

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO MÔN HỌC
LẬP TRÌNH ỨNG DỤNG DI ĐỘNG

Đề tài:

QUẢN LÝ THÔNG TIN LỚP HỌC

Sinh viên thực hiện	MSSV	Lớp
Vũ Đức Thái	20172804	ĐTVT.08-K62
Phan Thanh Tùng	20172904	ĐTVT.08-K62
Vũ Trọng Tới	20172858	ĐTVT.06-K62

Giảng viên hướng dẫn: PGS.TS. Đỗ Trọng Tuấn

Hà Nội, 1-2022

LỜI NÓI ĐẦU

Ngày nay công nghệ thông tin ngày càng phát triển, việc áp dụng các thành tựu của chúng mang lại hiệu và lợi ích to lớn trên mọi mặt của đời sống xã hội. Đặc biệt lập trình ứng dụng di động đang trở thành một xu thế mà nhiều công ty lớn đang tập trung phát triển, bởi vì đa phần người dùng hiện nay đều ưa thích sử dụng các thiết bị di động cầm tay giúp linh hoạt trong mọi hoạt động đời sống. Android là một hệ điều hành mã nguồn mở và cũng là một thị phần chiếm đa số của các thiết bị di động cũng như các sản phẩm nhúng hiện nay.

Ở Việt Nam, nhiều trường học đa số các giáo viên vẫn quản lý lớp học, học sinh sinh viên bằng các phương pháp ghi chép thủ công gây mất thời gian, phức tạp, hoặc là có trường hợp mất sổ ghi chép thì cũng không thể khôi phục lại được. Trong xã hội hiện đại điện thoại chạy Android đã trở nên quá phổ biến và giá thành ngày càng rẻ hoàn toàn có thể chạy các ứng dụng một cách nhanh chóng.

Nắm bắt được nhu cầu đó, cùng với sự hướng dẫn tận tình của PGS.TS Đỗ Trọng Tuấn mà nhóm chúng em phát triển ứng dụng *Quản lý thông tin lớp học* giúp cho giáo viên có thể dễ dàng, tiện lợi trong viên quản lý, theo dõi học sinh sinh viên trong lớp học.

Do kiến thức bản thân còn hạn hẹp cũng như chưa có kinh nghiệm làm việc với lập trình Android. Do vậy trong quá trình thực hiện nhóm em còn phạm nhiều sai sót cũng như chương trình thành phẩm có thể còn tồn tại lỗi không mong muốn. Rất mong nhận được sự góp ý của thầy để chúng em hoàn thiện sản phẩm của mình hơn.

Cuối cùng chúng em xin gửi lời cảm ơn sâu sắc đến thầy Đỗ Trọng Tuấn đã luôn tận tình hướng dẫn, giải đáp các thắc mắc và đồng hành cùng chúng em trong quá trình học tập nghiên cứu và thực hiện đề tài này.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

LỜI NÓI ĐẦU.....	i
DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT	i
DANH MỤC HÌNH VẼ.....	ii
DANH MỤC BẢNG BIỂU.....	iv
CHƯƠNG 1. PHÂN CÔNG CÔNG VIỆC.....	1
1.1 Công việc chính cần thực hiện.....	1
1.2 Nhân lực làm việc	1
1.3 Phân chia công việc	1
CHƯƠNG 2. TỔNG QUAN ĐỀ TÀI.....	5
2.1 Đặt vấn đề.....	5
2.2 Giới hạn đề tài.....	5
2.3 Mục tiêu xây dựng đề tài	5
2.3.1 Mục tiêu giữa kỳ	5
2.3.2 Mục tiêu cuối kỳ	6
CHƯƠNG 3. CƠ SỞ LÝ THUYẾT	7
3.1 Tổng quan về hệ điều hành android.....	7
3.1.1 Lịch sử phát triển	7
3.1.2 Kiến trúc của hệ điều hành android	9
3.2 Kiến thức về lập trình android	10
3.2.1 Activity.....	10
3.2.2 Fragments.....	12
3.2.3 ListView	15
3.2.4 Adapter	16
3.2.5 Cơ sở dữ liệu SQLite	17
3.2.6 Firebase	18
CHƯƠNG 4. PHÁT TRIỂN ỨNG DỤNG	19
4.1 Sơ đồ khối của hệ thống	19

4.2 Các chức năng cơ bản của ứng dụng	20
4.2.1 Giao diện đăng nhập.....	20
4.2.2 Quản lý thông tin lớp học	22
4.2.3 Quản lý thông tin sinh viên	25
4.2.4 Giao diện thông tin lịch học và hỗ trợ điểm danh	30
4.2.5 Giao diện thông tin ứng dụng.....	31
4.2.6 Một vài chức năng bổ xung tương tác với lớp học.....	32
CHƯƠNG 5. KIỂM THỬ ỨNG DỤNG	36
5.1 Môi trường kiểm thử.....	36
5.1.1 Chạy trên máy ảo.....	36
5.1.2 Chạy trên máy thật	36
5.2 Kịch bản kiểm thử cho ứng dụng và sửa lỗi chi tiết.....	36
5.2.1 Kiểm thử phần đăng nhập cho ứng dụng.....	36
5.2.2 Kiểm thử xung đột khi hiển thị thông báo lớp học theo thời gian thực	36
5.2.3 Kiểm thử xung đột dữ liệu trong SQLite database	38
5.2.4 Kiểm thử xung đột dữ liệu trong Firebase database.....	38
KẾT LUẬN.....	40
TÀI LIỆU THAM KHẢO	41

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Từ viết tắt	Tiếng anh	Tiếng việt
DB	Database	Cơ sở dữ liệu

DANH MỤC HÌNH VẼ

Hình 3-1 Ảnh minh họa biểu tượng của android	7
Hình 3-2 Các phiên bản của android.....	8
Hình 3-3 Kiến trúc của hệ điều hành android	9
Hình 3-4 Mô tả hoạt động của các activities trong android	11
Hình 3-5 Vòng đời của một activity.....	11
Hình 3-6 Mô tả hoạt động của fragments	12
Hình 3-7 Vòng đời của một fragment	13
Hình 3-8 Mô tả cách hoạt động của một listview	15
Hình 3-9 Mô tả cách hoạt động của một adapter	16
Hình 3-10 Biểu tượng của SQLite	17
Hình 3-11 Biểu tượng của firebase	18
Hình 4-1 Sơ đồ khối của hệ thống	19
Hình 4-2 Sơ đồ giao diện đăng nhập của hệ thống	20
Hình 4-3 Giao diện đăng nhập của ứng dụng	21
Hình 4-4 Giao diện đăng ký của ứng dụng	21
Hình 4-5 Sơ đồ quản lý thông tin lớp học.....	22
Hình 4-6 Giao diện quản lý danh sách các lớp học hiện tại.....	22
Hình 4-7 Giao diện thêm mới một lớp học	23
Hình 4-8 Giao diện cập nhật thông tin của một lớp học	24
Hình 4-9 Giao diện xác nhận xóa thông tin của một lớp học	24
Hình 4-10 Sơ đồ quản lý sinh viên.....	25
Hình 4-11 Giao diện quản lý danh sách sinh viên trong một lớp học.....	26
Hình 4-12 Giao diện thêm một sinh viên mới vào lớp học hiện tại.....	27
Hình 4-13 Giao diện cập nhật thông tin của sinh viên.....	28
Hình 4-14 Giao diện xác nhận xóa thông tin của một sinh viên.....	29
Hình 4-15 Giao diện tìm kiếm sinh viên trong một lớp học theo tên	30
Hình 4-16 Giao diện thông báo các lớp học theo thời gian thực	30

Hình 4-17 Giao diện hỗ trợ điểm danh sinh viên theo từng lớp.....	31
Hình 4-18 Giao diện thông tin ứng dụng	32
Hình 4-19 Giao diện chức năng gọi ngẫu nhiên một sinh viên trong lớp học ...	33
Hình 4-20 Giao diện chính của ứng dụng	35

DANH MỤC BẢNG BIỂU

Bảng 1-1 Bảng đánh giá nhân lực	1
Bảng 1-2 Bảng phân công công việc	4
Bảng 5-1 Danh sách các thiết bị dùng để kiểm thử	36
Bảng 5-2 Kịch bản kiểm thử phần hiển thị thông báo lớp học	37
Bảng 5-3 Kịch bản kiểm thử phần dữ liệu trong SQLite	38
Bảng 5-4 Kịch bản kiểm thử phần dữ liệu trong firebase	39

CHƯƠNG 1. PHÂN CÔNG CÔNG VIỆC

1.1 Công việc chính cần thực hiện

- Thống nhất môi trường làm việc (lưu trữ báo cáo, slide, video, code)
- Lựa chọn đề tài (đặt vấn đề, giới hạn đề tài, mục tiêu môn học)
- Xây dựng cơ sở lý thuyết của môn học
- Phát triển các chức năng cơ bản của ứng dụng
- Phát triển thêm một số chức năng tương tác với các đối tượng trong ứng dụng

1.2 Nhân lực làm việc

Tên	Điểm mạnh	Điểm yếu
Vũ Đức Thái	Có nền tảng lập trình C/C++, đọc tài liệu, phân công công việc.	Không
Phan Thanh Tùng	Có nền tảng lập trình C/C++, tìm tài liệu, thông thạo các ứng dụng vẽ minh họa.	Không
Vũ Trọng Tới	Có nền tảng lập trình Python, thông thạo các ứng dụng văn phòng, làm video.	Không

Bảng 1-1 Bảng đánh giá nhân lực

1.3 Phân chia công việc

Công việc chính	Người thực hiện	Chia công việc	Thời gian thực hiện	Trạng thái
Giai đoạn 1: Giữa kỳ				
Lựa	Cả nhóm	Tìm hiểu đề tài và chọn đề tài,	09/10/2021 – 14/10/2021	Hoàn

chọn đề tài		thống nhất mục tiêu và giới hạn đề tài		thành
	Thái	Viết báo cáo phân tổng quan đề tài	14/10/2021 – 16/10/2021	Hoàn thành
Thống nhất môi trường làm việc chung	Cả nhóm	Thống nhất môi trường làm việc chung (lưu trữ báo cáo, slide, tài liệu tham khảo, code)	18/10/2021 – 20/10/2021	Hoàn thành
Xây dựng cơ sở lý thuyết môn học	Thái	Viết báo cáo phân tổng quan về android	25/10/2021 – 1/11/2021	Hoàn thành
	Tới	Tìm hiểu lý thuyết và viết báo cáo phần Activity, Fragments	27/10/2021 – 3/11/2021	Hoàn thành
	Tùng	Tìm hiểu lý thuyết và viết báo cáo phần ListView, Adapter, Cơ sở dữ liệu SQLite	02/11/2021 – 16/11/2021	Hoàn thành
Phát triển các chức năng cơ bản của ứng dụng	Cả nhóm	Vẽ sơ đồ khối của hệ thống, phân tích các đối tượng, tương tác trong hệ thống	04/11/2021 – 7/11/2021	Hoàn thành
	Tùng	Code phần tạo API tương tác với cơ sở dữ liệu SQLite	05/11/2021 – 12/11/2021	Hoàn thành
	Thái	Code phần tạo API tương tác giữa các đối tượng trong chương trình và ListView	03/11/2021 – 09/11/2021	Hoàn thành
	Thái, Tùng	Code phần tạo giao diện, tương tác với thông tin của đối tượng classroom (thêm, sửa, xóa)	10/11/2021 – 16/11/2021	Hoàn thành
	Tới	Code phần tạo giao diện để tương tác với đối tượng sinh viên	17/11/2021 – 20/11/2021	Hoàn thành

	Thái, Tới	Code phần tương tác với thông tin của đối tượng student (thêm, sửa, xóa)	21/11/2021 – 23/11/2021	Hoàn thành
	Thái	Chỉnh sửa giao diện ứng dụng (thay đổi nền, định dạng lại listview)	23/11/2021 – 24/11/2021	Hoàn thành
	Tùng	Làm slide, quay video thuyết trình giữa kỳ	24/11/2021 – 30/11/2021	Hoàn thành
	Thái	Quay video giới thiệu các chức năng của ứng dụng (giả lập trên android studio)	25/11/2021 – 26/11/2021	Hoàn thành
	Tới	Làm video	30/11/2021 – 02/11/2021	Hoàn thành
	Cả nhóm	Hoàn thiện giữa kỳ	02/12/2021	Hoàn thành
Giai đoạn 2: Cuối kỳ				
Phát triển thêm các chức năng tương tác với lớp học	Tùng	Code chức năng xem lịch học các lớp trong phần thông báo của ứng dụng	03/12/2021 – 10/12/2021	Hoàn thành
	Tới	Code chức năng gọi ngẫu nhiên các sinh viên trong một lớp học	03/12/2021 – 09/01/2022	Hoàn thành
	Thái	Code hiệu ứng hoa rơi trong giao diện home của ứng dụng	03/12/2021 – 10/12/2021	Hoàn thành
	Tùng	Thêm chức năng đăng ký và đăng nhập sử dụng tài khoản email	10/12/2021 – 17/12/2021	Hoàn thành
	Tùng	Chuyển đổi cơ sở dữ liệu lưu trữ từ SQLite sang Firebase	18/12/2021 – 24/12/2021	Hoàn thành
	Tùng	Thêm chức năng điểm danh sinh viên trong lớp học hiện	27/12/2021 – 06/01/2022	Hoàn

		tại theo từng ngày		thành
Kiểm thử các chức năng và sửa lỗi chi tiết	Thái	Xây dựng các kịch bản và kiểm tra lỗi trên máy ảo	10/12/2021 – 07/01/2022	Hoàn thành
	Tối, Tùng	Xây dựng các kịch bản và kiểm tra lỗi trên máy thật	10/12/2021 – 07/01/2022	Hoàn thành
Hoàn thành bài tập lớn	Tùng	Viết báo cáo các phần Tùng làm trong nửa cuối kỳ	08/01/2022 - 11/01/2022	Hoàn thành
	Tối	Viết báo cáo các phần Tối làm trong nửa cuối kỳ	08/01/2022 - 13/01/2022	Hoàn thành
	Thái	Chỉnh sửa lại báo cáo, xuất file PDF	08/01/2022 - 14/01/2022	Hoàn thành
	Tùng	Làm slide thuyết trình cuối kỳ	14/01/2022 - 15/01/2022	Hoàn thành
	Thái, Tối	Quay video thuyết trình slide và demo ứng dụng trên máy di động	16/01/2022 - 18/01/2022	Hoàn thành
	Cả nhóm	Hoàn thành bài tập lớn	18/01/2022	Hoàn thành

Bảng 1-2 Bảng phân công công việc

CHƯƠNG 2. TỔNG QUAN ĐỀ TÀI

2.1 Đặt vấn đề

Hiện nay hầu hết các trường học đều sử dụng phương pháp quản lý truyền thống và gây ra tốn nhiều thời gian, trong khi điện thoại hiện nay đã trở nên cực kỳ phổ biến (hầu như chúng ta đều sở hữu từ một đến hai cái điện thoại thông minh), trong đó thì thị phần android là một thị phần lớn nhất trong thế giới di động. Do đó để hỗ trợ nhiều hơn cho việc quản lý sinh viên, chúng em quyết định xây dựng một ứng dụng quản lý thông tin lớp học mang tên “MyClass” giành cho giáo viên, giảng viên hoặc các đối tượng dạy học tự nhân với quy mô từ nhỏ đến lớn.

Bên cạnh đó, đề tài này cũng được tạo ra với mục đích học tập, chúng em sẽ cố gắng tiếp cận với càng nhiều công nghệ từ phổ biến đến chuyên sâu trong lập trình android, với sự giúp đỡ của thầy Đỗ Trọng Tuấn trên giảng đường và những khai phá trong quá trình trao đổi nhóm giữa các thành viên với nhau.

2.2 Giới hạn đề tài

Phần ứng dụng trong điện thoại chạy như một ứng dụng local, và phần lưu trữ chung sẽ có một server để tương tác, đóng vai trò như một ứng dụng remote. Khi người dùng nhập liệu trên điện thoại, tiến hành commit lên remote để đồng bộ thì sẽ được lưu trữ lại giống như một lịch sử lớp học. Ở những thiết bị khác khi người dùng đăng nhập vào hệ thống sẽ có quyền lấy lại được dữ liệu đã đồng bộ trên server.

Trong đề tài môn học này, chúng em giới hạn xây dựng ứng dụng ở local, tập trung vào phần tương tác ứng dụng local của người dùng với cơ sở dữ liệu lưu trong máy. Phần ứng dụng remote trên server liên quan đến một lĩnh vực khác ngoài môn học này, sẽ không xây dựng.

2.3 Mục tiêu xây dựng đề tài

2.3.1 Mục tiêu giữa kỳ

Trong khoảng thời gian giữa kỳ, chúng em lập kế hoạch đưa ra một vài mục tiêu như sau:

- Tìm hiểu và thực hành các kiến thức trên lớp.

- Biết cách tạo ra một API để tương tác với cơ sở dữ liệu SQLite (xây dựng back-end)
- Tương tác giữa cơ sở dữ liệu SQLite và View Group (thêm, sửa, xóa, các đối tượng)

2.3.2 Mục tiêu cuối kỳ

- Hoàn thiện phần đăng nhập cho sản phẩm (thêm phần xác thực người đăng nhập)
- Tạo ra một vài chức năng tương tác với lớp học (ví dụ như gọi ngẫu nhiên các bạn trong lớp, thông báo lớp học tiếp theo sẽ diễn ra, báo cáo thống kê...)
- Tạo thêm hiệu ứng về đồ họa cho ứng dụng

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

3.1 Tổng quan về hệ điều hành android

3.1.1 Lịch sử phát triển



Hình 3-1 Ảnh minh họa biểu tượng của android

Android là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành riêng cho các thiết bị di động có màn hình cảm ứng (điện thoại, máy tính bảng, đồng hồ thông minh, máy nghe nhạc,...). Cùng với việc được Google phát hành mã nguồn theo giấy phép Apache (một loại giấy phép ít bị ràng buộc), các nhà phát triển thiết bị, các nhà mạng, các lập trình viên nhanh chóng tiếp cận, điều chỉnh và phân phối Android một cách tự do.

Có thể nói hành động Google “cho không” Android là một bước tiến lớn trong việc biến đứa con này trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới. Với bản chất “mở” của mình thì Android đã thu hút được đông đảo các lập trình viên, chuyên viên phát triển ứng dụng, theo thống kê tháng 5 năm 2021 có khoảng hơn 3 tỉ các thiết bị chạy hệ điều hành Android trên toàn thế giới.



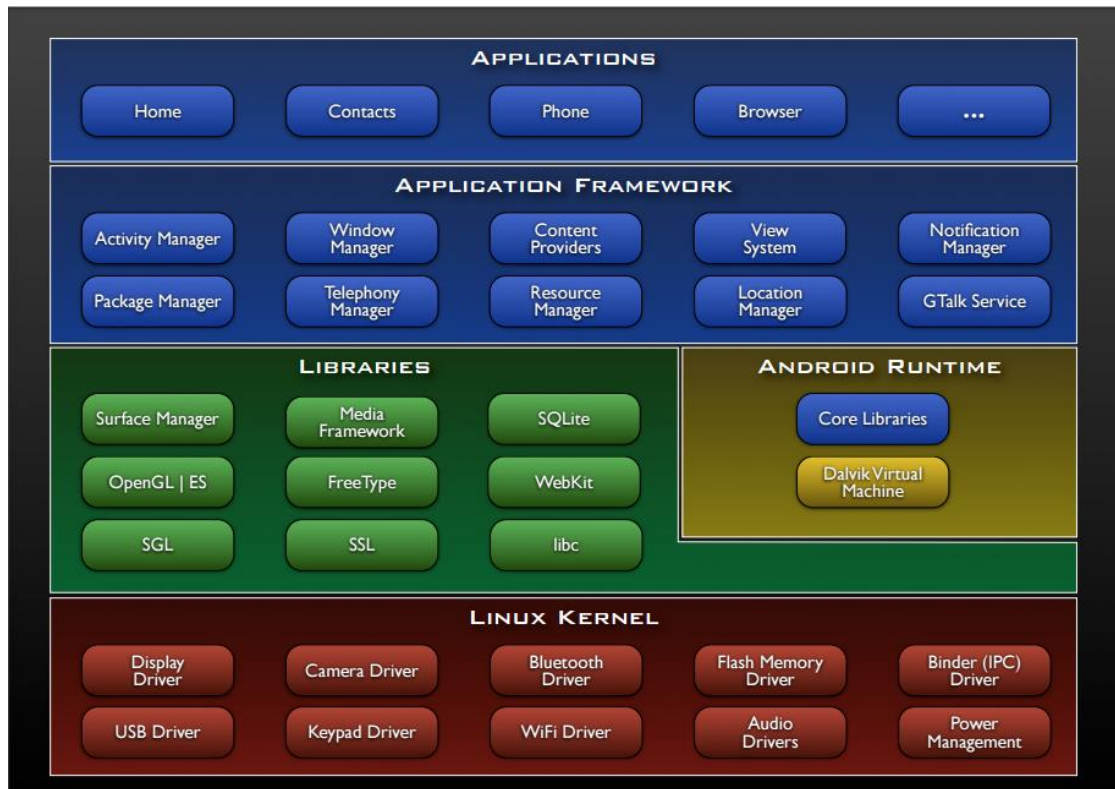
Hình 3-2 Các phiên bản của android

Các phiên bản android đến thời điểm hiện tại:

- Android 1.5 Cupcake
- Android 1.6 Donut
- Android 2.1 Eclair
- Android 2.2 Froyo
- Android 2.3 Gingerbread
- Android 3.2 Honeycomb
- Android 4.0 Ice Cream Sandwich
- Android 4.1 Jelly Bean
- Android 4.2 Jelly Bean
- Android 4.3 Jelly Bean
- Android 4.4 KitKat
- Android 5.0 Lollipop
- Android 6.0 Marshmallow
- Android 7.0 Nougat
- Android 8.0 Oreo
- Android 9 Pie

- Android 10
- Android 11
- Android 12

3.1.2 Kiến trúc của hệ điều hành android



Hình 3-3 Kiến trúc của hệ điều hành android

Tầng Applications: Là tầng chứa các ứng dụng như: trình duyệt, máy ảnh, thư viện, âm nhạc, điện thoại, game,...

Tầng Application Framework: Tầng Application Framework cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng trong các lớp Java. Các lập trình viên cũng được phép sử dụng các dịch vụ này trong các ứng dụng của họ. Tầng Application Framework bao gồm các dịch vụ chính sau:

- **Activity Manager:** Điều khiển các khía cạnh của vòng đời ứng dụng và Activity Stack.
- **Content Providers:** Cho phép các ứng dụng công bố và chia sẻ dữ liệu với các ứng dụng khác.
- **Resource Manager:** Cung cấp sự truy cập tới các resource được nhúng như chuỗi, thiết lập màu, UI layout.

- **Notifications Manager:** Cho phép các ứng dụng hiển thị thông báo tới người dùng.
- **View System:** Một tập hợp các view được sử dụng để tạo UI cho ứng dụng.

Tầng Android Runtime: Đây là thành phần thứ 3 trong cấu trúc, thuộc về lớp 2 tính từ dưới lên. Phần này cung cấp một thành phần quan trọng gọi là **Dalvik Virtual Machine** là một máy ảo Java đặc biệt, được thiết kế tối ưu cho Android.

Máy ảo Dalvik VM sử dụng các tính năng cốt lõi của Linux như quản lý bộ nhớ, đa luồng, mà thực chất là bên trong ngôn ngữ Java. Máy ảo Dalvik cho phép tất cả các ứng dụng Android chạy trong tiến trình riêng của nó.

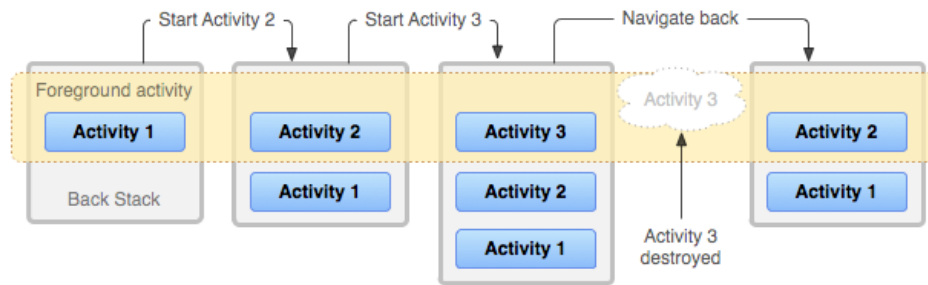
Tầng Libraries: Tầng này gồm các thư viện dựa trên Java. Nó bao gồm các Framework Library giúp xây dựng, vẽ đồ họa và truy cập cơ sở dữ liệu trở nên dễ dàng hơn. Chứa các bộ thư viện, API gần như là cốt lõi của Android, bao gồm bộ thư viện quản lý bề mặt cảm ứng (Surface Manager), OpenGL (phục vụ cho việc dựng đồ họa phức tạp), WebKit, bộ thư viện nổi tiếng libc, cơ sở dữ liệu SQLite hữu ích cho việc lưu trữ và chia sẻ dữ liệu, bộ thư viện thể phát, ghi âm về âm thanh hoặc video. Thư viện SSL chịu trách nhiệm cho bảo mật Internet.

Tầng Linux kernel: Linux Kernel là lớp thấp nhất, là nhân lõi của hệ điều hành, chứa các tập lệnh, driver giao tiếp giữa phần cứng và phần mềm của Android. Nó cung cấp các chức năng cơ bản như quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như: Camera, bàn phím, màn hình, ... Ngoài ra, nó còn quản lý mạng, driver của các thiết bị, điều này gỡ bỏ sự khó khăn về giao tiếp với các thiết bị ngoại vi.

3.2 Kiến thức về lập trình android

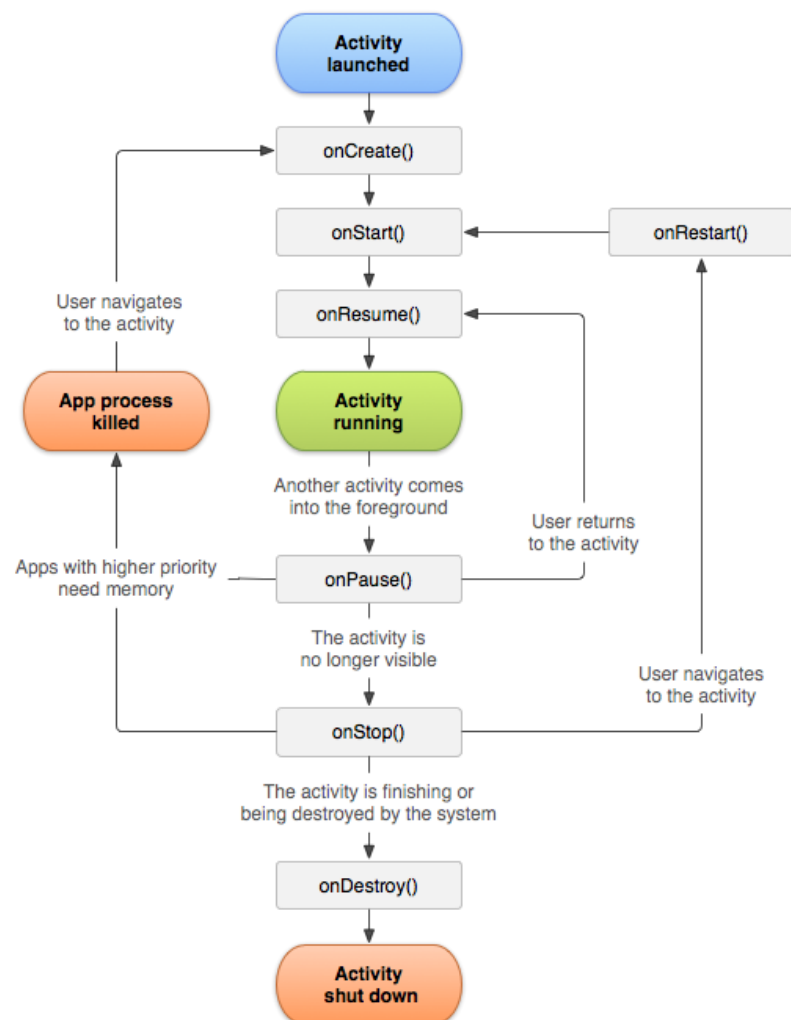
3.2.1 Activity

Activity trong Android được xem là 1 màn hình tương tác trong ứng dụng Android, Activity chịu trách nhiệm chuyển giao sự kiện cho các view trong nó và quản lý vòng đời (lifecycle) của nó, 1 ứng dụng Android có thể có 1 hoặc nhiều Activity, những Activity này có vòng đời độc lập với nhau và được duy trì trong Activity Back Stack.



Hình 3-4 Mô tả hoạt động của các activities trong android

Vòng đời của Activity:



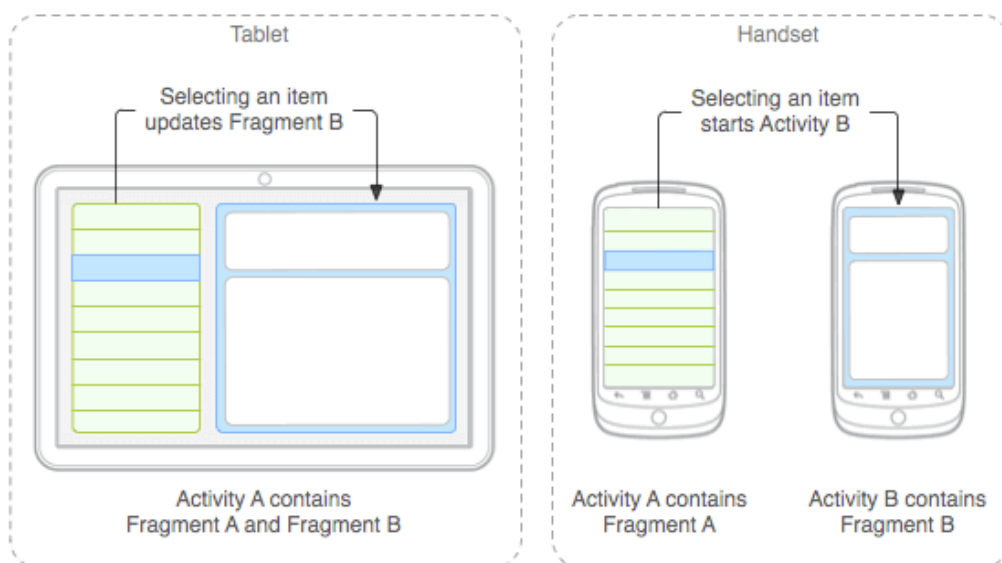
Hình 3-5 Vòng đời của một activity

- **onCreate()** tự động được gọi khi ứng dụng mới start nhưng chưa thể tương tác với người dùng. Trong phương thức này thường sử dụng để

setContentView(), khởi tạo view, inflate view từ xml đăng kí event cho view, khởi tạo Adapter cho ListView, RecyclerView, ...

- **onStart()** tự động được gọi sau khi onCreate() được gọi, phương thức này vẫn chưa tương tác được với người dùng trên UI.
- **onResume()** tự động được gọi sau khi onStart() được gọi, phương thức này tương tác được với UI.
- **onPause()** được sử dụng khá phổ biến trong vòng đời của Activity, thường dùng để lưu lại trạng thái của ứng dụng khi ứng dụng chuẩn bị dừng.
- **onStop()** tự động được gọi sau khi phương thức onPause() được gọi, phương thức này tự động được gọi khi nhấn nút HOME.
- **onRestart()** tự động được gọi khi Activity đã ngưng và đang chuẩn bị bắt đầu lại activity.
- **onDestroy()** tự động được gọi khi nhấn back từ Activity, hoặc gọi phương thức finish() của Activity.

3.2.2 Fragments



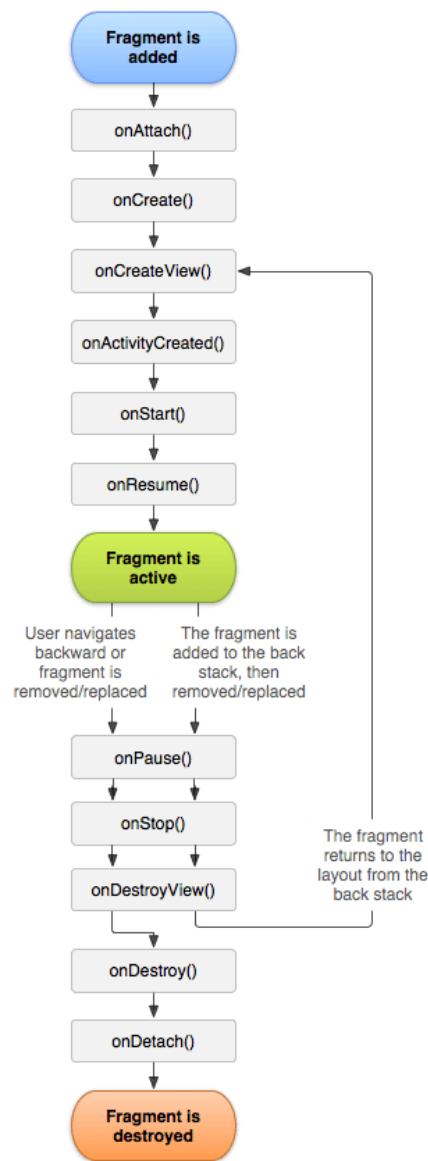
Hình 3-6 Mô tả hoạt động của fragments

Fragment là phần module của thiết kế hoạt động (*activity*), được sử dụng để thể hiện hành vi (*activity*) của giao diện người dùng (*UI*) trong một activity. Bằng cách sử dụng các fragment, chúng ta có thể tạo ra một thiết kế UI linh hoạt, có thể được điều chỉnh dựa trên kích thước màn hình của thiết bị như máy tính bảng, điện thoại thông minh. Ta có

thể xây dựng **multi-pane UI** bằng cách kết hợp nhiều fragment. Fragment có riêng các phương thức callback trong vòng đời và chấp nhận các sự kiện đầu vào của chính nó.

Ngoài ra, ta cũng có thể thêm hoặc xóa các fragment trong một activity trong khi activity đang chạy. Trong Android, fragment sẽ hoạt động như một activity phụ và chúng ta có thể sử dụng lại nó trong nhiều activity. Nói chung, fragment trong Android phải được đưa vào một activity do vòng đời của fragment đó sẽ luôn luôn bị ảnh hưởng bởi vòng đời của activity. Trong trường hợp nếu chúng ta tạm dừng một activity thì tất cả các fragment liên quan đến activity đó cũng sẽ bị dừng lại.

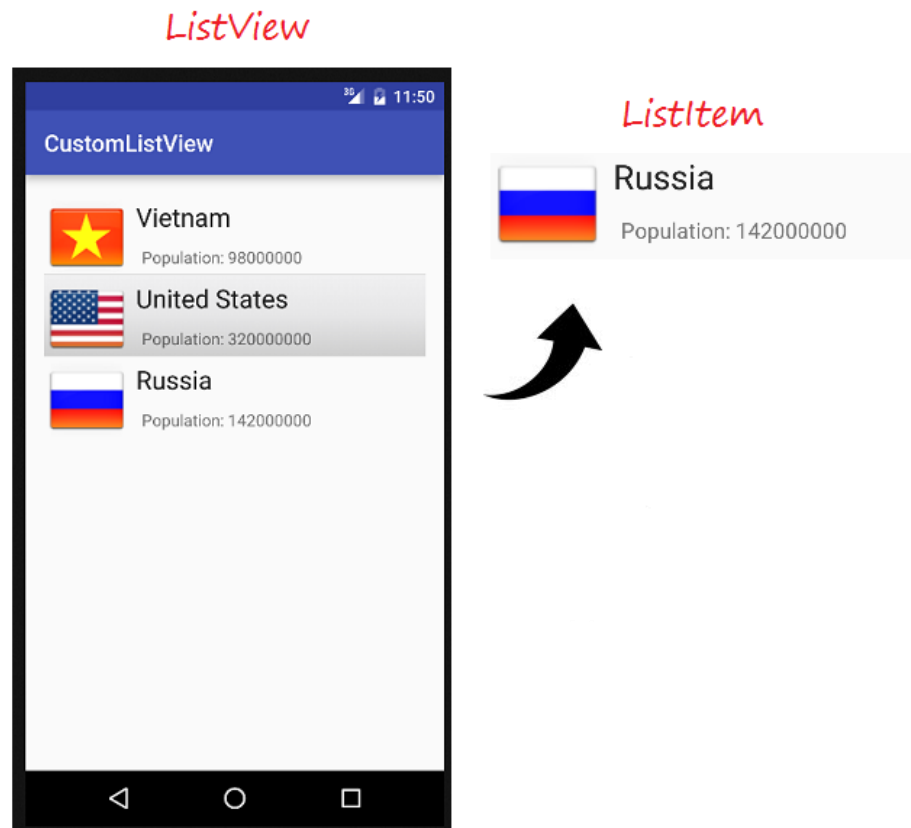
Vòng đời của một Fragment:



Hình 3-7 Vòng đời của một fragment

- **onAttach():** Được gọi khá sớm, ngay khi Activity chứa nó được kích hoạt. Hoặc ngay khi được gắn vào Activity. Callback này được gọi một lần duy nhất trong vòng đời Fragment.
- **onCreate():** Được gọi khi Fragment bắt đầu khởi tạo từ các dữ liệu đầu vào. Khác với onCreate() của Activity, rằng bạn có thể tạo giao diện cho màn hình ở callback này, thì với Fragment chúng ta còn phải đợi qua callback tiếp theo mới có thể tạo giao diện được. Callback này cũng được gọi một lần trong vòng đời Fragment.
- **onCreateView():** Khi Fragment bắt đầu vẽ UI lên màn hình, callback này được gọi. Nên chúng ta sẽ tận dụng callback này cho các thiết lập về giao diện. Bạn thấy rằng, theo như sơ đồ trên, thì callback này sẽ được gọi lại khi mà Fragment được gỡ ra khỏi Activity nhưng được đưa vào Back Stack, và được gọi lại hiển thị sau đó.
- **onActivityCreated():** Được gọi ngay sau khi onCreateView() được gọi. Nó báo hiệu trạng thái Activity chứa nó được khởi tạo hoàn toàn. Tuy ít được sử dụng hơn các callback khác, nhưng bạn cũng có thể tận dụng nó để thay đổi giao diện hay các tương tác với Activity chứa Fragment này thoải mái được rồi.
- **onStart():** Khi Fragment bắt đầu được nhìn thấy bởi người dùng và chuẩn bị nhận tương tác.
- **onResume():** Người dùng hoàn toàn nhìn thấy và tương tác được với Fragment.
- **onPause():** Được gọi như một dấu hiệu cho thấy rằng người dùng đang rời khỏi Fragment hiện tại.
- **onStop():** Fragment chính thức không còn được nhìn thấy nữa.
- **onDestroyView():** Chắc chắn là đối tượng View sẽ bị hủy ở callback này. Và do đó các khởi tạo view của bạn ở onCreateView() sẽ nhanh chóng không còn nữa. Nếu như Fragment được đưa vào Back Stack, thì khi được lấy ra lại sau đó, callback onCreateView() sẽ được gọi lại.
- **onDestroy():** Được gọi khi hủy fragment.
- **onDetach():** Được gọi đến báo hiệu Fragment sẽ được gỡ khỏi Activity đang chứa nó. Kết thúc vòng đời của Fragment.

3.2.3 ListView

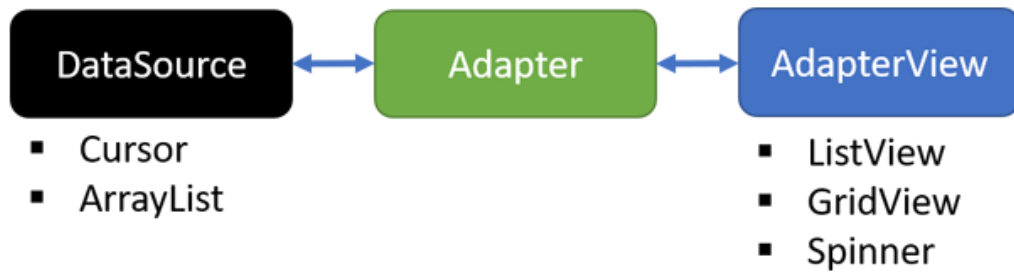


Hình 3-8 Mô tả cách hoạt động của một listview

ListView là một viewgroup, hiển thị các thành phần (elements) theo một danh sách, có thể cuộn được theo chiều thẳng đứng. ListView là một view quan trọng, nó được sử dụng rộng rãi trong các ứng dụng Android. Một ví dụ đơn giản của ListView là danh bạ liên lạc, nơi có một danh sách các địa chỉ liên lạc của bạn hiển thị trong một ListView.

Một ListView được tạo từ một danh sách các ListItem. ListItem là một dòng (row) riêng lẻ trong listview nơi mà dữ liệu sẽ được hiển thị. Bất kỳ dữ liệu nào trong listview chỉ được hiển thị thông qua listItem. Có thể coi listview như là một nhóm cuộn của các ListItem.

3.2.4 Adapter



Hình 3-9 Mô tả cách hoạt động của một adapter

Adapter là một đối tượng của một lớp cài đặt giao diện Adapter. Nó đóng vai trò như là một liên kết giữa một tập hợp dữ liệu và một Adapter View, một đối tượng của một lớp thừa kế lớp trừu tượng AdapterView. Tập hợp dữ liệu có thể là bất cứ điều gì mà trình bày dữ liệu một cách có cấu trúc. Mảng, các đối tượng List và các đối tượng Cursor thường sử dụng bộ dữ liệu.

Một Adapter có trách nhiệm lấy dữ liệu từ bộ dữ liệu và tạo ra các đối tượng View dựa trên dữ liệu đó. Các đối tượng View được tạo ra sau đó được sử dụng để gắn lên bất kỳ Adapter View mà ràng buộc với Adapter

Chúng ta có thể tạo các lớp Adapter riêng từ đầu, nhưng hầu hết các nhà phát triển muốn sử dụng hoặc thừa kế các lớp Adapter được cung cấp bởi Android SDK, chẳng hạn như ArrayAdapter và SimpleCursorAdapter.

3.2.5 Cơ sở dữ liệu SQLite



Hình 3-10 Biểu tượng của SQLite

SQLite là một cơ sở dữ liệu SQL mã nguồn mở, nó lưu trữ dữ liệu vào một tập tin văn bản trên một thiết bị. Nó mặc định đã được tích hợp trên thiết bị Android. Để truy cập dữ liệu này, bạn không cần phải thiết lập bất kỳ loại kết nối nào cho nó như JDBC, ODBC, ... SQLite được Richard Hipp viết dưới dạng thư viện bằng ngôn ngữ lập trình C.

SQLite có các ưu điểm sau:

- Tin cậy: các hoạt động transaction (chuyển giao) nội trong cơ sở dữ liệu được thực hiện trọn vẹn, không gây lỗi khi xảy ra sự cố phần cứng
- Tuân theo chuẩn SQL92 (chỉ có một vài đặc điểm không hỗ trợ)
- Không cần cài đặt cấu hình
- Kích thước chương trình gọn nhẹ, với cấu hình đầy đủ chỉ không đầy 300 kB
- Thực hiện các thao tác đơn giản nhanh hơn các hệ thống cơ sở dữ liệu khách/chủ khác
- Không cần phần mềm phụ trợ
- Phần mềm tự do với mã nguồn mở, được chú thích rõ ràng

3.2.6 Firebase



Hình 3-11 Biểu tượng của firebase

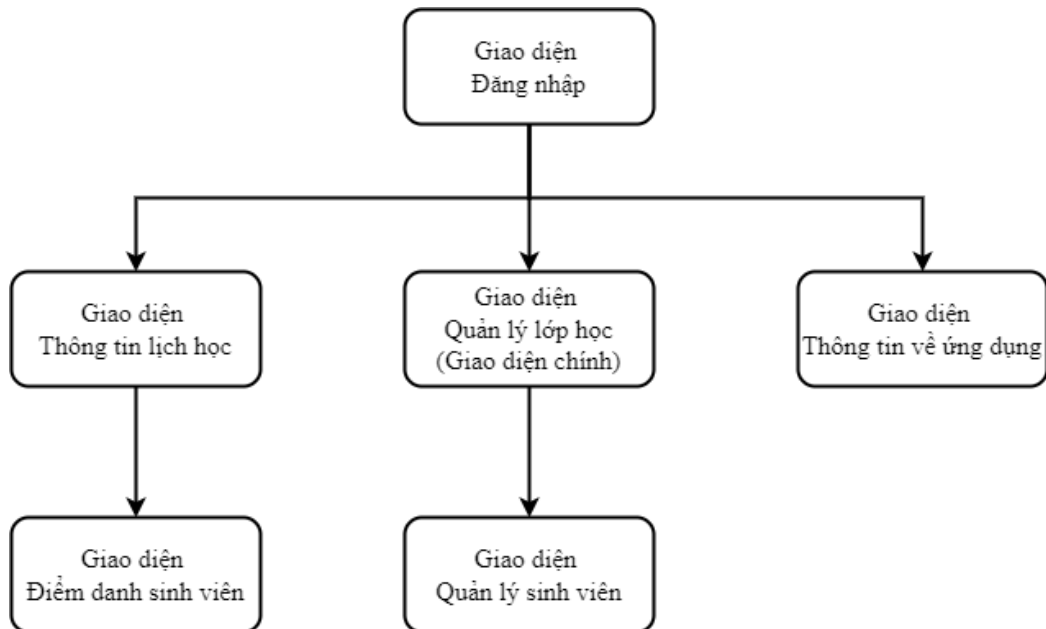
Gần một thập niên trước, **Firebase** ra đời với tiền thân là **Envolve**. Đây là một nền tảng đơn giản chuyên cung cấp những API cần thiết để tích hợp tính năng chat vào trang web. Bên cạnh ứng dụng nhắn tin trực tuyến, **Envolve** còn được người dùng sử dụng để truyền và đồng bộ hóa dữ liệu cho những ứng dụng khác như các trò chơi trực tuyến,... Do đó, các nhà sáng lập đã tách biệt hệ thống nhắn tin trực tuyến và đồng bộ dữ liệu thời gian thực thành hai phần riêng biệt. Trên cơ sở đó, năm 2012, **Firebase** ra đời với sản phẩm cung cấp là **dịch vụ Backend-as-a-Service**. Tiếp đến, vào năm 2014, Google mua lại **Firebase** và phát triển nó thành một dịch vụ đa chức năng được hàng triệu người sử dụng cho đến hiện nay.

Khi đăng ký một tài khoản trên **Firebase** để tạo ứng dụng, bạn đã có một cơ sở dữ liệu thời gian thực. Dữ liệu bạn nhận được dưới dạng JSON. Đồng thời nó cũng luôn được đồng bộ thời gian thực đến mọi kết nối client. Đối với các ứng dụng đa nền tảng, tất cả các client đều sử dụng cùng một cơ sở dữ liệu. Nó được tự động cập nhật dữ liệu mới nhất bất cứ khi nào các lập trình viên phát triển ứng dụng. Cuối cùng, tất cả các dữ liệu này được truyền qua kết nối an toàn SSL có bảo mật với chứng nhận 2048 bit.

Trong trường hợp bị mất mạng, dữ liệu được lưu lại ở local. Vì thế khi có mọi sự thay đổi nào đều được tự động cập nhật lên Server của **Firebase**. Bên cạnh đó, đối với các dữ liệu ở local cũ hơn với Server thì cũng tự động cập nhật để được dữ liệu mới nhất.

CHƯƠNG 4. PHÁT TRIỂN ỨNG DỤNG

4.1 Sơ đồ khối của hệ thống

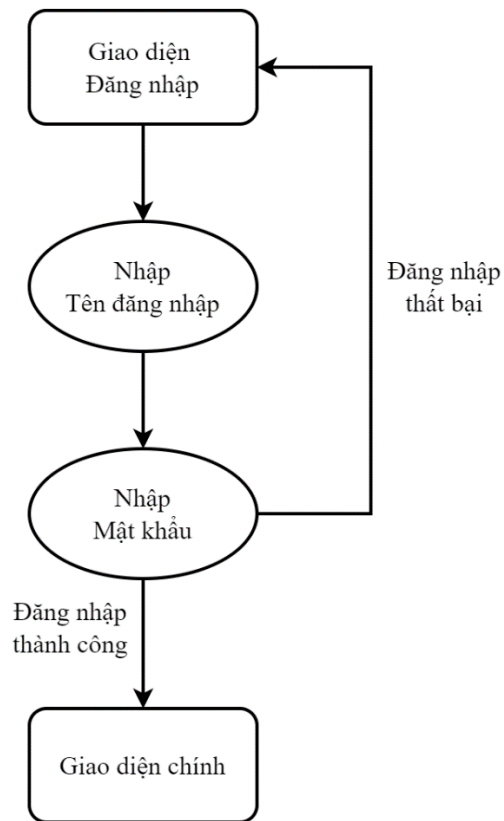


Hình 4-1 Sơ đồ khối của hệ thống

Hình 4.1 cho thấy hệ thống gồm 5 giao diện chính: Giao diện đăng nhập, giao diện thông tin lịch học, giao diện quản lý lớp học, giao diện thông tin về ứng dụng và giao diện quản lý sinh viên. Người dùng sau khi đăng nhập trên giao diện đăng nhập sẽ thao tác trên giao diện thông tin lịch học, giao diện quản lý lớp học, xem thông tin ứng dụng. Người dùng có thể truy cập vào giao diện quản lý sinh viên thông qua giao diện quản lý lớp học.

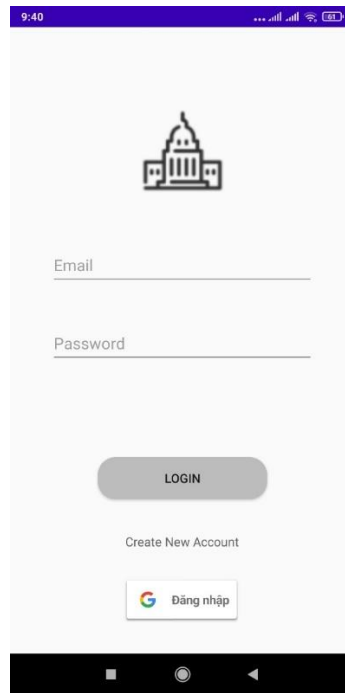
4.2 Các chức năng cơ bản của ứng dụng

4.2.1 Giao diện đăng nhập



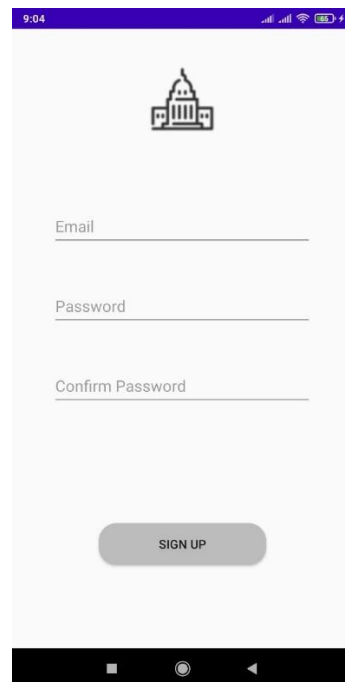
Hình 4-2 Sơ đồ giao diện đăng nhập của hệ thống

Hình 4.2 mô tả các bước đăng nhập vào ứng dụng bao gồm nhập tên đăng nhập, mật khẩu và thông báo phản hồi từ hệ thống.



Hình 4-3 Giao diện đăng nhập của ứng dụng

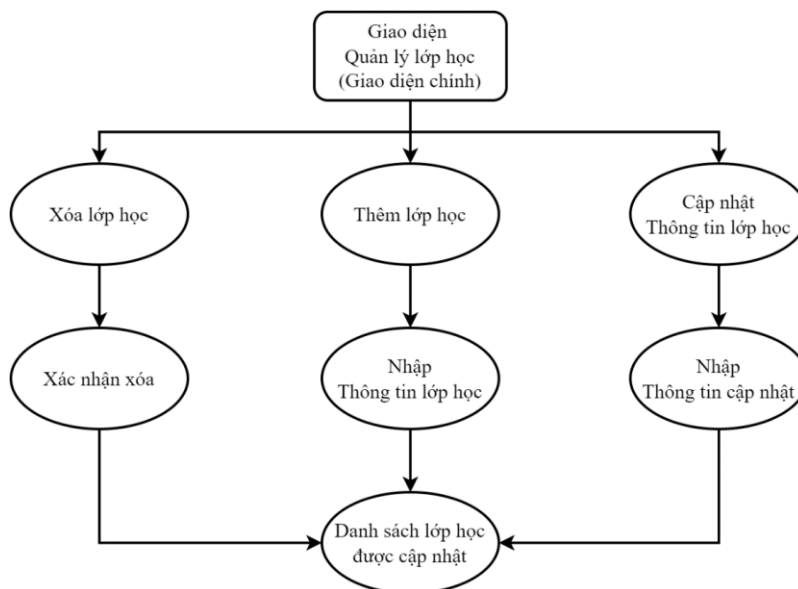
Hình 4.3 là giao diện đăng nhập của ứng dụng gồm mục nhập tên email đăng nhập, mật khẩu ở dạng ẩn nhằm tăng tính bảo mật. Nếu người dùng nhập sai tên email đăng nhập hoặc mật khẩu hệ thống sẽ báo lỗi và yêu cầu nhập lại, nhập đúng hệ thống sẽ đưa người dùng đến giao diện quản lý lớp học.



Hình 4-4 Giao diện đăng ký của ứng dụng

Hình 4.4 là giao diện đăng kí tài khoản của ứng dụng gồm mục nhập tên email đăng nhập, mật khẩu và xác nhận lại mật khẩu ở dạng ẩn nhằm tăng tính bảo mật.

4.2.2 Quản lý thông tin lớp học



Hình 4-5 Sơ đồ quản lý thông tin lớp học

Hình 4-5 mô tả các luồng chức năng chính của quản lý lớp học bao gồm: xóa lớp học, thêm lớp học mới, cập nhật thông tin lớp học và trả về danh sách lớp học. Chi tiết của từng chức năng tương tác với lớp học được viết chi tiết trong các phần sau.

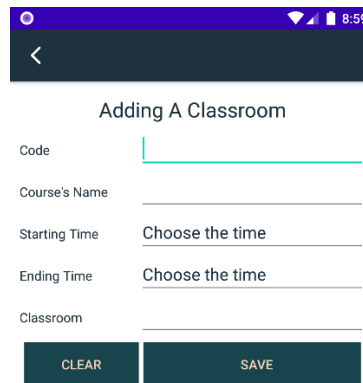
List Of Classrooms				
Order	Code	Course's Name	Timestream	Classroom
1	124863	Lập trình nâng cao	6:30-8:30	D5-502
2	124870	Đa phương tiện	9:30-11:30	D9-205
3	129268	Hệ điều hành	15:0-17:30	D8-106
4	129305	Xử lý ảnh số	12:30-15:0	D8-502
5	129310	Lập trình di động	9:30-11:45	D6-305

Hình 4-6 Giao diện quản lý danh sách các lớp học hiện tại

Hình 4-6 mô tả giao diện của danh sách lớp học hiện tại, ứng với từng tài khoản của người dùng. Tức là, mỗi người dùng sẽ quản lý một danh sách lớp riêng.

4.2.2.1 Chức năng thêm mới một lớp học

Để thêm mới một lớp học, ta nhấn vào nút có biểu tượng là dấu (+) trong giao diện quản lý danh sách các lớp học hiện tại, được mô tả trong Hình 4-6. Sau đó giao diện thêm một lớp học mới sẽ hiện ra.

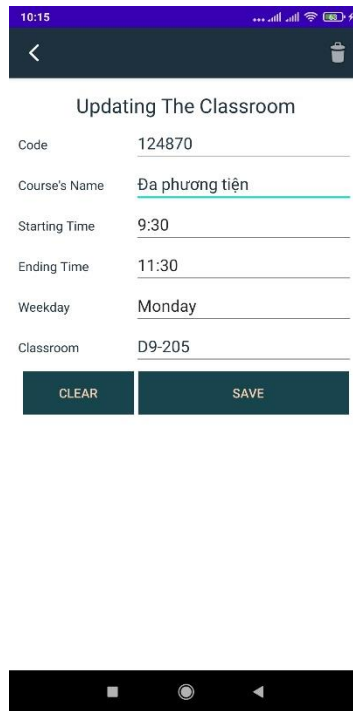


Hình 4-7 Giao diện thêm mới một lớp học

Hình 4-7 là giao diện thêm mới hoặc cập nhật thông tin lớp học, giao diện cho phép người dùng (giáo viên) nhập vào các thông tin của một lớp học bao gồm: Mã số lớp học, tên lớp học, thời gian bắt đầu, thời gian kết thúc lớp học, địa chỉ phòng học. Sau khi người dùng nhấn “Save” hệ thống sẽ trả về danh sách lớp học.

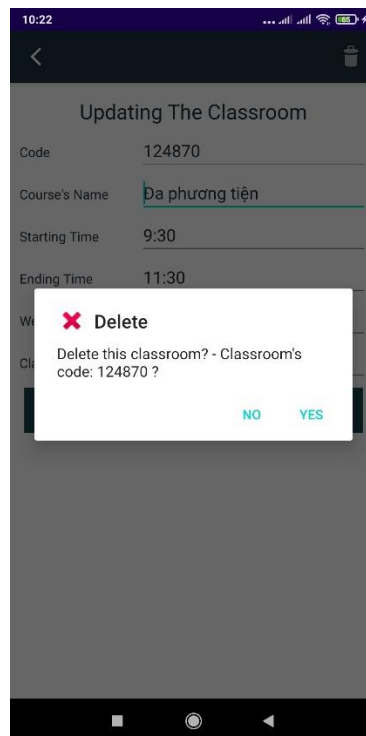
4.2.2.2 Chức năng cập nhật thông tin một lớp học

Để cập nhật thông tin cho một lớp học, ta nhấn giữ vào một item trong listview trong giao diện quản lý danh sách các lớp học hiện tại, được mô tả trong Hình 4-6. Sau đó giao diện cập nhật thông tin lớp học sẽ hiện ra.



Hình 4-8 Giao diện cập nhật thông tin của một lớp học

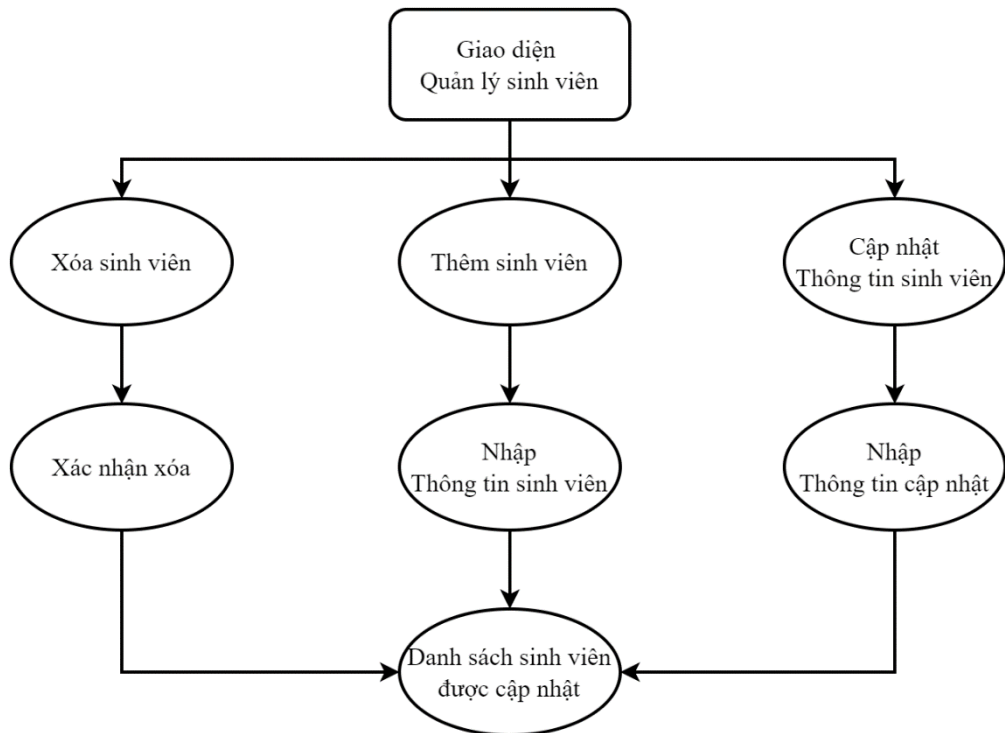
Hình 4-8 là giao diện cập nhật thông tin một lớp học, giao diện cho phép người dùng chỉnh sửa lại thông tin của một lớp học nếu nhập sai hoặc nhập thiếu. Sau khi cập nhật, để lưu lại, người dùng nhấn “Save” để lưu lại thông tin.



Hình 4-9 Giao diện xác nhận xóa thông tin của một lớp học

Hình 4-9 mô tả giao diện xác nhận xóa thông tin của một lớp học. Trong chức năng cập nhật thông tin lớp học cũng kèm theo chức năng xóa lớp học hiện tại. Để xóa một lớp học, từ giao diện thêm mới hoặc cập nhật thông tin lớp học, người dùng nhấn vào biểu tượng thùng rác trong giao diện cập nhật thông tin lớp học, hệ thống sẽ thông báo xác nhận và trả về giao diện danh sách lớp học nếu người dùng đồng ý xóa.

4.2.3 Quản lý thông tin sinh viên



Hình 4-10 Sơ đồ quản lý sinh viên

Hình 4-10 mô tả các chức năng chính của luồng quản lý sinh viên bao gồm: xóa sinh viên, thêm sinh viên mới, tìm kiếm sinh viên cập nhật thông tin sinh viên và trả về danh sách sinh viên sau các thao tác. Từ giao diện danh sách các lớp học Hình 4-6 người dùng nhấn vào một lớp học bất kỳ để vào giao diện quản lý sinh viên của chính lớp đó.

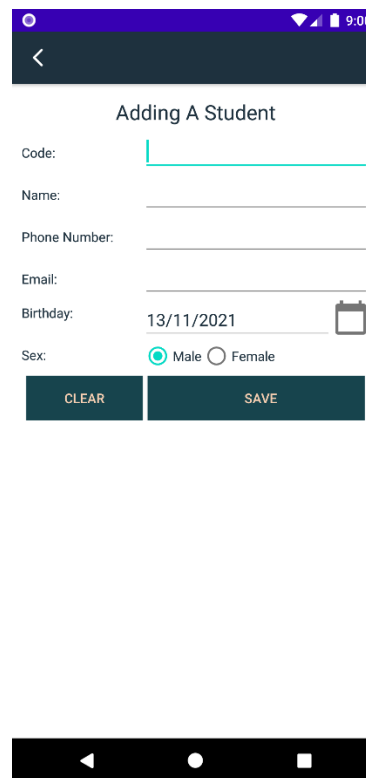


Order	Code	Name	Birthday	Sex
1	2017526	Trần Ngọc B	19/02/1999	♂
2	20170367	Nguyễn Thị C	18/03/1999	♀
3	20171234	Trần Duy A	11/03/1999	♂
4	20175265	Nguyễn Văn C	19/03/1999	♂
5	20175268	Nguyễn Văn A	18/03/1999	♂
6	20175287	Nguyễn Ngọc A	20/05/1999	♂
7	20175362	Nguyễn Ngọc C	04/05/1999	♂
8	20175393	Phan Văn A	17/09/1999	♂
9	20175837	Trần Thị A	04/05/1999	♀
10	20175926	Trần Ngọc C	16/01/1998	♂
11	20176285	Nguyễn Duy C	15/01/1998	♂
12	20176377	Nguyễn Duy C	19/08/1999	♂
13	20176386	Trần Thị A	16/04/1999	♂
14	20176497	Nguyễn Văn A	18/03/1999	♂
15	20185265	Trần Thị Ngọc M	20/01/2000	♀
16	20185362	Nguyễn Thị B	26/01/2000	♀
17	20186476	Nguyễn Văn B	16/03/2000	♂

Hình 4-11 Giao diện quản lý danh sách sinh viên trong một lớp học

Hình 4-11 mô tả giao diện quản lý danh sách sinh viên trong một lớp học, cũng tương tự như giao diện quản lý danh sách lớp học như ở trên. Trên thanh ActionBar hiển thị mã lớp của lớp hiện tại.

4.2.3.1 Chức năng thêm một sinh viên mới



The screenshot shows a mobile application interface for adding a new student. The screen is titled "Adding A Student" and contains the following fields and controls:

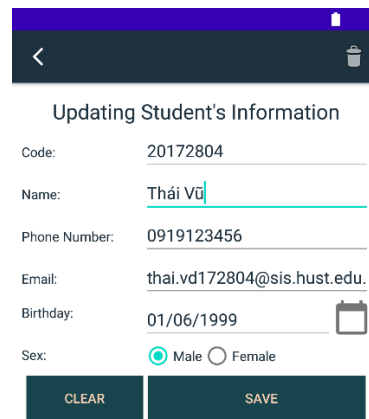
- Code:** A text input field.
- Name:** A text input field.
- Phone Number:** A text input field.
- Email:** A text input field.
- Birthday:** A date input field showing "13/11/2021" with a calendar icon.
- Sex:** Radio buttons for "Male" (selected) and "Female".
- Buttons:** "CLEAR" and "SAVE" buttons at the bottom.

The status bar at the top shows the time as 9:00. The bottom navigation bar is visible.

Hình 4-12 Giao diện thêm một sinh viên mới vào lớp học hiện tại

Hình 4-12 mô tả giao diện thêm một sinh viên mới vào lớp học hiện tại. Người dùng nhập vào các thông tin cơ bản của một sinh viên: Mã số sinh viên, tên, số điện thoại, địa chỉ email, ngày sinh, giới tính. Sau khi hoàn thành xong thông tin, nhấn vào “Save” để lưu lại thông tin của sinh viên này.

4.2.3.2 Chức năng cập nhật thông tin của một sinh viên



Updating Student's Information

Code: 20172804

Name: Thái Vũ

Phone Number: 0919123456

Email: thai.vd172804@sis.hust.edu.

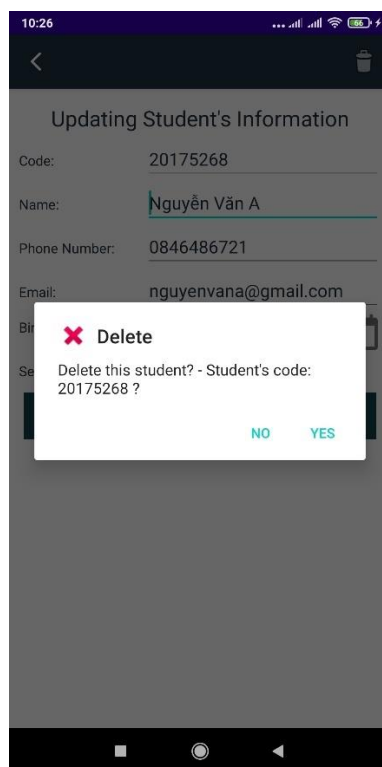
Birthday: 01/06/1999

Sex: ☒ Male ☐ Female

CLEAR SAVE

Hình 4-13 Giao diện cập nhật thông tin của sinh viên

Hình 4-13 là giao diện cập nhật thông tin một sinh viên. Cũng tương tự như giao diện cập nhật thông tin của lớp học, giao diện này cho phép người dùng chỉnh sửa lại thông tin của một sinh viên trong lớp học nếu nhập sai hoặc nhập thiếu. Sau khi cập nhật, để lưu lại, người dùng nhấn “Save” để lưu lại thông tin.

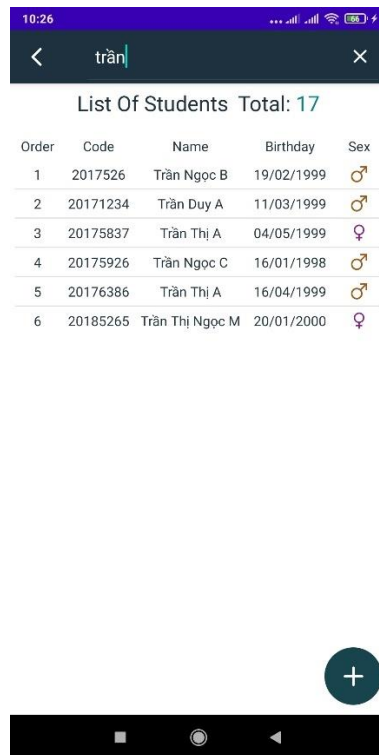


Hình 4-14 Giao diện xác nhận xóa thông tin của một sinh viên

Hình 4-14 mô tả giao diện xác nhận xóa thông tin của một sinh viên. Trong chức năng cập nhật thông tin sinh viên cũng kèm theo chức năng xóa sinh viên hiện tại. Để xóa một sinh viên, từ giao diện cập nhật thông tin sinh viên, người dùng nhấn vào biểu tượng thùng rác, hệ thống sẽ thông báo xác nhận và trả về giao diện danh sách sinh viên nếu người dùng đồng ý xóa.

4.2.3.3 Chức năng tìm kiếm sinh viên

Giao diện danh sách sinh viên (Hình 4-11) cho phép người dùng xem các thông tin cơ bản của một sinh viên, số lượng học sinh trong lớp học. Ngoài ra người dùng còn có thể tìm kiếm sinh viên thông qua tên bằng cách nhập tên sinh viên muốn tìm lên thanh tìm kiếm (nằm trên thanh ActionBar). Mô tả trong Hình 4-15.



Hình 4-15 Giao diện tìm kiếm sinh viên trong một lớp học theo tên

4.2.4 *Giao diện thông tin lịch học và hỗ trợ điểm danh*

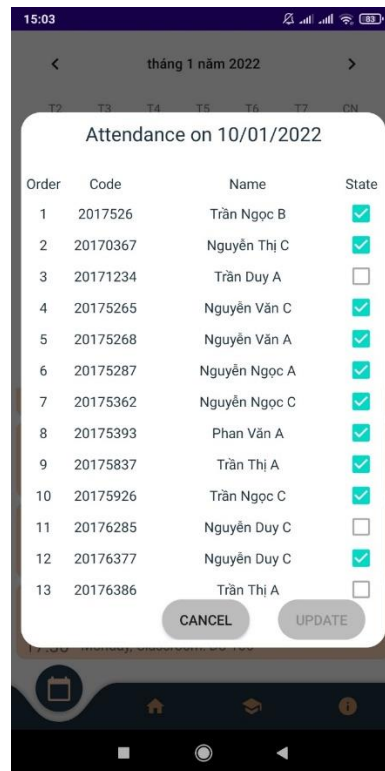
4.2.4.1 *Xem thông tin lịch học*



Hình 4-16 Giao diện thông báo các lớp học theo thời gian thực

Giao diện cho phép người dùng xem thông tin lịch học theo tuần, tháng một cách trực quan. Để xem thông tin lịch học, người dùng chọn vào ngày tương ứng trên giao diện lịch, các lớp học của ngày đó sẽ hiện ra dưới dạng ListView ở bên dưới giao diện lịch.

4.2.4.2 Hỗ trợ điểm danh



Hình 4-17 Giao diện hỗ trợ điểm danh sinh viên theo từng lớp

Giao diện nhằm giúp hỗ trợ người dùng điểm danh sinh viên theo mỗi buổi học. Để mở Dialog điểm danh, người dùng nhấn giữ vào một lớp học, một Dialog điểm danh sẽ xuất hiện, người dùng click chọn vào từng ListItem của từng sinh viên tương ứng để thay đổi trạng thái của State tương ứng với trạng thái có mặt hoặc vắng mặt của từng sinh viên. Sau đó người dùng click vào nút “UPDATE” để cập nhật danh sách điểm danh lên cơ sở dữ liệu

4.2.5 Giao diện thông tin ứng dụng



Hình 4-18 Giao diện thông tin ứng dụng

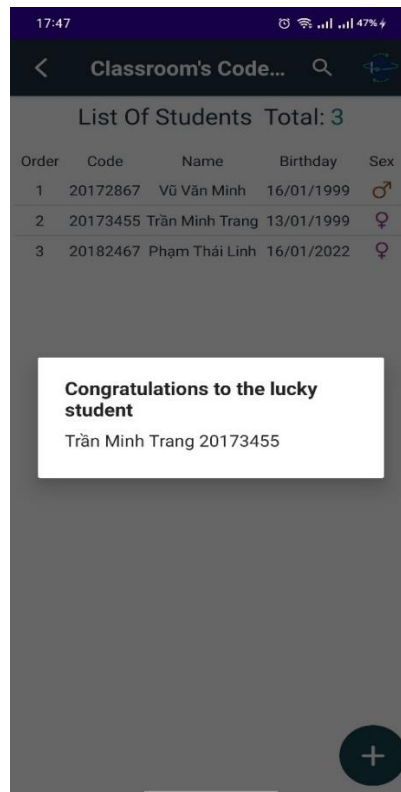
Hình 4-18 mô tả giao diện thông tin ứng dụng bao gồm các thông tin cơ bản về ứng dụng, nhà phát hành.

4.2.6 Một vài chức năng bổ xung tương tác với lớp học

4.2.6.1 Chức năng gọi ngẫu nhiên sinh viên trong lớp học

- Chức năng gọi ngẫu nhiên lớp học được mô tả dùng để gọi ngẫu nhiên sinh viên khi đang trong danh sách lớp học.
- Chuẩn bị: trước tiên chúng ta thêm một icon cho chức năng gọi ngẫu nhiên bằng cách tải ảnh hình xúc xắc, vào File → New → Image Asset, Trong thẻ “Foreground Layer”, chọn hình ảnh đã được cấp và thay đổi kích thước để có được hình dáng phù hợp cho icon, trong thẻ “Background Layer”, sử dụng background đã có sẵn, Click nút Next và sau đó là Finish, icon đã được tạo và chúng ta có thể sử dụng thông qua các lệnh gọi tương ứng.
- Tiếp đến ở trong “*studentsListActivity.java*” chúng ta tạo thêm một lựa chọn sự kiện xảy ra. Trong trường hợp người dùng nhấn icon gọi ngẫu nhiên thì hàm “*attendancerandom*” được gọi tới minh họa như trong Hình 4-19.

- Hàm “*attendancerandom*” thực hiện dùng hàm ramdom để lấy id sinh viên ngẫu nhiên và trả về thông tin sinh viên bao gồm tên và mã số sinh viên trong hộp AlertDialog.
- Giao diện gọi ngẫu nhiên sinh viên được hiển thị trong giao diện danh sách sinh viên như Hình 4-19.



Hình 4-19 Giao diện chức năng gọi ngẫu nhiên một sinh viên trong lớp học

4.2.6.2 Hiệu ứng hoa rơi trong phần đồ họa

- Chuẩn bị: trước tiên chúng ta thêm module cho phần tạo hiệu ứng hoa rơi bằng cách thêm đoạn mã
`“implementation 'com.plattysoft.leonids:LeonidsLib:1.3.2’”` vào phần dependencies của file *build.gradle (Module: ...)* trong *Gradle Scripts* sau đó ấn vào *sync now* để đồng bộ lại các module cho project
- Từ đây, chúng ta đã có thư viện để sử dụng, trong phần code, chúng ta thêm nó vào dự án “`import com.plattysoft.leonids.ParticleSystem`”
- Chúng ta có thể khởi tạo một hiệu ứng tùy ý dựa trên mỗi icon cho trước (chúng ta chuẩn bị trước những icon này, trong trường hợp này sẽ là các icon về hoa lá). Ví dụ ở đây là một đoạn code khởi tạo một đối tượng hoa giấy (confetti)

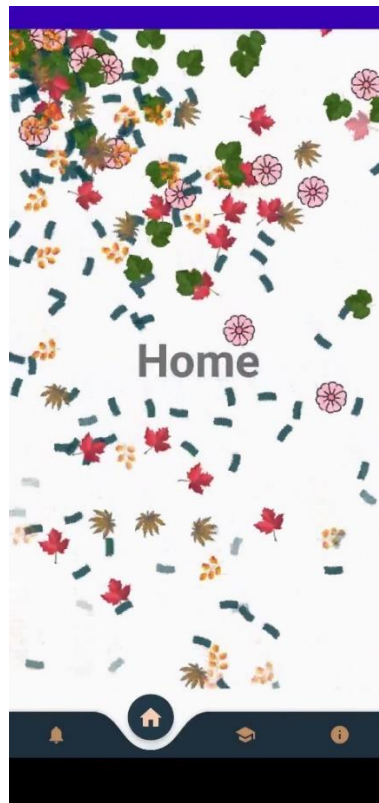
```

new ParticleSystem(getActivity(), 100, R.drawable.ic_aqua_confetti, 10000)
    .setFadeOut(2000)
    .setSpeedModuleAndAngleRange(0f, 0.1f, 0, 90)
    .setRotationSpeed(200)
    .setAcceleration(0.000015f, 90)
    .emit(v.findViewById(R.id.emiter_top_left), 8);

```

- Đầu tiên, khởi tạo một đối tượng với tham số truyền vào là một đối tượng activity của activity hiện tại, 100 là số lượng icon tối đa cho một lần vẽ, *R.drawable.ic_aqua_confetti* là một ảnh xạ định danh cho icon cần vẽ, 10000 là thời gian tồn tại của mỗi icon (tính theo ms).
- *.setFadeOut(2000)*: Thiết lập thời gian biến mất của icon.
- *.setSpeedModuleAndAngleRange(0f, 0.1f, 0, 90)*: Thiết lập tốc độ bé nhất đến lớn nhất của icon khi rơi (từ 0f -> 0.1f), được đo bằng số pixels/ms và góc rơi của cánh hoa (từ 0 -> 90 độ, tức là theo mặt nằm ngang rơi xuống thẳng đứng).
- *.setRotationSpeed(200)*: Thiết lập tốc độ xoay của mỗi icon (cánh hoa) trong không gian của ứng dụng, đo bằng đơn vị độ/giây.
- *.setAcceleration(0.000015f, 90)*: Thiết lập gia tốc cho cánh hoa và chiều của gia tốc.
- *.emit(v.findViewById(R.id.emiter_top_left), 8)*: Thiết lập vị trí cụ thể trên GroupView để thả đối tượng icon. Nếu một lúc nào đó con số vượt quá số lượng icon (cánh hoa) tối đa thì sẽ không có icon mới nào được tạo ra; tham số tiếp theo là số lượng các icon được thả ra trong 1s.

Kết quả đạt được của ứng dụng sau khi áp dụng giống như hình ở dưới đây (với nhiều icon về các loại hoa):



Hình 4-20 Giao diện chính của ứng dụng

CHƯƠNG 5. KIỂM THỬ ỨNG DỤNG

5.1 Môi trường kiểm thử

5.1.1 Chạy trên máy ảo

Ứng dụng được chạy trên Android Studio IDE, máy ảo với cấu hình sử dụng là: Pixel 5 (api: 26, resolution: 1080 x 2340: 440dpi, Target: Android 8.0, CPU/ABI: x86).

5.1.2 Chạy trên máy thật

Cấu hình của máy thật được sử dụng để kiểm thử là:

- Máy của Tới: Realme 5 pro.
- Máy của Tùng: Redmi 5 pro.

5.2 Kịch bản kiểm thử cho ứng dụng và sửa lỗi chi tiết

5.2.1 Kiểm thử phần đăng nhập cho ứng dụng

5.2.1.1 Kịch bản kiểm thử

Thiết bị	Thao tác	Tổng số lần thực hiện	Độ tin cậy	Ghi chú
Máy ảo	Nhập tài khoản và mật khẩu sai, bị yêu cầu nhập lại.	10	100%	Không
Máy ảo	Nhập tài khoản và mật khẩu đúng (tài khoản và mật khẩu là “admin”), đăng nhập thành công.	15	100%	Không
Máy thật	Đăng nhập và đăng xuất bằng tài khoản google.	20	100%	Không

Bảng 5-1 Danh sách các thiết bị dùng để kiểm thử

5.2.2 Kiểm thử xung đột khi hiển thị thông báo lớp học theo thời gian thực

5.2.2.1 Kịch bản kiểm thử

Thiết bị	Thao tác	Tổng số lần	Độ tin	Ghi chú
----------	----------	-------------	--------	---------

		thực hiện	cây	
Máy ảo	Nhập đầy đủ thông tin của 10 lớp học (tham số weekly giống nhau), kiểm tra hoạt động của phần notification.	7	100%	Không
Máy ảo	Nhập đầy đủ thông tin của một vài lớp học và thiếu thông tin giờ bắt đầu của một lớp học (các lớp học đều có tham số weekly giống nhau).	5	0% (lỗi)	Khi nhập thiếu thông tin về giờ bắt đầu, ứng dụng bị lỗi kiểm tra logic với thời gian bắt đầu của các lớp khác

Bảng 5-2 Kịch bản kiểm thử phần hiển thị thông báo lớp học

5.2.2.2 Sửa lỗi chi tiết

- Sửa lỗi phần nhập thông tin giờ bắt đầu bị thiếu của một lớp học:
 - Bắt người dùng phải nhập vào thời gian bắt đầu và kết thúc của một lớp học, nếu thông tin này bị thiếu thì sẽ không được thêm một lớp mới (sửa trong phần *EditClassroomActivity.java*)

```
String etStartTime = etStart.getText().toString();
if (etStartTime.length() == 0)
{
    etStart.setError("?");
    etStart.requestFocus();
    return;
}
String etEndTime = etEnd.getText().toString();
if (etEndTime.length() == 0)
{
    etEnd.setError("?");
    etEnd.requestFocus();
    return;
}
```

5.2.3 Kiểm thử xung đột dữ liệu trong SQLite database

5.2.3.1 Kịch bản kiểm thử

Thiết bị	Thao tác	Tổng số lần thực hiện	Độ tin cậy	Ghi chú
Máy ảo	Tạo ra 10 lớp học và thực hiện thao tác thêm, sửa, xóa, kiểm tra tính đúng đắn của thông tin các lớp học.	10	100%	Kiểm tra truy vấn cho database lớp học
Máy ảo	Tạo ra 1 lớp học bất kỳ và nhập thông tin của 20 sinh viên vào lớp học đó, thực hiện các thao tác thêm, sửa, xóa, kiểm tra tính đúng đắn của thông tin các sinh viên trong lớp học.	10	100%	Kiểm tra truy vấn cho database sinh viên
Máy ảo	Tạo ra 2 lớp học bất kỳ và nhập thông tin một sinh viên với mã số sinh viên giống nhau vào hai lớp đó. Tiến hành xóa một sinh viên đó trong 2 lớp học. Kiểm tra tính đúng đắn của dữ liệu.	3	0% (lỗi)	Khi xóa một sinh viên đi, thông tin sinh viên ở lớp kia bị sai (hoặc bị lỗi app)

Bảng 5-3 Kịch bản kiểm thử phần dữ liệu trong SQLite

5.2.3.2 Sửa lỗi chi tiết

- Chưa có kế hoạch sửa lỗi, vì trong tương lai sử dụng firebase để lưu trữ dữ liệu thông tin các đối tượng.

5.2.4 Kiểm thử xung đột dữ liệu trong Firebase database

5.2.4.1 Kịch bản kiểm thử

Thiết bị	Thao tác	Tổng số lần thực hiện	Độ tin cậy	Ghi chú
----------	----------	-----------------------	------------	---------

Máy thật	Đăng ký tài khoản mới với địa chỉ email riêng, kiểm tra các thông tin trong từng tài khoản riêng lẻ.	5	100%	Không có lỗi
Máy thật	Tạo ra 10 lớp học và thực hiện thao tác thêm, sửa, xóa, kiểm tra tính đúng đắn của thông tin các lớp học (ứng với từng tài khoản riêng lẻ).	5	100%	Kiểm tra truy vấn cho database của lớp học
Máy thật	Tạo ra 1 lớp học bất kỳ và nhập thông tin của 20 sinh viên vào lớp học đó, thực hiện các thao tác thêm, sửa, xóa, kiểm tra tính đúng đắn của thông tin các sinh viên trong lớp học (ứng với từng tài khoản riêng lẻ).	5	100%	Kiểm tra truy vấn cho database của sinh viên
Máy thật	Tạo ra 2 lớp học bất kỳ và nhập thông tin một sinh viên với mã số sinh viên giống nhau vào hai lớp đó. Tiến hành xóa một sinh viên đó trong 2 lớp học. Kiểm tra tính đúng đắn của dữ liệu (ứng với từng tài khoản riêng lẻ).	5	100%	Không có lỗi

Bảng 5-4 Kịch bản kiểm thử phần dữ liệu trong firebase

KẾT LUẬN

Sau quá trình thực hiện đề tài “*Quản lý thông tin lớp học*” nhóm chúng em đã học thêm được rất nhiều điều, nhóm đã có một cái nhìn tổng quan về android cũng như các kỹ thuật lập trình android. Kết hợp với những kiến thức trên lớp, chúng em đã lồng ghép vào trong dự án để ứng dụng trực tiếp những gì được học.

Các loại kỹ năng mềm như lập kế hoạch, phân chia công việc, tìm tài liệu, thuyết trình,... cũng được nhóm sử dụng rất nhiều trong quá trình thực thi nên khi kết thúc đề tài, những kỹ năng này của các thành viên trong nhóm tăng lên rất nhiều.

Chúng em xin chân thành gửi lời cảm ơn thầy Đỗ Trọng Tuấn đã giúp đỡ chúng em hoàn thiện đề tài và mở rộng một số kỹ năng về lập trình android, kiến trúc của một hệ thống phần mềm chạy trên di động và những kinh nghiệm quý giá trong việc phát triển ứng dụng. Trong quá trình thực hiện đề tài, chúng em được học hỏi những điều bổ ích, những kinh nghiệm thực tế, điều này giúp chúng em trưởng thành hơn về mặt chuyên môn cũng như kỹ năng mềm để có thể trở thành một kỹ sư tốt trong tương lai.

Chúng em xin chân thành cảm ơn !

TÀI LIỆU THAM KHẢO

- [1] "https://en.wikipedia.org," [Online]. Available: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)). [Accessed 20 11 2021].
- [2] "https://developer.mozilla.org," [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>. [Accessed 20 11 2021].
- [3] "https://openplanning.net," [Online]. Available: <https://openplanning.net/10433/android-sqlite-database>. [Accessed 20 11 2021].