

NHẬN XÉT CỦA GIẢNG VIÊN

MỤC LỤC

KÍ HIỆU VIẾT TẮT.....	i
MỤC LỤC HÌNH ẢNH.....	ii
MỤC LỤC BẢNG BIÊU	v
LỜI CẢM ƠN.....	vi
LỜI MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN VỀ LẬP TRÌNH ỨNG DỤNG DI ĐỘNG....	2
1.1 Ứng dụng gốc và ứng dụng đa nền tảng.....	2
1.1.1 Ứng dụng gốc.....	2
1.1.2 Ứng dụng đa nền tảng	2
1.1.3 So sánh ứng dụng gốc và ứng dụng đa nền tảng	3
1.2 Tổng quan về Flutter.....	4
1.2.1 Giới thiệu về Flutter	4
1.2.2 Giới thiệu về ngôn ngữ Dart	5
1.3 Tổng quan về React Native.....	5
1.3.1 Giới thiệu về React Native	5
1.3.2 Giới thiệu về ngôn ngữ JavaScript.....	5
CHƯƠNG 2. SO SÁNH FLUTTER, REACT NATIVE VỚI CÁC KHUNG LÀM VIỆC KHÁC (XAMARIN, IONIC).....	6
2.1 Flutter	6
2.2 React Native	7

2.3 Xamarin.....	8
2.4 Ionic	9
2.5 Kết luận	10
CHƯƠNG 3. ÚNG DỤNG MẠNG XÃ HỘI VÀ QUẢN LÝ DÀNH CHO BỘ MÔN VẬT LÝ TIN HỌC PC DASHBOARD.....	11
3.1 Ý tưởng đề tài.....	11
3.2 Giới thiệu sơ lược.....	11
3.3 Sơ đồ tính năng	12
3.4 Hệ thống máy chủ và cơ sở dữ liệu	13
3.4.1 Kết nối ứng dụng với máy chủ thông qua API	13
3.4.2 Truy cập máy chủ với tư cách quản trị viên.....	16
3.5 Mô tả chi tiết ứng dụng	21
3.5.1 Giao diện màn hình chờ	21
3.5.2 Giao diện màn hình đăng nhập và lấy lại mật khẩu	22
3.5.3 Giao diện màn hình điều khiển	23
3.5.4 Giao diện ngăn công cụ và các chức năng	24
3.5.5 Giao diện màn hình bộ môn	28
3.5.6 Giao diện màn hình lớp học, đăng bài và bình luận.....	28
3.5.7 Giao diện màn hình trò chuyện	31
3.5.8 Giao diện màn hình quản lý thông tin sinh viên và giảng viên.....	31
3.5.9 Giao diện màn hình quản lý học tập và giảng dạy	32
KẾT LUẬN.....	34
TÀI LIỆU THAM KHẢO.....	35

KÍ HIỆU VIẾT TẮT

API	Application Programming Interface
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
XML	eXtensible Markup Language
JSON	JavaScript Object Notation
REST	Representational State Transfer
OOP	Object Oriented Programming
SQL	Structured Query Language
UI	User Interface
UX	User Experience
IDE	Integrated Development Environment
QR	Quick Response
HTTP	Hypertext Transfer Protocol

MỤC LỤC HÌNH ẢNH

Hình 1.1: Ứng dụng đa nền tảng và ứng dụng gốc	3
Hình 1.2: Biểu tượng của Flutter	4
Hình 1.3: Biểu tượng của React Native	5
Hình 2.1: Biểu tượng của Xamarin	8
Hình 2.2: Biểu tượng của Ionic.....	9
Hình 3.1: Sơ đồ cấu trúc chung của ứng dụng	12
Hình 3.2: Sơ đồ chức năng của ngăn công cụ.....	12
Hình 3.3: Sơ đồ chức năng của bảng điều khiển	13
Hình 3.4: Sơ đồ gửi yêu cầu GET từ ứng dụng đến máy chủ.....	14
Hình 3.5: Sơ đồ gửi yêu cầu POST từ ứng dụng đến máy chủ.....	15
Hình 3.6: Sơ đồ gửi yêu cầu PUT từ ứng dụng đến máy chủ	15
Hình 3.7: Sơ đồ gửi yêu cầu DELETE từ ứng dụng đến máy chủ	16
Hình 3.8: Khởi chạy máy chủ cục bộ	17
Hình 3.9: Truy cập máy chủ đám mây.....	17
Hình 3.10: Chứng chỉ xác thực để truy cập cơ sở dữ liệu đám mây.....	18

Hình 3.11: Chứng chỉ xác thực để truy cập cơ sở dữ liệu cục bộ	18
Hình 3.12: Giao diện quản lý người dùng của quản trị viên.....	19
Hình 3.13: Giao diện quản lý bài đăng của quản trị viên	19
Hình 3.14: Giao diện trang chủ của máy chủ và hướng dẫn API	20
Hình 3.15: Hàm gọi API với Dio trong phiên bản sinh viên	20
Hình 3.16: Hàm gọi API với Axios trong phiên bản giảng viên	21
Hình 3.17: Biểu tượng của ứng dụng PC Dashboard	21
Hình 3.18: Giao diện màn hình chờ.....	22
Hình 3.19: Giao diện màn hình đăng nhập	22
Hình 3.20: Giao diện màn hình lấy lại mật khẩu	23
Hình 3.21: Giao diện bảng điều khiển	24
Hình 3.22: BottomNavigationBar của phiên bản sinh viên	24
Hình 3.23: Giao diện ngăn công cụ	25
Hình 3.24: Drawer của phiên bản sinh viên.....	25
Hình 3.25: Giao diện màn hình trang chủ.....	26
Hình 3.26: Giao diện màn hình cập nhật thông tin	26

Hình 3.27: Giao diện màn hình thay đổi mật khẩu	27
Hình 3.28: Giao diện màn hình nhà phát triển.....	27
Hình 3.29: Giao diện màn hình bộ môn.....	28
Hình 3.30: Giao diện màn hình lớp học.....	28
Hình 3.31: Giao diện màn hình đăng bài	29
Hình 3.32: Giao diện sửa và xóa bài đăng	29
Hình 3.33: Giao diện màn hình bình luận.....	30
Hình 3.34: Giao diện sửa và xoá bình luận.....	30
Hình 3.35: Giao diện màn hình trò chuyện phiên bản sinh viên	31
Hình 3.36: Giao diện màn hình danh sách sinh viên và giảng viên.....	31
Hình 3.37: Giao diện trang thông tin liên lạc	32
Hình 3.38: Giao diện màn hình thời khoá biểu.....	32
Hình 3.39: Giao diện màn hình thi cử phiên bản sinh viên	33

MỤC LỤC BẢNG BIỂU

Bảng 1.1: So sánh ứng dụng gốc và ứng dụng đa nền tảng4

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến các thầy cô bộ môn Vật lý Tin học đã tận tình giảng dạy, truyền đạt kiến thức cho em trong suốt những năm học vừa qua. Bên cạnh đó, em cũng xin chân thành cảm ơn giáo viên hướng dẫn, thầy Nguyễn Anh Huy và cô Nguyễn Anh Thư, đã tận tình hỗ trợ, chia sẻ kinh nghiệm cho em trong quá trình làm khóa luận tốt nghiệp.

Đề tài em được giao là một sự trải nghiệm mới mẻ, giúp em củng cố được những kiến thức cũng như có thêm được những kinh nghiệm bổ ích về Lập trình Di động, để có thể áp dụng vào đời sống thực tế cũng như công việc sau này. Do thời gian gấp rút và kiến thức còn hạn chế, nên chắc chắn sẽ không tránh khỏi sai sót. Vì vậy, em rất mong nhận được những lời góp ý và chỉ dẫn từ thầy cô để có thể hoàn thiện sản phẩm và tiếp tục phát triển đề tài sau này.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 24 tháng 07 năm 2020

Sinh viên thực hiện

Bùi Ngô Tôn Bách

LỜI MỞ ĐẦU

Với sự bùng nổ và phát triển của công nghệ hiện nay, điện thoại thông minh ngày càng được ưa chuộng bởi tính linh hoạt và tiện dụng, bao gồm đầy đủ các tính năng của một chiếc máy tính. Điều đó khiến người dùng có thể sử dụng mọi lúc mọi nơi để đáp ứng nhu cầu xem tin tức, liên lạc, giải trí, ... Do đó, điện thoại di động trở nên phổ biến trong thời đại 4.0. Nhận thấy được thực trạng đó, tôi chọn đề tài “Xây dựng ứng dụng mạng xã hội và quản lý dành cho bộ môn Vật lý Tin học”. Mục đích là tạo một kênh chung để liên lạc, kết nối giữa sinh viên và giảng viên, giúp hỗ trợ việc việc quản lý sinh viên, cũng như việc quản lý học tập và giảng dạy.

Đề tài có sự kết hợp giữa lập trình căn bản, lập trình hướng đối tượng, cơ sở dữ liệu, lập trình di động và các công nghệ mới, các thư viện hỗ trợ cho lập trình di động. Ngoài ra, tôi đã phải tìm hiểu và khảo sát các nhu cầu và xu hướng thiết kế giao diện và chức năng của các mạng xã hội lớn hiện nay trên thế giới, khảo sát về nhu cầu sử dụng một mạng xã hội riêng trong bộ môn Vật lý Tin học để có thể dễ dàng triển khai cũng như đưa ra giải pháp phù hợp nhất cho việc xây dựng một ứng dụng di động hoàn thiện, dễ sử dụng và thân thiện với người dùng. Nội dung khoá luận gồm ba chương:

- Chương 1: Tổng quan về lập trình ứng dụng di động
- Chương 2: So sánh Flutter, React Native với các khung làm việc khác (Xamarin, Ionic)
- Chương 3: Ứng dụng mạng xã hội và quản lý dành cho bộ môn Vật lý Tin học PC Dashboard

CHƯƠNG 1. TỔNG QUAN VỀ LẬP TRÌNH ỨNG DỤNG DI ĐỘNG

Trong cuộc sống hiện đại, thiết bị di động với ưu thế nhỏ gọn nhưng tích hợp cả một thế giới làm việc và giải trí đã trở thành người bạn đồng hành thân thiết của hầu hết mọi người, đặc biệt là giới trẻ năng động. Với sự tăng trưởng mạnh mẽ của thị trường này, không chỉ các hàng sản xuất di động mà cả các nhà phân phối và những doanh nghiệp công nghệ khác cũng chen chân vào cuộc đua phát triển ứng dụng di động để khai thác hết các tính năng của thiết bị di động. Trên thị trường ứng dụng di động hiện nay, hai hệ điều hành chiếm thị phần cao nhất là Android và iOS. Trong chương 1 này, tôi sẽ trình bày về xu hướng lập trình ứng dụng di động hiện nay.

1.1. Ứng dụng gốc và ứng dụng đa nền tảng

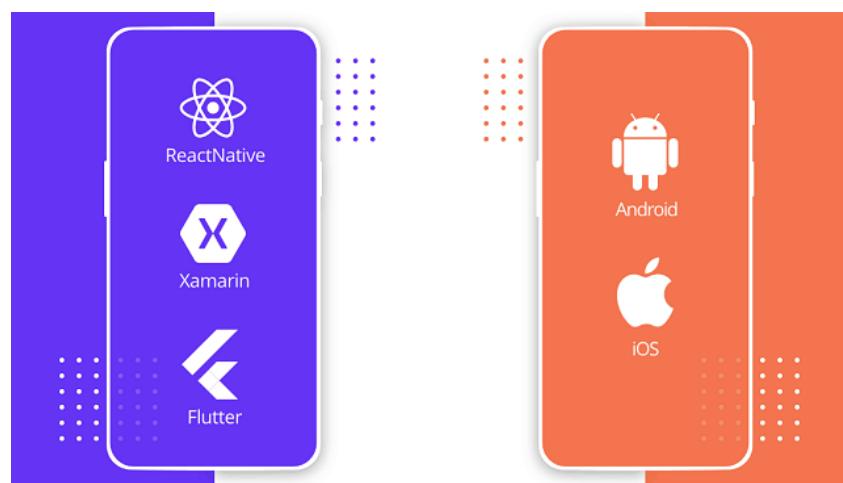
1.1.1. Ứng dụng gốc

Ứng dụng gốc là ứng dụng được phát triển theo ngôn ngữ lập trình dành riêng cho từng nền tảng, với Java, Kotlin dành cho Android và Objective-C, Swift dành cho iOS. Điều này khiến các ứng dụng đó chỉ tương thích với một nền tảng tương ứng và cho phép nhà phát triển tối ưu hóa hoàn toàn chúng để đạt hiệu suất tối đa cũng như mang lại trải nghiệm tốt nhất cho người dùng. Tuy nhiên, ứng dụng gốc vẫn là một phần không thể thiếu của lập trình ứng dụng di động, vẫn là phần cốt lõi của ứng dụng đa nền tảng.

1.1.2. Ứng dụng đa nền tảng

Ứng dụng đa nền tảng có thể cài đặt được trên điện thoại, máy tính, Web và các nền tảng khác, cho phép viết một lần nhưng có thể chạy được trên rất nhiều nền tảng, điều này giúp tiết kiệm được chi phí và thời gian phát triển. Việc bảo trì các ứng dụng đa nền tảng cũng dễ dàng hơn so với các ứng dụng gốc. Một số khung đa nền tảng nổi tiếng như React Native của Facebook, Flutter của Google và Xamarin của Microsoft. Trong những năm gần đây, lập trình ứng dụng di động có sự thay đổi đáng kể nhờ vào sự phát triển của ứng dụng đa nền tảng.

1.1.3. So sánh ứng dụng gốc và ứng dụng đa nền tảng



Hình 1.1: Ứng dụng đa nền tảng và ứng dụng gốc

	Ứng dụng gốc	Ứng dụng đa nền tảng
Ngôn ngữ	Ngôn ngữ riêng tùy theo từng nền tảng, với Java/ Kotlin cho Android và Objective-C/ Swift cho iOS.	Ngôn ngữ chung cho các nền tảng và tùy theo khung được sử dụng, với Dart cho Flutter, JavaScript/ TypeScript cho React Native, C# cho Xamarin.
Khả năng tái sử dụng mã	Mã nguồn được sử dụng riêng cho từng nền tảng, không thể sử dụng lại trên nền tảng khác.	Mặc dù dùng mỗi nền tảng sẽ có những thiết lập riêng, nhưng nhìn chung, chỉ cần viết một lần, mã nguồn có thể chạy được trên nhiều nền tảng khác nhau, như Android, iOS, Windows, MacOS, Linux, Web.

Giao diện đồ họa	Giao diện đẹp nhưng phụ thuộc nhiều vào nền tảng tương ứng nên rất khó để đồng bộ giữa các nền tảng. Mỗi nền tảng sẽ có sự khác biệt về giao diện nên việc thiết kế sẽ khó khăn.	Giao diện đẹp và ít phụ thuộc vào nền tảng nên dễ dàng đồng bộ giữa các nền tảng. Điều này khiến cho việc thiết kế trở nên dễ dàng hơn vì chỉ cần duy nhất một giao diện chung.
Hiệu năng	Hiệu năng cao và trải nghiệm người dùng tốt. Dễ dàng truy cập vào phần cứng của thiết bị.	Hiệu năng cao hay thấp tùy theo khung được sử dụng, nhưng nhìn chung đều thấp hơn của ứng dụng gốc.
Chi phí, thời gian và nguồn lực	Chi phí cao, thời gian dài và yêu cầu nguồn lực lớn vì mỗi nền tảng cần một bộ mã nguồn riêng.	Chi phí thấp, thời gian ngắn và chỉ cần nguồn lực nhỏ vì chỉ cần một bộ mã nguồn duy nhất để sử dụng cho tất cả các nền tảng

Bảng 1.1: So sánh ứng dụng gốc và ứng dụng đa nền tảng

1.2. Tổng quan về Flutter

1.2.1. Giới thiệu về Flutter

Flutter là một bộ công cụ phát triển giao diện người dùng cho phần mềm, mã nguồn mở và được phát triển bởi Google. Flutter được sử dụng để phát triển các ứng dụng cho Android, iOS, Windows, MacOS, Linux, Fuchsia và Web [4].



Hình 1.2: Biểu tượng của Flutter

1.2.2. Giới thiệu về ngôn ngữ Dart

Các ứng dụng Flutter được viết bằng ngôn ngữ Dart. Dart là một ngôn ngữ lập trình được tối ưu hóa cho các ứng dụng trên nhiều nền tảng. Dart được Google phát triển và được sử dụng để xây dựng các ứng dụng di động, máy tính, máy chủ, Web và thiết bị nhúng. Dart là một ngôn ngữ hướng đối tượng dựa trên lớp, có thể biên dịch thành mã gốc hoặc JavaScript. Dart cũng hỗ trợ các giao diện, mixin, lớp ảo, ... [9]

1.3. Tổng quan về React Native

1.3.1. Giới thiệu về React Native

React Native là một khung ứng dụng di động, mã nguồn mở được phát triển bởi Facebook. React Native được sử dụng để phát triển các ứng dụng cho Android, iOS và Web bằng cách cho phép các nhà phát triển sử dụng React cùng với các thành phần của nền tảng gốc. React (còn được gọi là React.js hay ReactJS) là một thư viện JavaScript dùng để xây dựng giao diện người dùng. React được phát triển bởi Facebook cùng cộng đồng gồm các nhà phát triển và các công ty [4].

1.3.2. Giới thiệu về ngôn ngữ JavaScript

JavaScript là một ngôn ngữ lập trình bậc cao, hướng đối tượng và đa mô hình, hỗ trợ các kiểu lập trình theo hướng sự kiện, chức năng, ... Bên cạnh HTML và CSS, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Mặc dù có sự tương đồng giữa JavaScript và Java, bao gồm tên ngôn ngữ, cú pháp và các thư viện tiêu chuẩn tương ứng, nhưng hai ngôn ngữ này rất khác nhau. JavaScript còn có một phiên bản nâng cấp là ngôn ngữ TypeScript, với hướng tiếp cận thuận hướng đối tượng và có sự tương đồng với Java hơn [10].



Hình 1.3: Biểu tượng của React Native

CHƯƠNG 2. SO SÁNH FLUTTER, REACT NATIVE VỚI CÁC KHUNG LÀM VIỆC KHÁC (XAMARIN, IONIC)

Việc phát triển ứng dụng di động gốc mang đến những ứng dụng hoàn thiện và hiệu năng cao. Vai trò của ứng dụng gốc trong lập trình di động là điều không thể phủ nhận. Thế nhưng con đường phát triển theo hướng này luôn đi kèm với sự tốn kém về thời gian và nguồn lực vì cùng một mã phải viết nhiều lần. Giải pháp để hoàn thành công việc một cách nhanh chóng mà không cần đầu tư nhiều nguồn lực chính là phát triển ứng dụng di động đa nền tảng. Hướng đi này cho phép các nhà phát triển viết mã một lần và áp dụng một vài phần của nó trên tất cả các nền tảng để tiết kiệm thời gian phát triển. Đó được coi là xu hướng và tương lai của lập trình di động. Tuy nhiên, các ứng dụng theo hướng này cung cấp hiệu suất kém hơn khi so sánh với các ứng dụng gốc được phát triển bằng ngôn ngữ gốc, mặc dù giao diện có nhiều sự đột phá. Để loại bỏ sự khác biệt giữa ứng dụng di động gốc và đa nền tảng, cộng đồng phát triển ứng dụng đang nỗ lực phát triển các khung và công cụ mới như Xamarin, React Native, Ionic và Flutter. Chúng đều là những khung đa nền tảng mạnh mẽ, được cung cấp bởi những công ty lớn.. Trong chương 2 này, tôi sẽ so sánh ưu và nhược điểm của bốn khung đa nền tảng phổ biến nhất hiện nay. Từ đó, tôi sẽ rút ra kết luận lí do sử dụng Flutter và React Native để thực hiện đề tài khoá luận này.

2.1. Flutter

Flutter sử dụng Dart để phát triển các ứng dụng chất lượng cao cho Android, iOS và Web. Dart là một ngôn ngữ lập trình cung cấp nhiều tiện ích và dựa trên C/C++ và Java. Dart là một trong những lý do để nhiều nhà phát triển ứng dụng ưu thích việc phát triển ứng dụng Flutter. Khi so sánh trên cơ sở hiệu năng của ứng dụng, Flutter đứng đầu so với các khung khác. Việc Flutter có lợi thế của Dart và không có cầu nối JavaScript để tương tác với các thành phần gốc của thiết bị khiến cho hiệu năng mà Flutter cung cấp là rất cao. Flutter cung cấp các giao diện người dùng tốt nhất. Mặc dù Ionic và Xamarin cũng cung cấp các ứng dụng đa nền tảng nhưng hiệu

quả và hiệu suất của chúng không thể so với Flutter và React Native. Xamarin và Ionic bị chậm, thiếu khả năng phản hồi nếu ứng dụng nặng và quá nhiều thành phần giao diện gốc được sử dụng. Hiện tại, Flutter là một khung mới trong cộng đồng và không quá phổ biến. Mặc dù Flutter vẫn có một số khuyết điểm nhưng việc sử dụng Flutter vẫn được đánh giá cao và có thể nhanh chóng chuyển từ ý tưởng sang nguyên mẫu rồi sang ứng dụng. Khi tương tác với thành phần gốc, người dùng thường có các chức năng có sẵn để sử dụng, giúp tăng tốc độ phát triển. Trong Flutter, người dùng cần thêm các tệp chuyên dụng cho cả nền tảng Android và iOS tùy thuộc vào quy tắc của mỗi nền tảng. Flutter là một khung nguồn mở. Các kỹ sư có thể sử dụng các hệ thống và thư viện miễn phí.

2.2. React Native

React Native sử dụng Javascript - là một trong những ngôn ngữ lập trình cấp cao, đa năng và phổ biến nhất. React Native kết hợp các ưu điểm của JavaScript và ReactJS, đồng thời được Facebook hỗ trợ. Ưu điểm của React Native là cho phép viết một vài thành phần trong Java, Kotlin hoặc Objective-C, Swift mỗi khi các nhà phát triển cần. Với việc sử dụng các mô-đun gốc và thư viện trong các ứng dụng React Native, người dùng có thể quản lý các hoạt động nặng về tính toán như xử lý video hoặc chỉnh sửa hình ảnh. Hiệu năng mà React Native cung cấp rất gần với các ứng dụng gốc vì React Native hiển thị các thành phần mã cụ thể cho các API gốc. React cũng cho phép các nhà phát triển sử dụng các mô-đun gốc được viết bằng ngôn ngữ gốc để viết mã cho các hoạt động phức tạp. Tuy nhiên, phần mã này không thể được sử dụng lại trên hai nền tảng và mục đích chính của chúng chỉ là cung cấp hiệu suất cao hơn. Các mô-đun React Native liên kết với các bộ điều khiển giao diện gốc, mang lại trải nghiệm tuyệt vời cho người dùng và rất gần với các ứng dụng gốc. React Native cũng sử dụng thư viện React với các thành phần giao diện mở rộng và hợp lý hóa việc phát triển giao diện. React Native đang đứng đầu về sự công nhận của thị trường và cộng đồng. Cộng đồng các nhà phát triển của React Native đang phát triển nhanh chóng và cho đến nay có nhiều kỹ sư React Native giàu kinh nghiệm. Điều này giúp dễ dàng khởi động một dự án React. React Native sử dụng một thư viện cực kì

phổ biến (React) và ngôn ngữ cần thiết nhất cho việc phát triển Web (JavaScript), đồng thời cũng cung cấp các ứng dụng gốc thực sự. Những điều này làm cho React Native trở thành một nền tảng mạnh mẽ và nổi tiếng. Khung này sử dụng các thành phần gốc được viết bằng Java, Kotlin hoặc Objective-C, Swift để nâng cao hiệu suất ứng dụng. Thế nhưng các thành phần gốc này không thể được sử dụng lại trên các nền tảng khác. Vì vậy, các nhà phát triển cần thực hiện một vài thao tác để thay đổi cơ sở mã chuyên biệt này. Tuy nhiên, React Native chấp nhận các thành phần gốc này hay bất cứ thứ gì còn lại của cơ sở mã có thể được sử dụng lại. React Native là một khung nguồn mở. Các kỹ sư có thể sử dụng các hệ thống và thư viện miễn phí.

2.3. Xamarin



Hình 2.1: Biểu tượng của Xamarin

Xamarin sử dụng C# với môi trường .NET để phát triển các ứng dụng Android, iOS và Mac. Bất cứ tính năng gì có thể thực hiện bằng ngôn ngữ gốc, nhà phát triển đều có thể làm bằng C# trong Xamarin. Tuy rằng các nhà phát triển không thể sử dụng các thư viện nguồn mở cho iOS và Android bằng Xamarin, nhưng có rất nhiều thư viện .NET có thể truy cập được để thay thế. Hiệu năng của Xamarin cũng được xem là gần với ứng dụng gốc. Xamarin có hai cách để xây dựng các ứng dụng di động là Xamarin.Android/ Xamarin.iOS và Xamarin.Forms. Các ứng dụng Xamarin.Android/ Xamarin.iOS hoạt động giống như ứng dụng gốc vì khả năng đa nền tảng của chúng chủ yếu tập trung vào việc chia sẻ môi trường thay vì cơ sở mã. Việc này giúp đạt được hiệu năng gốc, điều mà không thể thực hiện được với các giải pháp diễn giải mã khi chạy. Hướng Xamarin.Forms thì lại tập trung vào việc chia sẻ mã rộng với ít hành vi nền tảng cụ thể hơn. Việc này hoàn toàn làm giảm hiệu suất mã trong nhiều hoạt động so với các nền tảng khác nhau. Xamarin cho phép tạo giao

diện theo hai cách khác nhau là sử dụng Xamarin.Android/ Xamarin.iOS hoặc Xamarin.Forms. Với Xamarin.Android/ Xamarin.iOS, người dùng sẽ mất nhiều thời gian nhưng đảm bảo giao diện gốc về trải nghiệm. Với Xamarin.Forms, người dùng có thể làm cho quá trình phát triển nhanh hơn đáng kể và tiết kiệm được nhiều tài nguyên nhưng mất đi giao diện gốc. Xamarin.Forms có thể là một giải pháp tốt cho các dự án nội bộ và doanh nghiệp mà phần giao diện không quá cần thiết như trong các ứng dụng cộng đồng. Xamarin cũng là một khung khá phổ biến. Microsoft vẫn đang nỗ lực rất nhiều để phát triển cộng đồng Xamarin. Các nhà phát triển làm việc bên trong hệ sinh thái Microsoft có thể làm việc với sự hỗ trợ của Microsoft mà không cần phải thay đổi công nghệ. Xamarin yêu cầu chuyển đổi giữa các môi trường phát triển và tất cả các ứng dụng Xamarin được phát triển trong Visual Studio. Thông thường, có thể sử dụng lại mã nguồn với Xamarin.Forms để tăng tốc quá trình phát triển. Xamarin yêu cầu các nhà phát triển cài đặt IDE, được cung cấp trên cơ sở đăng ký. Tuy nhiên, Xamarin cũng cung cấp một phiên bản miễn phí cho sinh viên và liên doanh phi doanh nghiệp với tối đa năm người dùng.

2.4. Ionic



Hình 2.2: Biểu tượng của Ionic

Ionic sử dụng HTML5, CSS và JavaScript để phát triển các ứng dụng và sử dụng môi trường Cordova để truy cập các bộ điều khiển nền tảng gốc. Với việc sử dụng Ionic, người dùng cũng có thể sử dụng TypeScript để cải thiện chất lượng mã. Xét riêng về hiệu năng thì Ionic kém nhất. Hiệu năng của Ionic không gần với ứng dụng gốc như Xamarin, React Native hoặc Flutter cung cấp vì Ionic sử dụng các công nghệ Web để hiển thị một ứng dụng. Hướng đi này làm giảm đáng kể tốc độ. Ngoài ra, việc phát triển ứng dụng Ionic không sử dụng các thành phần gốc và chỉ có gắng tạo giao diện gốc bằng cách sử dụng các công nghệ Web. Giao diện Ionic hoàn toàn

không sử dụng các thành phần gốc và mọi thứ đều hiển thị trong HTML và CSS. Sau đó, Ionic áp dụng Cordova để cung cấp trải nghiệm di động. Các thành phần Angular đi kèm khung cũng cho phép các ứng dụng Ionic trông giống như các ứng dụng gốc. Đây là khung phổ biến thứ hai sau React. Ionic cho phép cho các nhà phát triển xây dựng các ứng dụng di động gốc theo cách nhanh nhất có thể. Một phần thiết yếu của các ứng dụng được phát triển trong Ionic là tính phổ biến của chúng. Dù hệ điều hành khác nhau nhưng Ionic vẫn sẽ hoạt động tốt tương tự trên mỗi hệ điều hành. Tuy nhiên, một số thành phần giao diện cần được thay đổi theo các quy tắc được quy định bởi một nền tảng cụ thể. Ionic là một khung nguồn mở miễn phí để phát triển các ứng dụng di động đa nền tảng. Phiên bản chuyên nghiệp có trả phí sẽ giúp đẩy nhanh quá trình phát triển hơn.

2.5. Kết luận

Cả bốn khung là Flutter, React Native, Xamarin và Ionic đều giúp giảm thời gian và chi phí phát triển ứng dụng. Mỗi khung đều hữu ích tùy thuộc vào sự ưu tiên và nhu cầu khi phát triển. Thế nhưng hiện nay, với sự cập nhật từ Google và Facebook, Flutter và React Native dần chiếm ưu thế và trở thành hai khung mạnh mẽ nhất. Flutter thì có cách tiếp cận độc đáo và hiệu năng cao nhờ vào việc biên dịch trực tiếp từ mã nguồn thành mã máy. Còn React Native thì đã khẳng định được vị trí trong lập trình một thời gian dài với cộng đồng lập trình viên đông đảo, sẵn sàng hỗ trợ và sửa lỗi. Dù vẫn còn một vài hạn chế nhưng nhìn chung, hai khung này đã, đang và sẽ là xu thế của lập trình di động nói chung, cũng như lập trình đa nền tảng nói riêng. Đó cũng là lí do để tài khoá luận chọn sử dụng hai khung này. Việc áp dụng công nghệ mới vào sản phẩm giúp cho việc triển khai và bảo trì trở nên thực tế và dễ dàng hơn, làm cho vòng đời sản phẩm lâu dài hơn.

CHƯƠNG 3. ÚNG DỤNG MẠNG XÃ HỘI VÀ QUẢN LÝ DÀNH CHO BỘ MÔN VẬT LÝ TIN HỌC PC DASHBOARD

Mạng xã hội một cụm từ không còn quá xa lạ đối với con người. Ngày nay hầu như không có ai là không sử dụng cho mình một mạng xã hội. Việc kết nối giữa các cá nhân và các tổ chức đã trở nên dễ dàng hơn thông qua mạng xã hội, giúp những thông tin từ tổ chức có thể đến được với cá nhân một cách nhanh chóng nhất. Ngoài ra, việc tích hợp thêm khả năng quản lý của các mạng xã hội lớn gần đây giúp cho tổ chức có thể quản lý được cá nhân, khiến cho mạng xã hội không chỉ là nơi để chia sẻ, mà còn là nơi để làm việc. Trong chương 3 này, tôi sẽ trình bày về sản phẩm thực tế của đề tài khoá luận.

3.1. Ý tưởng đề tài

Xuất phát từ thực trạng việc trao đổi thông tin và quản lý học tập, giảng dạy giữa các sinh viên và giảng viên còn hạn chế do thiếu một kênh chung để kết nối và liên lạc, ứng dụng PC Dashboard ra đời nhằm trở thành cầu nối để sinh viên và giảng viên có thể sử dụng. Ứng dụng như một mạng xã hội nhỏ dành cho bộ môn Vật lý Tin học, với các chức năng như trò chuyện, đăng tải tin tức, bình luận, xem tin tức từ bộ môn, lớp học, lịch học, lịch dạy cũng như kết quả thi.

3.2. Giới thiệu sơ lược

Ứng dụng PC Dashboard có hai phiên bản khác nhau:

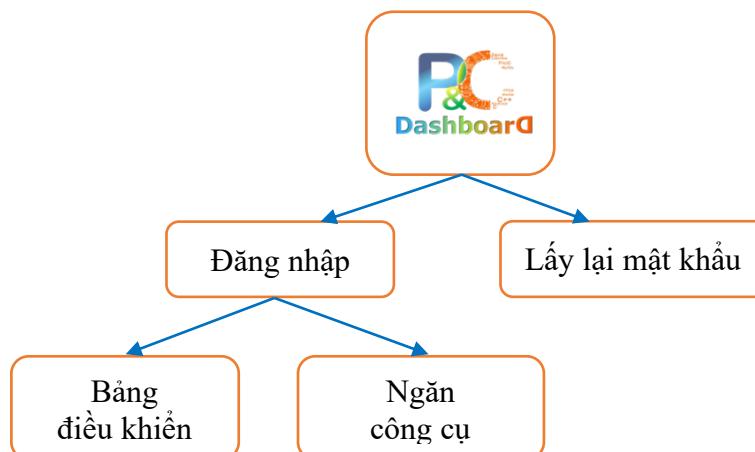
- Phiên bản dành riêng cho sinh viên được viết bằng Flutter, có thể cài đặt và sử dụng trên cả hai hệ điều hành Android và iOS. Với phiên bản này, ứng dụng tập trung vào việc kết nối, chia sẻ giữa các sinh viên với tính năng trò chuyện nhóm, đăng bài lớp và bình luận. Ngoài ra, sinh viên cũng có thể quản lý học tập và nhận tin tức từ bộ môn.

- Phiên bản dành riêng cho giảng viên được viết bằng React Native, có thể cài đặt và sử dụng trên cả hai hệ điều hành Android và iOS. Với phiên bản này, ứng dụng tập trung vào việc quản lý thông tin sinh viên để giảng viên có thể dễ dàng liên lạc

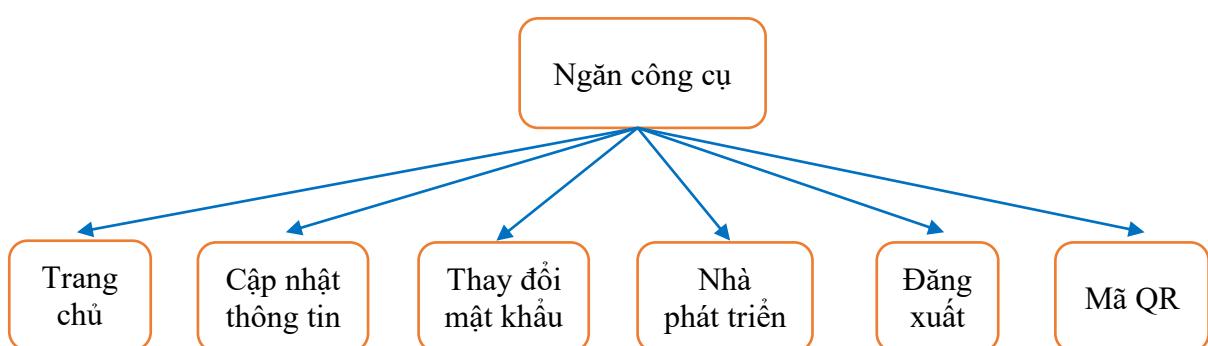
với sinh viên, đăng bài trên lớp và bình luận về bài viết. Ngoài ra, giảng viên cũng có thể quản lý giảng dạy và nhận tin tức từ bộ môn.

3.3. Sơ đồ tính năng

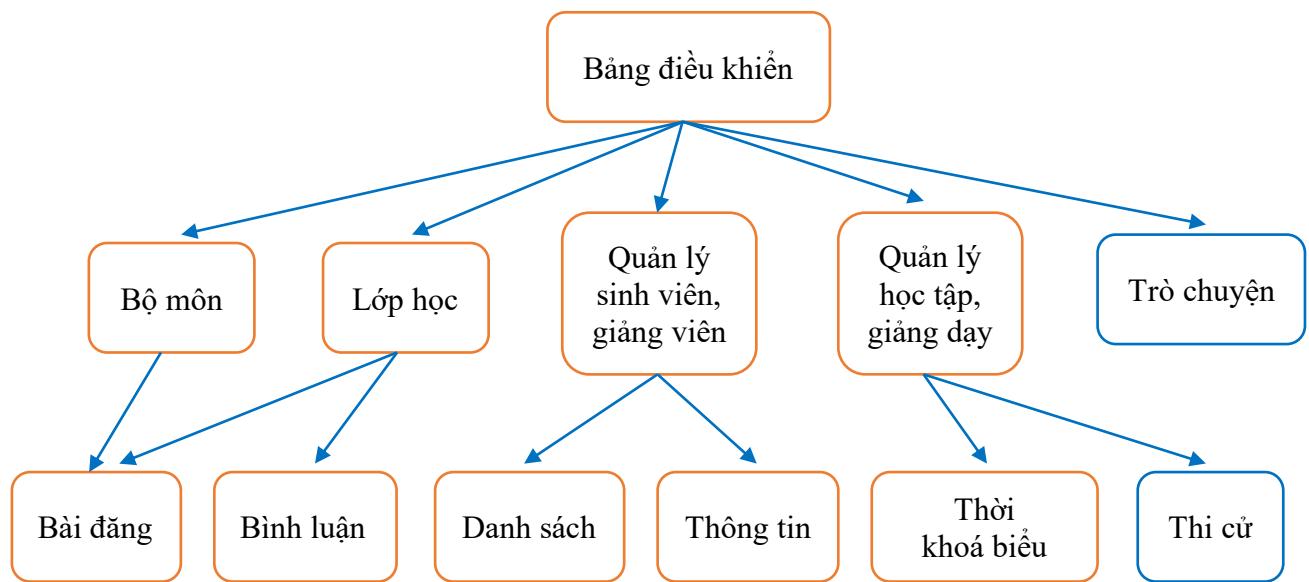
Ứng dụng mạng xã hội và quản lý dành cho bộ môn Vật lý Tin học PC Dashboard có các tính năng như hình dưới đây (Hình 3.1, Hình 3.2, Hình 3.3). Ở lần mở ứng dụng đầu tiên, sẽ có hai lựa là Đăng nhập và Lấy lại mật khẩu (Hình 3.1). Sau khi đăng nhập thành công, màn hình bảng điều khiển và ngăn công cụ sẽ xuất hiện. Trong ngăn công cụ có các mục Trang chủ, Cập nhật thông tin, Thay đổi mật khẩu, Nhà phát triển, Đăng xuất và mã QR (Hình 3.2). Trên bảng điều khiển có các mục Bộ môn, Lớp học (dùng để xem bài đăng và bình luận), Quản lý sinh viên, giảng viên (dùng để xem thông tin liên lạc), Quản lý học tập, giảng dạy (dùng để xem thời khoá biểu và điểm thi) và Trò chuyện (dành riêng cho phiên bản sinh viên).



Hình 3.1: Sơ đồ cấu trúc chung của ứng dụng



Hình 3.2: Sơ đồ chức năng của ngăn công cụ



Hình 3.3: Sơ đồ chức năng của bảng điều khiển

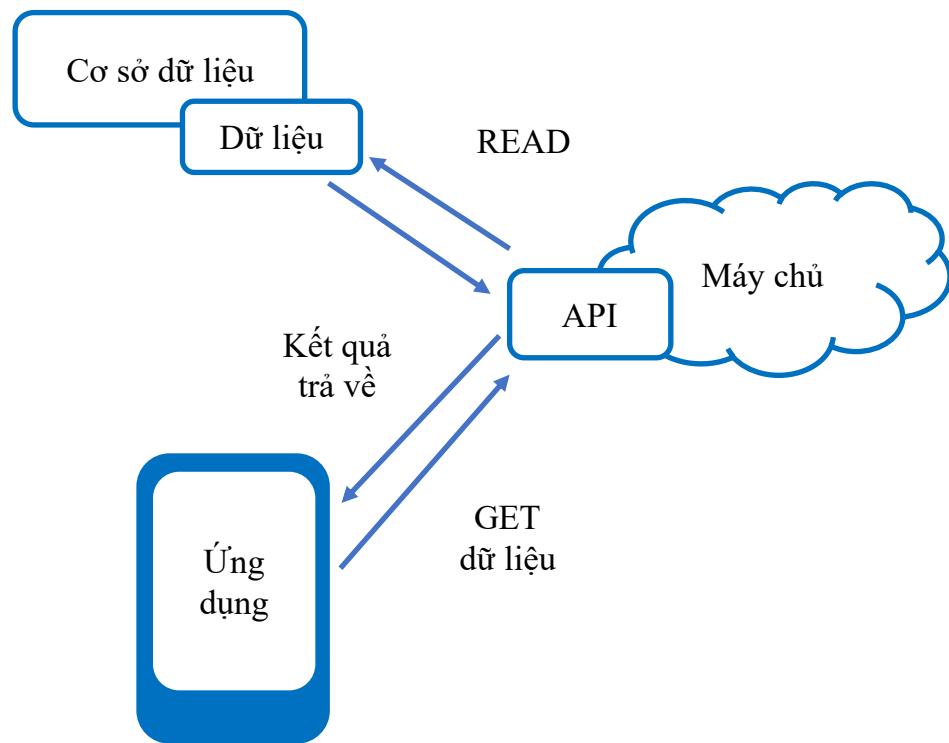
3.4. Hệ thống máy chủ và cơ sở dữ liệu

Ứng dụng PC Dashboard có máy chủ được xây dựng bằng khung Spring cho API và khung NodeJS cho WebSocket. Cơ sở dữ liệu của ứng dụng sử dụng PostgreSQL. Hơn nữa, máy chủ của ứng dụng được lưu trữ trên nền tảng đám mây Heroku, giúp việc truy cập ứng dụng có thể diễn ra mọi lúc, mọi nơi mà không phải bị phụ thuộc như khi sử dụng máy chủ cục bộ.

3.4.1 Kết nối ứng dụng với máy chủ thông qua API

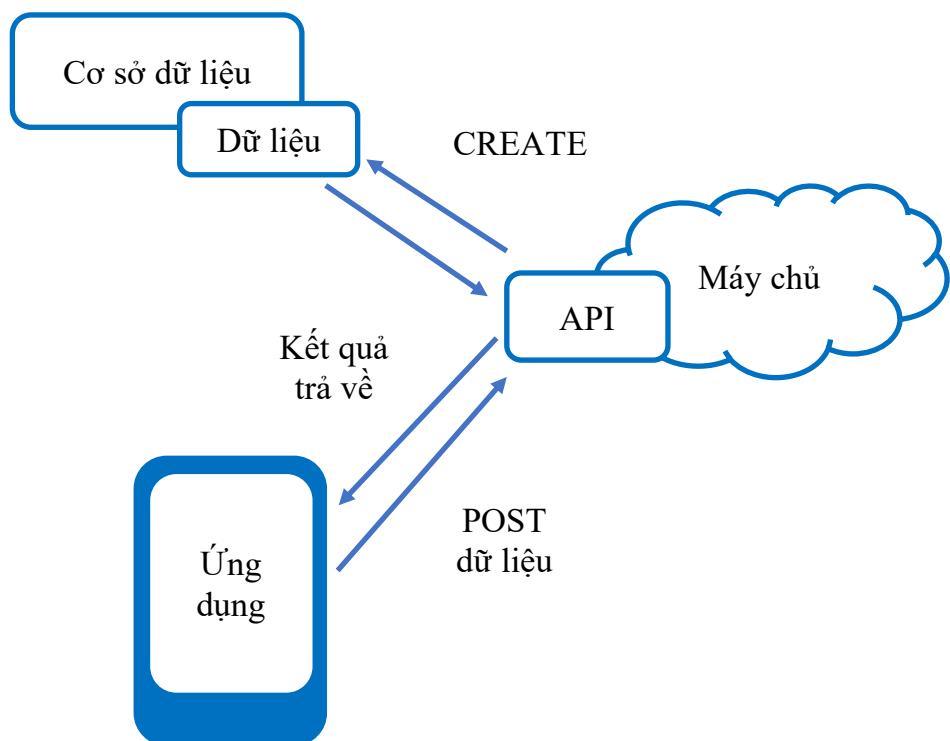
Dữ liệu được ứng dụng lấy từ máy chủ thông qua RESTfulAPI. API là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML. REST là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, ... đến một URL để xử lý dữ liệu. RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các tài nguyên.

Ứng dụng sử dụng các yêu cầu GET để chỉ truy xuất dữ liệu (Hình 3.4) - và không sửa đổi nó theo bất kỳ cách nào. Vì các yêu cầu GET không thay đổi trạng thái của dữ liệu nên đây được coi là các phương thức an toàn. Ứng dụng sẽ gửi yêu cầu GET dữ liệu lên máy chủ thông qua API, máy chủ sẽ đọc dữ liệu từ cơ sở dữ liệu và trả kết quả về cho ứng dụng hiển thị. Nếu tìm thấy dữ liệu trên máy chủ, thì nó sẽ trả về mã phản hồi HTTP 200 (thành công) - cùng với phần thân phản hồi, là nội dung JSON của dữ liệu. Trong trường hợp bài đăng không được tìm thấy trên máy chủ thì nó sẽ trả về mã phản hồi HTTP 404 (không tìm thấy). Tương tự, nếu xác định rằng chính yêu cầu GET không được hình thành chính xác thì máy chủ sẽ trả về mã phản hồi HTTP 400 (yêu cầu lỗi).

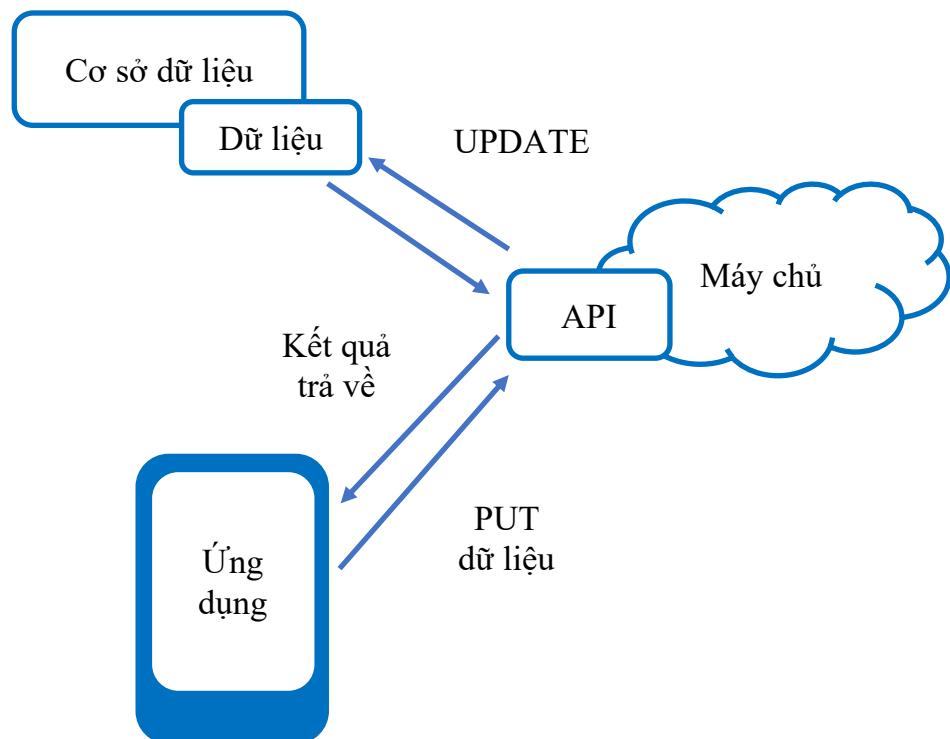


Hình 3.4: Sơ đồ gửi yêu cầu GET từ ứng dụng

API POST được sử dụng để tạo dữ liệu vào bảng cơ sở dữ liệu (Hình 3.5). Nếu dữ liệu đã được tạo trên máy chủ gốc, phản hồi sẽ có mã là HTTP 201 (đã tạo) và chứa kết quả mô tả trạng thái của yêu cầu và tham chiếu đến dữ liệu mới cùng tiêu đề vị trí.



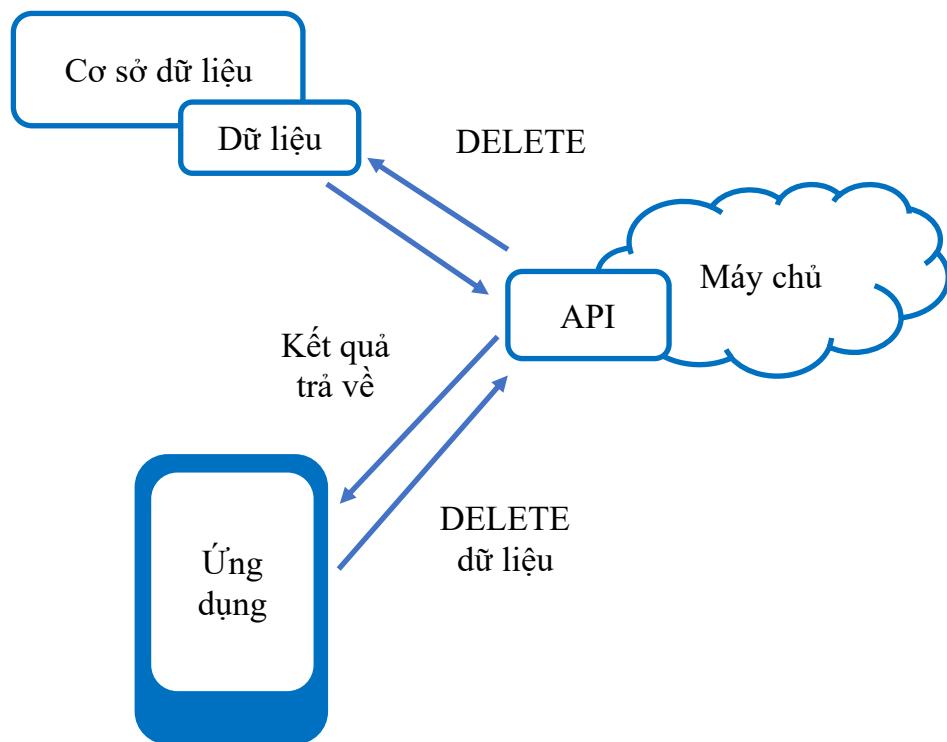
Hình 3.5: Sơ đồ gửi yêu cầu POST từ ứng dụng



Hình 3.6: Sơ đồ gửi yêu cầu PUT từ ứng dụng

Ứng dụng dùng API PUT để cập nhật bài đăng hiện có (Hình 3.6). Nếu dữ liệu không tồn tại, thì API có thể quyết định tạo dữ liệu mới hay không. Nếu dữ liệu mới được tạo bởi API PUT, máy chủ gốc sẽ thông báo cho người dùng thông qua mã phản hồi HTTP 201 (đã tạo) và nếu dữ liệu hiện có bị sửa đổi, thì mã phản hồi 200 (thành công).

API DELETE trong ứng dụng dùng để xoá dữ liệu (Hình 3.7). Nếu thành công, các yêu cầu DELETE có mã phản hồi HTTP 200 (thành công) kèm kết quả mô tả trạng thái. Khi người dùng DELETE một dữ liệu, nó sẽ bị xoá khỏi cơ sở dữ liệu. Việc gọi DELETE trên dữ liệu lần thứ hai sẽ trả về 404 (không tìm thấy) vì nó đã bị xoá.



Hình 3.7: Sơ đồ gửi yêu cầu DELETE từ ứng dụng

3.4.2 Truy cập máy chủ với tư cách quản trị viên

Quản trị viên có thể khởi chạy máy chủ cục bộ để kiểm thử, cập nhật tính năng và sửa lỗi bằng các lệnh của khung Spring trên IntelliJ IDE (Hình 3.8). Spring Boot là một nền tảng Java mã nguồn mở, dùng trong lập trình máy chủ để viết API.

```

Run - pc-dashboard
Run: PcDashboardApplication
Console Endpoints

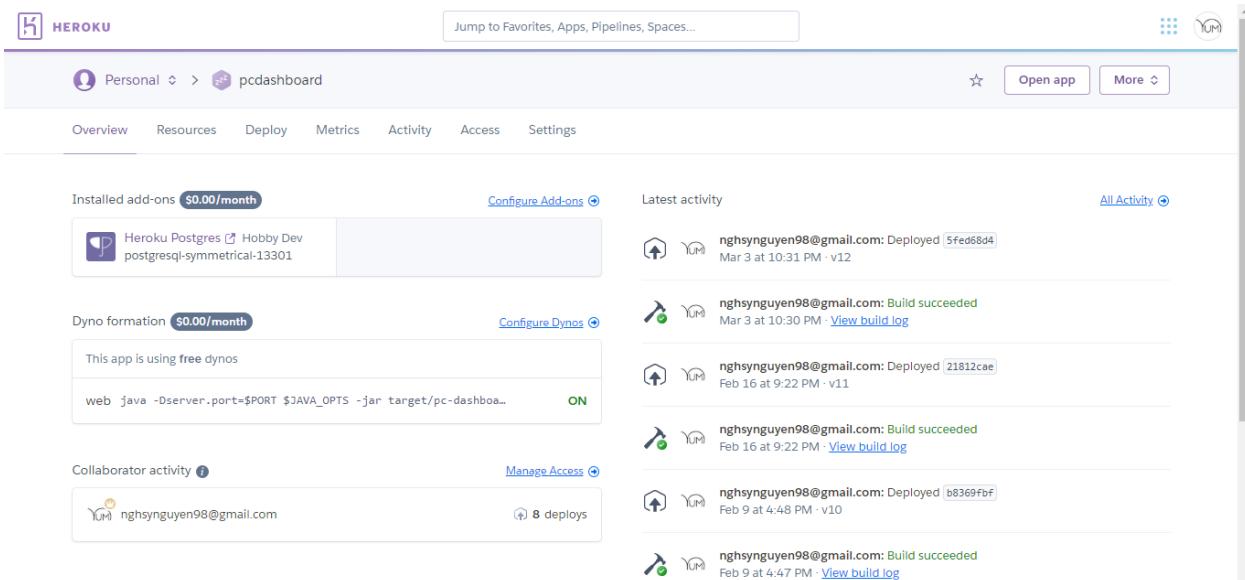
::: Spring Boot :::
(v2.1.8.RELEASE)

2020-06-23 21:26:42.627 INFO 10244 --- [ restartedMain] c.n.pcdashboard.PcDashboardApplication : Starting PcDashboardApplication on SY-NGUYEN with PID 10244 (C:\Users\PC\Desktop\pcdashboard\target\classes) started by PC in C:\Us...
2020-06-23 21:26:42.630 INFO 10244 --- [ restartedMain] c.n.pcdashboard.PcDashboardApplication : No active profile set, falling back to default profiles: default
2020-06-23 21:26:42.660 INFO 10244 --- [ restartedMain] o.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
2020-06-23 21:26:42.660 INFO 10244 --- [ restartedMain] o.DevToolsPropertyDefaultsPostProcessor : Found additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
2020-06-23 21:26:44.079 INFO 10244 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2020-06-23 21:26:44.398 INFO 10244 --- [ restartedMain] o.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 102ms. Found 9 repository interfaces.
2020-06-23 21:26:44.670 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration' of type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration] is defined in null: Bean 'org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration' of type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration] is defined in null
2020-06-23 21:26:44.794 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.security.config.annotation.configuration.ObjectPostProcessorConfiguration' of type [org.springframework.security.config.annotation.configuration.ObjectPostProcessorConfiguration] is defined in null: Bean 'org.springframework.security.config.annotation.configuration.ObjectPostProcessorConfiguration' of type [org.springframework.security.config.annotation.configuration.ObjectPostProcessorConfiguration] is defined in null
2020-06-23 21:26:44.795 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.security.config.annotation.configuration.AutowiredObjectPostProcessor' of type [org.springframework.security.config.annotation.configuration.AutowiredObjectPostProcessor] is defined in null: Bean 'org.springframework.security.config.annotation.configuration.AutowiredObjectPostProcessor' of type [org.springframework.security.config.annotation.configuration.AutowiredObjectPostProcessor] is defined in null
2020-06-23 21:26:44.711 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler#460072ea' of type [org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler#460072ea] is defined in null: Bean 'org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler#460072ea' of type [org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler#460072ea] is defined in null
2020-06-23 21:26:44.713 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null: Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null
2020-06-23 21:26:44.719 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null: Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null
2020-06-23 21:26:44.720 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null: Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null
2020-06-23 21:26:44.721 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null: Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null
2020-06-23 21:26:44.957 INFO 10244 --- [ restartedMain] trationDelegatedBeanPostProcessorChecker : Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null: Bean 'org.springframework.security.core.userdetails.UserDetailsService' of type [org.springframework.security.core.userdetails.UserDetailsService] is defined in null
2020-06-23 21:26:44.976 INFO 10244 --- [ restartedMain] org.apache.catalina.core.StandardService : Tomcat initialized with port(s): 8080 (http)
2020-06-23 21:26:44.976 INFO 10244 --- [ restartedMain] org.apache.catalina.core.StandardService : Starting service [Tomcat]
2020-06-23 21:26:44.976 INFO 10244 --- [ restartedMain] org.apache.catalina.core.StandardService : Initializing Spring embedded WebApplicationContext
2020-06-23 21:26:45.104 INFO 10244 --- [ restartedMain] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 2441 ms
2020-06-23 21:26:45.153 INFO 10244 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2020-06-23 21:26:45.153 INFO 10244 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2020-06-23 21:26:45.171 INFO 10244 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000041: Hibernate Core [5.4.3.Final]
2020-06-23 21:26:45.099 INFO 10244 --- [ restartedMain] org.hibernate.Version : HHH000042: Hibernate Commons Annotations [5.1.0.Final]
2020-06-23 21:26:45.082 INFO 10244 --- [ restartedMain] org.hibernate.Version : HHH000043: Using dialect: org.hibernate.dialect.PostgreSQLDialect
2020-06-23 21:26:45.096 INFO 10244 --- [ restartedMain] org.hibernate.annotations.common.Version : HHH000044: Using dialect: org.hibernate.dialect.PostgreSQLDialect
2020-06-23 21:26:45.096 INFO 10244 --- [ restartedMain] org.hibernate.annotations.common.Version : HHH000045: Using dialect: org.hibernate.dialect.PostgreSQLDialect
2020-06-23 21:27:49.244 INFO 10244 --- [ restartedMain] o.s.b.a.JpaLocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2020-06-23 21:27:49.256 INFO 10244 --- [ restartedMain] j.localContainerEntityManagerFactoryBean : LiveReload server is running on port 35729
2020-06-23 21:27:49.675 INFO 10244 --- [ restartedMain] o.s.b.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2020-06-23 21:27:49.199 INFO 10244 --- [ restartedMain] c.p.services.Implement.FcsServiceImpl : Start init
2020-06-23 21:27:49.199 INFO 10244 --- [ restartedMain] c.p.services.Implement.FcsServiceImpl : Firebase application has been initialized
2020-06-23 21:27:49.327 WARN 10244 --- [ restartedMain] akeWebConfiguration$SpanWebMvcConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly config...
2020-06-23 21:27:49.406 INFO 10244 --- [ restartedMain] o.s.s.web.DefaultServletFilterChain : Creating filter chain: any request, [org.springframework.web.context.request.async.WebAsyncManagerIntegrationFilter@6570b...
2020-06-23 21:27:49.512 INFO 10244 --- [ restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-06-23 21:27:49.823 INFO 10244 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2020-06-23 21:27:49.829 INFO 10244 --- [ restartedMain] c.pcdashboard.PcDashboardApplication : Started PcDashboardApplication in 67.487 seconds (VM running for 68.778)

```

Hình 3.8: Khởi chạy máy chủ cục bộ

Quản trị viên cũng có thể đưa máy chủ lên đám mây bằng cách sử dụng nền tảng Heroku. Heroku là nền tảng đám mây cho phép các lập trình viên xây dựng, triển khai, quản lý và mở rộng ứng dụng. Heroku hỗ trợ triển khai máy chủ đám mây một cách miễn phí (Hình 3.9). Ứng dụng sẽ có hai dịch vụ máy chủ, một là Spring cho API, hai là NodeJS cho WebSocket. NodeJS là một mã nguồn được xây dựng dựa trên nền tảng JavaScript V8 Engine và được sử dụng để xây dựng các ứng dụng Web.



Hình 3.9: Truy cập máy chủ đám mây

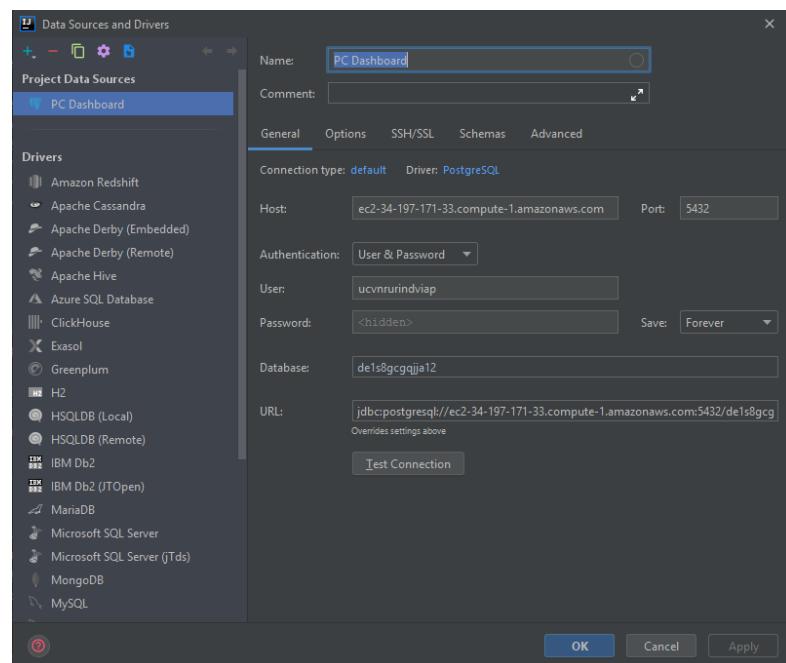
Mỗi máy chủ đám mây sẽ có một chứng chỉ xác thực riêng. Điều này đảm bảo tính bảo mật của dự án cũng như quyền truy cập hệ thống. Thông tin về chứng chỉ xác thực sẽ hiển thị trong tài khoản Heroku của quản trị viên (Hình 3.10). Quản trị viên phải điền thông tin này ở máy chủ cục bộ bằng IntelliJ IDE để có thể truy cập vào cơ sở dữ liệu (Hình 3.11).

The screenshot shows the 'ADMINISTRATION' section of the Heroku dashboard. Under 'Database Credentials', it displays the following information for a PostgreSQL database:

- Host:** ec2-34-197-171-33.compute-1.amazonaws.com
- Database:** de1s8gcgqjja12
- User:** ucvnrurindviap
- Port:** 5432
- Password:** 53660bbddd9a43bc22052edcc9fe27a535345b0515e31eb7a4bc91fe661853be
- URI:** postgres://ucvnrurindviap:53660bbddd9a43bc22052edcc9fe27a535345b0515e31eb7a4bc91fe661853be@ec2-34-197-171-33.compute-1.amazonaws.com:5432/de1s8gcjja12
- Heroku CLI:** heroku pg:psql postgresql-symmetrical-13301 --app pcdashboard

Below this, there are sections for 'Reset Database' and 'Destroy Database'.

Hình 3.10: Chứng chỉ xác thực để truy cập cơ sở dữ liệu đám mây



Hình 3.11: Chứng chỉ xác thực để truy cập cơ sở dữ liệu cục bộ

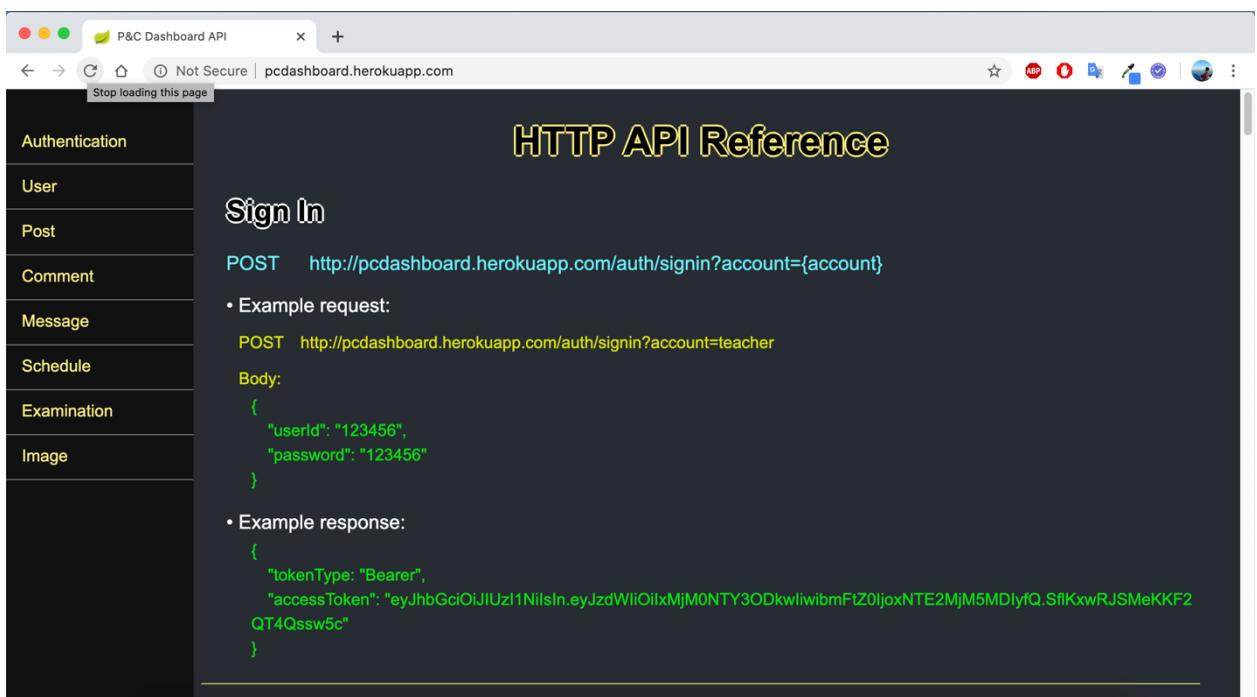
Khi hệ thống xác thực thành công, quản trị viên có thể truy cập vào cơ sở dữ liệu để quản lý người dùng, bài đăng (Hình 3.12, Hình 3.13). Việc thêm, sửa, xoá dữ liệu có thể thực hiện nhờ API với hướng dẫn được hiển thị ở trang chủ của máy chủ (Hình 3.14).

1	10 http://pcdashboard.herokuapp.com/image..._K16	khoi@gmail.com	Huỳnh Thị Nhàn	0 \$2a\$10\$5KqjaATKgsZnKce0\$8gVed...	0990xxxxxx	1613094		
2	15 http://pcdashboard.herokuapp.com/image..._K16	quyen@gmail.com	Cao Võ Nhà Quản	0 \$2a\$10\$91xt1lGDlUJZEKjh3mb...	0990xxxxxx	1613163		
3	3 http://pcdashboard.herokuapp.com/image..._K16	phanhantungd@gmail.com	Phan Thành Tùng	0 \$2a\$10\$gukYNuTxMvdCxdlWw7leF...	09912522913	1613240		
4	14 http://pcdashboard.herokuapp.com/image..._K16	phung@gmail.com	Nguyễn Thị Kim Phụng	0 \$2a\$10\$e4hAZ2huU8bxky25h2Aneg...	0990xxxxxx	1613154		
5	5 http://pcdashboard.herokuapp.com/image..._K16	nang@gmail.com	Võ Thành Nam	0 \$2a\$10\$172Qdqeuaefet6jPjztt4en...	0990xxxxxx	1613118		
6	7 http://pcdashboard.herokuapp.com/image..._K16	chibag@gmail.com	Phan Ngũ Chỉ Bảo	0 \$2a\$10\$NFBu0IX0dfegQ1Bwtf0...	0990xxxxxx	1613015		
7	1 http://pcdashboard.herokuapp.com/image..._K16	nghsynguyen8@gmail.com	Nguyễn Hồng Sỹ Nguyễn	0 \$2a\$10\$xs5uirc/ZDmbtA0dxQxVO2...	0793648763	1613124		
8	8 http://pcdashboard.herokuapp.com/image..._K16	haidang@gmail.com	Tống Hải Đăng	0 \$2a\$10\$yjq3kULJKvGCF1j81KH...	0990xxxxxx	1613026		
9	12 http://pcdashboard.herokuapp.com/image..._K16	nguyen@gmail.com	Nguyễn Trần Khanh Nguyễn	0 \$2a\$10\$g141ne9wVVcz66.UKvVug...	0990xxxxxx	1613125		
10	4 http://pcdashboard.herokuapp.com/image..._K16	hoaphuctrphvnhhung@gmail.com	Mai Hoài Phúc	0 \$2a\$10\$u0ixp7FKeKuJFKolt0...	039505949	1613151		
11	13 http://pcdashboard.herokuapp.com/image..._K16	phuc@gmail.com	Hồ Việt Phúc	0 \$2a\$10\$u0ixp7FKeKuJFKolt0...	0990xxxxxx	1613148		
12	6 http://pcdashboard.herokuapp.com/image..._K16	rnuhang@gmail.com	Võ Thị Như Hằng	0 \$2a\$10\$u0aqvVnQvTEzprm3K...	0990xxxxxx	1613083		
13	5 http://pcdashboard.herokuapp.com/image..._K16	taiphuoc@gmail.com	Trần Tài Phúc	0 \$2a\$10\$VDNgkTx30.lv7CUXnewH...	0990xxxxxx	1613152		
14	9 http://pcdashboard.herokuapp.com/image..._K16	hieu@gmail.com	Cao Chí Hiếu	0 \$2a\$10\$ee0p3U2zU795tzBBP...	0990xxxxxx	1613062		
15	2 http://pcdashboard.herokuapp.com/image..._K16	bachflutter@gmail.com	Bùi Ngũ Tôn Bách	0 \$2a\$10\$3anvxJQhpksaa7PKewPeC...	0000000000	1613081		
16	16 http://pcdashboard.herokuapp.com/image..._K16	giathinh@gmail.com	Ngô Gia Thịnh	0 \$2a\$10\$Hmrdd6GUkjYj2K5luzy0...	0990xxxxxx	1613183		
17	17 http://pcdashboard.herokuapp.com/image..._K16	duothinh@gmail.com	Nguyễn Đức Thịnh	0 \$2a\$10\$526sbwobeg1e3hU2hcxiu...	0990xxxxxx	1613184		
18	18 http://pcdashboard.herokuapp.com/image..._K16	thog@gmail.com	Nguyễn Quốc Thanh	0 \$2a\$10\$9IdF1DTCHpqIwpj7n...	0990xxxxxx	1613186		
19	19 http://pcdashboard.herokuapp.com/image..._K16	thuy@gmail.com	Nguyễn Thị Lê Thúy	0 \$2a\$10\$91tX5Y7lePh0hrSm1te...	0990xxxxxx	1613195		
20	20 http://pcdashboard.herokuapp.com/image..._GV	hvtuan@hcmus.edu.vn	Huỳnh Văn Tuấn	0 \$2a\$10\$1qQbhnnjuTe00121Yey...	9876543210	0441		
21	21 http://pcdashboard.herokuapp.com/image..._GV	hthyen@hcmus.edu.vn	Hứa Thị Hoàng Yến	0 \$2a\$10\$tur1161801ZQyYe10p1xg...	0993376733	0442		
22	22 http://pcdashboard.herokuapp.com/image..._GV	hbvnh@hcmus.edu.vn	Hồ Văn Bình	0 \$2a\$10\$AZR6hufTklbc1pfpH...	0918012299	0434		
23	23 http://pcdashboard.herokuapp.com/image..._GV	pnthuan@hcmus.edu.vn	Phan Nguyệt Thuận	0 \$2a\$10\$5g5q9bzttzUdf5bd6j...	0997621310	1307		
24	24 http://pcdashboard.herokuapp.com/image..._GV	nhtanh@hcmus.edu.vn	Nguyễn Huỳnh Tuấn Anh	0 \$2a\$10\$9EB0vurkeR.H0Kp30...	0993253250	0433		
25	25 http://pcdashboard.herokuapp.com/image..._GV	nc1nh@hcmus.edu.vn	Nguyễn Chí Linh	0 \$2a\$10\$NMHT7VSChvFrgUKS2zeub...	0363364418	2342		
26	26 http://pcdashboard.herokuapp.com/image..._GV	htanh@hcmus.edu.vn	Huỳnh Thành Nhât	0 \$2a\$10\$ANC2LuuqaaKYG0Re501...	0919786248	0439		
27	27 http://pcdashboard.herokuapp.com/image..._GV	nathuh@hcmus.edu.vn	Nguyễn Anh Thư	0 \$2a\$10\$sohkv0lAUUagyj.751cou...	0976360697	2558		
28	28 http://pcdashboard.herokuapp.com/image..._GV	dvtie@hcmus.edu.vn	Đặng Văn Liệt	0 \$2a\$10\$11zx21h557fa0xs8R02...	0963360027	7375		
29	29 http://pcdashboard.herokuapp.com/image..._GV	ntnhat@gmail.com	Nguyễn Tiến Nhật	0 \$2a\$10\$Pc7DyNCRI1B01rnIon/V...	0990xxxxxx	1613133		
30	30 http://pcdashboard.herokuapp.com/image..._GV	ntnhang@gmail.com	Nguyễn Minh Thành	0 \$2a\$10\$1de1wx2f3mXrC0H8e150...	0990xxxxxx	1613176		
31	31 http://pcdashboard.herokuapp.com/image..._GV	nvatuan@gmail.com	Nguyễn Võ Anh Tuấn	0 \$2a\$10\$Pz2md0Hmf228n0X/CQh0u...	0990xxxxxx	1613238		
32	32 http://pcdashboard.herokuapp.com/image..._GV	vttanh@gmail.com	Vương Trần Trí Anh	0 \$2a\$10\$V1NVE01ndgubr109650...	0990xxxxxx	1713018		
33	33 http://pcdashboard.herokuapp.com/image..._GV	robaog@gmail.com	Nguyễn Quốc Bảo	0 \$2a\$10\$4Fu4puhElx/kF6u8b0...	0990xxxxxx	1713019		
34	35 http://pcdashboard.herokuapp.com/image..._GV	minhong@gmail.com	Hà Hữu Hòn	0 \$2a\$10\$V1Yh7C2XgqR8k0...	0990xxxxxx	1713050		
35	36 http://pcdashboard.herokuapp.com/image..._GV	pohon@gmail.com	Phạm Quỳnh Hương	0 \$2a\$10\$C23661fFLnSHW7v/Eu0...	0990xxxxxx	1713066		
36	37 http://pcdashboard.herokuapp.com/image..._GV	reky@gmail.com	Nguyễn Cao Kỳ	0 \$2a\$10\$quKKL0X.M93tH13lu...	0990xxxxxx	1713066		
37	38 http://pcdashboard.herokuapp.com/image..._GV	cdoic@gmail.com	Cam Đức Lộc	0 \$2a\$10\$8u1g8881xsnpmtokak1...	0990xxxxxx	1713072		
38	39 http://pcdashboard.herokuapp.com/image..._GV	ntman@gmail.com	Nông Thành Mẫn	0 \$2a\$10\$RvJR9esgJn2DETso.9u...	0990xxxxxx	1713081		

Hình 3.12: Giao diện quản lý người dùng của quản trị viên

1	a0bbfbf7-e7d3-47f3-80ac-856e47cc52bf	2020-11-14 08:38:48.682000	9003270	http://pcdashboard.herokuapp.com/image/3be931c0-deca-4533-be77-2146e5359b33
2	81951496-6b71-4897-83b3-53a3f0cc3931	2020-02-09 08:41:02.365000	9003275	http://pcdashboard.herokuapp.com/image/bc90b176-99d5-4aca-91c6-83d7688a23e3

Hình 3.13: Giao diện quản lý bài đăng bộ môn của quản trị viên



Hình 3.14: Giao diện trang chủ của máy chủ và hướng dẫn API

Trong ứng dụng, tôi sử dụng thư viện Dio và Axios cho việc gửi yêu cầu HTTP lên máy chủ. Dio là một thư viện HTTP cho ngôn ngữ Dart, được phát triển dựa trên Retrofit. Axios là một thư viện HTTP cho ngôn ngữ JavaScript, thường được sử dụng cho cả NodeJS và ReactJS (Hình 3.15, Hình 3.16).

```
Future<List<DepartmentResponse>> fetchList(int number) async {
  try {
    String token = await Preferences.loadToken();
    Response response = await Dio()
      .get(Configs.baseUrl + Configs.departmentPath,
        queryParameters: {'number': number},
        options: Options(headers: {'Authorization': token}))
      .timeout(const Duration(milliseconds: 3000));
    List<DepartmentResponse> posts = (response.data as List)
      .map((item) => DepartmentResponse.fromJson(item))
      .toList();
    return posts;
  } catch (e) {
    print(e);
    return null;
  }
}
```

Hình 3.15: Hàm gọi API với Dio trong phiên bản sinh viên

```

export const loadDepartmentPosts = (number) => {
  return async (dispatch) => {
    let token = await Preferences.loadToken()
    Axios({
      method: 'GET',
      url: Configs.baseUrl + Configs.departmentPath,
      headers: {
        'Authorization': token
      },
      params: { 'number': number }
    }).then(response => {
      dispatch(saveDepartmentPosts(response.data))
    }).catch(error => {
      console.log(error)
    })
  }
}

```

Hình 3.16: Hàm gọi API với Axios trong phiên bản giảng viên

3.5. Mô tả chi tiết ứng dụng

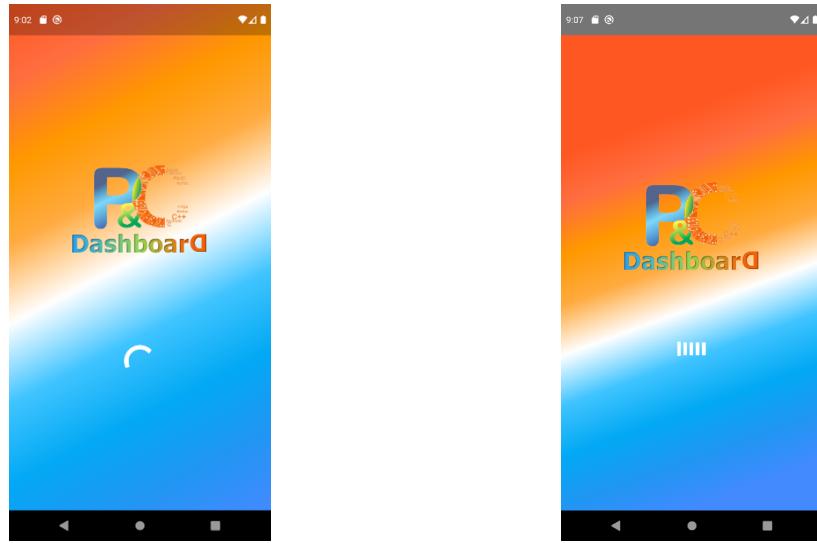
3.5.1 Giao diện màn hình chờ



Hình 3.17: Biểu tượng của ứng dụng PC Dashboard

Biểu tượng của ứng dụng cũng chính là biểu tượng của bộ môn Vật lý Tin học, được thiết kế lại để phù hợp với ứng dụng, với tông màu cam và xanh dương (Hình 3.17). Màn hình chờ được hiển thị khi vừa mở ứng dụng, nhằm thông báo đến người dùng việc chờ để ứng dụng có thể tải trước dữ liệu (Hình 3.18). Dữ liệu tải trước là thông tin đăng nhập lần trước, dùng để kiểm tra việc đăng nhập của người dùng. Nếu người dùng đã từng đăng nhập trước đây và chưa đăng xuất, ứng dụng sẽ vào thẳng bảng điều khiển mà không cần thông qua màn hình đăng nhập. Điều này giúp người dùng không phải đăng nhập lại nữa, làm đơn giản hơn quá trình sử dụng. Thông tin đăng nhập và con dấu được lưu trữ dưới thiết bị, với một hạn sử dụng. Khi

quá mốc thời gian này, con dấu sẽ hết hạn và buộc người dùng đăng nhập lại. Việc đăng nhập lại này giúp bảo mật tài khoản, xác thực người dùng, tránh việc đánh cắp tài khoản và thông tin. Bên cạnh đó, tính năng này giúp việc trải nghiệm ứng dụng trở nên quen thuộc hơn, giống như các mạng xã hội lớn.



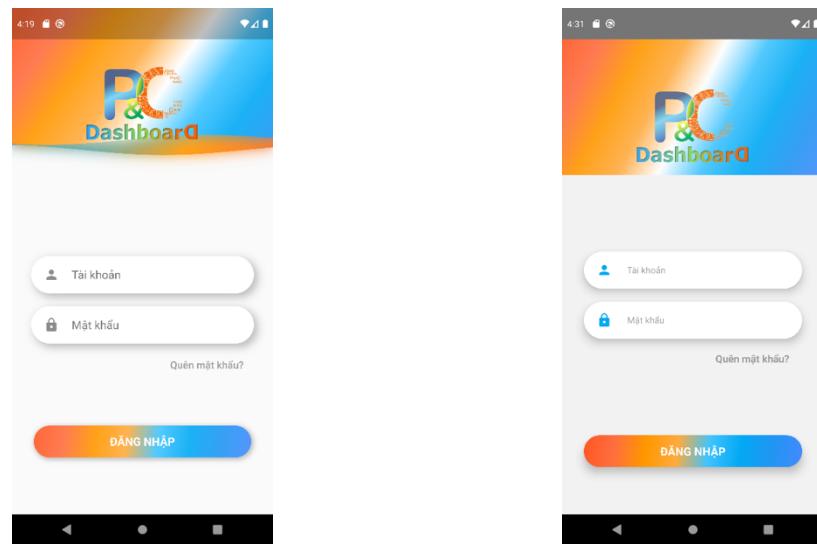
(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.18: Giao diện màn hình chờ

3.5.2 Giao diện màn hình đăng nhập và lấy lại mật khẩu

Ở màn hình đăng nhập, người dùng có thể điền tên tài khoản (mã số sinh viên/mã số giảng viên) và mật khẩu để tiến hành đăng nhập vào ứng dụng (Hình 3.19). Tài



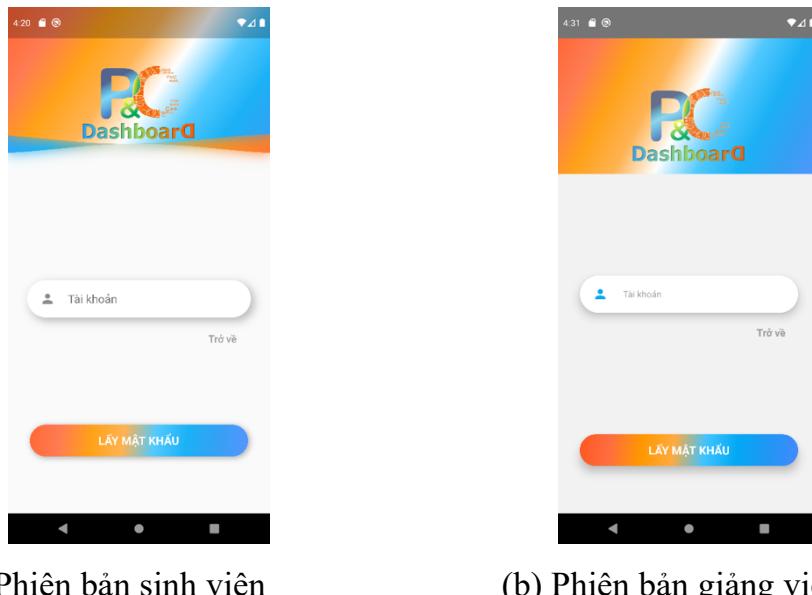
(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.19: Giao diện màn hình đăng nhập

khoản đăng nhập sẽ được đăng ký và cung cấp bởi quản trị viên thông qua trình quản lý của máy chủ.

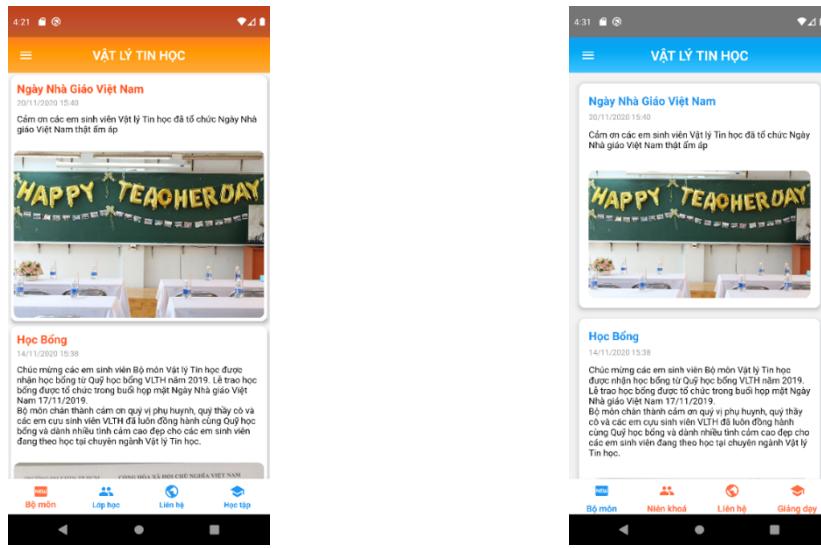
Nếu người dùng quên mật khẩu, chỉ cần nhập tên tài khoản ở màn hình lấy lại mật khẩu và xác nhận, hệ thống sẽ gửi một email hướng dẫn khôi phục mật khẩu về tài khoản mail tương ứng của người dùng (Hình 3.20). Trong email sẽ có một đường dẫn. Khi người dùng nhấp vào, trang tạo mới mật khẩu sẽ được mở ra và người dùng chỉ cần nhập mật khẩu mới vào. Việc gửi email lấy lại mật khẩu là hoàn toàn tự động nhờ dịch vụ của Google thông qua Gmail. Do đó, người dùng cần cung cấp chính xác email khi quản trị viên đăng ký để tránh việc thất lạc tài khoản sau này.



Hình 3.20: Giao diện màn hình lấy lại mật khẩu

3.5.3 Giao diện màn hình điều khiển

Sự khác nhau giữa hai phiên bản bắt đầu thể hiện rõ từ đây. Với phiên bản sinh viên, tông màu chủ đạo sẽ là màu cam (Hình 3.21). Với phiên bản giảng viên, tông màu chủ đạo sẽ là màu xanh da trời. Điều này vừa phù hợp với đối tượng sử dụng khi sinh viên cần giao diện với màu sắc tươi trẻ, còn giảng viên cần giao diện với màu sắc nhẹ nhàng và nhưng vẫn tuân theo tông màu của biểu tượng bộ môn Vật lý Tin học. Bảng điều khiển sẽ có bốn mục cho người dùng lựa chọn, tương ứng với bốn chức năng là bộ môn, lớp học, liên hệ và học tập/ giảng dạy.



(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.21: Giao diện bảng điều khiển

Bảng điều khiển sử dụng BottomNavigationBar để hiển thị 4 mục và cho phép người dùng lựa chọn (Hình 3.22).

```

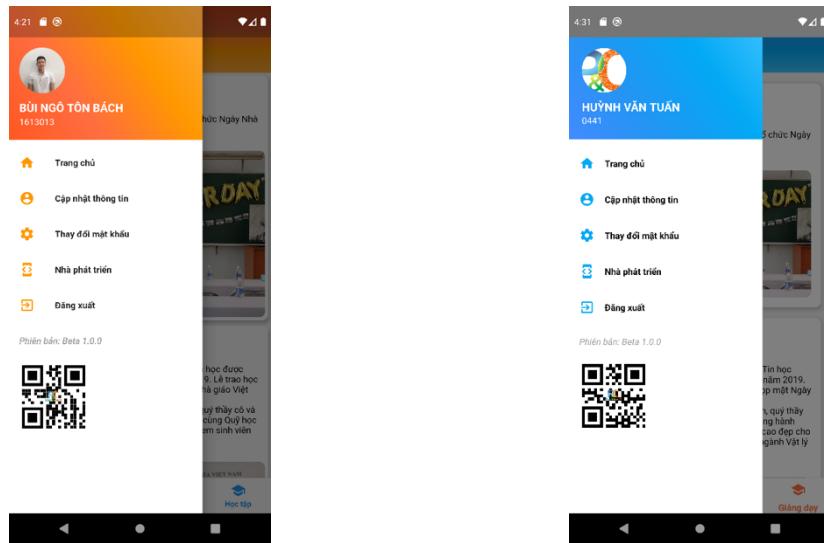
bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(...), // Text, BottomNavigationBarItem
    BottomNavigationBarItem(...), // Text, BottomNavigationBarItem
    BottomNavigationBarItem(...), // Text, BottomNavigationBarItem
    BottomNavigationBarItem(...), // Text, BottomNavigationBarItem
  ],
  selectedTextColor: Colors.deepOrangeAccent,
  unselectedTextColor: Colors.blue,
  type: BottomNavigationBarType.fixed,
  currentIndex: currentIndex,
  backgroundColor: Colors.white,
  onTap: (index) {
    setState(() {
      currentIndex = index;
    });
  },
), // BottomNavigationBar

```

Hình 3.22: BottomNavigationBar của phiên bản sinh viên

3.5.4 Giao diện ngăn công cụ và các chức năng

Ngăn công cụ sẽ xuất hiện trên màn hình điều khiển khi người dùng thao tác vuốt màn hình từ trái sang phải (Hình 3.23).



(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.23: Giao diện ngăn công cụ

Drawer hỗ trợ thao tác vuốt trái sang phải để mở ngăn công cụ (Hình 3.24). Mỗi mục trên Drawer sẽ là một ListTile, với hành vi onTap sẽ mở ra màn hình tương ứng.

```

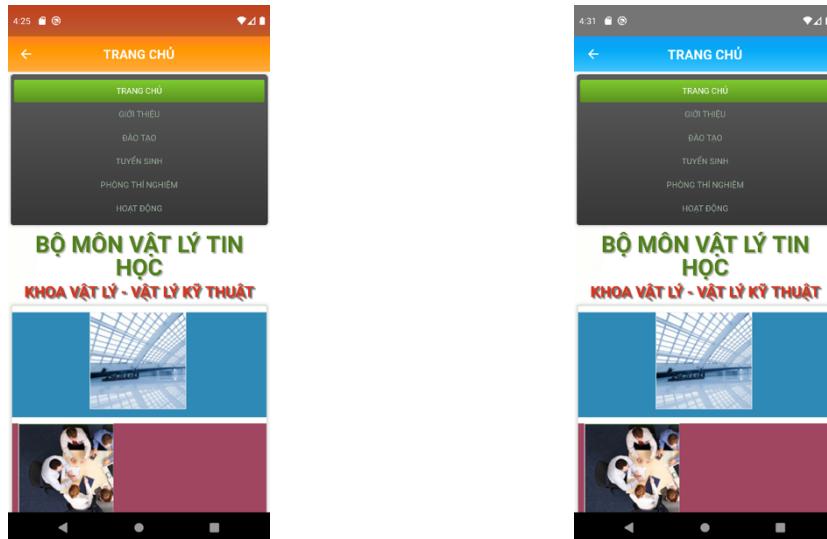
  |- child: Drawer(
    |- child: ListView(
      |   // Important: Remove any padding from the ListView.
      |   padding: EdgeInsets.zero,
      |   children: <Widget>[
      |     InkWell(
      |       child: UserAccountsDrawerHeader(...), // UserAccountsDrawerHeader
      |     ), // InkWell
      |     ListTile(...), // ListTile
      |     Padding(...) // Padding
      |   ], // <Widget>[]
    ), // ListView
  ), // Drawer

```

Hình 3.24: Drawer của phiên bản sinh viên

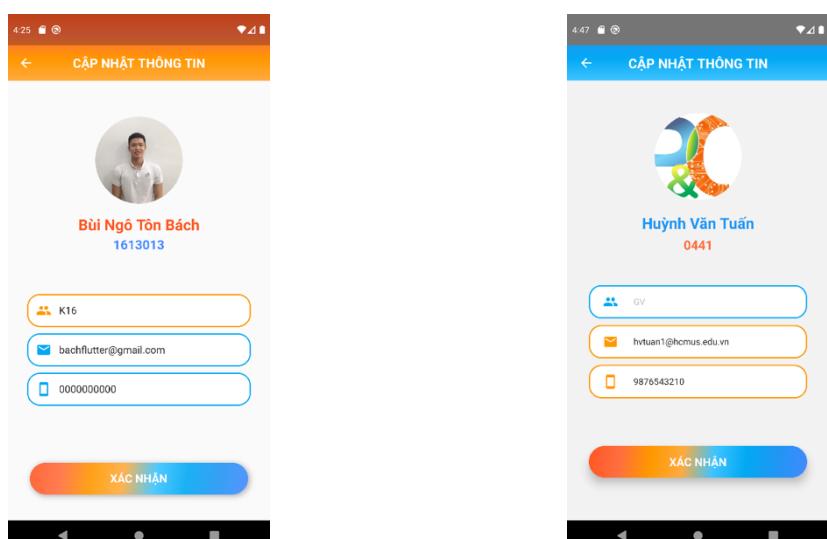
Ở phía trên cùng của ngăn công cụ là thông tin tài khoản đang đăng nhập vào ứng dụng, bao gồm ảnh đại diện, họ và tên, mã số sinh viên/ mã số giảng viên. Ở phía dưới cùng của ngăn công cụ còn hiển thị mã QR dùng để quét điểm danh.

Mục Trang chủ cho phép người dùng truy cập vào trang Web của bộ môn Vật lý Tin học để theo dõi những thông báo mới nhất từ bộ môn (Hình 3.25).



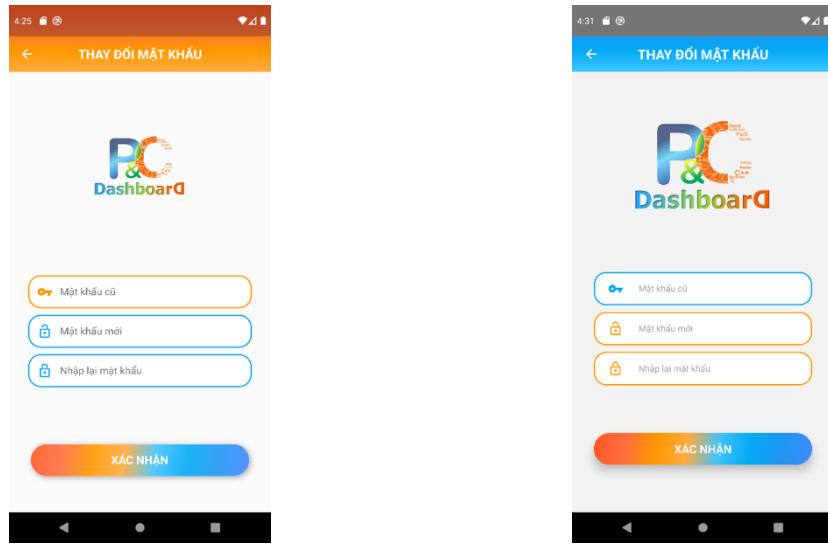
Hình 3.25: Giao diện màn hình trang chủ

Mục Cập nhật thông tin cho phép người dùng sử dụng chức năng cập nhật thông tin cho tài khoản, giúp cho việc quản lý thông tin về email, số điện thoại, ... luôn là dữ liệu mới nhất, đảm bảo cho việc liên lạc một cách nhanh chóng (Hình 3.26).



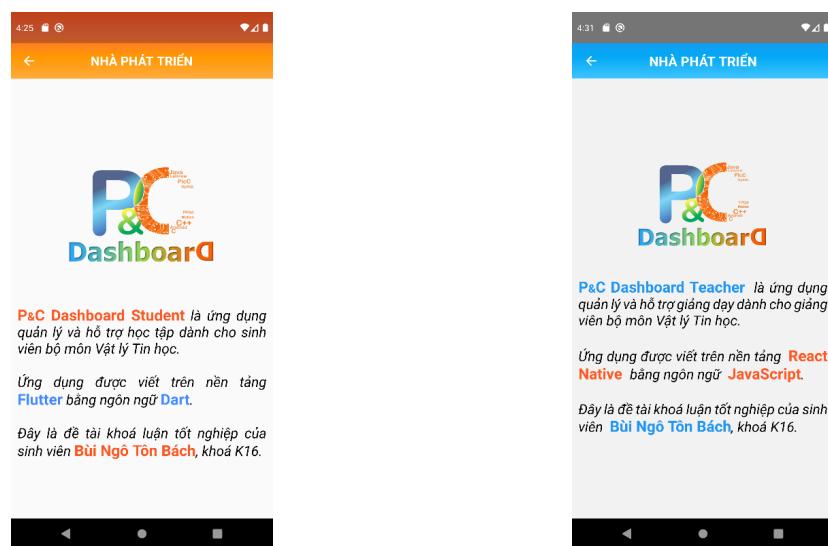
Hình 3.26: Giao diện màn hình cập nhật thông tin

Mục Thay đổi mật khẩu cho phép người dùng sử dụng chức năng thay đổi mật khẩu cho tài khoản, giúp cho việc bảo mật thông tin và an toàn dữ liệu, tránh việc mất tài khoản khi để lộ mật khẩu (Hình 3.27).



Hình 3.27: Giao diện màn hình thay đổi mật khẩu

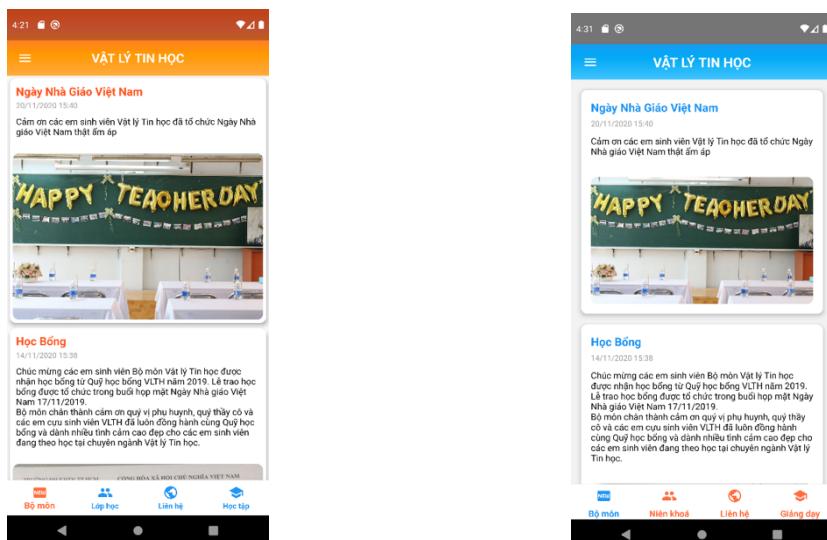
Mục Nhà phát triển cho phép người dùng xem được thông tin của lập trình viên cũng như ứng dụng (Hình 3.28). Nút Đăng xuất cho phép thoát tài khoản ra khỏi ứng dụng.



Hình 3.28: Giao diện màn hình nhà phát triển

3.5.5 Giao diện màn hình bộ môn

Tất cả bài viết chứa tin tức liên quan đến khoa và bộ môn sẽ được người quản trị đăng tải và hiển thị tại màn hình bộ môn. Sinh viên và giảng viên đều có thể xem được thông báo. Điều này giúp thông tin từ khoa và bộ môn có thể đến được người dùng một cách nhanh nhất (Hình 3.29).

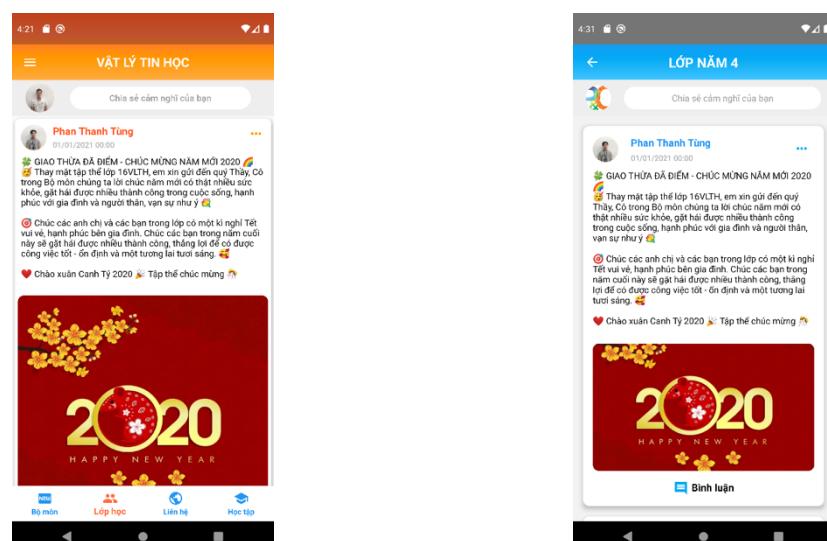


(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.29: Giao diện màn hình bộ môn

3.5.6 Giao diện màn hình lớp học, đăng bài và bình luận

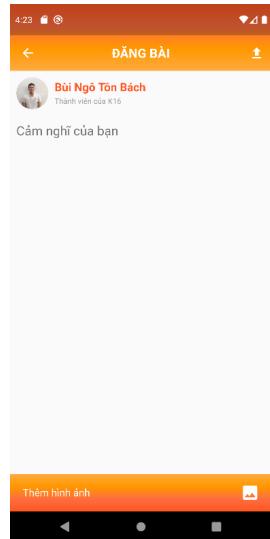


(a) Phiên bản sinh viên

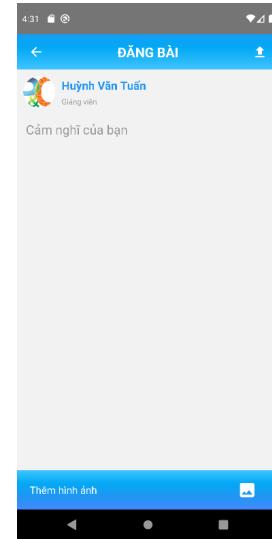
(b) Phiên bản giảng viên

Hình 3.30: Giao diện màn hình lớp học

Sinh viên chỉ có thể truy cập vào nhóm lớp của niên khóa tương ứng. Còn giảng viên có thể truy cập vào nhóm lớp của cả năm ba và năm tư để thông báo và quản lý được tất cả sinh viên của mình (Hình 3.30). Trong nhóm lớp, người dùng có thể đăng bài. Bên dưới bài đăng có thể đính kèm ảnh (Hình 3.31).



(a) Phiên bản sinh viên



(b) Phiên bản giảng viên

Hình 3.31: Giao diện màn hình đăng bài

Ngoài ra, người dùng có thể sửa hoặc xóa bài đăng bằng cách nhấn vào biểu tượng ở góc ba chấm trên bên phải của mỗi bài đăng (Hình 3.32). Một cửa sổ sẽ hiện



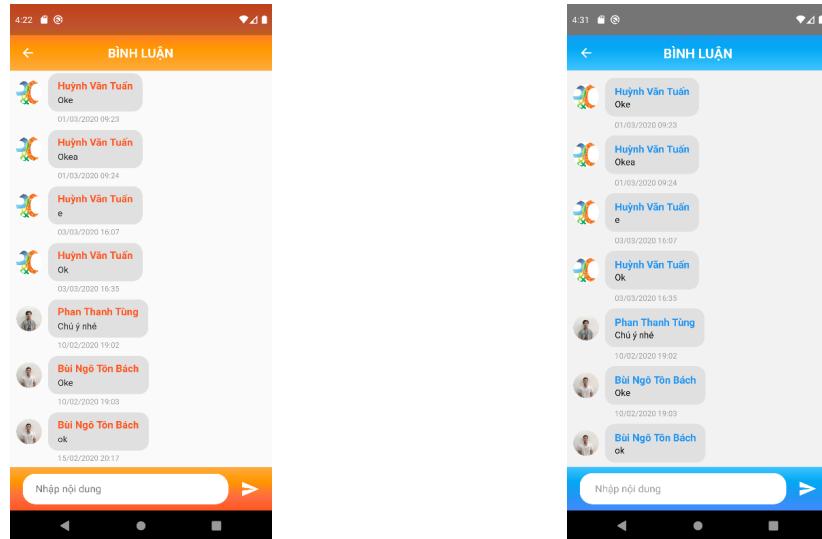
(a) Phiên bản sinh viên



(b) Phiên bản giảng viên

Hình 3.32: Giao diện sửa và xóa bài đăng

lên từ phía dưới màn hình, cho phép người dùng chọn lựa thao tác như sửa, xoá bài đăng hoặc huỷ.

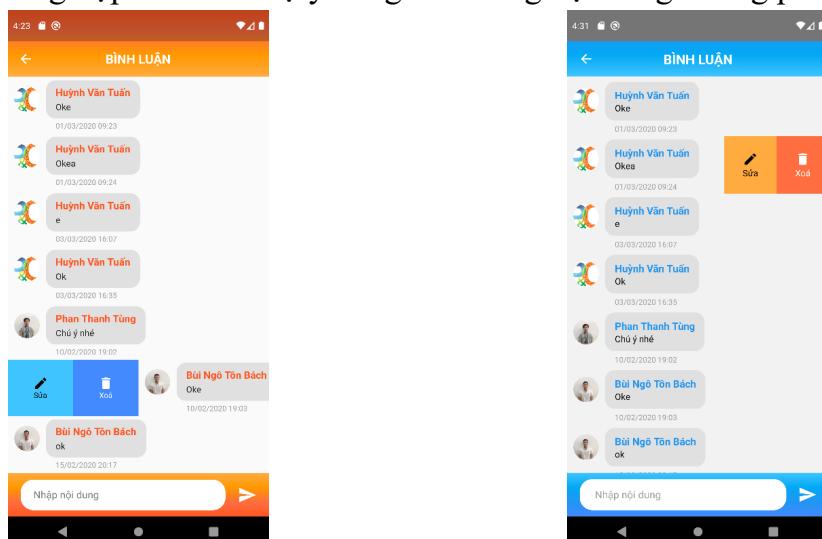


(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.33: Giao diện màn hình bình luận.

Khi bấm vào nút bình luận trên mỗi bài đăng, người dùng sẽ vào màn hình bình luận (Hình 3.33). Người dùng cũng có thể sửa hoặc xóa bình luận bằng thao tác vuốt màn hình (Hình 3.34). Sinh viên chỉ có thể sửa và xoá bài đăng, bình luận của bản thân. Còn viên còn có thể xoá bài đăng, bình luận của bản thân và cả sinh viên. Điều này tránh trường hợp sinh viên tự ý đăng tải những nội dung không phù hợp.



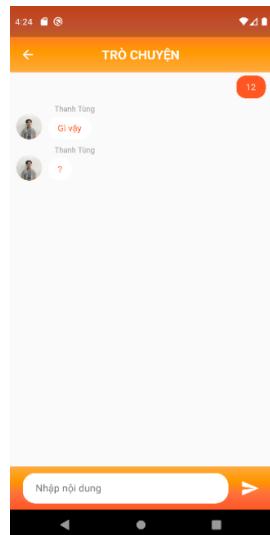
(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.34: Giao diện sửa và xóa bình luận

3.5.7 Giao diện màn hình trò chuyện

Ở phiên bản cho sinh viên, ứng dụng có thêm chức năng trò chuyện nhóm, giúp tăng sự trao đổi giữa các thành viên trong lớp khi mà có thể nói chuyện, bàn luận về các chủ đề trong học thuật cũng như ngoài đời sống (Hình 3.35).



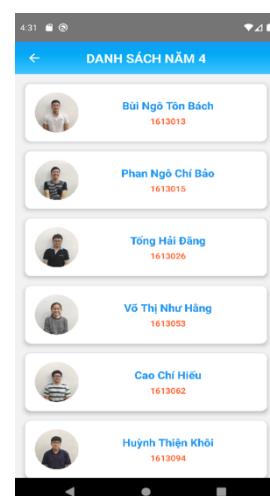
Hình 3.35: Giao diện màn hình trò chuyện phiên bản sinh viên

3.5.8 Giao diện màn hình quản lý thông tin sinh viên và giảng viên

Ứng dụng cung cấp danh sách sinh viên và danh sách giảng viên, giúp việc quản lý danh sách sinh viên từng lớp và điểm danh của giảng viên trở nên dễ dàng (Hình 3.36).



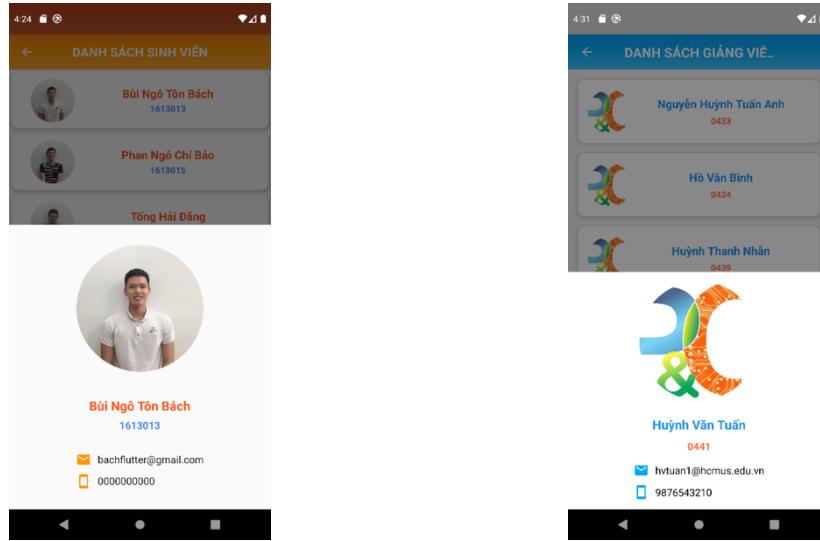
(a) Phiên bản sinh viên



(b) Phiên bản giảng viên

Hình 3.36: Giao diện màn hình danh sách sinh viên và giảng viên

Người dùng cũng có thể xem thông tin từng sinh viên, giảng viên và liên lạc khi cần thiết (Hình 3.37). Khi nhấn vào email hoặc số điện thoại, ứng dụng sẽ chuyển đến chức năng soạn thảo email hoặc quay số điện thoại, giúp cho người dùng có thể dễ dàng liên lạc mà không cần thoát ứng dụng.



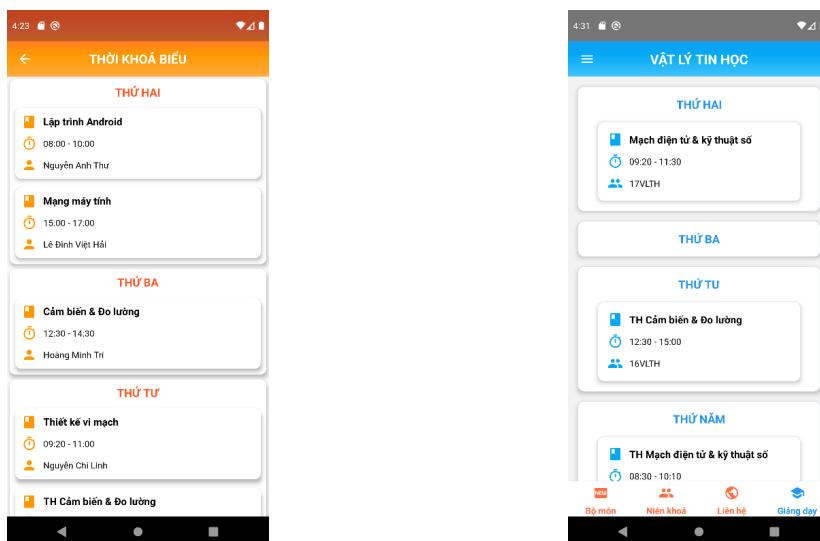
(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.37: Giao diện trang thông tin liên lạc

3.5.9 Giao diện màn hình quản lý học tập và giảng dạy

Ứng dụng cung cấp thông tin về thời khóa biểu để sinh viên và giảng viên có thể nắm được lịch học và lịch dạy của bản thân (Hình 3.38).



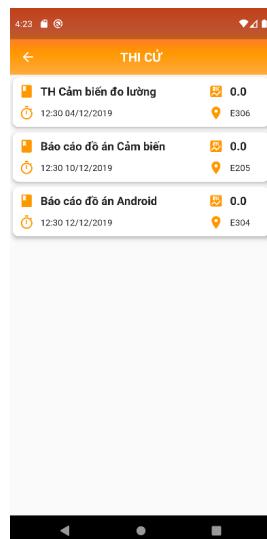
(a) Phiên bản sinh viên

(b) Phiên bản giảng viên

Hình 3.38: Giao diện màn hình thời khóa biểu

Thời khoá biểu sẽ thay đổi nếu có sự cập nhật từ quản trị viên. Sinh viên và giảng viên cần thường xuyên truy cập để có thể nắm được lịch học tập và giảng dạy mới nhất.

Ngoài ra, ở phiên bản cho sinh viên, sinh viên còn có thể xem được lịch thi, điểm thi của bản thân, từ đó có kế hoạch ôn tập phù hợp để đạt được kết quả học tập tốt nhất (Hình 3.39). Thời gian thi, địa điểm thi sẽ được cập nhật trước nhằm đảm bảo sinh viên nắm được thông tin môn thi để có mặt. Khi có kết quả thi, ứng dụng sẽ cập nhật điểm số cho sinh viên.



Hình 3.39: Giao diện màn hình thi cử phiên bản sinh viên

KẾT LUẬN

Ưu điểm

Ứng dụng có giao diện thân thiện với người dùng và dễ dàng sử dụng. Thiết kế và màu sắc giữa hai phiên bản phù hợp với đối tượng người dùng hướng đến. Ứng dụng chạy được trên nhiều nền tảng, hiệu năng cao và trải nghiệm người dùng tốt. Chức năng đa dạng, đáp ứng đủ nhu cầu của sinh viên và giảng viên bộ môn Vật lý Tin học. Bên cạnh đó, mã nguồn ứng dụng sạch, dễ đọc và dễ hiểu, phù hợp để đưa vào giảng dạy.

Khuyết điểm

Cơ sở dữ liệu vẫn chưa cập nhật đầy đủ, còn thiếu thông tin của sinh viên và giảng viên. Bên cạnh đó, ứng dụng vẫn còn một số lỗi trong quá trình sử dụng có thể gây dừng và nhà phát triển chưa triển khai lên cửa hàng ứng dụng

Hướng phát triển

Nhà phát triển sửa những lỗi phát sinh trong quá trình sử dụng và tích hợp thêm một số chức năng để ứng dụng trở nên phong phú hơn. Ngoài ra, ứng dụng sẽ được triển khai lên cửa hàng để cho phép sinh viên và giảng viên có thể tải về và cài đặt.

Mã nguồn ứng dụng

Phiên bản dành cho sinh viên:



<https://github.com/tonbach18598/Flutter-PCDashboard>

Phiên bản dành cho giảng viên:



<https://github.com/tonbach18598/ReactNative-PCDashboard>

TÀI LIỆU THAM KHẢO

- [1] Bùi Ngô Tôn Bách, Phan Thanh Tùng, Nguyễn Hồng Sỹ Nguyên. Báo cáo đồ án môn lập trình di động. Bộ môn Vật lý Tin học. Khoa Vật lý – Vật lý Kỹ thuật. Trường Đại học Khoa học Tự nhiên. Đại học Quốc gia Thành phố Hồ Chí Minh, tháng 12 năm 2019.
- [2] Trang chủ tài liệu về lập trình Android [Online].
<https://developer.android.com/docs>, ngày 27 tháng 12 năm 2019
- [3] Trang chủ tài liệu về lập trình iOS [Online].
<https://developer.apple.com/documentation/>, ngày 05 tháng 05 năm 2020
- [4] Trang chủ tài liệu về lập trình Flutter [Online]. <https://flutter.dev/docs>, ngày 06 tháng 05 năm 2020
- [5] Trang chủ tài liệu về lập trình React Native [Online].
<https://reactnative.dev/docs/getting-started>, ngày 05 tháng 05 năm 2020
- [6] Trang chủ tài liệu về lập trình Xamarin [Online].
<https://docs.microsoft.com/en-us/xamarin/>, ngày 23 tháng 05 năm 2019
- [7] Trang chủ tài liệu về lập trình Ionic [Online].
<https://ionicframework.com/docs>, ngày 02 tháng 04 năm 2020
- [8] SimbirSoft. Creating a Mobile App in 2020: Native vs. Cross-platform Development [Online]. <https://medium.com/simbirsoft/creating-a-mobile-app-in-2020-native-vs-cross-platform-development-d90f25cef188>, ngày 29 tháng 01 năm 2020
- [9] Trang chủ của ngôn ngữ Dart [Online]. <https://dart.dev/>, ngày 12 tháng 06 năm 2020
- [10]Trang chủ của ngôn ngữ JavaScript [Online]. <https://javascript.com>, ngày 18 tháng 06 năm 2020