



NHIỆM VỤ ĐỒ ÁN MÔN HỌC

Họ và tên : **ĐẶNG QUỐC LAI** MSSV : **1811545103**
Chuyên ngành : **Kỹ thuật phần mềm** Lớp : **18DTH2A**
Email : **danglai.mail@gmail.com** SĐT : **077.249.0922**
Tên đề tài : **Tìm hiểu Ngôn Ngữ Android và Lập trình Ứng dụng tìm kiếm quán ăn ngon.**

Giảng viên hướng dẫn : **ThS. Giang Hào Côn**

Thời gian thực hiện : Từ ngày **17/10/2020** đến **30/12/2020**.

Nhiệm vụ/nội dung (mô tả chi tiết nội dung, yêu cầu, phương pháp...):

- Tìm hiểu về lập trình Android và các kiến thức cơ bản về SQLite.
- Xây dựng giao diện và xử lý sự kiện trên các giao diện:
 - + Đăng nhập.
 - + Màn hình chính.
 - + Màn hình Chi tiết quán ăn.
 - + Màn hình Tìm kiếm.
 - + Màn hình Xem bình luận về quán ăn.
- Viết báo cáo đồ án, in và đóng cuốn theo biểu mẫu qui định.

Nội dung và yêu cầu đã được thông qua Bộ môn.

TP. HCM, ngày ... tháng ... năm 20...

TRƯỞNG BỘ MÔN

GIẢNG VIÊN HƯỚNG DẪN

(Ký tên)

(Ký tên)

ThS. Phạm Văn Đăng

ThS. Giang Hào Côn

LỜI CẢM ƠN

Tôi chân thành cảm ơn Giảng viên hướng dẫn ThS. Giang Hào Côn, người đã hướng dẫn tận tình, đóng góp các ý kiến chuyên môn cho báo cáo lần này. Thầy đã động viên tinh thần tôi cố gắng hoàn thành những nghiên cứu đặt ra. Thầy cũng cung cấp một số tài liệu liên quan đến đề án mà tôi đang nghiên cứu và ân cần nhắc nhở tôi đến tiến độ thực hiện báo cáo này.

Tôi cũng chân thành gửi lời cảm ơn đến các Thầy, Cô khoa Công nghệ thông tin trường Đại học Nguyễn Tất Thành đã giảng dạy, hướng dẫn, trang bị các kiến thức cho tôi trong 2 năm học vừa qua, từ các kiến thức cơ bản đến các vấn đề chuyên sâu.

Tôi xin gửi lời cảm ơn đến anh, chị, bạn bè, đồng nghiệp bằng nhiều hình thức khác nhau đã giúp đỡ tôi trong quá trình học tập tại trường cũng như trong thời gian hoàn thành báo cáo.

Tôi xin gửi lời cảm ơn đến anh, chị, bạn bè, đồng nghiệp bằng nhiều hình thức khác nhau đã giúp đỡ tôi trong quá trình học tập tại trường cũng như trong thời gian hoàn thành báo cáo này.

Sinh viên thực hiện

(Ký tên)

ĐẶNG QUỐC LAI

LỜI NÓI ĐẦU

Ngày nay, khi công nghệ thông tin gần như xuất hiện ở mọi phương diện đời sống của con người, từ sinh hoạt cá nhân cho đến công việc cộng đồng thì một thiết bị điện thoại thông minh là không thể thiếu. Đồng thời, khi mức sống con người càng nâng cao thì nhu cầu ăn uống cũng ngày càng phức tạp. Vì vậy, việc sử dụng một thiết bị thông minh để phục vụ nhu cầu ăn uống là xu thế tất yếu trong tương lai. Khác với kiểu ăn uống truyền thống, ngày nay, chúng ta sử dụng thiết bị di động để tìm kiếm, thậm chí đặt đồ ăn online để có người giao tới nhà, nên việc phát triển một phần mềm hỗ trợ người dùng tìm kiếm, đặt đồ ăn là vô cùng cần thiết.

Phần mềm không chỉ giúp người dùng tìm kiếm quán ăn gần, hay lọc các quán ăn theo món, theo địa chỉ, giá tiền,... mà còn giúp người dùng đăng tải bình luận về quán ăn. Việc sử dụng bình luận để đánh giá quán ăn không chỉ giúp tạo sự cạnh tranh công bằng giữa các quán mà còn giúp người dùng có thể dễ dàng chọn quán ăn phù hợp với mình hơn.

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm đồ án:

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày ... tháng ... năm 20...

GIẢNG VIÊN HƯỚNG DẪN

(Ký tên)

ThS. Giang Hào Côn

BẢNG PHÂN CÔNG THỰC HIỆN ĐỒ ÁN

	SINH VIÊN 1	SINH VIÊN 2
Họ tên	Đặng Quốc Lai	
MSSV	1811545103	
Nhiệm vụ	<ul style="list-style-type: none">- Thiết kế giao diện các màn hình.- Thiết kế Database, tạo database mẫu cho ứng dụng.- Xử lý sự kiện trên các màn hình.- Làm báo cáo cuối kỳ.	

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	1
1.1. Giới thiệu đề tài.....	1
1.2. Cơ sở khoa học và tính thực tiễn của đề tài.....	2
1.3. Mục tiêu đề tài.....	3
1.4. Phạm vi đề tài.....	3
1.5. Phương pháp nghiên cứu và phạm vi nghiên cứu.....	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	4
2.1. Giới thiệu tổng quan lập trình Android	4
2.1.1. Sự ra đời của Android.....	4
2.2. Các thành phần ứng dụng của Android	4
2.2.1. Activity.....	4
2.2.2. Intent	5
2.2.3. View	5
2.2.4. Service.....	5
2.2.5. Broadcast Receiver	5
2.2.6. Content provider	6
2.2.7. Context	6
2.2.8. Notification.....	6
2.2.9. Toast.....	6
2.2.10. Alert Dialog.....	7
2.3. Các loại Layout trong ứng dụng Android.....	7
2.3.1. Linear Layout	7
2.3.2. Relative Layout	7

2.3.3. Frame Layout	7
2.3.4. Table Layout.....	7
2.3.5. Absolute Layout	8
2.3.6. ListView/RecyclerView.....	8
2.3.7. GridView	8
2.4. Các điều khiển (View/Widget) trong Android	8
2.4.1. EditText.....	8
2.4.2. ImageView	9
2.4.3. Button.....	9
2.4.4. CheckBox	9
2.4.5. Switch / SwitchCompat.....	10
2.4.6. RadioButton và RadioGroup.....	10
2.4.7. TextView	10
2.5. Giao diện và xử lý sự kiện trong Android	11
2.5.1. Giao diện (Activity)	11
2.5.2. Xử lý sự kiện trong Android	12
2.6. Tài nguyên Ứng dụng trong Android	12
2.6.1. Intent Filter	12
2.6.2. AndroidManifest.....	12
2.7. Đa ngôn ngữ và xử lý tập tin.....	13
2.7.1. Đa ngôn ngữ	13
2.7.2. Xử lý tập tin.....	13
2.7.3. Tập tin XML.....	13
2.8. Asset và SharedPreferences	14

2.8.1. Asset.....	14
2.8.2. SharedPreferences	14
2.9. SQLite.....	14
2.9.1. Giới Thiệu SQLite	14
2.9.2. Truy vấn trong SQLite	15
2.9.3. Phương thức Query()	15
2.9.4. Các phương thức với Cursor	16
2.9.5. Các phương thức khác:	16
2.10. Content Provider và Thread	16
2.10.1. Content Provider.....	16
2.10.2. Đa tiến trình trong Android.....	17
CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT ỨNG DỤNG	19
3.1. Thiết kế	19
3.1.1. Mô hình chức năng BFD, DFD.....	19
3.1.2. Mô hình dữ liệu ERD.....	21
3.2. Cài đặt ứng dụng	22
3.2.1. Màn hình đăng nhập	22
3.2.2. Màn hình chính.....	23
3.2.3. Màn hình xem chi tiết quán ăn.....	24
3.2.4. Màn hình tìm kiếm quán ăn	25
3.2.5. Màn hình xem đánh giá quán ăn	26
KẾT LUẬN.....	27
TÀI LIỆU THAM KHẢO.....	28

DANH MỤC HÌNH ẢNH

Hình 3.1 Sơ đồ BFD.....	19
Hình 3.2 Sơ đồ DFD	20
Hình 3.3 Sơ đồ ERD	21
Hình 3.4 Màn hình đăng nhập	22
Hình 3.5 Màn hình chính.....	23
Hình 3.6 Màn hình xem chi tiết quán ăn.....	24
Hình 3.7 Các màn hình tìm kiếm quán ăn.....	25
Hình 3.8 Màn hình xem đánh giá quán ăn	26

DANH SÁCH TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	CSDL	Cơ sở dữ liệu
2	MVC	Model – Control – View
3	HTML	Hypertext Markup Language
4	BFD	Business Function Diagram
5	DFD	Data Flow Diagram
6	ERD	Entity Relationship Diagram

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1. Giới thiệu đề tài

Ngày nay, khi công nghệ thông tin gần như xuất hiện ở mọi phương diện đời sống của con người, từ sinh hoạt cá nhân cho đến công việc cộng đồng thì một thiết bị điện thoại thông minh là không thể thiếu. Đồng thời, khi mức sống con người càng nâng cao thì nhu cầu ăn uống cũng ngày càng phong phú. Không chỉ là nhu cầu ăn no mặc ấm mà còn là ăn ngon mặc đẹp. Nhưng cuộc sống cứ cuốn con người theo guồng công việc và nhiều người hiện đại không có thời gian để tự tay vào bếp nấu ăn nhưng vẫn mong muốn được ăn ngon và tiện lợi. Nếu như trước đây mọi người phải đến tận cửa hàng để mua đồ ăn hoặc mua mang về thì bây giờ việc giao hàng tận nơi mang lại nhiều thuận tiện cho khách. Thay vì đi lại giữa điều kiện thời tiết không thuận lợi, xe cộ đông đúc khách hàng có thể đặt món thông qua ứng dụng giao đồ ăn. Quá trình để có một bữa ăn đơn giản chỉ là sử dụng chiếc điện thoại thông minh hoặc máy tính có kết nối internet, lựa chọn món, đặt hàng và ngồi chờ món ăn được đưa đến tận tay. Bên cạnh đó các hàng quán cũng quen với việc bán online nên việc làm đồ và đóng gói cũng tối ưu hơn, đảm bảo được chất lượng sản phẩm khi tới tay người dùng. Hơn nữa, khách hàng mục tiêu của dịch vụ đặt giao đồ ăn đa phần là người trẻ: học sinh, sinh viên, dân văn phòng, v.v... Nhóm khách hàng này thành thạo công nghệ, thích những trải nghiệm mới lạ và không có nhiều thời gian rảnh. Chính nhờ khả năng phục vụ một lượng lớn khách hàng như vậy nên ngành dịch vụ đặt món và giao hàng qua ứng dụng ngày một phát triển và thay đổi cách thức mua bán của thị trường nhanh như vậy.

Một trong những lý do khác tác động trực tiếp đến sự phát triển nhanh chóng của dịch vụ đặt giao đồ ăn là dịch bệnh Covid19 và lệnh giãn cách xã hội. Người người, nhà nhà không thể đến trực tiếp tại cửa hàng, bằng cách thuận tiện nhất là đặt giao đồ ăn đã đáp ứng đủ nhu cầu ăn uống của khách và cứu sống không ít hàng quán. Qua đó cũng chứng minh sự thuận tiện khi dịch vụ đặt giao hàng ngày càng phát triển, mang lại nhiều giá trị cho người dùng.

Bên cạnh đó, những lợi ích có thể kết hợp giữa nhiều bên tạo ra hiệu ứng thu hút khách hàng như:

- Nhiều khuyến mãi, ưu đãi.

- Hỗ trợ ship và bình ổn giá.
- Thời gian ship nhanh và hỗ trợ liên tục.
- Khách hàng theo dõi được đơn hàng.
- Mua ít cũng hỗ trợ giao cho khách hàng.
- Khách hàng có thể đánh giá trực tiếp chất lượng món ăn cũng như dịch vụ ship hàng trực tiếp trên app.

Vì vậy, việc sử dụng một thiết bị thông minh để phục vụ nhu cầu này là một lựa chọn thông minh của khách hàng. Hiểu được xu hướng này, qua quá trình tìm hiểu và phân tích, em quyết định sẽ thiết kế và xây dựng một ứng dụng đặt đồ ăn online trên hệ điều hành Android với mong muốn có thể ứng dụng được vào thực tế và giải quyết nhu cầu của khách hàng tìm kiếm dễ dàng hơn.

Một vài lợi ích mà ứng dụng này mang lại như: không chỉ giúp người dùng tìm kiếm quán ăn gần, hay lọc các quán ăn theo món, theo địa chỉ, giá tiền,... mà còn giúp người dùng dễ dàng đánh giá chất lượng của quán ăn hay các món ăn của quán một cách khách quan và thực tế nhất có thể. Điều này không chỉ giúp tạo sự cạnh tranh công bằng giữa các quán mà còn giúp người dùng có niềm tin vào quán ăn mà mình lựa chọn.

1.2. Cơ sở khoa học và tính thực tiễn của đề tài

Như đã giới thiệu trên, đi từ thực tiễn, cuộc sống công nghệ 4.0 ngày càng phát triển và nhu cầu của người tiêu dùng ngày càng số hóa. Mua hàng online, đặt xe online, đặt phòng online... và đặt đồ ăn online hoàn toàn nằm trong nhu cầu thiết yếu của chúng ta. Đặc biệt, ở những thành phố lớn như Hà Nội và thành phố Hồ Chí Minh, nhu cầu số hóa lại càng cao hơn so với các vùng khác. Khi dịch vụ kinh doanh ẩm thực đang mở rộng khắp nơi, nhu cầu kết nối giữa người mua và người bán tăng cao. Nhiều website, fanpage ra đời nhằm giới thiệu, quảng bá cho các quán ăn, nhà hàng bằng nhiều hình thức khác nhau: chạy quảng cáo, viết review... thì hình thức đặt đồ ăn online qua app cũng vô cùng phổ biến. Chẳng những giúp người dùng đáp ứng được nhu cầu mà còn giúp họ có cái nhìn khách quan hơn về các quán ăn, nhà hàng và cả những app đặt đồ ăn uy tín, tiện lợi. Với chiến lược lấy khách hàng là trọng tâm, thấu hiểu và

giải quyết nhu cầu của khách hàng, những nhà phát triển ứng dụng phải luôn bắt kịp xu hướng và cải tiến công nghệ sao cho đem lại sự tiện lợi nhất cho khách hàng. Nhiều app đặt đồ ăn ra đời tuy nhiên vẫn còn một số thiếu sót và chưa thực sự hài lòng khách hàng và đem lại tiện lợi nhất cho người bán. Hiểu được những vấn đề đang tồn tại, em đánh giá đề tài mình đang thực hiện hoàn toàn có khả năng phát triển trong thực tiễn. Mặc dù, lúc bắt đầu xây dựng sẽ có nhiều khó khăn và thiếu sót, nhưng em sẽ cố gắng hoàn thiện từng ngày để trở thành chiếc app tốt nhất trong mắt người tiêu dùng.

1.3. Mục tiêu đề tài

Xây dựng được ứng dụng đặt đồ ăn online chạy trên nền Android có tính thực tiễn cao, có khả năng triển khai ứng dụng vào thực tế, giúp được nhiều quán ăn, nhà hàng cũng như khách hàng có thể thuận tiện trong mua – bán sản phẩm và dịch vụ ẩm thực. Những chức năng của app sẽ đáp ứng cơ bản nhu cầu của người dùng và giao diện thu hút, thân thiện, phù hợp với chủ đề được chọn. Người dùng sử dụng app sẽ đánh giá tích cực và đưa ra góp ý để cải thiện app ngày một hoàn thiện hơn.

1.4. Phạm vi đề tài

- Xây dựng được các màn hình bao gồm: đăng nhập ứng dụng, màn hình chính, màn hình xem chi tiết cửa hàng, màn hình xem các đánh giá của các khách hàng trước.
- Thiết kế được cơ sở dữ liệu, thêm dữ liệu mẫu cho ứng dụng.
- Xử lý sự kiện, chuyển đổi qua lại giữa các màn hình.

1.5. Phương pháp nghiên cứu và phạm vi nghiên cứu

- Khảo sát lấy yêu cầu người sử dụng điện thoại về các ứng dụng mua bán trên mobile để xây dựng ý tưởng, khảo sát các trang web chuyên về ăn uống như foody, now,...
- Tiếp đến, tiến hành nghiên cứu công nghệ phát triển ứng dụng chạy trên hệ điều hành Android. Xây dựng phía client một app có thể cài đặt trên điện thoại hệ điều hành Android đáp ứng các yêu cầu cơ bản của hệ thống.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu tổng quan lập trình Android

2.1.1. Sự ra đời của Android

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 iPhone đầu tiên và 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”.

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.

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. Chiếc điện thoại Android đầu tiên được bán vào năm 2008.

Năm 2010, Google khởi đầu dòng thiết bị Nexus với thiết bị đầu tiên của HTC là Nexus One.

Năm 2013, ra mắt loạt thiết bị phiên bản GPE.

Năm 2014, Google công bố Android Wear, hệ điều hành dành cho các thiết bị đeo được.

2.2. Các thành phần ứng dụng của Android

2.2.1. Activity

- Activity là một màn hình, nơi người dùng có thể tương tác với ứng dụng, ví dụ: chụp hình, xem bản đồ,...
- Một ứng dụng có thể có một hoặc nhiều Activity, Activity được khởi chạy đầu tiên khi ứng dụng hoạt động được gọi là “MainActivity”.
- Activity có thể hiển thị ở chế độ toàn màn hình, hoặc ở dạng cửa sổ với một kích thước nhất định.
- Các Activity có thể gọi đến các Activity khác, Activity được gọi sẽ nhận được tương tác ở thời điểm đó.

2.2.2. Intend

- Intend là đối tượng mang thông điệp, cho phép tạo ra các yêu cầu hành động giữa các thành phần trong ứng dụng, hoặc giữa các ứng dụng với nhau.
- Intend được sử dụng nhiều trong 3 trường hợp sau:
 - + Khởi động Activity.
 - + Khởi động Service.
 - + Chuyển phát thông tin cho Broadcast Receiver.

2.2.3. View

- View được sử dụng để tạo ra các điều khiển trên màn hình cho phép nhận các tương tác từ người dùng cũng như hiển thị các thông tin cần thiết.
- View gồm 2 loại:
 - + View: các điều khiển đơn lẻ.
 - + ViewGroup: có thể chứa View hay 1 View group khác. View Group là cấp cao nhất của giao diện hiển thị ra màn hình. Có nghĩa là khi khai báo bằng Java Code muốn định chiều rộng, chiều cao 1 cái view thì khai báo thông qua View Group nếu như view không hỗ trợ các thuộc tính đó.

2.2.4. Service

- Service được sử dụng để thực thi các tác vụ cần nhiều thời gian, thực hiện ở chế độ ngầm và thường không cần giao diện hiển thị.
- Service có thể được khởi chạy và hoạt động xuyên suốt cả khi ứng dụng không hoạt động.
- Một số tác vụ cần thực hiện bằng Service:
 - + Trình diễn các tập tin đa truyền thông như nhạc, phim...
 - + Kết nối và thực hiện tải các nội dung thông qua Internet.
 - + Truy xuất đọc ghi tập tin.

2.2.5. Broadcast Receiver

- Thành phần ứng dụng cho phép truyền tải các thông báo trên phạm vi toàn hệ thống. Không có giao diện nhưng có thể thực hiện thông báo qua thanh trạng thái.
- Broadcast Receiver truyền thông báo ở hai dạng:
 - + Hệ thống: các thông báo được truyền trực tiếp từ hệ thống như: tắt màn hình, pin yếu, thay đổi kết nối...
 - + Ứng dụng: xây dựng các truyền thông báo đến các thành phần trong ứng dụng như: khởi động Service, tải nội dung đến ứng dụng...

2.2.6. Content provider

- Content Provider xây dựng cách thức truy xuất tập hợp các dữ liệu ứng dụng, dữ liệu có thể lưu trữ ở nhiều dạng như: SQLite, tập tin, tài nguyên Web hoặc bất kì thư mục lưu trữ nào.
- Có thể sử dụng Content Provider để xây dựng các ứng dụng sử dụng chung nguồn tài nguyên hoặc sử dụng riêng.
- Trong Android, một số Content Provider được xây dựng sẵn:
 - + Danh bạ.
 - + Tài nguyên đa truyền thông.
 - + Lịch.

2.2.7. Context

- Context là một lớp cơ bản chứa hầu hết thông tin về môi trường ứng dụng của android. thuộc gói android.content (android.content.Context).
- Context cung cấp các phương thức để các lớp khác có thể tương tác với hệ thống Android.
- Cho phép truy cập tới các nguồn tài nguyên (resources) đã được định nghĩa và các lớp khác.
- Hầu hết các lớp có liên quan tới UI (layout, button, textview, imageview, listview,...) đều phải “super” tới Context.
- Context giúp chúng ta dễ dàng truy cập và tương tác tới các tài nguyên của hệ thống, các thông tin, các dịch vụ (services), các thông số cấu hình, dữ liệu, danh bạ, cuộc gọi, kết nối, chế độ rung (vibrator), ...

2.2.8. Notification

- Notification được xây dựng cho mục đích gửi các thông báo đến người dùng thông qua thanh trạng thái.
- Giao diện Notification không thuộc giao diện ứng dụng, nhưng có thể tùy chỉnh giao diện Notification thông qua các phương thức có sẵn.

2.2.9. Toast

- Toast chính là một khối xây dựng cơ bản (gọi tắt là View) có chứa một thông điệp nhỏ, hiển thị nhanh chóng cho người dùng biết, nó giống như một thông báo nổi trên ứng dụng.
- Toast có các đặc điểm như sau:
 - + Có thể được tạo và hiển thị trong Activity hoặc trong Service.
 - + Không cho phép người sử dụng tương tác.
 - + Khi hiển thị sau khoảng thời gian nào đó sẽ tự đóng lại.

- Toast có 2 giá trị mặc định và ta nên sử dụng 2 giá trị này, không nên gõ con số cụ thể nào vào cả, hằng số `Toast.LENGTH_SHORT` hiển thị trong 2 giây, `Toast.LENGTH_LONG` hiển thị trong 3.5 giây.
- Cách Tạo Toast:
`Toast.makeText(MainActivity.this, "Content", Toast.LENGTH_SHORT).show();`

2.2.10. Alert Dialog

Alert Dialog là một cửa sổ nhỏ có chức năng nhắc người dùng đưa ra một quyết định hoặc nhập thông tin bổ sung. Alert Dialog không lấp kín màn hình và thường được sử dụng cho các sự kiện yêu cầu người dùng phải thực hiện một hành động trước khi có thể đi tiếp.

2.3. Các loại Layout trong ứng dụng Android

2.3.1. Linear Layout

- Linear Layout là loại layout hay được sử dụng nhất vì tính đơn giản của nó. `LinearLayout` sẽ bố trí các view theo dạng khối và không đè lên nhau.
- Linear Layout có hai chiều bố trí bố cục là:
 - + Vertical Orientation: các view bên trong sẽ được sắp xếp theo chiều dọc.
 - + Horizontal Orientation: các view bên trong sẽ được sắp xếp theo chiều ngang.

2.3.2. Relative Layout

Relative Layout là loại layout mà ở đó vị trí các widget (`Text Views`, `Buttons`,...) phụ thuộc vào vị trí các widget khác hoặc view cha.

2.3.3. Frame Layout

Là loại layout này thì các thành phần con của nó mặc định sẽ nằm ở phía trên-trái của màn hình và Widget sau sẽ đè lên Widget trước.

2.3.4. Table Layout

Cho phép chia các thành phần con thành các hàng, cột như table trong HTML.

2.3.5. Absolute Layout

Absolute Layout trong Android giúp xác định vị trí chính xác (các tọa độ x/y) của view con. Absolute Layout là ít linh động và khó duy trì hơn các loại Layout mà không có vị trí tuyệt đối khác.

2.3.6. ListView/RecyclerView

- ListView là layout để hiển thị các view con dưới dạng danh sách. Để xây dựng được danh sách thì ListView cần thêm một Adapter nữa.
- RecyclerView cũng giống với List View, nhưng được bổ sung thêm nhiều tính năng hơn. Đây là một trong những Layout rất hữu ích và rất hay được sử dụng trong các ứng dụng cần hiển thị dữ liệu có cấu trúc giống nhau như: Đọc báo, quản lý file,...

2.3.7. GridView

- GridView là một View Layout ở đó các item (hình ảnh, file,...) được đặt trong một Grid.
- Về cơ bản cách xây dựng GridView rất giống với Listview, cũng cần một Adapter để có thể đưa dữ liệu hiển thị trên GridView.

2.4. Các điều khiển (View/Widget) trong Android

2.4.1. EditText

- EditText (hoặc AppCompatActivity trong thư viện Support Design) là loại View hiện thị một hộp (chữ nhật) cho phép người dùng nhập dữ liệu (chữ, số... có thể không chế nhập liệu là text, số, phone, ngày tháng ...).
- Các thiết lập dữ liệu đầu vào:
 - + Date: nhập định dạng ngày tháng.
 - + Datetime: nhập định dạng ngày giờ.
 - + Number: chỉ cho nhập số.
 - + Text: nhập liệu bình thường.
 - + TextEmailAddress: nhập định dạng email.

- + TextMultiLine: nhập nhiều dòng.
- + TextPassword: nhập bình thường nhưng sẽ bị ẩn để che mật khẩu.

2.4.2. ImageView

- ImageView để hiển thị các hình ảnh trong lập trình Android, ImageButton nút bấm hiển thị ảnh, ảnh được thu phóng phù hợp với mục đích sử dụng, nguồn ảnh Bitmap hay Drawable.
- ImageView là loại View dùng để hiển thị tài nguyên hình ảnh như các ảnh Bitmap, các ảnh Drawable. Nó cũng cung cấp các chức năng tùy biến khác nhau như đổ màu nhuộm (tint) vào ảnh, co kéo/cắt ảnh khi hiển thị trên View.
- Các thuộc tính quan trọng của ImageView gồm:
 - + scaleType: dùng để thiết lập thu phóng ảnh, nhận các giá trị như: fitXY, center,...
 - + src: chỉ đến đường dẫn của hình ảnh, bitmap,...
 - + background: dùng để trang trí bên ngoài cho ảnh, có thể là border-radius,...

2.4.3. Button

- Button là một loại View, nó hiển thị nút bấm để chờ người dùng bấm vào. Button kế thừa từ TextView nên các thuộc tính, thiết lập cho TextView ở phần trước là có hiệu quả như đối với Button.
- Để đặt sự kiện cho button có 2 cách:
 - + Sử dụng thuộc tính android:onclick của XML.
 - + Đặt sự kiện ở file *.java tương ứng.

2.4.4. CheckBox

- CheckBox mở rộng từ CompoundButton tạo là một loại Button cho phép hiển thị hộp kiểm, hiển thị thông tin trạng thái checked hay unchecked, mã để tùy biến CheckBox.
- CompoundButton là một lớp cơ abstract mở rộng từ TextView (Button), từ lớp này nó được mở rộng để xây dựng các View là: CheckBox, RadioButton, Switch, ToggleButton.
- Các lớp mở rộng CheckBox, RadioButton, Switch, ToggleButton lưu lại các thuộc tính, ứng xử tương tự như TextView, Button nhưng có thêm hai trạng thái là checked và unchecked.
- Các thuộc tính quan trọng của CheckBox:
 - + android:checked thiết lập trạng thái checked, unchecked với giá trị "true" hay "false".

- + `android:button` để gán Drawable vào View (trạng thái cho CheckBox, RadioButton).
- + `android:buttonTint` để gán màu Tint.

2.4.5. Switch / SwitchCompat

- Trong Android Switch / SwitchCompat được sử dụng để bật tắt qua lại giữa hai trạng thái on / off, tùy chọn về màu sắc kích thước, gán ảnh vào Switch.
- Switch là loại View, mở rộng từ CompoundButton, nên nó có các thiết lập, thuộc tính, triển khai code và XML tương tự như TextView, Button, CheckBox.
- Loại View này cho phép bật tắt qua lại giữa hai trạng thái bằng cách bấm, hoặc kéo một cái công tắc gạt (thumb) trượt trên một đường ngăn (track). Có hai phiên bản Switch (`android.widget.Switch`) và SwitchCompat (`android.support.v7.widget.SwitchCompat`), để đảm bảo tính năng hiện đại tương thích trên nhiều thiết bị khi sử dụng Library Support nên sử dụng SwitchCompat.

2.4.6. RadioButton và RadioGroup

- Sử dụng RadioButton và RadioGroup để tạo ra giao diện tùy chọn một trong các gợi ý đưa ra, ví dụ sử dụng RadioButton tạo ra câu hỏi trắc nghiệm một lựa chọn.
- RadioButton cũng là loại control biểu diễn trạng thái checked/unchecked. Có điểm khác với CheckBox, Switch đó là khi người dùng nhấn vào chọn nó, nó sẽ chuyển sang checked nếu đang là unchecked, nhưng chiều ngược lại nếu nó đang là checked thì không thể bấm vào nó để chuyển sang trạng thái unchecked (Tuy nhiên có thể thiết lập bằng code). Thường RadioButton sẽ sử dụng cùng với RadioGroup, lúc đó sẽ có vài RadioButton bên trong RadioGroup và ở một thời điểm người dùng chỉ có thể chọn một.
- Để thay đổi layout RadioGroup theo hướng ngang hay đứng dùng thuộc tính `android:orientation` gán bằng Vertical (đứng) hay Horizontal (ngang)
- Màu sắc icon trạng thái của RadioButton thay đổi bằng thuộc tính `android:buttonTint`.

2.4.7. TextView

- TextView là một View cho phép hiển thị các dòng chữ (text) trên màn hình, nó có nhiều thuộc tính tùy mục đích sử dụng mà áp dụng, như thiết lập cỡ chữ, font chữ, màu chữ...
- Các thuộc tính quan trọng của TextView:

- + id: Thiết lập id cho TextView.
- + capitalize: Nếu được thiết lập, xác định rằng TextView này có một phương thức đầu vào thuần văn bản và nên tự động viết hoa nhưng gì người dùng soạn thảo.
- + gravity: Nếu được thiết lập, xác định rằng TextView này có một phương thức đầu vào thuần văn bản và nên tự động viết hoa nhưng gì người dùng soạn thảo.
- + textSize: Chính sửa kích thước chữ.
- + textColor: Chính sửa màu sắc chữ.

2.5. Giao diện và xử lý sự kiện trong Android

2.5.1. Giao diện (Activity)

- Activity là các giao diện màn hình cho ứng dụng Android.
- Các Activity trong hệ thống được quản lý như 1 ngăn xếp activity (activity stack). Khi 1 activity mới bắt đầu nó được đặt lên đầu của ngăn xếp và trở thành Running Activity (activity đang chạy), đồng thời activity trước đó sẽ nằm ngay phía dưới trong ngăn xếp đó, và sẽ không trở nên visible (nhìn thấy) cho đến khi activity ở trên thoát ra khỏi ngăn xếp.
- Activity gồm 4 trạng thái chính:
 - + Nếu activity ở phía trên của màn hình (hay ở trên cùng của ngăn xếp), thì nó đang ở trạng thái active (hoạt động) / running (đang chạy). Ví dụ khi ta cần gọi điện thì activity bấm số đó đang ở trạng thái active.
 - + Nếu activity không thể tương tác nhưng vẫn nhìn thấy (khi mà bị che bởi 1 activity khác nhưng người dùng vẫn có thể nhìn thấy nó ở phía sau) thì activity này đang ở trạng thái paused (tạm dừng). Khi ở trạng thái này Activity có thể bị xóa bỏ bởi hệ thống khi thiết bị thiếu bộ nhớ.
 - + Nếu activity hoàn toàn bị che khuất bởi activity khác thì nó đang ở trạng thái stopped (đã dừng). Activity này vẫn giữ được tất cả trạng thái và thông tin, nhưng không còn hiển thị với người dùng và thường xuyên bị xóa bỏ bởi hệ thống khi thiếu bộ nhớ. Ví dụ khi tắt màn hình thì activity vào trạng thái stopped.
 - + Nếu activity ở trạng thái paused (tạm dừng) hay stopped (đã dừng), hệ thống có thể xóa bỏ activity đó khỏi bộ nhớ bằng cách yêu cầu nó tự kết thúc hoặc xóa bỏ tiến trình

của nó. Khi activity đó hiển thị lại với người dùng thì sẽ được khởi tạo lại và khôi phục lại trạng thái trước đó.

- Vòng đời của Activity:
 - + Entire lifetime: Từ phương thức onCreate() cho tới onDestroy(): từ lúc Activity được gọi ra cho đến lúc bị hủy.
 - + Visible lifetime: Từ sau khi gọi onStart() cho tới lúc gọi onStop() : trong trường hợp này ta vẫn có thể thấy màn hình Activity.
 - + Foreground lifetime: Xảy ra từ khi gọi onResume() cho tới lúc gọi onPause(): trong suốt thời gian này Activity luôn nằm ở trên cùng và ta có thể tương tác được với nó.

2.5.2. Xử lý sự kiện trong Android

- Đặt sự kiện trong file *.java
Đầu tiên tạo một biến trong file *.java sử dụng phương thức findViewById để ánh xạ đến button trong file xml. Sau đó chèn sự kiện bằng phương thức setOnClickListener.
- Đặt sự kiện trực tiếp trong file *.xml
Thêm thuộc tính vào thẻ XML như sau: android:onClick="functionHello". Trong đó, functionHello là hàm được viết trong file *.java.

2.6. Tài nguyên Ứng dụng trong Android

2.6.1. Intent Filter

- Trong khai báo Activity trên AndroidManifest.xml có thể có thẻ <intent-filter>, thẻ này dùng chỉ rõ các loại Intent gửi tới Activity mà Activity có thể chấp nhận.
- Khi tạo ra Implicit Intent, hệ thống Android sẽ tìm kiếm các thành phần thích hợp để start bằng cách so sánh nội dung của Intent với các <intent-filter> khai báo trong file manifest của ứng dụng khác trên thiết bị. Nếu Intent phù hợp với một <intent-filter>, hệ thống khởi động thành phần đó và truyền vào đối tượng Intent.

2.6.2. AndroidManifest

- Lưu trữ thông tin tên gói ứng dụng, tồn tại duy nhất một tên gói cho mỗi ứng dụng.
- Lưu trữ thông tin tên gói ứng dụng, tồn tại duy nhất một tên gói cho mỗi ứng dụng.
- Định nghĩa tiến trình quản lý các thành phần ứng dụng.
- Định nghĩa các quyền sử dụng API và truy xuất ứng dụng khác.

- Qui định các yêu cầu khi được ứng dụng khác truy xuất.
- Khai báo cấp độ API tối thiểu xây dựng ứng dụng.
- Khai báo các thư viện có liên quan.

2.7. Đa ngôn ngữ và xử lý tập tin

2.7.1. Đa ngôn ngữ

- Đa ngôn ngữ trong android giúp cho ứng dụng android có thể hiển thị giao diện với nhiều ngôn ngữ khác nhau tùy thuộc vào việc người dùng lựa chọn ngôn ngữ hiển thị trên thiết bị.
- Mặc định android sử dụng ngôn ngữ tiếng Anh là ngôn ngữ chính và nạp chuỗi từ res \Rightarrow values \Rightarrow strings.xml. Do đó, nếu ta muốn bổ sung thêm những ngôn ngữ khác, ta cần phải tạo thư mục với tên được đặt theo quy tắc là value-mã ngôn ngữ theo từng quốc gia.

2.7.2. Xử lý tập tin

- Xử lý tập tin trong Android thường có 2 công việc là Lưu trữ và đọc dữ liệu.
- Lưu trữ dữ liệu trong android cung cấp nhiều lựa chọn trong việc lưu trữ dữ liệu như:
 - + Bộ nhớ trong (internal storage)
 - + Lưu trữ dữ liệu vào bộ nhớ ngoài (external storage)
 - + Lưu trữ dữ liệu với shared preferences
 - + Lưu trữ dữ liệu sử dụng sqlite.

2.7.3. Tập tin XML

- XML viết tắt của cụm từ “Extensible Markup Language” được dịch là “ngôn ngữ đánh dấu mở rộng”. Đây là một dạng ngôn ngữ đánh dấu có chức năng mô tả và truyền dữ liệu khác nhau.
- Dùng để mô tả dữ liệu. Các thẻ tag trong file XML chưa được xác định trước. Do đó, người dùng phải tự định nghĩa trong quá trình tạo file XML.
- Tác dụng chính của file XML là đơn giản việc chia sẻ tài nguyên dữ liệu giữa các platform và hệ thống khác nhau, đặc biệt những hệ thống được kết nối mạng. Chính vì thế, XML có tác dụng rất to lớn trong việc trao đổi, chia sẻ giữa các hệ thống.
- XMLPullParser:

- + Là 1 trong số 3 cách Parser dữ liệu trên thì Android khuyến cáo chúng ta sử dụng XMLPullParser, bởi vì nó dễ sử dụng và tốn ít bộ nhớ.
- + Cấu trúc của một file XML sẽ bao gồm những cặp thẻ và trong cặp thẻ đó có thể có các thuộc tính hoặc có thể có các giá trị nằm trong cặp thẻ <xml> </xml>

2.8. Asset và SharedPreferences

2.8.1. Asset

- Là thư mục dùng lưu trữ dữ liệu, các dữ liệu dạng chỉ đọc, không được ghi. Trước khi làm việc cần tạo sẵn dữ liệu chép vào thư mục này để chương trình sử dụng khi cần.
- Trình tự tạo thư mục Assets:
 - + Right Click vào thư mục gốc của App và chọn: New > Folder > Assets Folder.
 - + Tại mục Target Source Set > chọn Main.
 - + Chọn Finish.

2.8.2. SharedPreferences

- SharedPreferences trong Android là một class Interface cho phép đọc và lưu dữ liệu bằng các cặp key và value dưới dạng file xml trong ứng dụng Android. Ngoài ra, dữ liệu của nó cũng có thể lưu là ở dưới dạng nguyên thủy như: int, float, string, boolean, long.
- Vì ổ chứa gốc của SharedPreferences cũng chính là ứng dụng Android, nên khi ta xóa ứng dụng hoặc dữ liệu thì file này cũng hoàn toàn biến mất. Để lưu được, ta phải chuyển qua kiểu String rồi save bằng thư viện Gson.

2.9. SQLite

2.9.1. Giới Thiệu SQLite

- SQLite hỗ trợ cú pháp truy vấn giống SQL.
- SQLite hỗ trợ các kiểu dữ liệu như: TEXT (giống như kiểu chuỗi), INTEGER (giống như kiểu Long), REAL (giống như kiểu Double). Tất cả các kiểu dữ liệu khác cần được chuyển đổi thành một trong ba kiểu dữ liệu trên trước khi lưu vào Database.
- SQLite được nhúng vào các thiết bị Android. Sử dụng Sqlite Database trong Android không yêu cầu cài đặt cũng như xin quyền truy xuất.

- Khi ứng dụng tạo một Database thì đường dẫn mặc định được tạo ra là: data/data/tên_ứng_dụng/databases/filename.
- Để tạo và cập nhật một database trong ứng dụng ta cần tạo ra một class kế thừa lớp SQLiteOpenHelper.
- Trong phương thức khởi tạo của class này tại phương thức super() truyền vào tên database và Version của database.
- Trong class này cần phải override 2 phương thức để tạo và cập nhật database là onCreate và onUpgrade. Hai phương thức này sẽ nhận đối tượng SQLiteDatabase như là 1 tham số.
- Class SQLiteOpenHelper cung cấp getReadableDatabase() và getWritableDatabase() để truy cập đến đối tượng SQLiteDatabase.

2.9.2. Truy vấn trong SQLite

- Tạo bảng: Create Ten_bang (Tên_Cột Kiểu_Dữ_Liệu(Kích_thuoc)).
- Xóa một bảng: Drop Table Ten_bang.
- Thêm một dòng vào bảng: Insert Into Ten_bang(Cac_cot) Values (Gia_tri_tuong_ung).
- Sửa một dòng trong bảng: Update Ten_bang Set Tên_Cột = Giá_Trị Where Điều_kiện.
- Xóa một dòng trong bảng: Delete From Ten_bang Where Điều_kiện.
- Lấy ra các dòng trong bảng: Select Ten_cot From Ten_bang Where Điều_kiện.
- Lấy ra tất cả dòng trong bảng: Select * From Ten_bang.

2.9.3. Phương thức Query()

Cú pháp: query(String table,

String[] column,

String selection, String[] selectionArgs,

String groupBy,

String having,

String orderBy)

Trong đó:

- table: là tên bảng cần truy vấn.

- column: là các cột cần trả về dữ liệu, nếu là null thì lấy tất cả các cột.
- selection và SelectionArgs là điều kiện truy vấn.
- groupBy là các thuộc tính gom nhóm.
- having chỉ điều kiện.
- orderBy chỉ định dạng sắp xếp.

2.9.4. Các phương thức với Cursor

Tên phương thức	Chức năng
getCount()	Trả về số dòng của bảng kết quả
moveToFirst()	Di chuyển lên dòng đầu của bảng
moveToNext()	Di chuyển sang dòng tiếp theo
isAfterLast()	Kiểm tra xem đã đến cuối bảng
getString(), getInt()	Lấy thông tin theo tên cột hoặc chỉ mục

2.9.5. Các phương thức khác:

- Insert(): Sử dụng phương thức này để insert một bản ghi vào CSDL. Ví dụ. ta có một đối tượng database thuộc kiểu SQLiteDatabase. Dùng đối tượng ContentValues để đưa dữ liệu vào bảng. Đối tượng này có các phương thức put (tên cột , dữ liệu) . Sau đó gọi phương thức insert để đưa đối tượng (dòng này) vào bảng.
- Delete(): Dùng để xóa một hoặc một số record trong bảng theo một điều kiện.
- ExecSQL(): Phương thức này dùng để thực thi một câu lệnh SQL trực tiếp.

2.10. Content Provider và Thread

2.10.1. Content Provider

- Content Provider cung cấp cơ chế truy cập dữ liệu giữa các ứng dụng khác nhau trên cùng một thiết bị Android.
- Sử dụng Content Provider giống hình thức giao tiếp client/server trong đó ứng dụng truy cập dữ liệu đóng vai trò là client và Content Provider đóng vai trò là server.
- Để truy cập dữ liệu của một ứng dụng khác thì từ ứng dụng của mình chúng ta sử dụng đối tượng ContentResolver. Đối tượng này sẽ cung cấp các chức năng như insert(), update(), delete(), query().

- Các phương thức của Content Provider:

Tên phương thức	Chức năng
query()	Truy xuất dữ liệu từ ContentProvider. Sử dụng các tham số để truy vấn dữ liệu và trả về Cursor.
onCreate()	Khởi tạo ContentProvider. ContentProvider sẽ không được khởi tạo cho đến khi ContentResolver truy cập.
insert()	Phương thức này được gọi khi một hàng mới được thêm vào cơ sở dữ liệu Provider.
update()	Phương thức này được gọi khi cập nhật vào cơ sở dữ liệu Provider.
delete()	Phương thức này được gọi khi xóa các hàng trong cơ sở dữ liệu Provider.

2.10.2. Đa tiến trình trong Android

- Thread
 - + Thread là một tiến trình đơn vị xử lý của máy tính có thể thực hiện một công việc riêng biệt.
 - + Multi-Thread là khái niệm cho nhiều tiến trình chạy đồng thời, một ứng dụng Java ngoài luồng chính có thể có các luồng khác thực thi đồng thời làm ứng dụng chạy nhanh và hiệu quả hơn.
 - + Có 2 loại Thread là: Worker Thread và Main Thread.
- Ưu điểm của đa luồng:
 - + Mỗi luồng có thể dùng chung và chia sẻ tài nguyên trong quá trình chạy.
 - + Phân tách các luồng: luồng chính thì chạy trên giao diện người dùng, các luồng phụ làm nhiệm vụ gửi thông tin đến luồng chính.
- Nhược điểm của đa luồng:
 - + Càng nhiều luồng thì xử lý càng phức tạp.
 - + Khó quản lý, dễ gây ra deadlock.
- Handler:

- + Handler là một đối tượng Android cung cấp dùng để liên kết, trao đổi giữa các Thread với nhau, là trao đổi giữa Thread sinh ra Handler và các Thread khác. Thường là Main Thread (UI Thread) với các Worker Thread (Background Thread).
- + Handler có nhiệm vụ gửi và thực thi các Message hoặc Runnable tới Message Queue của Thread sinh ra nó (Handler). Handler luôn được gắn kết với một Thread (Thread sinh ra nó) cũng với Message Queue (của Thread đó). Các Message và Runnable sẽ được thực thi khi đi ra khỏi Message Queue. Có 2 nhiệm vụ mà Handler thường làm đó là:

(1) Lên lịch thực thi các Message và Runnable ở các thời điểm trong tương lai.

(2) Sắp xếp một hành động được thực hiện trong một Thread khác.

- Async Task

- + AsyncTask là phương tiện khác để xử lý công việc sử dụng background thread và giao tiếp với UI thread mà không dùng Thread hay Handler.
- + Trong AsyncTask<Params, Progress, Result> có 3 đối số Generic Type:

Params	Là giá trị được truyền vào khi gọi thực thi tiến trình và nó sẽ được truyền vào doInBackground
Progress	Là giá trị (biến) dùng để update giao diện điện lúc tiến trình thực thi, biến này sẽ được truyền vào hàm onProgressUpdate
Result	Là biến dùng để lưu trữ kết quả trả về sau khi tiến trình thực hiện xong

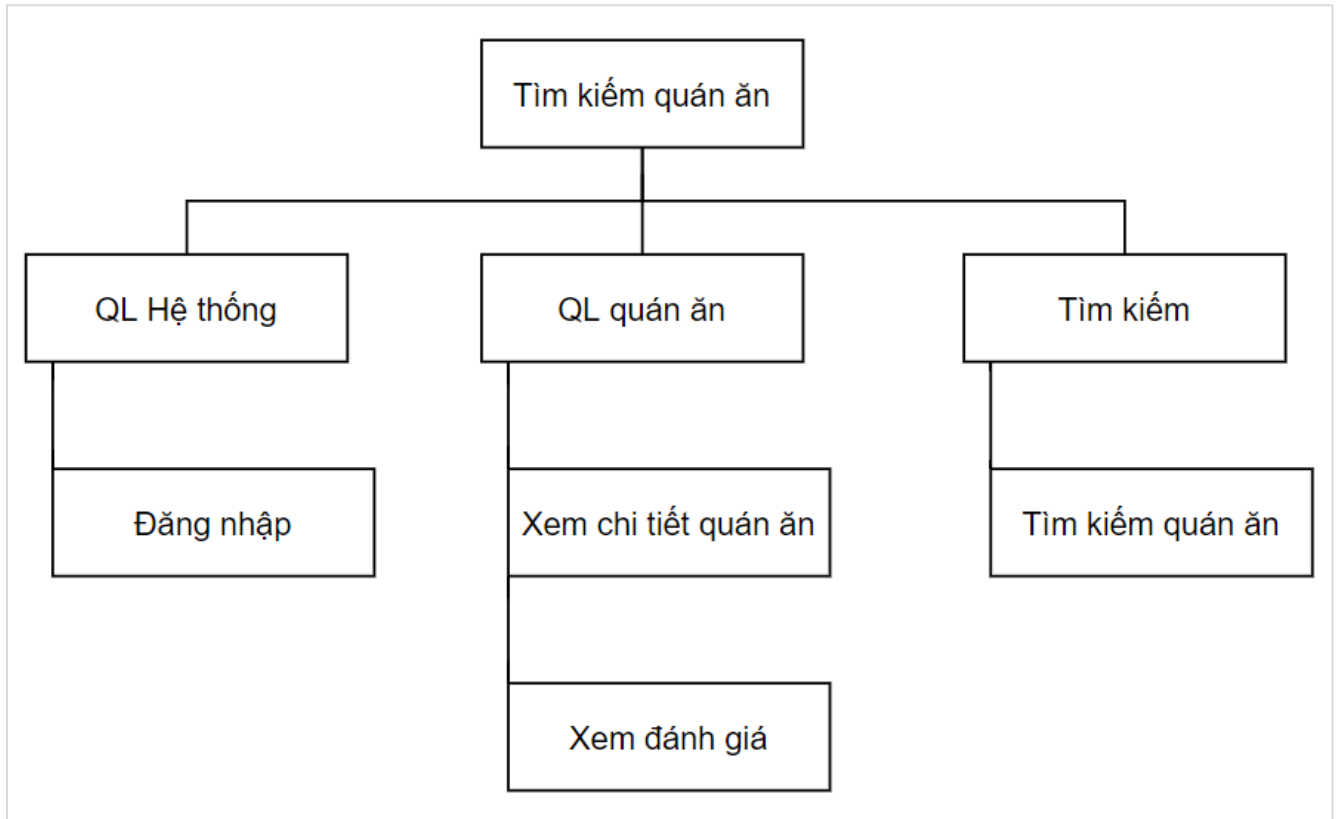
- Thông thường, 1 AsyncTask chứa 4 hàm:

onPreExecute()	Tự động được gọi đầu tiên khi tiến trình được kích hoạt.
doInBackground()	Được thực thi trong quá trình tiến trình chạy nền, thông qua hàm này để ta gọi hàm onProgressUpdate để cập nhật giao diện (gọi lệnh publishProgress). Ta không thể cập nhật giao diện trong hàm doInBackground().
onProgressUpdate()	Dùng để cập nhật giao diện lúc runtime.
onPostExecute()	Sau khi tiến trình kết thúc thì hàm này sẽ tự động xảy ra. Ta có thể lấy được kết quả trả về sau khi thực hiện tiến trình kết thúc ở đây.

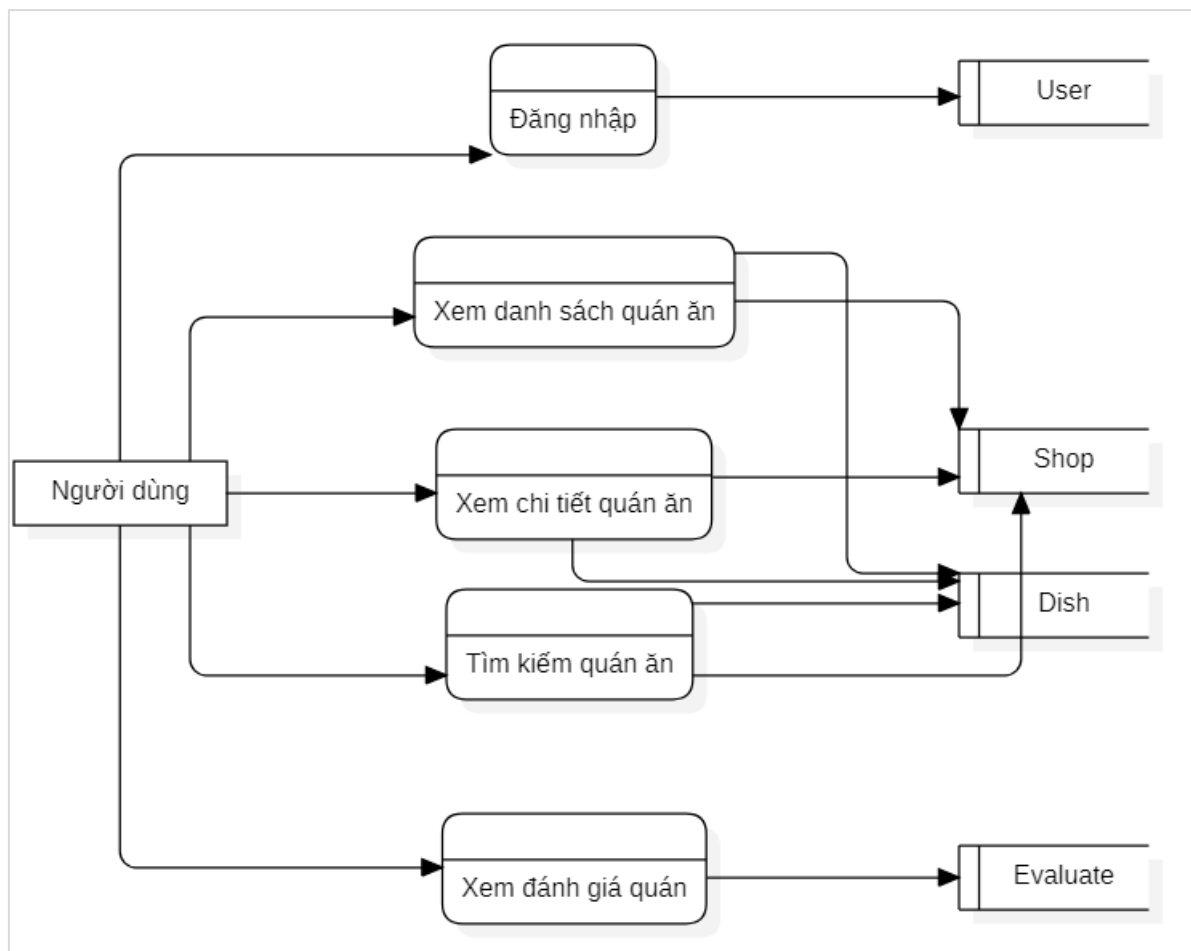
CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT ỨNG DỤNG

3.1. Thiết kế

3.1.1. Mô hình chức năng BFD, DFD



Hình 3.1 Sơ đồ BFD



Hình 3.2 Sơ đồ DFD

KẾT LUẬN

Kết quả đạt được

- Hoàn thiện được các giao diện màn hình.
- Thiết kế được cơ sở dữ liệu, thêm dữ liệu mẫu cho ứng dụng.
- Kết nối và đưa được cơ sở dữ liệu lên giao diện.

Hạn chế

- Giao diện chưa hoàn chỉnh nên không hoàn hảo như bản thiết kế.
- Chưa xây dựng được giao diện và chức năng đăng ký thành viên.
- Chưa xây dựng được phần dành cho quản trị viên để thêm, xóa, sửa quán ăn.

Hướng phát triển

- Khắc phục các hạn chế hiện tại như: giao diện, phần dành cho quản trị viên,...
- Sử dụng cơ sở dữ liệu khác như MySQL, PostgreSQL,... để tạo backend kết hợp ứng dụng di động với website như trang foody hoặc now, giúp người dùng có thể truy cập trên cả app và web.
- Đưa ứng dụng lên CH Play để thành sản phẩm chính thức.

TÀI LIỆU THAM KHẢO

- [1] Android - SQLite Database,
https://www.tutorialspoint.com/android/android_sqlite_database.htm, tham khảo ngày:
24/12/2020.
- [2] Khám phá lịch sử của hệ điều hành Android, <https://viettimes.vn/kham-pha-lich-su-cua-he-dieu-hanh-android-nguon-goc-bieu-tuong-va-ten-goi-toan-keo-banh-post83632.html>, tham khảo ngày: 28/12/2020.
- [3] ListView | Android Developers,
<https://developer.android.com/reference/android/widget/ListView>, tham khảo ngày:
22/12/2020.
- [4] SharedPreferences,
<https://developer.android.com/reference/android/content/SharedPreferences>, tham khảo ngày:
26/12/2020.
- [5] Use ConstraintLayout to design your Android views,
<https://developer.android.com/codelabs/constraint-layout#4>, tham khảo ngày: 22/12/2020.