 6: Khi bạn chạy lệnh trên, nó sẽ phân tích hệ thống và hiển thị báo cáo của nó, như trong hình dưới đây. Tại đây, bạn sẽ tìm thấy thông tin chi tiết của tất cả các công cụ còn thiếu, cần thiết để chạy Flutter cũng như các công cụ phát triển có sẵn nhưng không được kết nối với thiết bị.



```
Command Prompt - flutter doctor
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>flutter doctor

Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.10.6, on Microsoft Windows [Version 10.0.19045.2364], locale en-US)
    ! Warning: `flutter` on your path resolves to C:\Users\ACER\flutter, which is not inside your current Flutter SDK
      checkout at C:\src\flutter. Consider adding C:\src\flutter\bin to the front of your path.
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Chrome - develop for the web
[!] Visual Studio - develop for Windows (Visual Studio Community 2019 16.11.19)
    X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop
      development with C++" workload, and include these components:
        MSVC v142 - VS 2019 C++ x64/x86 build tools
        - If there are multiple build tool versions available, install the latest
        C++ CMake tools for Windows
        Windows 10 SDK
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.80.1)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 2 categories.

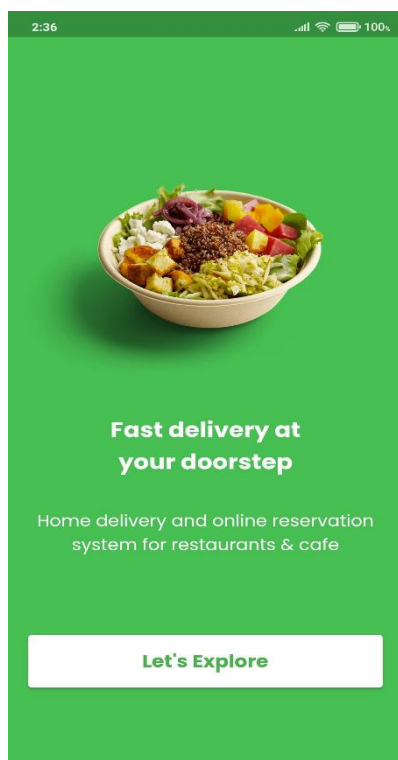
C:\Users\ACER>
```

- ✚ Bước 7: Cài đặt thành công.

3.2. Giao diện Project

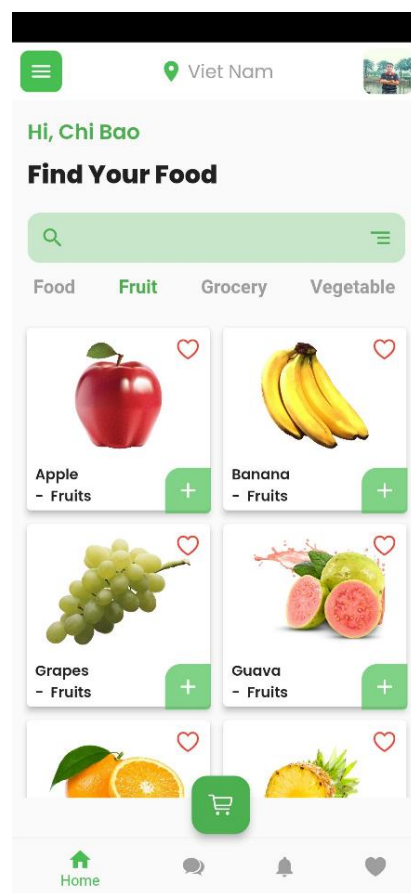
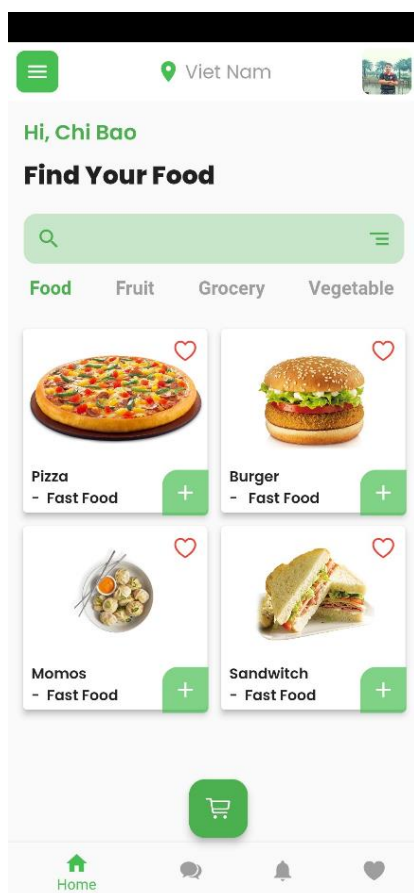
3.2.1. Giao diện chính

Khi bạn vừa khởi động ứng dụng, hệ thống sẽ hiển thị giao diện chính của ứng dụng.

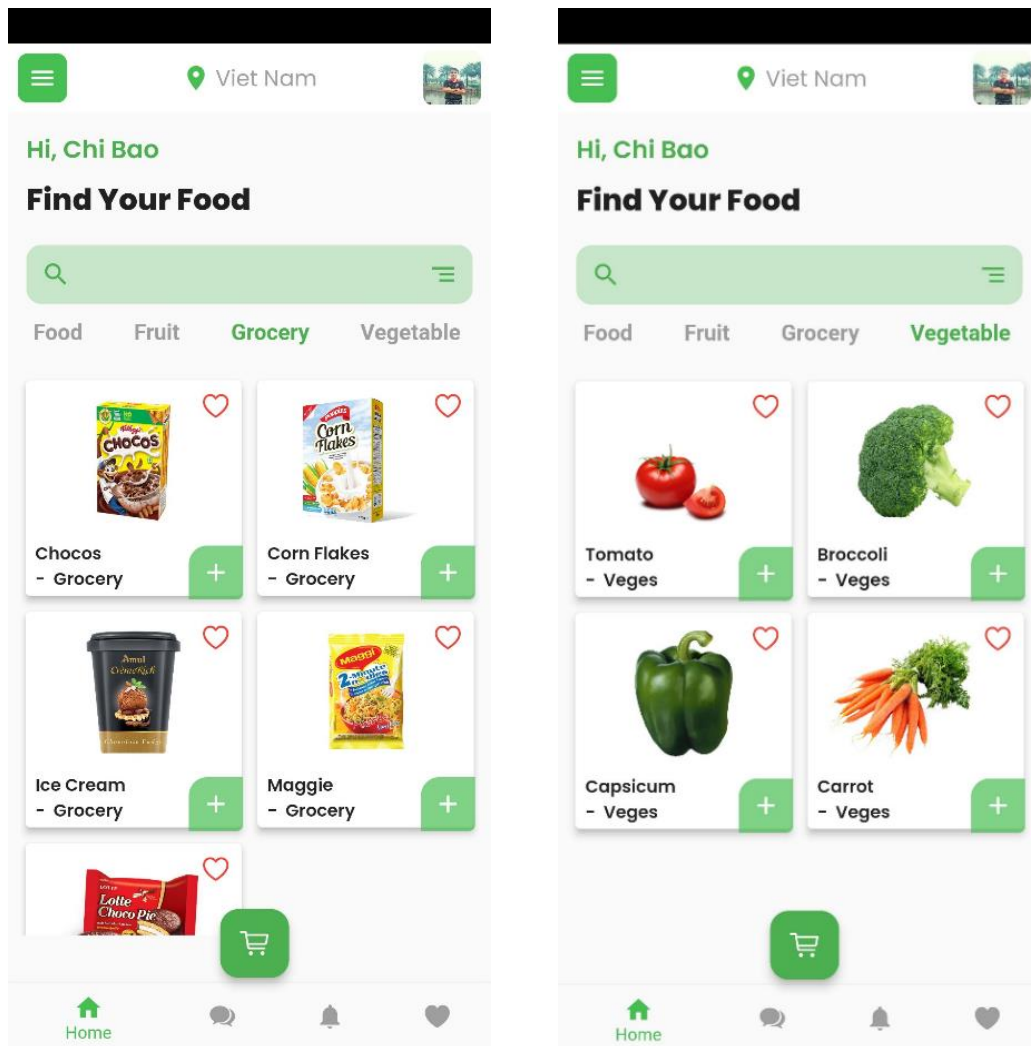


Hình 3. 1. Giao diện chính của hệ thống

3.2.2. Giao diện Home (Sản phẩm)





Hình 3. 2. Giao diện trang Home "Food - Fruit"

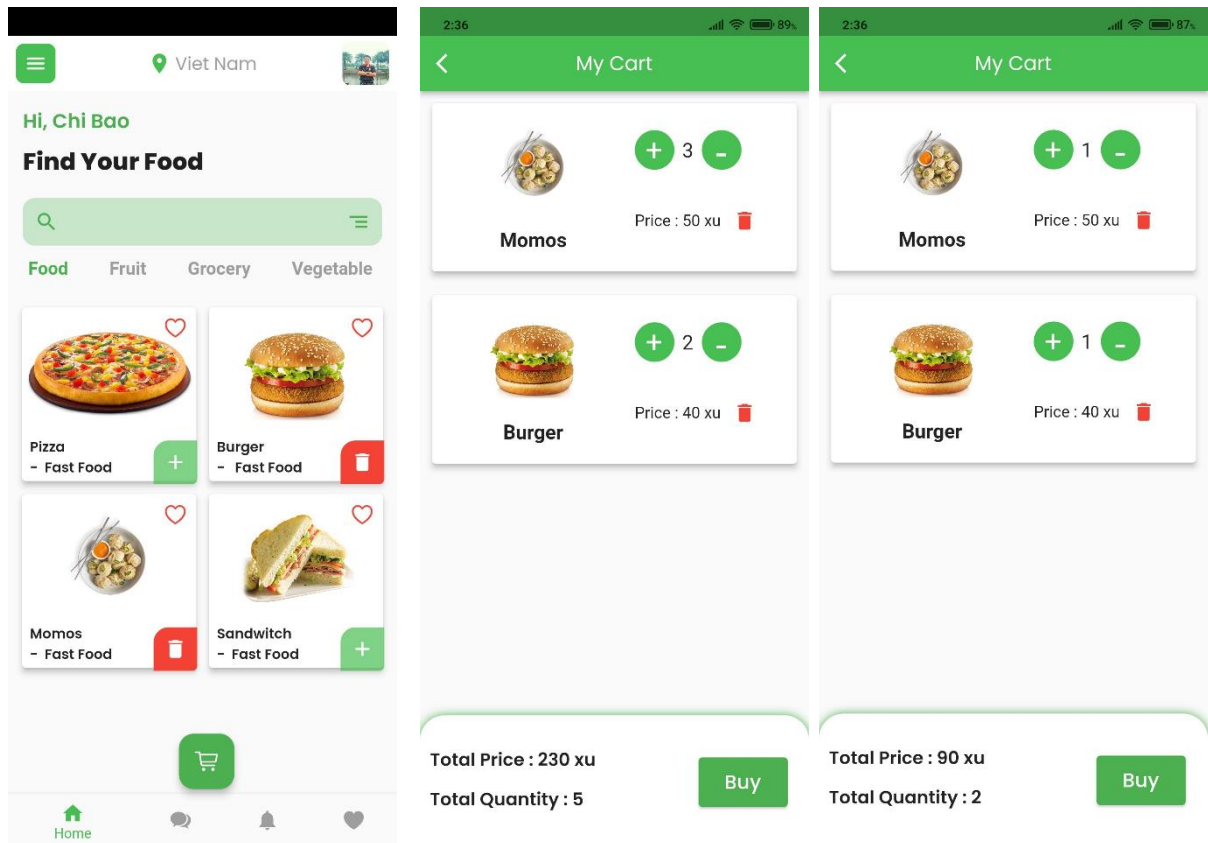


Hình 3. 3. Giao diện trang Home "Grocery - Vegetable"

3.2.3. Giao diện giỏ hàng (Hóa đơn)





- ✚ Khi bạn chọn mua một sản phẩm, bạn nhấn vào dấu **+** trên sản phẩm đó
- ✚ Khi nút đó chuyển sang **+** thì bạn đã chọn mua sản phẩm thành công
- ✚ Bạn chọn vào **+** để xem hàng hóa mà bạn đã chọn
- ✚ Bạn có thể thêm – sửa – xóa số lượng sản phẩm
- ✚ Sau khi bạn kiểm tra các thông tin hàng hóa → Nhấn nút **Buy** để thanh toán các hàng hóa đó.

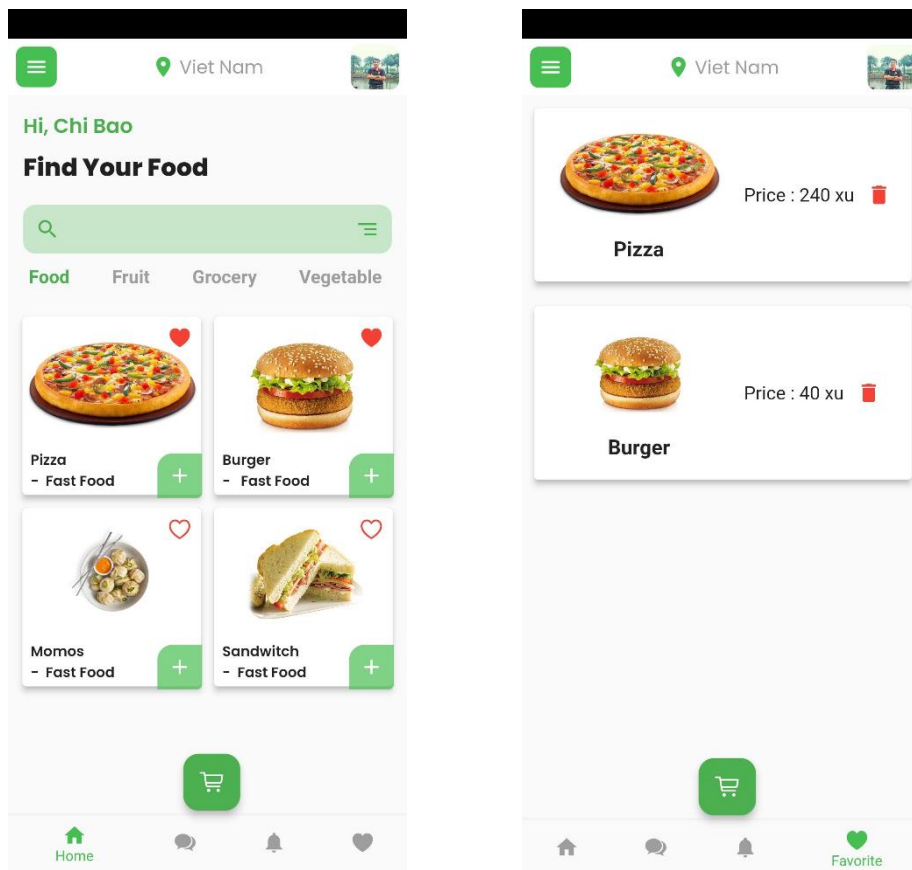
→ Ngoài ra, trong quá trình chọn hàng hóa cần mua, bạn có thể xóa hàng hóa mà bạn đã chọn → Nhấn vào → nút  chuyển thành → Xóa  khỏi giỏ hàng thành công.



Hình 3. 4. Giao diện giỏ hàng (xem - sửa - xóa) hàng hóa

3.2.4. Giao diện sản phẩm yêu thích

- Trang Home bạn có thể nhấn vào  để thêm lượt thích vào sản phẩm mà bạn ưng nhất
- Khi trang sản phẩm đó hiện  → Thêm lượt thích thành công
- Bạn nhấn vào  ở cuối màn hình để kiểm tra các hàng hóa mà bạn đã thích
- Bạn có thể xóa lượt thích khi nhấn vào 



Hình 3. 5. Giao diện yêu thích (xem - sửa - xóa) hàng hóa

3.3. Điều hướng màn hình

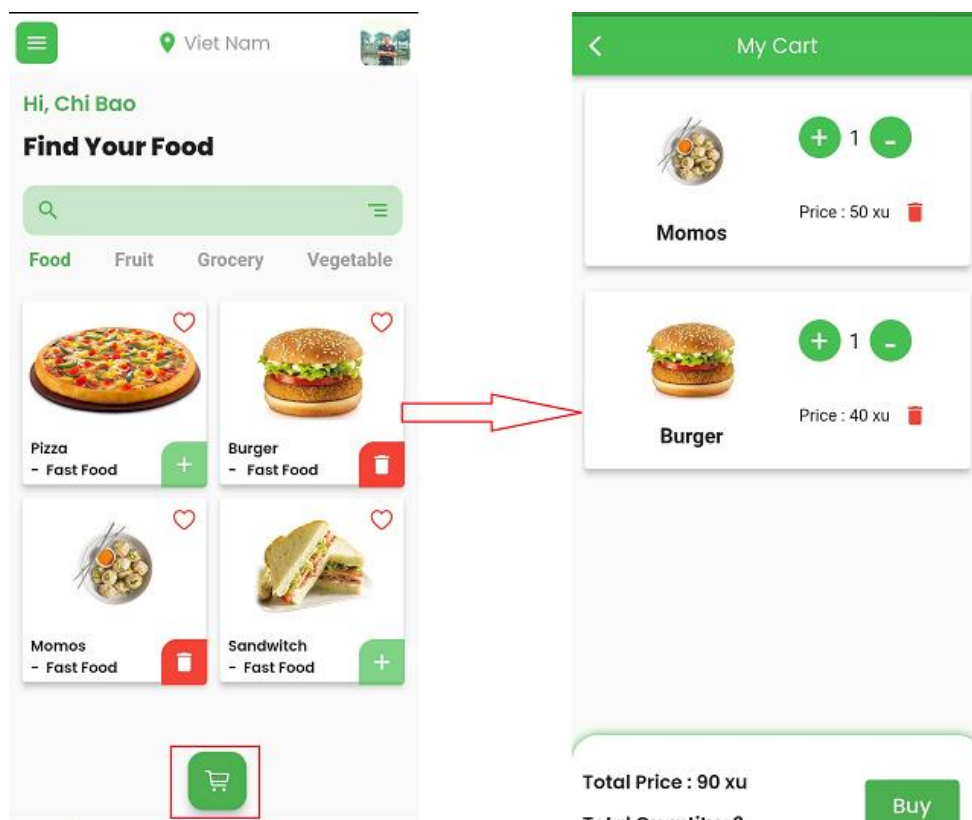
3.3.1. Home → My Cart

- ✚ Bạn mở Android Studio → Chọn Project “**food-Ordering-App-master**”
- ✚ Bạn mở file dart: `..\lib\views\screens\home_page.dart`
- ✚ Có dòng `Navigator.pushNamed(context, 'cart_screen');` → Đây là code điều hướng đến trang My Cart (cart_screen)
- ✚ Giải thích các dòng code liên quan:
 - ✓ **onPressed:** Đây là hàm xử lý sự kiện khi người dùng nhấn vào nút. Trong trường hợp này, khi nút được nhấn, ứng dụng sẽ thực hiện việc điều hướng đến màn hình "cart_screen".
 - ✓ **backgroundColor:** Đây là màu nền của nút "FloatingActionButton". Trong ví dụ này, màu nền được đặt là màu xanh lá cây.

- ✓ **shape:** Đây là hình dáng của nút. Trong trường hợp này, nút có hình chữ nhật với các góc được bo tròn.
- ✓ **child:** Đây là nội dung của nút "FloatingActionButton". Trong ví dụ này, nội dung là một biểu tượng (icon) của giỏ hàng được thể hiện bằng icon "shopping_cart" của Cupertino Icons, màu sắc của icon là màu trắng.

```
child: FloatingActionButton(  
  onPressed: () {  
    // Get.toNamed("/CartScreen");  
    Navigator.pushNamed(context, 'cart_screen');  
  },  
  backgroundColor: Colors.green,  
  shape: const RoundedRectangleBorder(  
    borderRadius: BorderRadius.all(  
      Radius.circular(15.0),  
    ), // BorderRadius.all  
  ), // RoundedRectangleBorder  
  child: const Icon(  
    CupertinoIcons.shopping_cart,  
    color: Colors.white,  
  ), // Icon  
)
```

🔗 Kết quả:

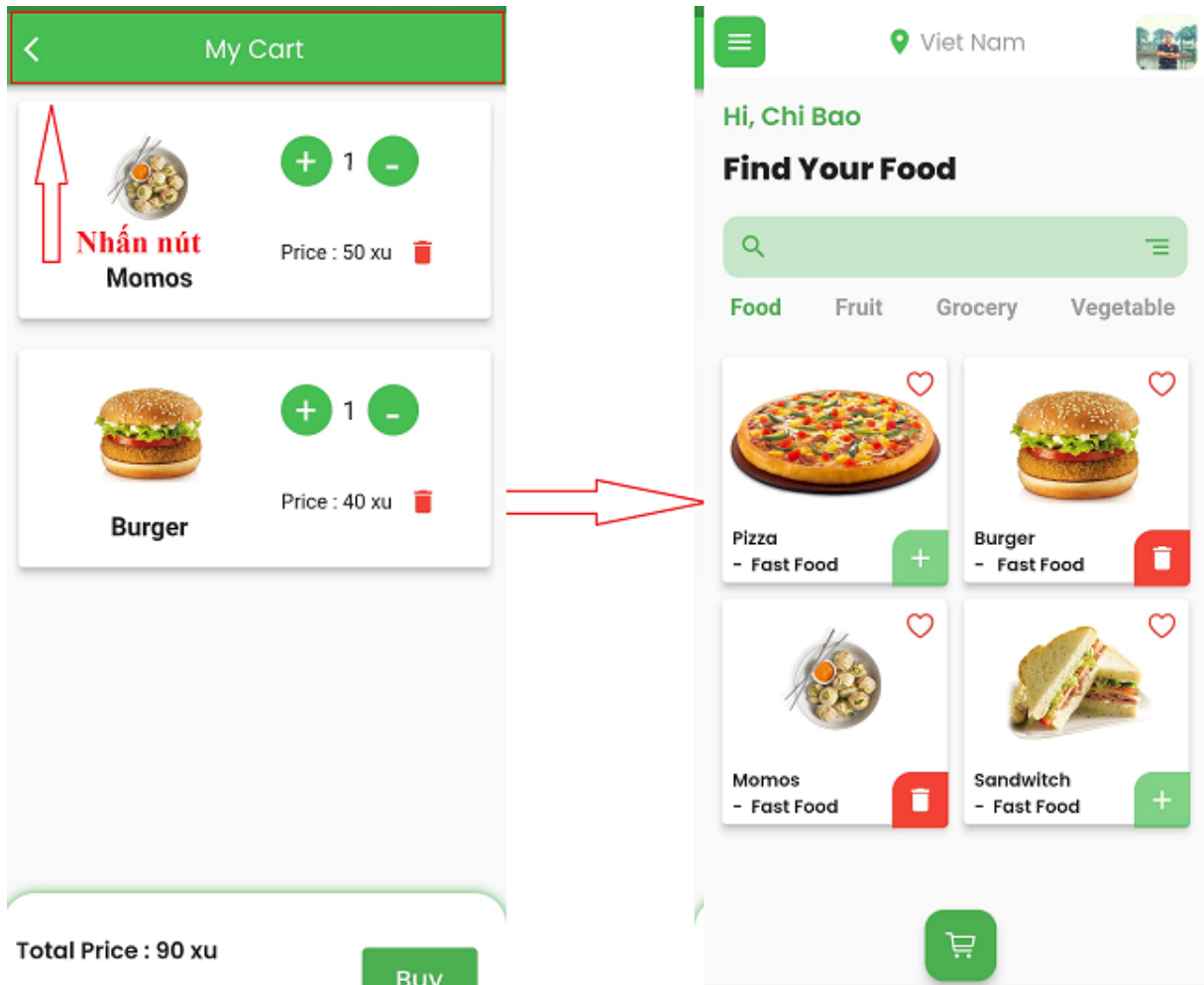


3.3.2. My Cart → Home

- + Bạn mở Android Studio → Chọn Project “**food-Ordering-App-master**”
- + Bạn mở file dart: `..\lib\views\screens\cart_screen.dart`
- + Có dòng **`Navigator.pop(context);`**; → Đây là code điều hướng đến trang trước đó “Home” (`home_page`)
- + Giải thích các dòng code liên quan:
 - ✓ **appBar:** Đây là phần định nghĩa của thanh AppBar. Thông qua việc đặt các thuộc tính trong AppBar, bạn có thể tùy chỉnh cách mà nó sẽ hiển thị trong giao diện của bạn.
 - ✓ **title:** Đây là tiêu đề của AppBar, trong trường hợp này, tiêu đề được đặt là “My Cart”. Để tạo kiểu cho tiêu đề, bạn sử dụng Google Fonts để áp dụng kiểu văn bản theo một kiểu cụ thể.
 - ✓ **centerTitle:** Thuộc tính này xác định xem tiêu đề có nên được căn giữa trên AppBar hay không. Trong trường hợp này, tiêu đề sẽ được căn giữa.
 - ✓ **backgroundColor:** Đây là màu nền của AppBar, được đặt là màu xanh lá cây (`Color(0xff48bf53)`).
 - ✓ **leading:** Đây là một phần tử hiển thị bên trái của AppBar. Trong trường hợp này, bạn sử dụng một `IconButton` để tạo một nút trở lại (nút quay lại). Khi người dùng nhấn vào nút này, `Navigator.pop(context);` được gọi để thực hiện việc quay lại trang trước đó trong ứng dụng.

```
appBar: AppBar(  
  title: Text(  
    "My Cart",  
    style: GoogleFonts.poppins(textStyle: const TextStyle()),  
  ), // Text  
  centerTitle: true,  
  backgroundColor: const Color(0xff48bf53),  
  leading: IconButton(  
    onPressed: () {  
      Navigator.pop(context);  
    },  
    icon: const Icon(Icons.arrow_back_ios),  
  ), // IconButton
```


🚦 Kết quả:



3.4. Lập trình bất đồng bộ

- 🚦 Bạn mở Android Studio → Chọn Project “**food-Ordering-App-master**”
- 🚦 Bạn mở file dart: `..\lib\helpers\db_heper.dart`
- 🚦 Các đoạn code bất đồng bộ:

```
Future<void> initDB() async {  
  var directoryPath = await getDatabasesPath();  
  String path = join(directoryPath, dbname);  
  
  db = await openDatabase(path, version: 1,  
    onCreate: (Database db, int version) async {  
    String query =  
      "CREATE TABLE IF NOT EXISTS $TableName($colId INTEGER PRIMARY  
KEY AUTOINCREMENT, $colname TEXT, $colprice NUMERIC, $colimage  
TEXT, $colquantity INTEGER, $collike TEXT)";  
    }
```

```
    await db.execute(query);  
  });
```

```
Future<int?> insertRecode({required Product data}) async {  
  await initDB();  
  
  String query =  
    "INSERT INTO  
$TableName($colname,$colprice,$colimage,$colquantity,$collike)VALUES (  
?, ?, ?, ?, ?)";  
  
  List args = [data.name, data.price, data.image, data.quantity,  
data.like];  
  int? id = await db?.rawInsert(query, args);  
  
  return id;  
}
```

```
Future<List<Product>> fetchAllRecode() async {  
  await initDB();  
  String query = "SELECT * FROM $TableName";  
  
  List<Map<String, dynamic>> data = await db!.rawQuery(query);  
  
  List<Product> allProduct =  
    data.map((e) => Product.fromMap(data: e)).toList();  
  
  return allProduct;  
}
```

```
Future<int> deleteRecode({required int id}) async {  
  await initDB();  
  String query = "DELETE FROM $TableName WHERE $colId=?";  
  List aegs = [id];  
  
  return db!.rawDelete(query, aegs);  
}
```

```
Future<int> updateRecode({required Product data, required int id})  
async {  
  await initDB();  
  String query =  
    "UPDATE $TableName SET $colname=?, $colprice=?, $colimage=?,  
$colquantity=?, $collike=? WHERE $colId=?";  
  List args = [  
    data.name,  
    data.price,  
    data.image,  
    data.quantity,  
    data.like,  
    id  
  ];  
}
```

```
return await db!.rawUpdate(query, args);  
}
```

```
Future<List<Product>> fetchSearchedRecode({required String data})  
async {  
  await initDB();  
  
  String query = "SELECT * FROM $TableName WHERE $colname LIKE  
'%$data%'";  
  
  List<Map<String, dynamic>> ProductData = await db!.rawQuery(query);  
  
  List<Product> allProduct =  
    ProductData.map((e) => Product.fromMap(data: e)).toList();  
  
  return allProduct;  
}
```

✚ Giải thích các đoạn mã bất đồng bộ:

1. **Future<void> initDB() async:** Đoạn mã này là một hàm bất đồng bộ (async) để khởi tạo cơ sở dữ liệu SQLite. Nó sử dụng `await` để đợi cho việc mở cơ sở dữ liệu và thực hiện truy vấn tạo bảng khi cần thiết.
2. **Future<int?> inserRecode({required Product data}) async:** Đoạn mã này là một hàm bất đồng bộ (async) để thêm một bản ghi mới vào cơ sở dữ liệu. Nó sử dụng `await` để đợi cho việc khởi tạo cơ sở dữ liệu trước khi thực hiện truy vấn chèn.
3. **Future<List<Product>> fetchAllRecode() async:** Đoạn mã này là một hàm bất đồng bộ (async) để lấy tất cả các bản ghi từ cơ sở dữ liệu. Nó sử dụng `await` để đợi cho việc khởi tạo cơ sở dữ liệu trước khi thực hiện truy vấn lấy dữ liệu.
4. **Future<int> deleteRecode({required int id}) async:** Đoạn mã này là một hàm bất đồng bộ (async) để xóa một bản ghi khỏi cơ sở dữ liệu dựa trên ID. Nó sử dụng `await` để đợi cho việc khởi tạo cơ sở dữ liệu trước khi thực hiện truy vấn xóa.
5. **Future<int> updateRecode({required Product data, required int id}) async:** Đoạn mã này là một hàm bất đồng bộ (async) để cập nhật một bản ghi trong cơ sở dữ liệu. Nó sử dụng `await` để đợi cho việc khởi tạo cơ sở dữ liệu trước khi thực hiện truy vấn cập nhật.

6. **Future<List<Product>> fetchSearchedRecode({required String data})**
async: Đoạn mã này là một hàm bất đồng bộ (async) để lấy các bản ghi từ cơ sở dữ liệu dựa trên một tìm kiếm. Nó sử dụng **await** để đợi cho việc khởi tạo cơ sở dữ liệu trước khi thực hiện truy vấn tìm kiếm.

CHƯƠNG IV. KẾT LUẬN VÀ KIẾN NGHỊ

4.1. Kết quả đạt được

- ✚ Đã sử dụng được ngôn ngữ lập trình Dart kết hợp với công cụ lập trình Android Studio.
- ✚ Có thêm kiến thức về lập trình Android (điều hướng và lập trình bất đồng bộ)
- ✚ Đã thiết kế và cài đặt thành công ứng dụng.

→ Vì kỹ năng lập trình còn hạn chế nên ứng dụng vẫn chưa thật sự hoàn thiện và đầy đủ chức năng như mong muốn.

4.2. Ưu điểm

- ✚ Giao dễ dễ sử dụng
- ✚ Hoàn thành các yêu cầu được giao
- ✚ Về các chức năng: thông tin hàng hóa, giỏ hàng, xem món hàng đã like đã hoàn thiện
- ✚ Hoàn thành chức năng: thêm, xóa, sửa số lượng hàng hóa trong giỏ hàng

4.3. Nhược điểm

- ✚ Do việc hạn chế về thời gian, và khả năng design giao diện còn chưa đẹp hoàn thiện nhất và còn một số sai sót trong công đoạn thiết kế.
- ✚ Chưa xây dựng được nhiều chức năng khác.

4.4. Hướng phát triển

- ✚ Trong thời gian tới em sẽ design giao diện ứng dụng đẹp hơn.
- ✚ Tiếp tục hoàn thiện các chức năng và bổ sung thêm các chức năng mới nhằm mang lại sự tiện dụng nhất cho người dùng.
- ✚ Trong tương lai, em hi vọng sẽ cố gắng hoàn thiện tốt đề tài này và cố gắng đáp ứng yêu cầu của hệ thống hơn.

4.5. Kiến nghị

Do thời gian có hạn, kinh nghiệm và vốn hiểu biết của chúng em còn nhiều hạn chế nên chương trình vẫn chưa được tối ưu. Chính vì vậy, chúng em mong thầy cô hướng dẫn cho đề tài của chúng em được tốt hơn.

TÀI LIỆU THAM KHẢO

- [1] Duy, T. T. (2020, April 6). *Bài 1. Cách cài đặt Android Studio phiên bản năm 2020*. Retrieved 8 6, 2023, from <https://duythanhcse.wordpress.com/2020/04/06/bai-1-cach-cai-dat-android-studio-phien-ban-2020/>
- [2] Hien, D. T. (2019, 12 20). *Flutter - Lập trình bắt đồng bộ với Future, async, await*. Retrieved 8 6, 2023, from <https://viblo.asia/p/flutter-lap-trinh-bat-dong-bo-voi-future-async-await-oOVIYpLaZ8W#L%C3%A0m%20Vi%E1%BB%87c%20v%E1%BB%9Bi%20Futures:%20async%20v%C3%A0%20Await>
- [3] Hoa, K. (2021, 21 Jul). *Tìm hiểu lập trình bắt đồng bộ trong Dart & Flutter*. Retrieved 8 6, 2023, from <https://200lab.io/blog/lap-trinh-bat-dong-bo-trong-dart-flutter/>
- [4] Minh, T. (2022, 08 23). *Genymotion là gì? Hướng dẫn cách tải và cách sử dụng phần mềm giả lập Android Genymotion*. Retrieved 8 6, 2023, from <https://didongviet.vn/dchannel/genymotion/>
- [5] Pham, X. V. (2023, 05). *SDK là gì-so sánh SDK và API*. Retrieved 8 6, 2023, from <https://viblo.asia/p/sdk-la-gi-so-sang-sdk-va-api-WAyK8Gv95xX>
- [6] Xuan, D. (2021, 5 3). *Tự học Flutter / Tìm hiểu về Navigation and Routing trong Flutter*. (Share To Learn More) Retrieved 8 6, 2023, from <https://cafedev.vn/tu-hoc-flutter-tim-hieu-ve-navigation-and-routing-trong-flutter/>