

HỌC VIỆN KỸ THUẬT MẬT MÃ
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN HỌC LẬP TRÌNH ANDROID CƠ BẢN

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI LIỆU TRÊN CLOUD
SERVER**

Sinh viên thực hiện:

Nguyễn Tiến Nguyên – CT040136

Phạm Xuân Mạnh – CT040131

Nguyễn Minh Trường – CT040150

Giảng viên hướng dẫn:

ThS. Bùi Thị Như

Khoa Công nghệ thông tin – Học viện Kỹ thuật mật mã

Hà Nội, 2023

MỤC LỤC

MỤC LỤC	2
DANH MỤC HÌNH ẢNH	i
DANH MỤC BẢNG BIỂU	ii
LỜI MỞ ĐẦU.....	iii
CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID	1
1.1 Lịch sử ra đời.....	1
1.2 Giao diện Android	2
1.3 Nhân Linux.....	2
1.4 Quản lý bộ nhớ	3
1.5 Lịch cập nhật	4
1.6 Cộng đồng mã nguồn mở	4
1.7 Bảo mật và tính riêng tư	5
1.8 Các phiên bản của Android	6
CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG	10
2.1 Ngôn ngữ lập trình C#.....	10
2.1.1 Giới thiệu về C#.....	10
2.1.2 Đặc trưng của ngôn ngữ lập trình C#	10
2.1.3 Ưu điểm nổi bật của ngôn ngữ lập trình C#	11
2.2 Ngôn ngữ lập trình Dart	11
2.2.1 Giới thiệu về Dart	11
2.2.2 Ưu nhược điểm của Dart	12
a) Ưu điểm	12
b) Nhược điểm	12
2.2.3 Các tính năng của ngôn ngữ Dart	12
2.3 Framework Flutter	13
2.3.1 Giới thiệu về Flutter	13
2.3.2 Ưu nhược điểm của Flutter.....	13
CHƯƠNG 3: MÔI TRƯỜNG LẬP TRÌNH.....	15
3.1 Giới thiệu Visual Studio Code và Visual Studio.....	15
3.1.1 Visual Studio Code (VS Code).....	15

3.1.2	Visual Studio	15
CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG		16
4.1	Sơ đồ Use Case tổng quát.....	16
4.2	Use Case phân rã	16
4.2.1	Use Case cập nhật thông tin	16
4.2.2	Use Case thao tác file	17
4.3	Biểu đồ hoạt động.....	18
4.3.1	Đăng ký.....	18
4.3.2	Đăng nhập.....	19
4.3.3	Cập nhật thông tin.....	20
4.3.4	Quên mật khẩu.....	21
4.3.5	Đăng xuất.....	21
4.3.6	Upload file	22
4.3.7	Xóa file	22
4.3.8	Xem file	Error! Bookmark not defined.
4.3.9	Upload drive	23
4.3.10	Đăng nhập bằng mã QR	24
4.3.11	Đổi mật khẩu	25
4.3.12	Download file.....	26
4.4	Đặc tả Use case.....	26
4.4.1	Use Case đăng ký tài khoản mới	26
4.4.2	Use Case đăng nhập.....	27
4.4.3	Use Case quên mật khẩu.....	27
4.4.4	Use Case cập nhật thông tin cá nhân	28
4.4.5	Use Case đổi mật khẩu	28
4.4.6	Use Case xóa file	29
4.4.7	Use Case upload file	29
4.4.8	Use Case upload file lên drive.....	29
4.4.9	Use Case login QR	30
4.4.10	Use Case tạo folder	30
4.5	Cơ sở dữ liệu	31
CHƯƠNG 5: THỰC NGHIỆM		33

5.1 Sản phẩm thực tế	33
5.2 Đánh giá ứng dụng	35
KẾT LUẬN.....	36
TÀI LIỆU THAM KHẢO	37

DANH MỤC HÌNH ẢNH

Hình 4.1 Use Case cập nhật thông tin	16
Hình 4.2 Use Case thao tác file	17
Hình 4.3 Bdhd đăng ký	18
Hình 4.4 Bdhd đăng nhập.....	19
Hình 4.5 Bdhd cập nhật thông tin	20
Hình 4.6 Bdhd quên mật khẩu.....	21
Hình 4.7 Bdhd đăng xuất.....	21
Hình 4.8 Bdhd upload file	22
Hình 4.9 Bdhd xóa file	22
Hình 4.10 Bdhd xem file	Error! Bookmark not defined.
Hình 4.11 Bdhd upload driver.....	23
Hình 4.12 Bdhd đăng nhập bằng mã QR	24
Hình 4.13 Bdhd đổi mật khẩu	25
Hình 4.14 Bdhd download file	26
Hình 4.15 Bảng AppUser	31
Hình 4.16 Bảng Cache	32

DANH MỤC BẢNG BIỂU

Bảng 4.1 Đặc tả use case đăng ký tài khoản mới.....	27
Bảng 4.2 Đặc tả use case đăng nhập	27
Bảng 4.3 Đặc tả use case quên mật khẩu	28
Bảng 4.4 Đặc tả use case cập nhật thông tin cá nhân.....	28
Bảng 4.5 Đặc tả use case đổi mật khẩu	28
Bảng 4.6 Đặc tả use case xóa file.....	29
Bảng 4.7 Đặc tả use case upload file.....	29
Bảng 4.8 Đặc tả use case upload file lên drive	30
Bảng 4.9 Đặc tả use case login QR	30
Bảng 4.10 Đặc tả use case tạo folder	31

LỜI MỞ ĐẦU

Trong thời đại 4.0 hiện nay, điện thoại di động đang trở thành một thiết bị không thể thiếu trong cuộc sống của con người. Chúng không chỉ được sử dụng để liên lạc mà còn trở thành công cụ làm việc và giải trí hàng ngày. Với sự phổ biến và tiện ích của điện thoại di động, việc xây dựng các ứng dụng cho nền tảng Android đã trở thành một ngành công nghiệp mới đầy tiềm năng, mở ra nhiều cơ hội phát triển đáng kể.

Hệ điều hành Android, ra đời và phát triển trong vài năm trở lại đây, đã thể hiện sự kế thừa và phát triển từ những hệ điều hành trước đó. Android kết hợp các công nghệ tiên tiến và mang lại nhiều ưu điểm vượt trội. Nhanh chóng trở thành đối thủ cạnh tranh mạnh mẽ với các hệ điều hành di động khác và được người dùng ưa chuộng. Sự phát triển của Android đã tạo ra một cơ sở vững chắc cho việc phát triển ứng dụng trên nền tảng này.

Hiện nay, các ứng dụng trên hệ điều hành Android đa dạng và phục vụ nhiều mục đích khác nhau. Chúng có thể là ứng dụng giải trí như trò chơi, xem phim, nghe nhạc, đọc truyện; ứng dụng học tập như học tiếng Anh, toán học; ứng dụng công việc như quản lý dự án, làm việc nhóm, và còn rất nhiều mục đích khác.

Nhóm chúng em đã lựa chọn đề tài "**Xây dựng ứng dụng quản lý tài liệu trên Cloud Server**". Đề tài này nhằm mục đích nghiên cứu và phát triển một ứng dụng giúp người dùng quản lý và truy cập tài liệu từ xa thông qua việc lưu trữ trên máy chủ đám mây (Cloud Server). Với xu hướng ngày càng phụ thuộc vào công nghệ đám mây và sự phổ biến của việc làm việc từ xa, đề tài này hứa hẹn đáp ứng nhu cầu quản lý tài liệu hiệu quả và tiện lợi cho người dùng trong môi trường kết nối mạng rộng.

Trong quá trình làm báo cáo, chúng em nhận thấy kiến thức và kinh nghiệm của chúng em còn hạn chế. Do đó, không thể tránh khỏi những sai sót và thiếu sót. Chúng em rất mong nhận được sự đóng góp và hỗ trợ từ các thầy cô giáo và các bạn để báo cáo của chúng em trở nên hoàn thiện hơn.

CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID

1.1 Lịch sử ra đời

Vào tháng 10/2003, trước khi thuật ngữ “điện thoại thông minh” được hầu hết công chúng sử dụng và vài năm trước khi Apple công bố chiếc điện thoại iPhone đầu tiên cũng như hệ điều hành iOS, công ty Android Inc được thành lập ở Palo Alto, California. Bốn người sáng lập là Rich Miner, Nick Sears, Chris White và Andy Rubin. Vào thời điểm thành lập, ông Rubin nói rằng Android Inc sẽ phát triển “thiết bị di động thông minh hơn về vị trí và sở thích của chủ sở hữu”[1].

Theo PC World, Rubin sau này đã tiết lộ trong một bài phát biểu vào năm 2013 tại Tokyo rằng hệ điều hành Android đã được ra mắt để cải thiện hệ điều hành của máy ảnh số. Rõ ràng, nhóm nghiên cứu tại Android đã không nghĩ ngay từ đầu về việc tạo ra một hệ điều hành có thể phục vụ như phần cốt lõi của một hệ thống máy tính di động hoàn chỉnh.

Nhưng sau đó, thị trường máy ảnh kỹ thuật số giảm sút, Android Inc đã quyết định chuyển sang hệ điều hành trên điện thoại di động. Như Rubin đã nói vào năm 2013, “Cùng một nền tảng, cùng hệ điều hành chúng tôi xây dựng cho máy ảnh, nó đã trở thành Android cho điện thoại di động.”

Năm 2005, chương lớn tiếp theo trong lịch sử của Android được thực hiện khi Google mua lại công ty gốc. Ông Andy Rubin và các thành viên sáng lập khác vẫn tiếp tục phát triển hệ điều hành dưới quyền chủ sở hữu mới của họ. Quyết định này được đưa ra để sử dụng Linux làm nền tảng cho hệ điều hành Android và điều đó cũng có nghĩa là Android sẽ được cung cấp miễn phí cho các nhà sản xuất điện thoại di động của bên thứ ba. Google và nhóm Android cảm thấy công ty có thể kiếm tiền với các dịch vụ khác sử dụng hệ điều hành, bao gồm cả ứng dụng.

Năm 2007, Apple công bố chiếc iPhone đầu tiên với thế giới và đã tự tay thiết lập một kỷ nguyên mới dành cho điện thoại di động. Cùng thời điểm, Google vẫn đang làm việc hết công suất với Android và đảm bảo mọi thông tin đều tuyệt mật. Đến tháng 11 cùng năm, công ty bắt đầu từ từ hé lộ kế hoạch cạnh tranh “thắng mặt” với Apple và các nền tảng di động khác. Google đã tận dụng sự hình thành của một thứ có tên là Liên minh Thiết bị cầm tay mở (Open Handset Alliance), vốn dĩ có sự góp mặt của nhiều nhà sản xuất điện thoại như HTC, Motorola, nhà sản xuất chip như Qualcomm và Texas Instruments, cũng như nhà mạng lớn như T-Mobile.

Hệ điều hành Android được chính thức ra mắt từ năm 2007 cùng với tuyên bố thành lập Liên minh Thiết bị cầm tay mở. Tháng 9/2008 smartphone chạy Android đầu tiên đã trình làng - T-Mobile G1, có tên khác là HTC Dream. Vào thời gian đầu, rất nhiều tính năng cơ bản bị thiếu sót như: bàn phím ảo, cảm ứng đa điểm và tính năng mua ứng dụng vẫn chưa xuất hiện. Tuy nhiên, một số tính năng cũng như giao diện đặc sản của hệ điều

hành này khởi nguồn từ chiếc G1 và trở thành những yếu tố không thể thiếu trên Android sau này.

1.2 Giao diện Android

Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình [1]. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà, thường dùng tính năng rung của thiết bị để tạo phản hồi rung cho người dùng. Những thiết bị phần cứng bên trong như gia tốc kế, còn quay hồi chuyển và cảm biến khoảng cách được một số ứng dụng sử dụng để phản hồi một số hành động khác của người dùng, ví dụ như điều chỉnh màn hình từ chế độ hiển thị dọc sang chế độ hiển thị ngang tùy theo vị trí của thiết bị, hoặc cho phép người dùng lái xe đua bằng xoay thiết bị giống như đang điều khiển vô lăng.

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng (icon) và tiện ích (widget); biểu tượng ứng dụng sẽ mở ứng dụng tương ứng, còn tiện ích hiển thị những nội dung sống động, cập nhật tự động như dự báo thời tiết, hộp thư của người dùng, hoặc những mẫu tin thời sự ngay trên màn hình chính. Màn hình chính có thể gồm nhiều trang xem được bằng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích. Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi “chủ đề” của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone chẳng hạn. Phần lớn những nhà sản xuất và một số nhà mạng thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể “kéo” xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện. Trong các phiên bản đời đầu, người dùng có thể nhấn vào thông báo để mở ra ứng dụng tương ứng, về sau này các thông tin cập nhật được bổ sung thêm tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi nhỡ mà không cần phải mở ứng dụng gọi điện ra. Thông báo sẽ luôn nằm đó cho đến khi người dùng đã đọc hoặc xóa nó đi.

1.3 Nhân Linux

Android có một hạt nhân dựa trên nhân Linux phiên bản 2.6 [1]. Kể từ Android 4.0 Ice Cream Sandwich (bánh ngọt kẹp kem) trở về sau, là phiên bản 3.x, với middleware, thư viện và API viết bằng C, còn phần mềm ứng dụng chạy trên một nền tảng ứng dụng gồm các thư viện tương thích với Java dựa trên Apache Harmony. Android sử dụng máy ảo

Dalvik với một trình biên dịch động để chạy “mã dex” (Dalvik Executable) của Dalvik, thường được biên dịch sang Java bytecode. Nền tảng phần cứng chính của Android là kiến trúc ARM. Người ta cũng hỗ trợ x86 thông qua dự án Android x86, và Google TV cũng sử dụng một phiên bản x86 đặc biệt của Android.

Nhân Linux dùng cho Android đã được Google thực hiện nhiều thay đổi về kiến trúc so với nhân Linux gốc. Android không có sẵn X Window System cũng không hỗ trợ các thư viện GNU chuẩn, nên việc chuyển các ứng dụng hoặc thư viện Linux có sẵn sang Android rất khó khăn. Các ứng dụng C đơn giản và SDL cũng được hỗ trợ bằng cách chèn những đoạn shim Java và sử dụng tương tự JNI, như khi người ta chuyển Jagged Alliance 2 sang Android.

Một số tính năng cũng được Google đóng góp ngược vào nhân Linux, đáng chú ý là tính năng quản lý nguồn điện có tên wakelock, nhưng bị những người lập trình chính cho nhân từ chối vì họ cảm thấy Google không có ý định sẽ tiếp tục bảo trì đoạn mã do họ viết. Google thông báo vào tháng 4 năm 2010 rằng họ sẽ thuê hai nhân viên để làm việc với cộng đồng nhân Linux, nhưng Greg Kroah-Hartman, người bảo trì nhân Linux hiện tại của nhánh ổn định đã nói vào tháng 12 năm 2010 rằng ông ta lo ngại rằng Google không còn muốn đưa những thay đổi của mình vào Linux dòng chính nữa. Một số lập trình viên Android của Google tỏ ý rằng “nhóm Android thấy chán với quy trình đó”, vì nhóm họ không có nhiều người và có nhiều việc khẩn cấp cần làm với Android hơn.

Vào tháng 12 năm 2011, nhắm tới việc đưa một số driver, bản vá và tính năng của Android ngược vào nhân Linux, bắt đầu từ Linux 3.3, Linux cũng đưa tính năng autosleep (tự nghỉ hoạt động) và wakelocks vào nhân 3.5, sau nhiều nỗ lực phối trộn trước đó. Tương tác thì vẫn vậy nhưng bản hiện thực trên Linux dòng chính cho phép hai chế độ nghỉ: bộ nhớ (dạng nghỉ truyền thống mà Android sử dụng), và đĩa (là ngủ đông trên máy tính để bàn). Việc trộn sẽ hoàn tất kể từ nhân 3.8, Google đã công khai kho mã nguồn trong đó có những đoạn thử nghiệm đưa Android về lại nhân 3.8.

Việc Android có được xem là một bản phân phối Linux hay không vẫn còn là vấn đề tranh cãi, tuy được Linux Foundation và Chris DiBona, trưởng nhóm mã nguồn mở Google ủng hộ. Một số khác, như linux-magazine.com thì không đồng ý, do Android không hỗ trợ nhiều công cụ GNU, trong đó có glibc.

1.4 Quản lý bộ nhớ

Vì các thiết bị Android chủ yếu chạy bằng pin, nên Android được thiết kế để quản lý bộ nhớ (RAM) để giảm tối đa điện năng tiêu thụ, trái với hệ điều hành máy tính để bàn luôn cho rằng máy tính sẽ có nguồn điện không giới hạn. Khi một ứng dụng Android không còn được sử dụng, hệ thống sẽ tự động nhưng nó trong bộ nhớ - trong khi ứng dụng về mặt kỹ thuật vẫn “mở”, những ứng dụng này sẽ không tiêu thụ bất cứ tài nguyên nào (như năng lượng pin hay năng lượng xử lý) và nằm đó cho đến khi nó được cần đến. Cách làm như vậy có lợi ích kép là vừa làm tăng khả năng phản hồi nói chung cả thiết bị Android, vì ứng

dụng không nhất thiết phải đóng rồi mở lại từ đầu, vừa đảm bảo các ứng dụng nền không làm tiêu hao năng lượng một cách không cần thiết.

Android quản lý các ứng dụng trong bộ nhớ một cách tự động: khi bộ nhớ thấp, hệ thống sẽ hủy ứng dụng và tiến trình không được hoạt động lâu nhất, sắp theo thời điểm cuối mà chúng được sử dụng (tức là cũ nhất sẽ bị tắt trước). Tiến trình này được thiết kế ẩn đi với người dùng, để người dùng không cần phải quản lý bộ nhớ hoặc tự tay tắt các ứng dụng. Tuy nhiên, sự che giấu này của hệ thống quản lý bộ nhớ Android đã dẫn đến sự thịnh hành của các ứng dụng tắt chương trình của bên thứ ba trên cửa hàng Google Play; những ứng dụng kiểu như vậy được cho là có hại nhiều hơn có lợi.

1.5 Lịch cập nhật

Google đưa ra các bản cập nhật lớn cho Android theo chu kỳ từ 6 đến 9 tháng, mà phần lớn thiết bị đều có thể nhận được qua sóng không dây. So với các hệ điều hành cạnh tranh khác, như iOS, các bản cập nhật Android thường mất thời gian lâu hơn để đến với các thiết bị. Với những thiết bị không thuộc dòng Nexus và Pixel, các bản cập nhật thường đến sau vài tháng kể từ khi phiên bản được chính thức phát hành. Nguyên nhân của việc này một phần là do sự phong phú về phần cứng của các thiết bị Android, nên người ta phải mất thời gian điều chỉnh bản cập nhật cho phù hợp, vì mã nguồn chính thức của Google chỉ chạy được trên những thiết bị Nexus chủ lực của họ. Chuyển Android sang những phần cứng cụ thể là một quy trình tốn thời gian và công sức của các nhà sản xuất thiết bị, những người luôn ưu tiên các thiết bị mới nhất và thường bỏ rơi các thiết bị cũ hơn. Do đó, những chiếc điện thoại thông minh thế hệ cũ thường không được cập nhật nếu nhà sản xuất quyết định rằng nó không đáng để bỏ thời gian, bất kể chiếc điện thoại đó có khả năng chạy bản cập nhật hay không. Vấn đề này còn trầm trọng hơn khi những nhà sản xuất điều chỉnh Android để đưa giao diện và ứng dụng của họ vào, những thứ này cũng sẽ phải làm lại cho mỗi bản cập nhật. Sự chậm trễ còn được đóng góp bởi nhà mạng, sau khi nhận được bản cập nhật từ nhà sản xuất, họ còn điều chỉnh thêm cho phù hợp với nhu cầu rồi thử nghiệm kỹ lưỡng trên hệ thống mạng của họ trước khi chuyển nó đến người dùng.

1.6 Cộng đồng mã nguồn mở

Android có một cộng đồng các lập trình viên và những người đam mê rất năng động. Họ sử dụng mã nguồn Android để phát triển và phân phối những phiên bản chỉnh sửa của hệ điều hành. Các bản Android do cộng đồng phát triển thường đem những tính năng và cập nhật mới vào nhanh hơn các kênh chính thức của nhà sản xuất, tuy không được kiểm thử kỹ lưỡng cũng như không có đảm bảo chất lượng nhưng cung cấp sự hỗ trợ liên tục cho các thiết bị cũ không còn nhận được bản cập nhật chính thức hoặc mang Android vào những thiết bị ban đầu chạy một hệ điều hành khác như HP Touchpad. Các bản Android của cộng đồng thường được root sẵn và có những điều chỉnh không phù hợp với những người dùng không rành rẽ, như khả năng ép xung hoặc tăng/giảm điện áp bộ xử lý của thiết bị. CyanogenMod là firmware của cộng đồng được sử dụng phổ biến nhất, và hoạt động như một tổ chức của số đông khác.

Trước đây, nhà sản xuất thiết bị và nhà mạng tỏ ra thiếu thiện chí với việc phát triển firmware của bên thứ ba. Những nhà sản xuất lo ngại rằng các thiết bị chạy phần mềm không chính thức sẽ hoạt động không tốt và dẫn đến tốn tiền hỗ trợ. Hơn nữa, các firmware đã thay đổi như CyanogenMod đôi khi còn cung cấp những tính năng như truyền tải mạng (tethering) mà người dùng bình thường phải trả tiền cho nhà mạng để được sử dụng. Kết quả là nhiều thiết bị bắt đầu đặt ra hàng rào kỹ thuật như khóa bootloader hay hạn chế quyền truy cập root. Tuy nhiên, khi phần mềm do cộng đồng phát triển ngày càng trở nên phổ biến, và sau một thông báo của Thư viện Quốc hội Hoa Kỳ cho phép “jailbreak” (vượt rào) thiết bị di động thì các nhà sản xuất và nhà mạng mới tỏ ra mềm mỏng hơn với các nhà phát triển bên thứ ba, thậm chí một số hãng như HTC, Motorola, Samsung và Sony, còn hỗ trợ và khuyến khích phát triển. Kết quả của việc này là dần dần nhu cầu tìm ra các hạn chế phần cứng để cài đặt được firmware không chính thức đã bớt đi do ngày càng nhiều thiết bị được phát hành với bootloader đã mở khóa sẵn hoặc có thể mở khóa, tương tự như điện thoại dòng Nexus, tuy nhiên thông thường họ sẽ yêu cầu người dùng từ bỏ chế độ bảo nếu muốn làm như vậy.

1.7 Bảo mật và tính riêng tư

Các ứng dụng Android chạy trong một “hộp cát”, là một khu vực riêng rẽ với hệ thống và không được tiếp cận đến phần còn lại của tài nguyên hệ thống, trừ khi nó được người dùng trao quyền truy cập một các công khai khi cài đặt. Trước khi cài đặt ứng dụng, Play Store sẽ hiện thị tất cả các quyền mà ứng dụng đòi hỏi.

Hệ thống hộp cát và yêu cầu quyền làm giảm bớt ảnh hưởng của lỗi bảo mật hoặc lỗi chương trình có trong ứng dụng, nhưng sự bối rối của lập trình viên và tài liệu hướng dẫn còn hạn chế đã dẫn tới những ứng dụng đòi hỏi những quyền không cần thiết, do đó làm giảm đi hiệu quả của hệ thống này. Một số công ty bảo mật, như Lookout Mobile Security, AVG Technologies và McAfee đã phát hành những phần mềm diệt virus cho các thiết bị Android. Phần mềm này không có hiệu quả vì cơ chế hộp cát vẫn áp dụng cho các ứng dụng này, do vậy làm hạn chế khả năng quét sâu vào hệ thống để tìm nguy cơ.

Một nghiên cứu của một công ty bảo mật Trend Micro đã liệt kê tình trạng lạm dụng dịch vụ trả tiền là hình thức phần mềm ác ý phổ biến nhất trên Android, trong đó tin nhắn SMS sẽ bị gửi đi từ điện thoại bị nhiễm đến một số điện thoại trả tiền mà người dùng không hề hay biết. Loại phần mềm ác ý khác hiển thị quảng cáo không mong muốn và gây khó chịu trên thiết bị. Đe dọa bảo mật trên Android được cho là tăng rất nhanh theo cấp số mũ. Tuy nhiên các kỹ sư của Google phản bác rằng hiểm họa từ phần mềm độc hại và virus đã bị thổi phồng bởi các công ty bảo mật đang lợi dụng sự sợ hãi để bán phần mềm của họ cho người dùng.

Google hiện đang sử dụng bộ quét phần mềm độc hại Google Bouncer để theo dõi và quét các ứng dụng trên Play Store. Nó sẽ đánh dấu các phần mềm bị nghi ngờ và cảnh báo cho người dùng về những vấn đề có thể xảy ra trước khi họ tải nó về máy. Android phiên bản 4.2 được phát hành vào năm 2012 cùng với các tính năng bảo mật được cải thiện,

bao gồm một bộ quét phần mềm độc hại được cài sẵn vào hệ thống, hoạt động cùng với Google Play nhưng cũng có thể quét các ứng dụng được cài đặt từ nguồn thứ ba. Một hệ thống cảnh báo sẽ thông báo cho người dùng một ứng dụng cố gắng gửi tin nhắn vào số tính tiền, chặn tin nhắn đó lại cho đến khi người dùng xác nhận cho phép nó.

Điện thoại thông minh Android có khả năng báo cáo vị trí của điểm truy cập kết nối Wi-Fi, phát hiện ra việc di chuyển của người dùng điện thoại, để xây dựng những cơ sở dữ liệu có chứa vị trí của hàng trăm triệu điểm truy cập. Những cơ sở này tạo nên một bản đồ điện tử để tìm vị trí điện thoại thông minh. Bản chất mã nguồn mở của Android cho phép nhưng nhà thầu bảo mật lấy những thiết bị sẵn có rồi điều chỉnh để sử dụng ở mức độ bảo mật cao hơn.

1.8 Các phiên bản của Android

Kể từ khi tung ra phiên bản Android đầu tiên năm 2008 đến nay hệ điều hành Android đã vượt qua mốc 2 tỷ thiết bị [3]. Qua 15 năm phát triển, Google đã ghi được những bước tiến đáng kể, thể hiện ở con số cứ 10 điện thoại thông minh được bán ra trên toàn cầu thì có 9 chiếc là chạy hệ điều hành Android.

- **Phiên bản Android 1.0**

Android 1.0 (2008) bao gồm một số tính năng cơ bản như liên lạc, tin nhắn và các ứng dụng cơ bản khác như nghe nhạc và xem hình ảnh.

- **Phiên bản Android 1.5 Cupcake**

Android 1.5 Cupcake (2009) là bản cập nhật lớn đầu tiên của Android. Cupcake đã thêm các tiện ích cho màn hình chính, bản phím ảo, quay video trong máy ảnh, chức năng sao chép và dán vào trình duyệt web.

- **Phiên bản Android 2.0 Eclair**

Android 2.0 Eclair (2009) hỗ trợ tài khoản Google, cho phép người dùng tìm kiếm nội dung theo từ khóa trong các tin nhắn văn bản, đồng thời bổ sung hỗ trợ đa chạm và máy ảnh được cải tiến với đèn flash, zoom số.

- **Phiên bản Android 2.2 Froyo**

Android 2.2 Froyo (2010) giới thiệu Flash Player 10.1, cho phép điện thoại phát video và phát trực tuyến âm thanh. Máy ảnh flash tương thích Bluetooth giúp người dùng có thể sử dụng điện thoại của mình làm điểm phát sóng Wi-Fi.

- **Phiên bản Android 2.3 Gingerbread**

Android 2.3 Gingerbread (2011), Android được biết đến rộng rãi hơn nhờ tính năng giao tiếp trường gần (NFC), cho phép điện thoại thông minh kết nối với các thiết bị lân cận khác. Phiên bản này cũng cho phép gọi video bằng máy ảnh trước và thêm trình quản lý tải xuống.

- **Phiên bản Android Honeycomb 3.0**

Android Honeycomb 3.0 (2011) là bản cập nhật đầu tiên chỉ dành cho máy tính bảng, hỗ trợ đồ họa 3D, các tab trình duyệt cạnh nhau, trò chuyện video với Google Talk, chia sẻ kết nối Bluetooth và chế độ toàn màn hình trong thư viện ảnh.

- **Phiên bản Android Ice Cream Sandwich 4.0**

Android Ice Cream Sandwich 4.0 (2011) sáp nhập các hệ điều hành điện thoại và máy tính bảng. ICS cũng thêm nhận dạng khuôn mặt để mở khóa điện thoại, phản hồi văn bản, tự động trả lời các cuộc gọi bị từ chối và hiệu ứng video trực tiếp trong camera.

- **Phiên bản Android Jelly Bean 4.1 - 4.3.1**

Android Jelly Bean 4.1 (2012) có hiệu suất nhanh hơn, mượt mà hơn nhờ “Project Butter”, cho phép người dùng tương tác nhiều hơn, thông báo có thể mở rộng, trình duyệt mặc định là Chrome... Hai phiên bản còn lại cũng có tên là Jelly Bean, được phát hành tương ứng vào tháng 10/2012 và tháng 7/2013, trong đó phiên bản 4.2 gồm tối ưu hóa, hỗ trợ nhiều người dùng cho máy tính bản, widget cho màn hình khóa, tùy chỉnh nhanh và screen saver, còn phiên bản 4.3 gồm các cải tiến và cập nhật nội bộ cho nền tảng Android.

- **Phiên bản Android 4.4 KitKat**

Android 4.4 KitKat (2013) đã được tích hợp Google Now cho phép người dùng truy cập thông tin thời tiết, địa điểm, giá vé máy bay và nhiều thông tin khác một cách dễ dàng hơn. Nó cũng có nhiều cải tiến về bảo mật, bao gồm tính năng SELinux để ngăn chặn các cuộc tấn công bảo mật, cũng như các tính năng bảo mật mới như tính năng bảo mật truy cập vào thiết bị.

Android 4.4 KitKat đã được tối ưu hóa để chạy nhanh hơn trên các thiết bị có cấu hình thấp hơn, bao gồm các thiết bị có bộ nhớ RAM ít hơn 1GB. Mặc dù giao diện người dùng mới với nhiều thay đổi trong các biểu tượng và màu sắc, cũng như các tính năng đồ họa mới như các hiệu ứng mờ và lồng nhau. Đồng thời nó đã bổ sung tính năng hỗ trợ in trực tuyến, cho phép người dùng in tài liệu, hình ảnh và nội dung khác từ thiết bị Android của họ mà không cần kết nối với máy tính.

- **Phiên bản Android 5.0 Lollipop**

Android 5.0 Lollipop (2014) có thêm tính năng tạm dừng hoạt động cho phép người dùng tạm dừng các ứng dụng không được sử dụng để tăng tuổi thọ pin và tăng tốc độ hệ thống. Vấn đề bảo mật trong phiên bản này cũng được cải tiến với tính năng xác thực hai yếu tố và tính năng bảo mật truy cập vào thiết bị.

- **Phiên bản Android 6.0 Marshmallow**

Android 6.0 Marshmallow (2015) bắt đầu có chế độ Doze để tiết kiệm pin. Đồng thời bổ sung thêm hỗ trợ tích hợp cho đầu đọc vân tay, USB Type-C và chế độ 4K cho các ứng dụng.

- **Phiên bản Android 7.0 Nougat**

Android 7.0 Nougat (2016) có thêm tính năng Multi-Window cho phép người dùng chia màn hình thành hai phần để chạy hai ứng dụng khác nhau cùng một lúc. Và một vài tính năng khác như: tính năng “Doze on the Go” - tăng tuổi thọ cho thiết bị bằng cách tự động ngủ khi thiết bị không được sử dụng và đang di chuyển; “Direct Reply” - cho phép người dùng trả lời tin nhắn mà không cần mở ứng dụng nhắn tin; “Quick Switch” - cho phép người dùng nhanh chóng chuyển giữa các ứng dụng đang mở bằng cách nhấn và giữ nút “back”; “Data Saver” - cho phép người dùng điều chỉnh các ứng dụng để giới hạn lưu lượng dữ liệu được sử dụng khi mạng không tốt.

- **Phiên bản Android 8.0 Oreo**

Android 8.0 Oreo (2017) được tối ưu hóa để cho phép các thiết bị chạy nhanh hơn và độ bền pin tốt hơn. Đồng thời có khả năng phân tích và quản lý bộ nhớ tốt hơn, điều này sẽ giúp cho các thiết bị chạy mượt hơn và không bị giật lag.

- **Phiên bản Android Pie 9.0**

Android 9 Pie (2018) là phiên bản phát hành lớn thứ 9 của hệ điều hành Android. Ở Android 9, Google đã đưa ra tính năng Adaptive Battery nhằm tối ưu hóa tuổi thọ pin trên điện thoại. Dựa vào AI, tính năng này sẽ dự đoán các ứng dụng mà người dùng có thể sẽ không sử dụng trong 1 khoảng thời gian tới. Từ đó, thiết bị của bạn sẽ hạn chế cung cấp pin cho những ứng dụng này. Vì vậy, thời lượng pin trên smartphone của người dùng sẽ tránh bị tiêu hao vô những ứng dụng không cần thiết mà chỉ tập trung vào các ứng dụng ưa thích của người dùng.

- **Phiên bản Android 10**

Khác với các phiên bản Android từ thế hệ 7 trở đi sẽ được đặt tên theo 1 món bánh hoặc kẹo ngọt nào đó thì hệ điều hành thế hệ thứ 10 được Google phát hành vào tháng 9/2019 có cái tên khá đơn giản là Android 10, không đặt theo tên 1 món ăn nào cả. Nhắc đến Android 10 không thể nào không nhắc tới Dark Mode, cho phép người dùng bật và tắt chế độ này ngay trên menu cài đặt nhanh một cách dễ dàng. Trên Android 10 chúng ta sẽ không còn thấy sự xuất hiện của 3 nút bấm quen thuộc mà thay vào đó là 1 thanh ngang nhỏ dùng để điều hướng theo cử chỉ của người dùng. Với Android 10, người dùng có thể vuốt lên để trở về màn hình chính, vuốt từ cạnh trái hoặc cạnh phải để quay lại, khá giống trên các thiết bị iPhone.

- **Phiên bản Android 11**

Nối tiếp Android 10 thì vào 09/2020, Google đã chính thức công bố hệ điều hành di động thế hệ thứ 11 với tên gọi cũng đơn giản không kém phần là Android 11. Có lẽ ở các giao diện tùy chỉnh khác đã có sẵn tính năng này từ trước nhưng với các smartphone sử dụng Android thuần thì phải đến thế hệ thứ 11 này, người dùng mới được quay màn hình trực tiếp mà không cần thông qua ứng dụng thứ 3. Tính năng này cho phép người dùng vừa quay màn hình vừa ghi âm giọng nói cũng như âm thanh của hệ thống cùng 1 lúc. Người dùng sử dụng smartphone đang chạy Android 11 có quyền cho phép các ứng dụng truy cập

vào 1 vài quyền riêng tư nhưng chỉ duy nhất 1 lần mà thôi. Sau khi sử dụng xong, hệ thống sẽ ngay lập tức thu hồi quyền đó trong lần chạy tiếp theo. Ngoài các tính năng kể trên, phiên bản hệ điều hành Android 11 còn cung cấp cho người dùng thêm 1 vài tính năng hay ho khác như: Tăng kích thước quay video mở rộng, bong bóng chat, thay đổi giao diện của trình phát nhạc, đề xuất các ứng dụng hay dùng, lịch sử thông báo...

- **Phiên bản Android 12**

Android 12 chính thức đã được Google ra mắt vào 20/10/2021. Một trong những thay đổi đáng chú ý nhất mà Google đã thực hiện trong Android 12 là thiết kế Material You. Bố cục và thiết kế mới mang tính biểu cảm và trông to hơn so với các bản phát hành Android trước đây. Ngoài ra Android 12 mang đến nhiều công cụ khác nhau, cho phép bạn phối hợp màu sắc và mở rộng ra ngoài các biểu tượng ứng dụng, menu, widget... Google cũng đã thiết kế lại các widget, tương lai công ty sẽ phát hành widget cho các ứng dụng của bên thứ 3. Tính năng mới đáng nói trên Android 12 là nó cho phép người dùng khởi tạo chơi game ngay khi tải về. Nói 1 cách dễ hiểu, người dùng có thể chơi ngay sau khi vừa nhấn tải game về. Điều này giúp giảm thiểu thời gian đợi chờ tải về của các tựa game có dung lượng lớn.

CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG

2.1 Ngôn ngữ lập trình C#

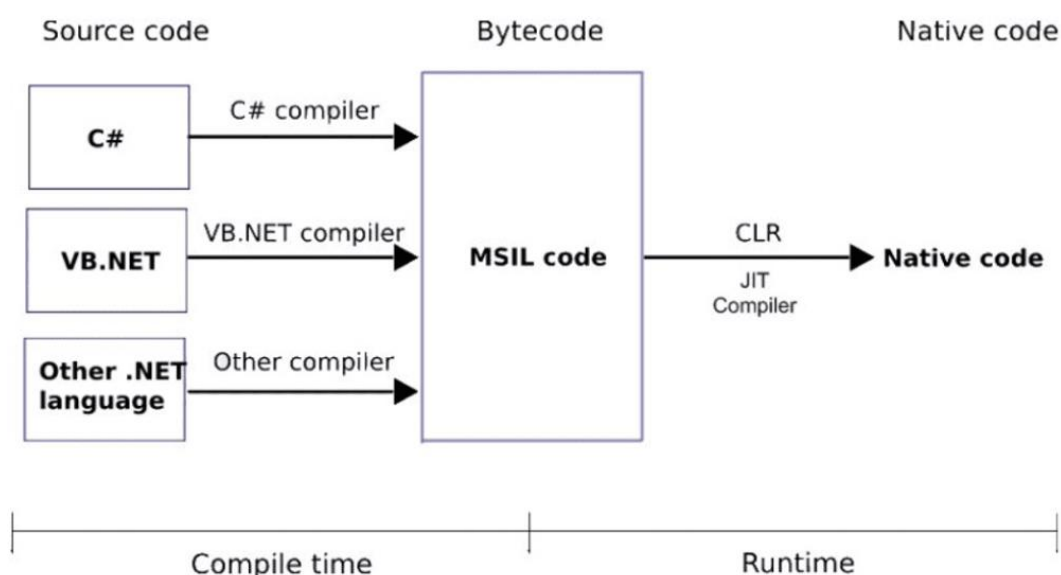
2.1.1 Giới thiệu về C#

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.

Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code thành native code.



C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

2.1.2 Đặc trưng của ngôn ngữ lập trình C#

Đơn giản: Đặc trưng đầu tiên của C# là loại bỏ những vấn đề phức tạp đã có trong Java và C++ như macro, template, tính đa kế thừa, lớp cơ sở ảo (hay còn gọi virtual base

class). Các cú pháp, toán tử, biểu thức và cả tính năng của C# khá tương đương Java và C++ song đã qua cải tiến nên đơn giản hơn nhiều.

Hiện đại: C# sở hữu nhiều khả năng như xử lý ngoại lệ, tự động trong thu gom bộ nhớ, bảo mật mã nguồn, dữ liệu mở rộng,... Đây là tất cả những đặc điểm được mong chờ ở một ngôn ngữ lập trình hiện đại.

Hướng đối tượng: C# là một trong những ngôn ngữ được đánh giá là thuần hướng đối tượng. Nó sở hữu cả 4 tính chất quan trọng, đặc trưng là tính kế thừa, tính đóng gói, tính trừu tượng và tính đa hình.

Ít từ khóa: Một trong những đặc trưng cơ bản của C# là ít từ khóa. Từ khóa được dùng trong ngôn ngữ này chỉ nhằm mục đích mô tả thông tin. Tuy ít từ khóa song C# vẫn rất mạnh mẽ. Lập trình viên có thể sử dụng nó để thực hiện mọi nhiệm vụ.

Mã nguồn mở: C# là một trong những ngôn ngữ lập trình mã nguồn mở, được phát triển, điều hành một cách độc lập với Microsoft. Đây là một trong những nét độc đáo khiến ngôn ngữ này được biết đến và ưa chuộng rộng rãi.

Đa nền tảng: C# là ngôn ngữ được sử dụng trong lập trình trên nhiều nền tảng. Các ứng dụng hoặc website được xây dựng bằng ngôn ngữ này có thể hoạt động tốt trên nhiều nền tảng như Windows, Linux và Mac.

Tiến hóa: C# vẫn đang được nâng cấp và cho ra mắt các phiên bản mới với nhiều tính năng vượt trội và khả năng làm việc mạnh mẽ hơn. Hiện C# có thể làm việc với console, điện toán đám mây, phần mềm học máy,...

2.1.3 Ưu điểm nổi bật của ngôn ngữ lập trình C#

C# ‘gây sốt’ trong giới công nghệ nhờ những ưu điểm nổi bật như:

- C# gần gũi với Java và C++, nhờ vậy mà nó kế thừa được tất cả các ‘tinh hoa’ của hai ngôn ngữ này. Lập trình viên có kiến thức về hai ngôn ngữ trên có thể dùng C# dễ dàng.
- Cộng đồng những người sử dụng C# đang phát triển với tốc độ chóng mặt. Lập trình viên có thể tham khảo và tìm kiếm thông tin dễ dàng.
- C# có khả năng tạo ra mọi ứng dụng và phổ biến trong giới lập trình. Đặc biệt là lập trình game.

2.2 Ngôn ngữ lập trình Dart

2.2.1 Giới thiệu về Dart

Dart là ngôn ngữ lập trình hướng đối tượng, được Google phát hành có mục đích chung với mã nguồn mở và được ECMA chấp thuận làm tiêu chuẩn. Ngôn ngữ này được sử dụng để xây dựng các ứng dụng web, di động trên nhiều nền tảng khác nhau. Hiện tại, Dart hỗ trợ hầu hết các tính năng phổ biến của ngôn ngữ lập trình như lớp, giao diện, chức năng,... tương tự những ngôn ngữ khác.

Ngôn ngữ Dart được sử dụng để phát triển, xây dựng các ứng dụng web, di động với hiệu suất cao. Khắc phục những hạn chế của JavaScript như yêu cầu cơ sở mã nguồn mở, không có khả năng mở rộng,... Một số ứng dụng phổ biến và quan trọng của Dart có thể kể đến như:

- Đóng vai trò là ngôn ngữ lập trình cơ bản cho khung Flutter, và được sử dụng để xây dựng các ứng dụng di động có thể mở rộng.
- Do là ngôn ngữ có mục đích chung, nó được sử dụng để xây dựng, lập trình các ứng dụng di động gốc cho iOS, Android hoặc cho máy tính để bàn, máy chủ.

2.2.2 Ưu nhược điểm của Dart

a) Ưu điểm

Dart có những ưu điểm nổi bật như sau:

- Là ngôn ngữ mở rộng, linh hoạt, tạo điều kiện tích cực cho việc biên dịch trở nên nhanh chóng hơn.
- Có tính ổn định tốt, được sử dụng để xây dựng các ứng dụng về thời gian thực với hiệu suất cao.
- Hỗ trợ cả biên dịch Vừa đúng lúc (JIT) và biên dịch Trước thời hạn (AOT).
- Có thể thích ứng nhanh chóng với các quy trình công việc có sự thay đổi.

b) Nhược điểm

Một số hạn chế khi sử dụng Dart như:

- Là ngôn ngữ lập trình mới nên hiện tại cộng đồng người sử dụng, hỗ trợ có quy mô nhỏ, chưa có quá nhiều tài nguyên phục vụ cho công việc học tập.
- Chỉ bao gồm một lớp đối tượng duy nhất, không hỗ trợ quá trình lặp lại mã.
- Không hỗ trợ đổi tên hàm mà không thực hiện viết câu lệnh gán mới.

2.2.3 Các tính năng của ngôn ngữ Dart

Hướng đối tượng: Dart sử dụng dữ liệu dưới dạng đối tượng thay vì coi dữ liệu là hàm hoặc logic và hỗ trợ cả các khái niệm lập trình hướng đối tượng cơ bản.

Không đồng bộ: Dart không có tính đồng bộ nhưng cho phép đồng thời nâng cao. Hiểu đơn giản là bạn có thể chạy đồng thời nhiều tác vụ với Dart bằng cách sử dụng thể độc lập (phân lập).

Các thư viện tích hợp: Dart bao gồm các thư viện tích hợp mở rộng như Input – Output (IO), Software Development Kit (SDK),... Bạn có thể tìm các đoạn mã code viết sẵn trong những thư viện này và tối ưu theo mục đích của mình.

Hỗ trợ đa nền tảng: Dart có thể hoạt động trên nhiều hệ điều hành khác nhau Windows, Linux, macOS cũng như nhiều hệ điều hành khác bởi tính năng Máy ảo Dart.

2.3 Framework Flutter

2.3.1 Giới thiệu về Flutter

Flutter là một framework mã nguồn mở phát triển bởi Google, được sử dụng để xây dựng ứng dụng di động, web và desktop đa nền tảng. Nó cho phép nhà phát triển tạo giao diện người dùng đẹp và tương tác, sử dụng một ngôn ngữ lập trình duy nhất là Dart.

2.3.2 Ưu nhược điểm của Flutter

Mỗi giải pháp đều có những thế mạnh và yếu điểm riêng, Flutter cũng không ngoại lệ. Nhưng trước tiên phải kể đến một số lợi ích vượt trội khi sử dụng Flutter:

- Flutter chạy nhanh giúp tiết kiệm thời gian, công sức và tiền bạc. Giống như bất kỳ công nghệ đa nền tảng nào khác, Flutter cho phép bạn sử dụng cùng một cơ sở mã để xây dựng các ứng dụng iOS và Android riêng biệt. Điều này sẽ đẩy nhanh toàn bộ quá trình phát triển mà không cần phân tích hai cơ sở mã khác nhau cho cùng một nền tảng.
- "Hot reload" (tải nóng/ nhanh) của Flutter giải pháp bạn thực hiện các thay đổi đối với mã code và xem được kết quả ngay lập tức trong bản xem trước ứng dụng mà không cần đọc lại mã. Bằng cách này, bạn có thể dễ dàng sửa lỗi và thử nghiệm với các phần tử và tính năng UI khác nhau.
- Tùy chỉnh toàn bộ & kết xuất nhanh nhờ cấu trúc phân lớp của Flutter. Ứng dụng này cung cấp quyền kiểm soát mọi pixel trên màn hình cũng như không giới hạn người dùng thêm và tạo hoạt ảnh trong thiết kế đồ họa, video, văn bản và điều khiển.
- Flutter cũng áp dụng cho web và cung cấp tài liệu thích hợp cho phép bạn kiểm tra cách các điều khiển gốc hoạt động.
- Flutter tách UI khỏi các điều khiển gốc giúp loại bỏ lỗi không tương thích (dù ít xảy ra) từ phía nhà sản xuất. UI riêng biệt cũng tự động đem đến một sự đồng nhất trên tất cả các phiên bản hệ thống.

Mặc dù có nhiều ưu điểm, song qua thử nghiệm, Flutter cũng có những yếu điểm nhất định.

- Flutter vẫn chưa thật hoàn thiện. Vì là một ứng dụng mới nên Flutter vẫn chưa đạt mức hoàn hảo. Thực tế, nhiều tính năng nâng cao của Flutter vẫn chưa được hỗ trợ; nhiều thư viện chưa được thử nghiệm chính thức còn tồn tại hạn chế khi so sánh với các bản sao gốc (như là Google Maps).
- Dart khá "non nớt". Về cơ bản Dart khá giống với Swift và Kotlin, nhưng có ít tính năng hơn, hoặc những tính năng hiện có chưa được toàn diện.
- Các ứng dụng Flutter khá "nặng". Chúng chiếm nhiều dung lượng và mất nhiều thời gian để tải xuống hoặc cập nhật.
- Giao diện không giống 100% so với phiên bản gốc. Về cơ bản, Flutter không tạo ra các thành phần gốc mà sao chép không hoàn toàn các thiết kế Material Design của Android và các thành phần riêng của iOS bằng thư viện Cupertino. Thư viện này sẽ

hiển thị, đặc biệt với các phiên bản hệ thống chứa các trường văn bản hoặc các nút - những thành phần biến đổi bên ngoài nhưng không thay đổi bên trong Flutter.

- Hướng dẫn phát triển ứng dụng Flutter chưa được đồng nhất, điều này có thể gây nhiều khó khăn khi xây dựng các phần mềm mang tính phức tạp.
- Framework thay đổi nhanh chóng gây khó khăn cho việc duy trì mã. Thêm vào đó, Flutter chưa chắc sẽ được ứng dụng trong tương lai khi Google liên tục loại bỏ các dự án của ứng dụng này.

CHƯƠNG 3: MÔI TRƯỜNG LẬP TRÌNH

3.1 Giới thiệu Visual Studio Code và Visual Studio

Visual Studio Code và Visual Studio là hai môi trường phát triển tích hợp (Integrated Development Environment - IDE) phổ biến được sử dụng trong lập trình.

3.1.1 Visual Studio Code (VS Code)

Visual Studio Code là một trình biên tập mã nguồn mở và miễn phí được phát triển bởi Microsoft. Đây là một IDE nhẹ nhưng mạnh mẽ, hỗ trợ đa nền tảng (Windows, macOS, Linux) và rất linh hoạt cho các ngôn ngữ lập trình khác nhau.

Tính năng chính của Visual Studio Code bao gồm:

- Giao diện người dùng đơn giản và thân thiện với người dùng.
- Hỗ trợ nhiều ngôn ngữ lập trình phổ biến như C++, C#, Java, Python, JavaScript, HTML, CSS và nhiều ngôn ngữ khác.
- Cung cấp IntelliSense (gợi ý mã) thông minh và gỡ lỗi tích hợp.
- Hỗ trợ tích hợp Git và quản lý phiên bản.
- Khả năng mở rộng với các extension và theme để tùy chỉnh theo nhu cầu của người dùng.
- Hỗ trợ tích hợp Terminal để thao tác với dòng lệnh trong môi trường phát triển.

3.1.2 Visual Studio

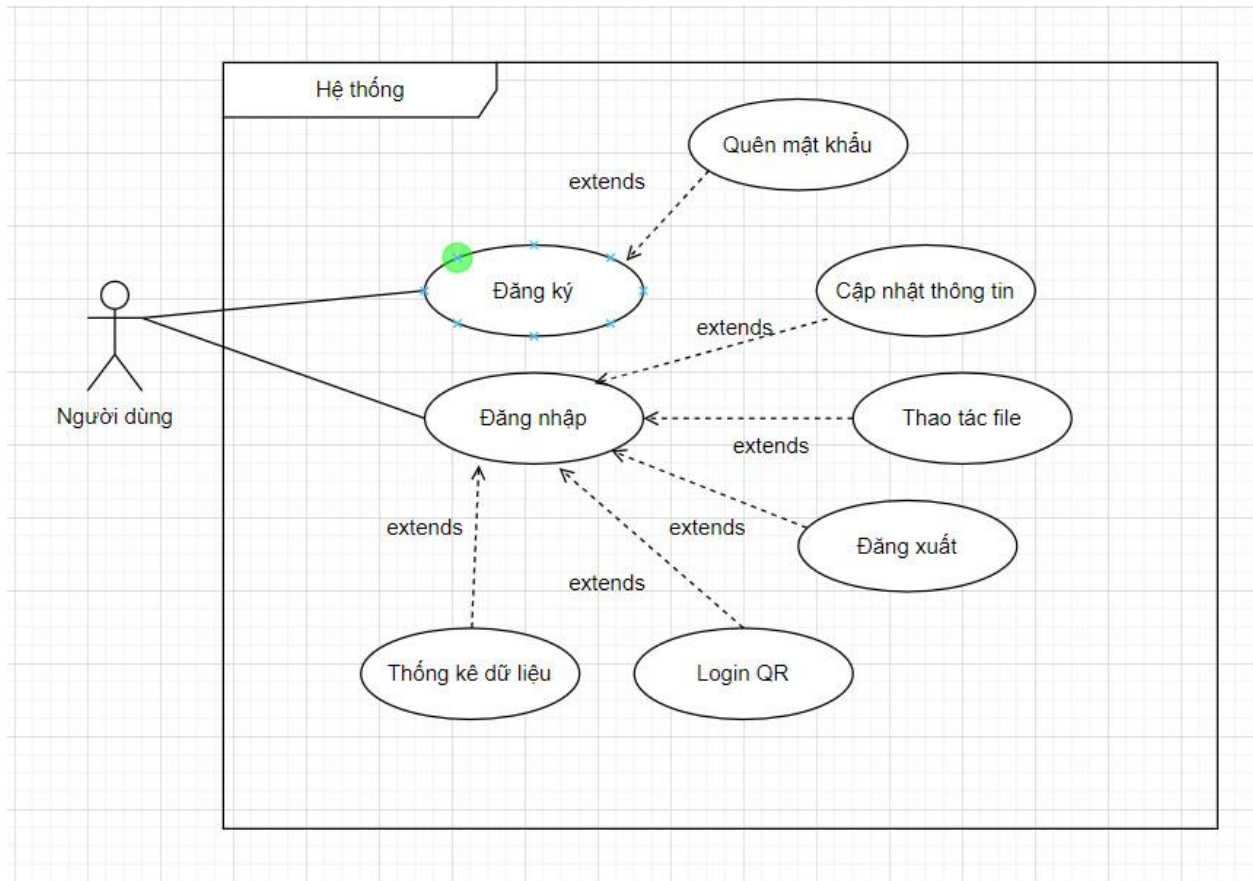
Visual Studio là một IDE phát triển bởi Microsoft, chủ yếu được sử dụng cho việc phát triển ứng dụng trên nền tảng Windows. Visual Studio cung cấp một loạt các công cụ và tính năng mạnh mẽ để hỗ trợ quá trình phát triển phần mềm, từ thiết kế, viết mã, gỡ lỗi cho đến triển khai ứng dụng.

Tính năng chính của Visual Studio bao gồm:

- Hỗ trợ đa ngôn ngữ và nền tảng, bao gồm C++, C#, Visual Basic, JavaScript, HTML, CSS và nhiều ngôn ngữ khác.
- Giao diện đồ họa dễ sử dụng và linh hoạt.
- Hỗ trợ IntelliSense và gỡ lỗi tích hợp, giúp tăng năng suất và chất lượng phát triển.
- Cung cấp công cụ và môi trường phát triển đầy đủ cho việc xây dựng ứng dụng di động, web, desktop và các loại ứng dụng khác.
- Hỗ trợ tích hợp Git và quản lý phiên bản.
- Có khả năng mở rộng với các extension và công cụ của bên thứ ba.

CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

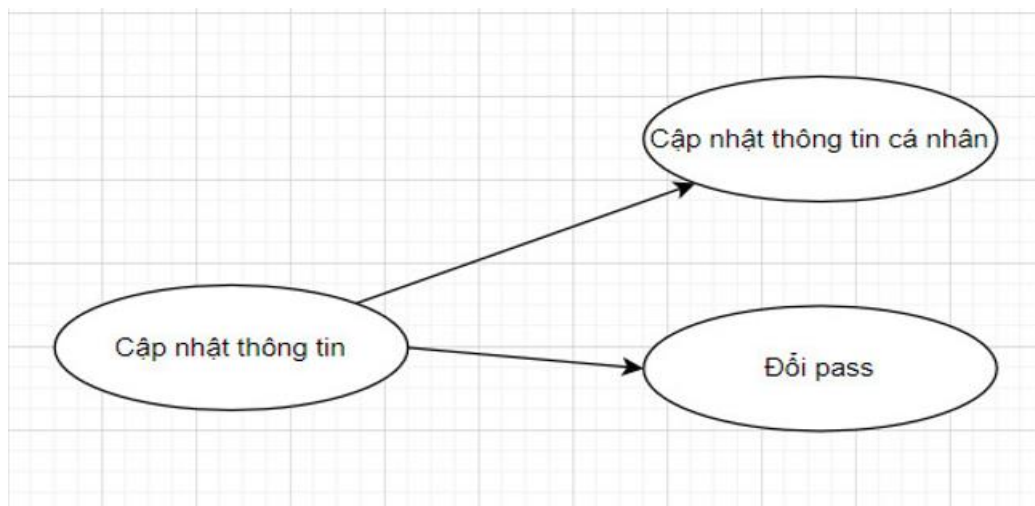
4.1 Sơ đồ Use Case tổng quát



Hình 3.1 Use Case tổng quát

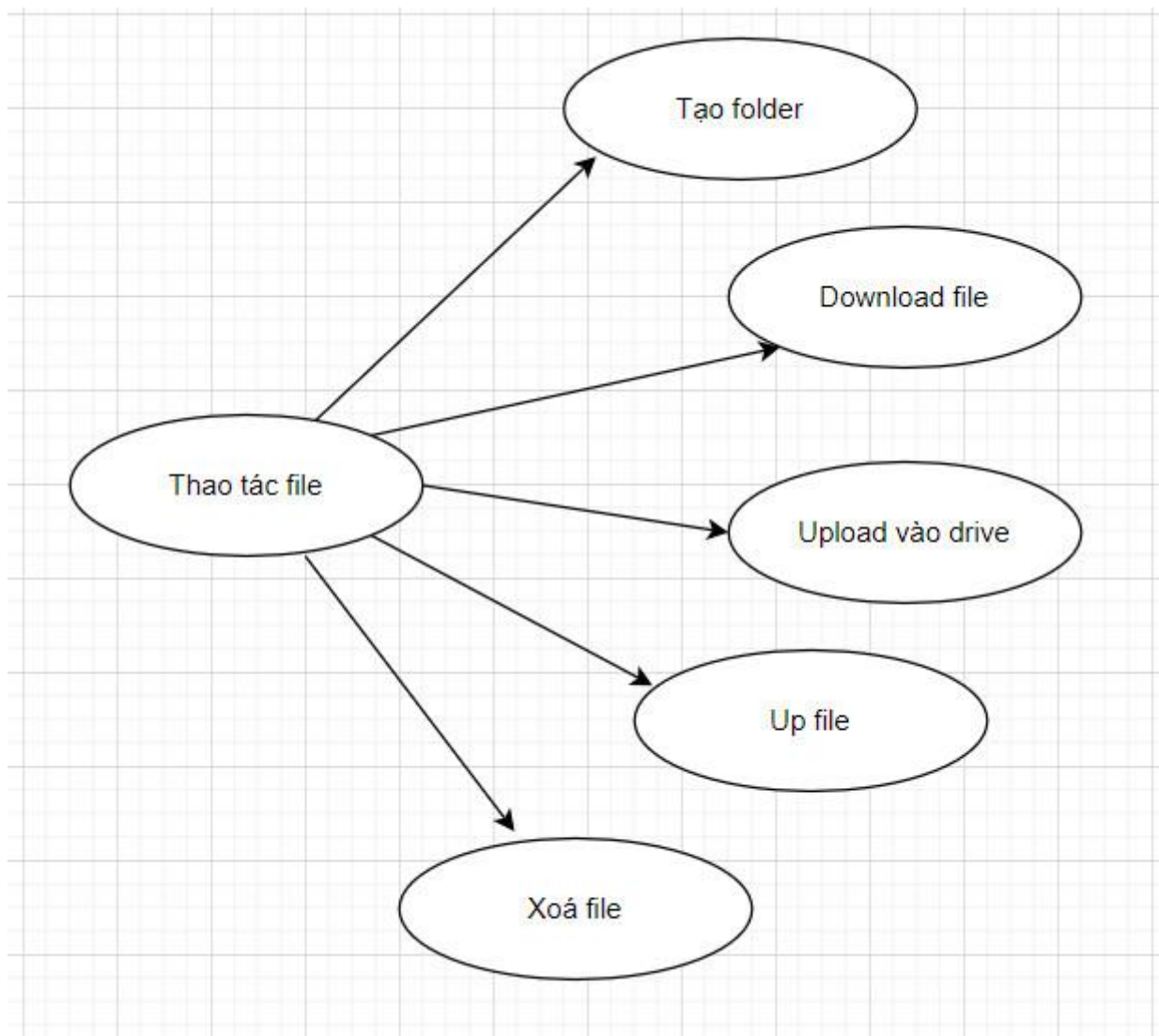
4.2 Use Case phân rã

4.2.1 Use Case cập nhật thông tin



Hình 4.1 Use Case cập nhật thông tin

4.2.2 Use Case thao tác file



Hình 4.2 Use Case thao tác file

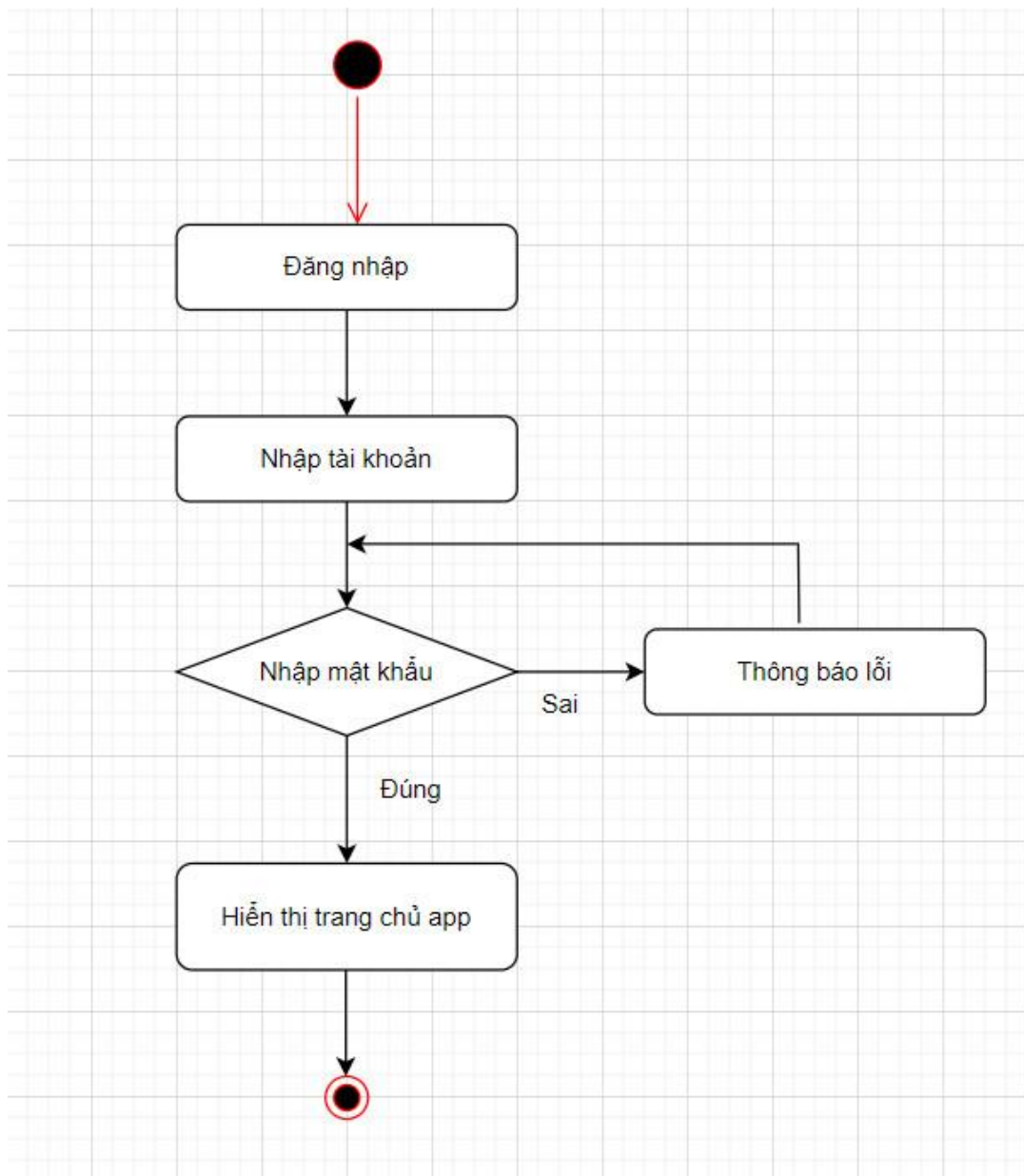
4.3 Biểu đồ hoạt động

4.3.1 Đăng ký



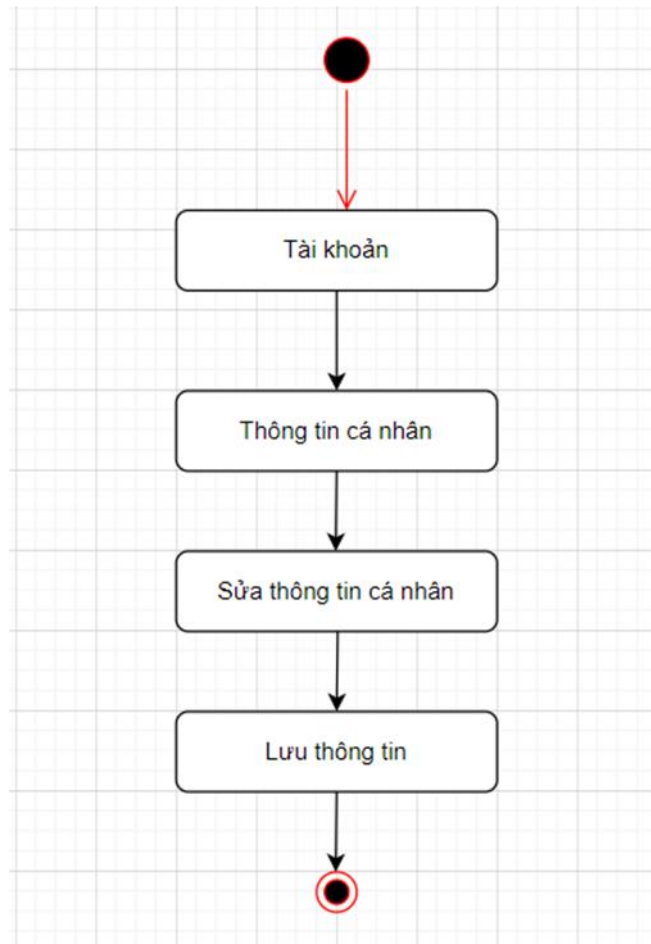
Hình 4.3 Bdhd đăng ký

4.3.2 Đăng nhập



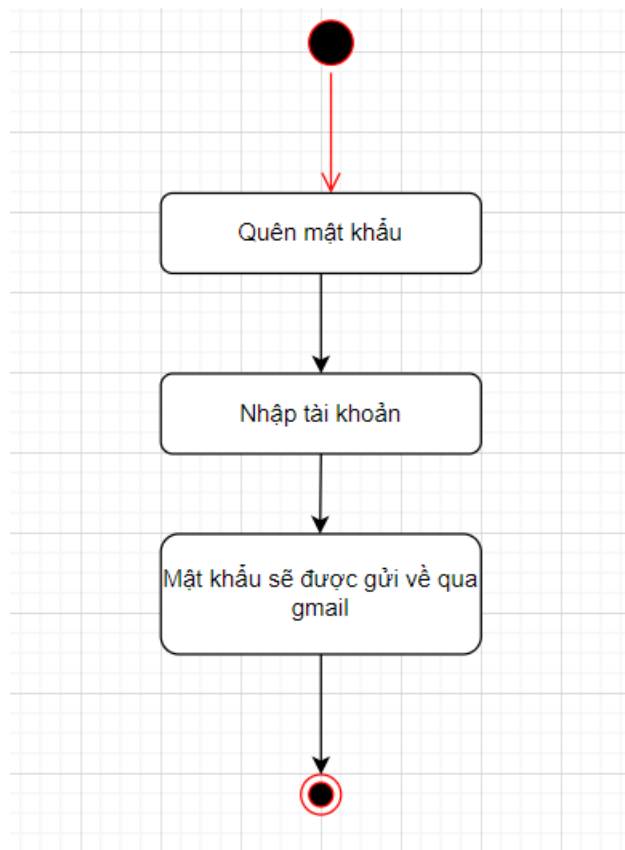
Hình 4.4 Bdhd đăng nhập

4.3.3 Cập nhật thông tin



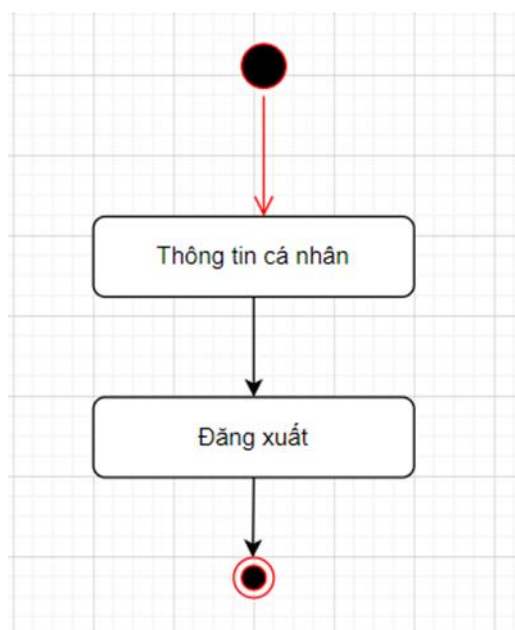
Hình 4.5 Bdhđ cập nhật thông tin

4.3.4 Quên mật khẩu



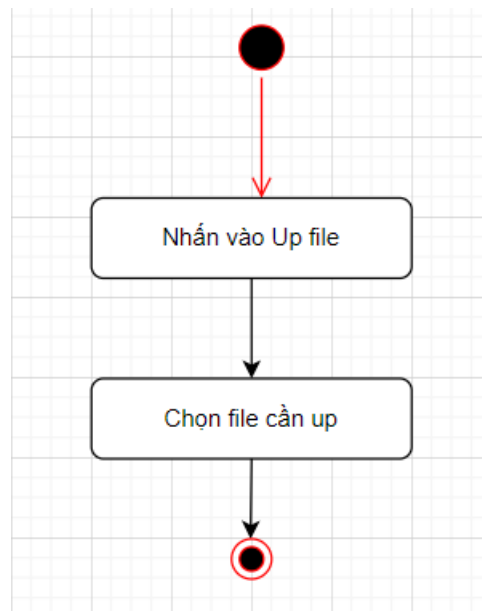
Hình 4.6 Bdhd quên mật khẩu

4.3.5 Đăng xuất



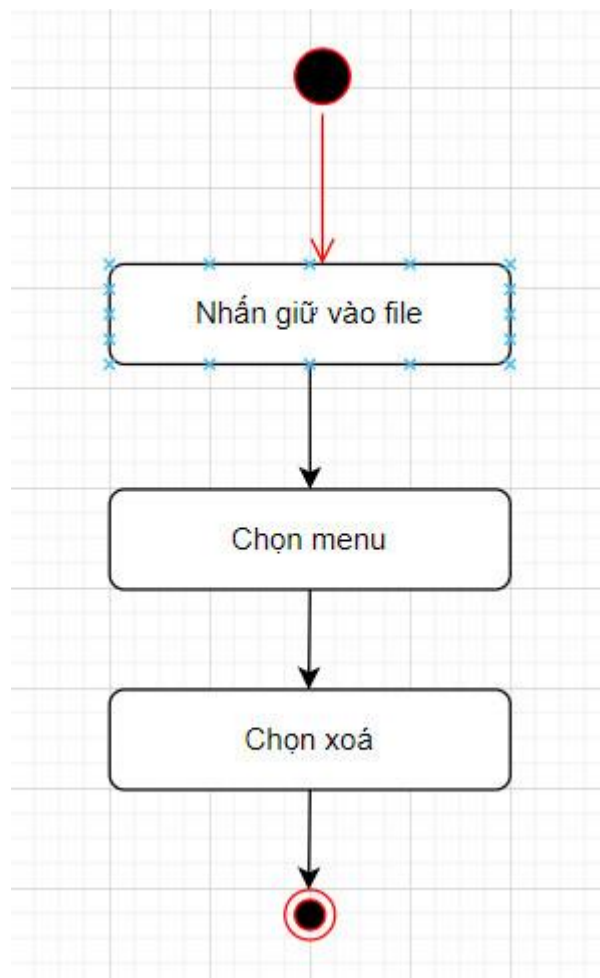
Hình 4.7 Bdhd đăng xuất

4.3.6 Upload file



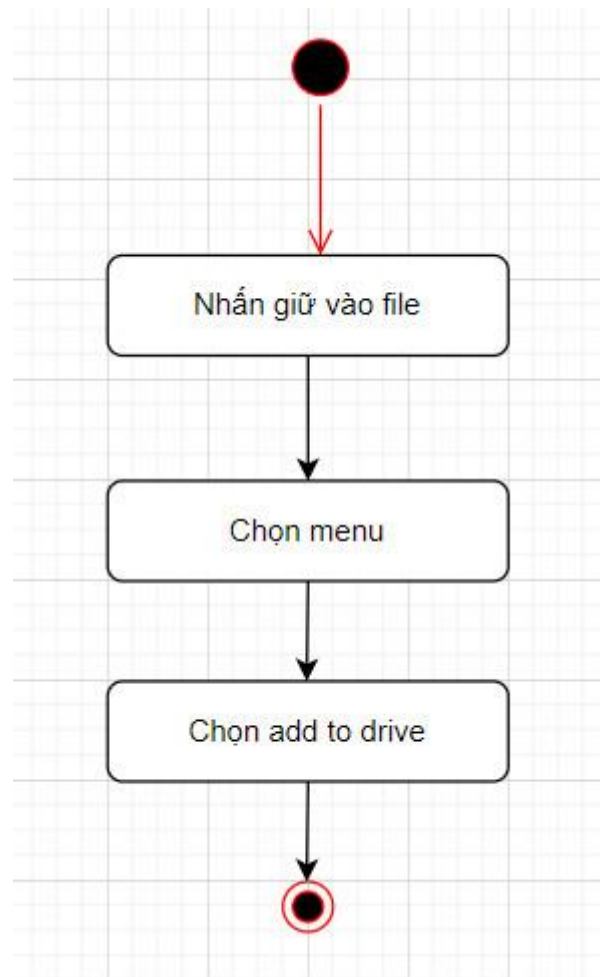
Hình 4.8 Bdhd upload file

4.3.7 Xóa file



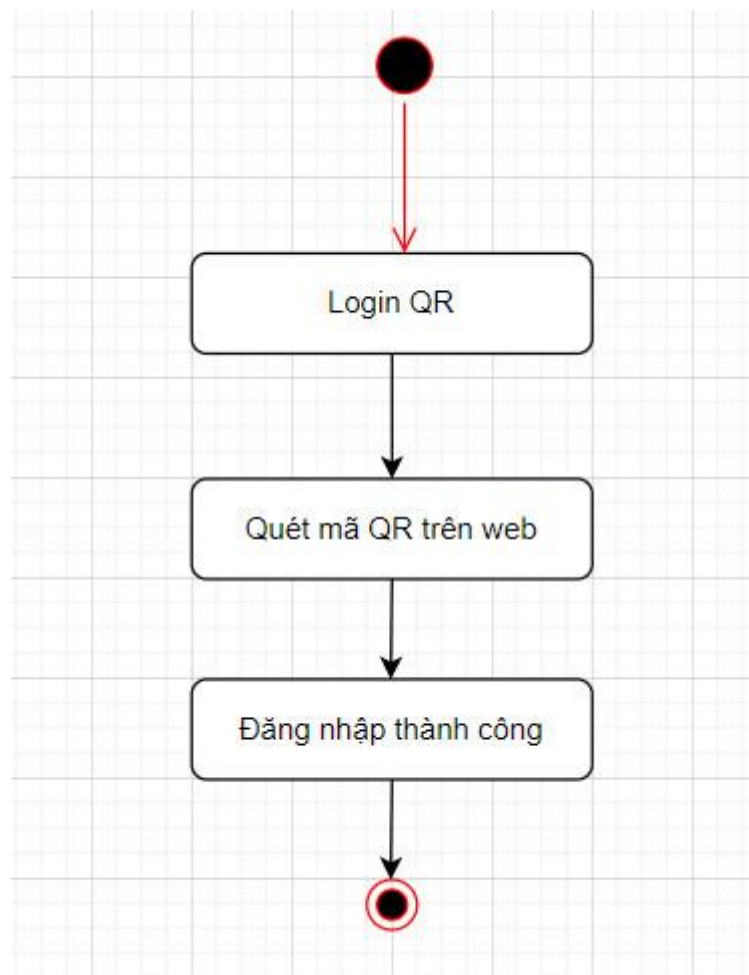
Hình 4.9 Bdhd xóa file

4.3.8 Upload drive



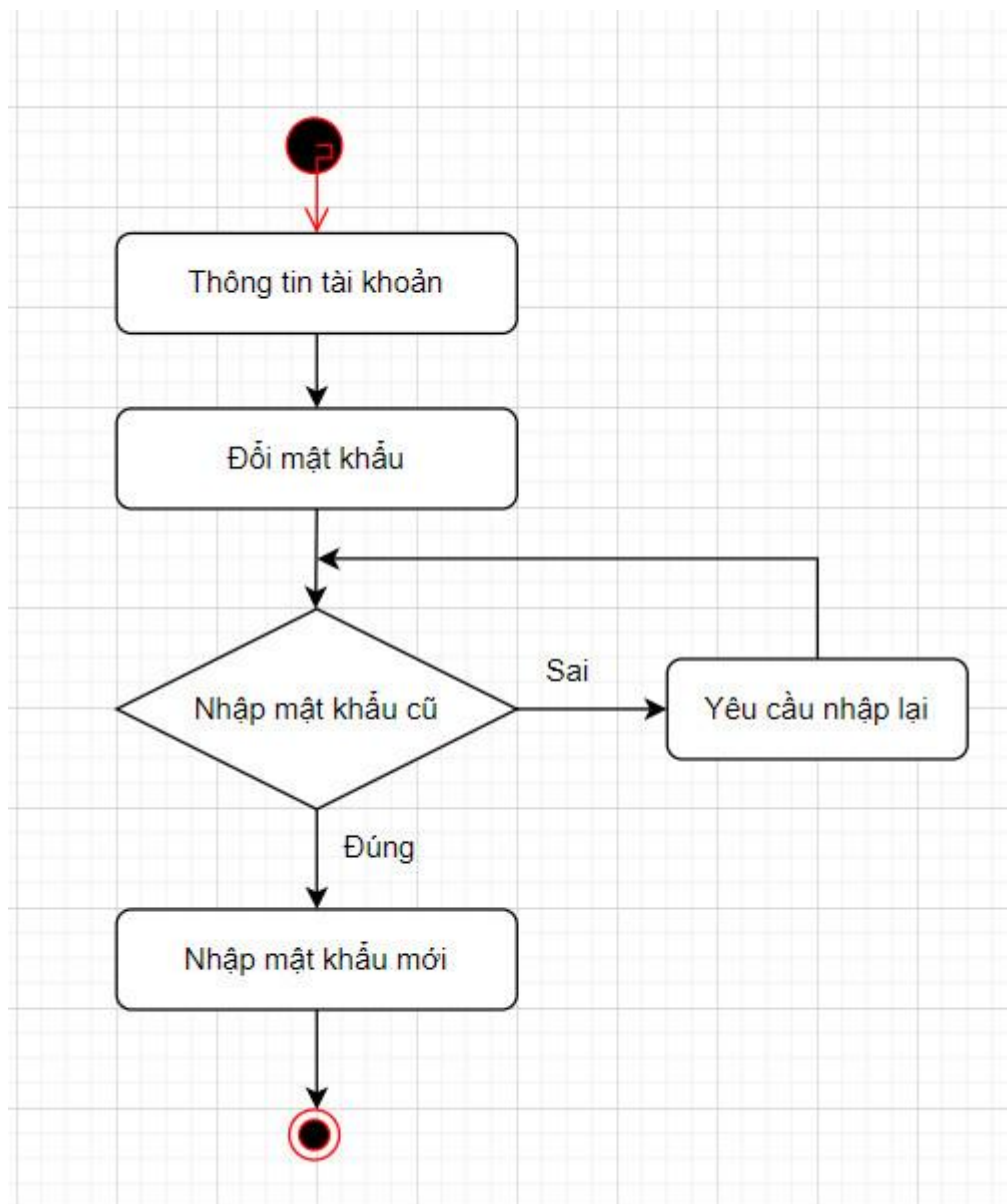
Hình 4.10 Bdhd upload driver

4.3.9 Đăng nhập bằng mã QR



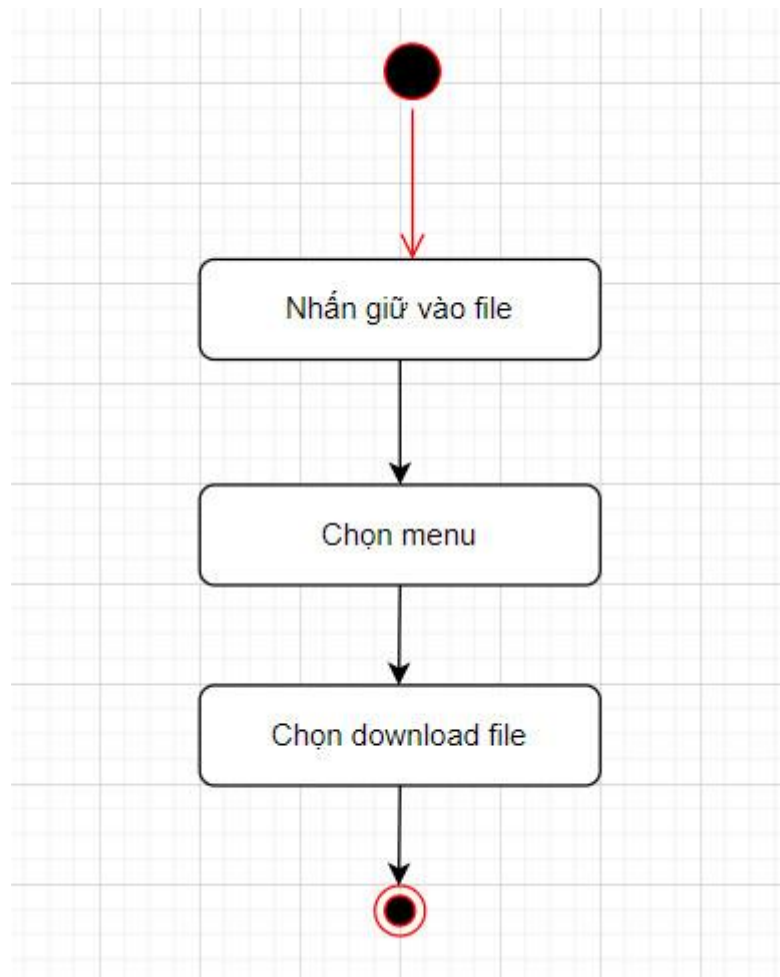
Hình 4.11 Bdhd đăng nhập bằng mã QR

4.3.10 Đổi mật khẩu



Hình 4.12 Bdhd đổi mật khẩu

4.3.11 Dowload file



Hình 4.13 Bdhd download file

4.4 Đặc tả Use case

4.4.1 Use Case đăng ký tài khoản mới

Use case	Đăng ký tài khoản mới
Actor	Người dùng
Brief Description	Use case đăng ký tài khoản mới mô tả quá trình tạo tài khoản mới
Pre conditions	Không
Basic Flows	Người dùng chọn chức năng đăng ký Hệ thống hiển thị các thông tin để người dùng nhập

	Sau khi điền thông tin ấn đăng ký, hệ thống sẽ gửi gmail xác thực tài khoản về email đã được điền thông tin ở trên Người dùng check gmail để nhập mã xác thực hoàn thành quá trình đăng ký mới
Alternative Flows	Đăng ký thành công

Bảng 4.1 Đặc tả use case đăng ký tài khoản mới

4.4.2 Use Case đăng nhập

Use case	Đăng nhập
Actor	Người dùng
Brief Description	Use case đăng nhập mô tả quá trình đăng nhập của người dùng
Pre conditions	Không
Basic Flows	Người dùng nhập thông tin tài khoản và mật khẩu để đăng nhập Hệ thống sẽ check tài khoản và mật khẩu với data trong hệ thống Nếu sai mật khẩu, tài khoản hệ thống yêu cầu người dùng nhập lại Đúng mật khẩu, login sẽ chuyển đến trang chủ của app
Alternative Flows	Đăng nhập thành công

Bảng 4.2 Đặc tả use case đăng nhập

4.4.3 Use Case quên mật khẩu

Use case	Quên mật khẩu
Actor	Quản lý
Brief Description	Use case quên mật khẩu mô tả quá trình lấy lại mật khẩu
Pre conditions	Đã có tài khoản
Basic Flows	Người dùng chọn quên mật khẩu Hệ thống hiển thị form để người dùng có thể nhập thông tin tài khoản Sau khi điền thông tin ấn lấy lại mật khẩu, hệ thống gửi gmail mật khẩu mới đến tài khoản gmail đã đăng ký Người dùng sẽ có lại mật khẩu được gửi thông qua gmail

Alternative Flows	Lấy lại mật khẩu thành công
-------------------	-----------------------------

Bảng 4.3 Đặc tả use case quên mật khẩu

4.4.4 Use Case cập nhật thông tin cá nhân

Use case	Cập nhật thông tin cá nhân
Actor	Người dùng
Brief Description	Use case cập nhật thông tin cá nhân mô tả quá trình sửa đổi thông tin của người dùng
Pre conditions	Người dùng đã đăng nhập thành công
Basic Flows	Người dùng chọn vào tài khoản Hệ thống hiển thị các chức năng trong mục tài khoản Chọn cập nhật thông tin Hệ thống hiển thị form thông tin để người dùng sửa Sau khi điền thông tin cần sửa ấn lưu hệ thống sẽ cập nhật thông tin người dùng
Alternative Flows	Cập nhật thành công

Bảng 4.4 Đặc tả use case cập nhật thông tin cá nhân

4.4.5 Use Case đổi mật khẩu

Use case	Đổi mật khẩu
Actor	Người dùng
Brief Description	Use case đổi mật khẩu mô tả quá trình đổi mật khẩu của người dùng
Pre conditions	Người dùng đã đăng nhập thành công
Basic Flows	Người dùng chọn tài khoản Hệ thống hiển thị các chức năng trong mục tài khoản Chọn đổi mật khẩu Hiện form để người dùng điền thông tin mật khẩu cũ, mật khẩu mới và xác nhận lại mật khẩu mới Nếu nhập sai mật khẩu cũ, hệ thống yêu cầu nhập lại Sau khi điền thông tin ấn đổi mật khẩu hệ thống thông báo đổi mật khẩu thành công
Alternative Flows	Đổi mật khẩu thành công

Bảng 4.5 Đặc tả use case đổi mật khẩu

4.4.6 Use Case xóa file

Use case	Xoá file
Actor	Người dùng
Brief Description	Use case xoá file mô tả quá trình xoá file của người dùng
Pre conditions	Người dùng đã đăng nhập thành công
Basic Flows	Người dùng chọn danh sách file Hệ thống hiển thị tất cả các file có trong hệ thống Chọn file cần xoá, nhấn giữ vào file Chọn menu Hệ thống hiển thị các chức năng có thể thực hiện Chọn xoá file Sau khi xoá file, hệ thống cập nhật lại danh sách file
Alternative Flows	Xoá file thành công

Bảng 4.6 Đặc tả use case xóa file

4.4.7 Use Case upload file

Use case	Upload file
Actor	Người dùng
Brief Description	Use case upload file mô tả quá trình tải file lên của người dùng
Pre conditions	Người dùng đã đăng nhập thành công
Basic Flows	Người dùng chọn danh sách file Hệ thống hiển thị tất cả các file có trong hệ thống Chọn upload file Hệ thống sẽ truy cập vào các tệp trong máy Chọn các file cần upload và ấn chọn Sau khi upload file, hệ thống cập nhật lại danh sách file
Alternative Flows	Upload file thành công

Bảng 4.7 Đặc tả use case upload file

4.4.8 Use Case upload file lên drive

Use case	Upload file lên driver
Actor	Người dùng

Brief Description	Use case upload file mô tả quá trình tải file lên drive của người dùng
Pre conditions	Người dùng đã đăng nhập thành công
Basic Flows	<p>Người dùng chọn danh sách file</p> <p>Hệ thống hiển thị tất cả các file có trong hệ thống</p> <p>Chọn vào file cần upload nhấn giữ để hiện các mục chức năng thao tác</p> <p>Chọn menu</p> <p>Chọn upload</p> <p>Sau khi upload, file sẽ được cập nhật lên drive</p>
Alternative Flows	Upload file thành công

Bảng 4.8 Đặc tả use case upload file lên drive

4.4.9 Use Case login QR

Use case	Login QR
Actor	Người dùng
Brief Description	Use case login QR mô tả quá trình login trên web thông qua quét mã QR
Pre conditions	Người dùng đã đăng nhập thành công
Basic Flows	<p>Người dùng chọn quét mã QR</p> <p>Hệ thống truy cập vào máy ảnh để có thể quét</p> <p>Quét mã QR trên web tài khoản sẽ được login trực tiếp trên web</p> <p>Login thành công chuyển đến trang chủ của web</p>
Alternative Flows	Login QR thành công

Bảng 4.9 Đặc tả use case login QR

4.4.10 Use Case tạo folder

Use case	Tạo Folder
Actor	Người dùng
Brief Description	Use case tạo folder mô tả quá trình tạo 1 folder mới của người dùng
Pre conditions	Người dùng đã đăng nhập thành công
Basic Flows	Người dùng chọn danh sách file

	Hệ thống hiển thị tất cả các file có trong hệ thống Chọn thêm mới Hệ thống hiển thị danh mục thao tác Chọn tạo folder, sau đó nhập tên của folder Sau khi tạo folder, hệ thống cập nhật lại danh sách file
Alternative Flows	Tạo folder thành công

Bảng 4.10 Đặc tả use case tạo folder

4.5 Cơ sở dữ liệu

	Column Name	Data Type	Allow Nulls
▶🔑	Id	nvarchar(450)	<input type="checkbox"/>
	FamilyName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	GivenName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Gender	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Picture	nvarchar(MAX)	<input checked="" type="checkbox"/>
	GoogleRefreshToken	nvarchar(MAX)	<input checked="" type="checkbox"/>
	UserName	nvarchar(256)	<input checked="" type="checkbox"/>
	NormalizedUserName	nvarchar(256)	<input checked="" type="checkbox"/>
	Email	nvarchar(256)	<input checked="" type="checkbox"/>
	NormalizedEmail	nvarchar(256)	<input checked="" type="checkbox"/>
	EmailConfirmed	bit	<input type="checkbox"/>
	PasswordHash	nvarchar(MAX)	<input checked="" type="checkbox"/>
	SecurityStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ConcurrencyStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>
	PhoneNumber	nvarchar(MAX)	<input checked="" type="checkbox"/>
	PhoneNumberConfirmed	bit	<input type="checkbox"/>
	TwoFactorEnabled	bit	<input type="checkbox"/>
	LockoutEnd	datetimeoffset(7)	<input checked="" type="checkbox"/>
	LockoutEnabled	bit	<input type="checkbox"/>
	AccessFailedCount	int	<input type="checkbox"/>
			<input type="checkbox"/>

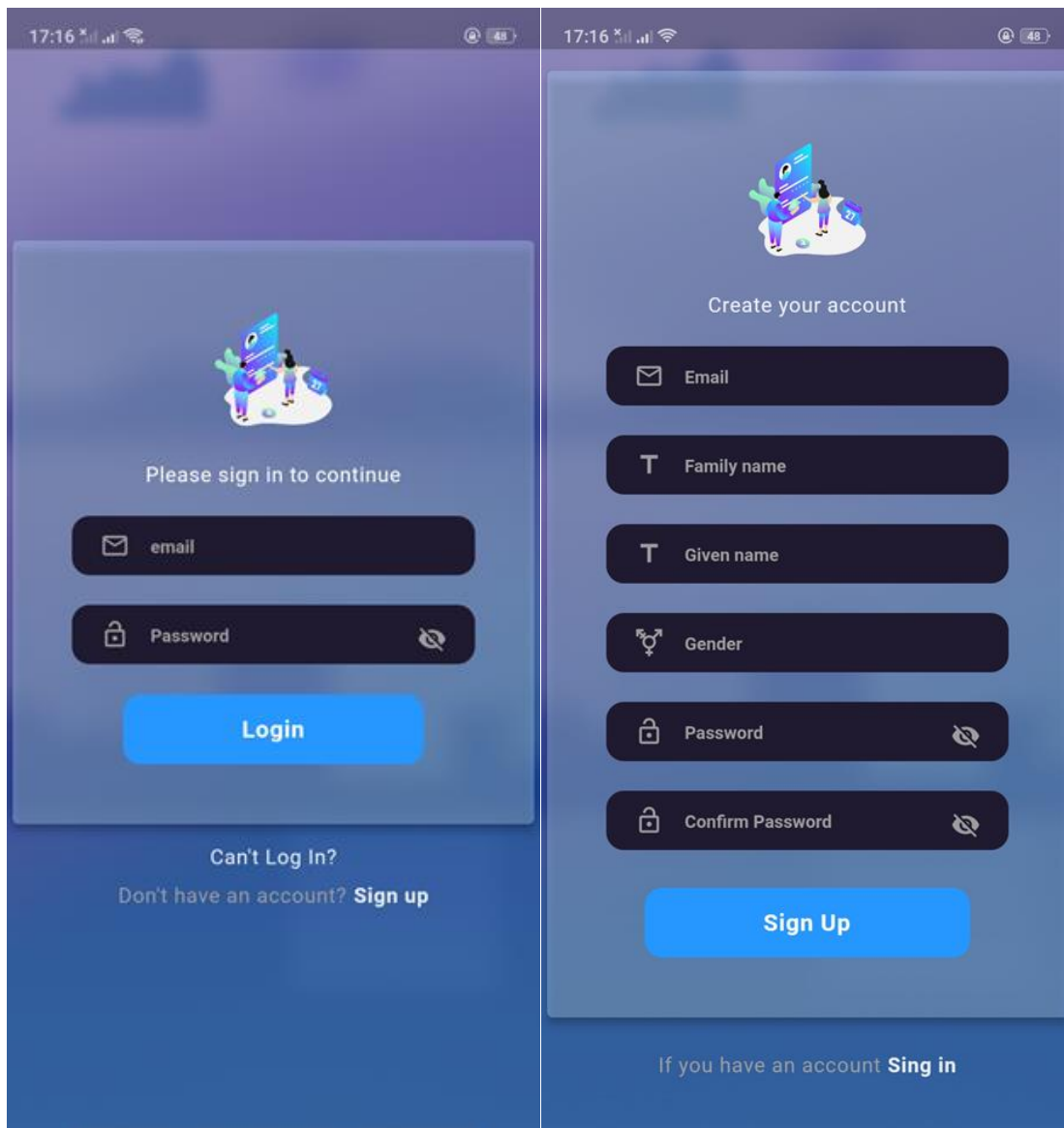
Hình 4.14 Bảng AppUser

Column Name	Data Type	Allow Nulls
Id	nvarchar(900)	<input type="checkbox"/>
Value	varbinary(MAX)	<input type="checkbox"/>
ExpiresAtTime	datetimeoffset(7)	<input type="checkbox"/>
SlidingExpirationInSecon...	bigint	<input checked="" type="checkbox"/>
AbsoluteExpiration	datetimeoffset(7)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Hình 4.15 Bảng Cache

CHƯƠNG 5: THỰC NGHIỆM

5.1 Sản phẩm thực tế



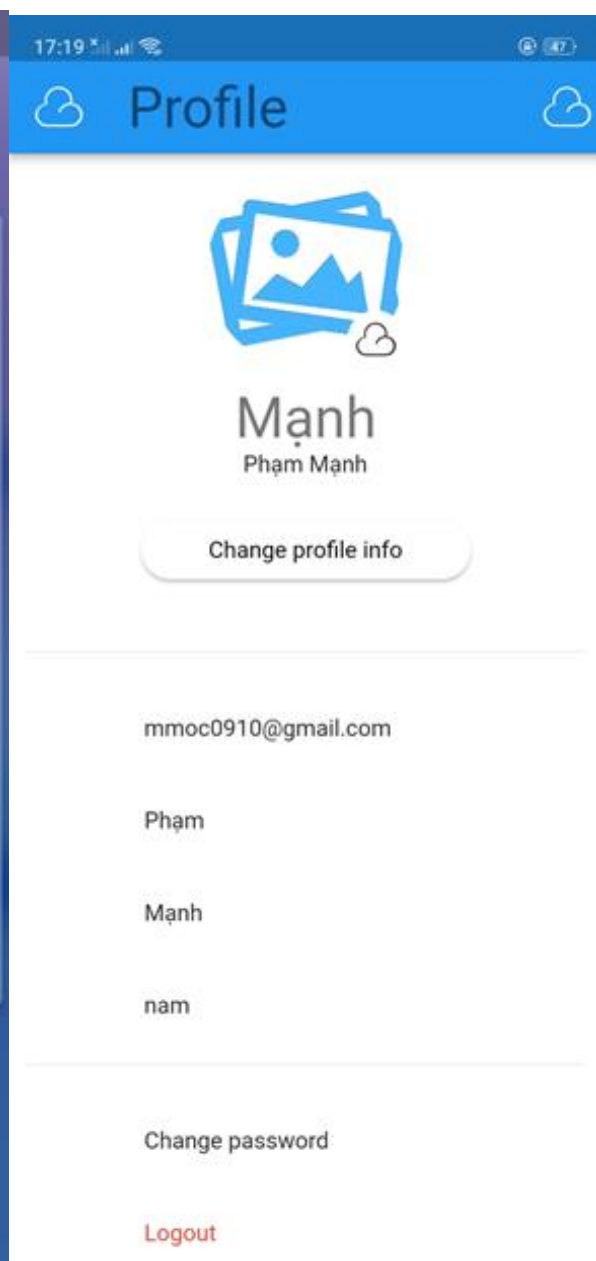
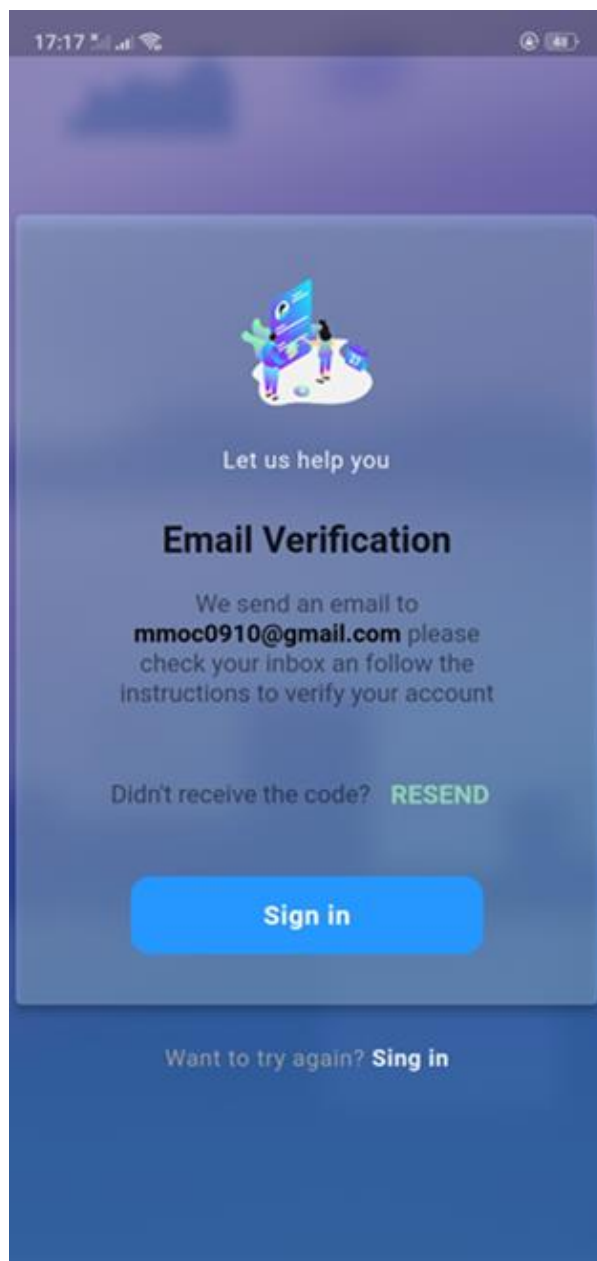
The image displays two side-by-side mobile application screens with a purple-to-blue gradient background. Both screens feature a status bar at the top showing the time 17:16, signal strength, and battery level at 48%.

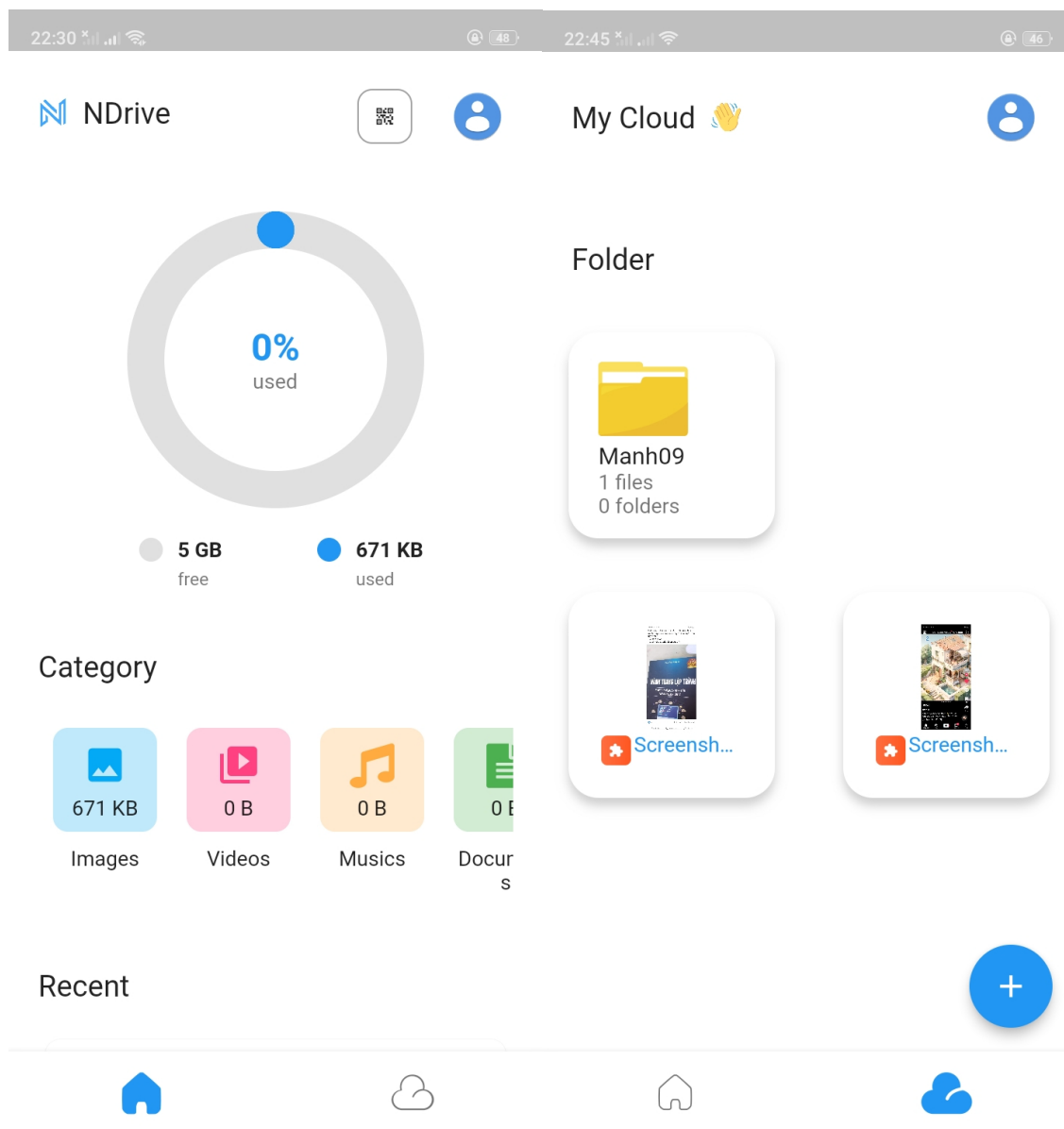
Left Screen (Login):

- Header: "Please sign in to continue" with an illustration of people at a computer.
- Form fields: "email" (with an envelope icon) and "Password" (with a lock icon and a toggle for visibility).
- Button: A blue "Login" button.
- Footer: "Can't Log In?" and "Don't have an account? **Sign up**".

Right Screen (Sign Up):

- Header: "Create your account" with an illustration of people at a computer.
- Form fields: "Email" (with an envelope icon), "Family name" (with a 'T' icon), "Given name" (with a 'T' icon), "Gender" (with a gender icon), "Password" (with a lock icon and a toggle for visibility), and "Confirm Password" (with a lock icon and a toggle for visibility).
- Button: A blue "Sign Up" button.
- Footer: "If you have an account **Sing in**".





5.2 Đánh giá ứng dụng

Ưu điểm:

- App dễ cài đặt và sử dụng
- App hoạt động đúng với yêu cầu đặt ra

Nhược điểm:

- App vẫn còn ít chức năng
- Do đang sử dụng api nên còn nhiều hạn chế

KẾT LUẬN

Sau một thời gian tìm hiểu đề tài “Xây dựng ứng dụng quản lý tài liệu trên Cloud Server” nhóm chúng em đã thực hiện được nội dung đề tài của theo yêu cầu đặt ra. Với mục đích là tìm hiểu về hệ điều hành Android, hiểu biết về ngôn ngữ lập trình mới cho điện thoại di động Android và tìm hiểu các ứng dụng của hệ điều hành Android trong thế giới di động ngày nay. Thông qua việc tìm hiểu và nắm bắt lý thuyết về hệ điều hành Android, bước đầu đã xây dựng thành công chương trình thực nghiệm với các chức năng cơ bản.

Do thời gian có hạn cùng với kiến thức của bản thân còn hạn chế nên chúng em chưa tìm hiểu sâu về các kiến thức liên quan tới hệ điều hành Android nên chương trình vẫn còn rất nhiều hạn chế, chúng em sẽ cố gắng phát triển thêm để chương trình thân thiện, dễ sử dụng và nhiều chức năng hơn.

TÀI LIỆU THAM KHẢO

- [1] Hồ Thị Thảo Trang, “Giáo trình Android”, khoa Công Nghệ Thông Tin, đại học Mỏ Địa Chất, Hà Nội.
- [2] Dawn Griffiths & David Griffiths. (2017). “Head First Android Development” 2nd Edition.
- [3] John Horton. (2015). “Android Programming for Beginners”.
- [4] Dave Smith & Jeff Friesen. (2014). “Android Recipes: A Problem-Solution Approach” 3rd Edition.