|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**  **KHOA CÔNG NGHỆ THÔNG TIN**  Description: C:\Documents and Settings\Administrator\Desktop\logo dai hoc_khong nen.png  **ĐỒ ÁN MÔN HỌC/LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG**  **ỨNG DỤNG BÁN BALO**  **Giảng viên hướng dẫn: ths.BÙI TIẾN ĐỨC**  **Sinh viên thực hiện: TRƯƠNG HÙNG VỸ**  **MSSV: 2100002452**  **Chuyên ngành: KỸ THUẬT PHẦN MỀM**  **Môn học: LẬP TRÌNH ỨNG DỤNG**  **Khóa: 2021**  **Tp.HCM, tháng năm** |

|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**  **KHOA CÔNG NGHỆ THÔNG TIN**  **Description: C:\Documents and Settings\Administrator\Desktop\logo dai hoc_khong nen.png**  **ĐỒ ÁN MÔN HỌC/ LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG**  **ỨNG DỤNG BÁN BALO**  **Giảng viên hướng dẫn: ths.BÙI TIẾN ĐỨC**  **Sinh viên thực hiện: TRƯƠNG HÙNG VỸ**  **MSSV: 2100002452**  **Chuyên ngành: KỸ THUẬT PHẦN MỀM**  **Môn học: LẬP TRÌNH ỨNG DỤNG**  **Khóa: 2021**  **Tp.HCM, tháng năm** |

# LỜI CẢM ƠN

Lời đầu tiên xin cảm ơn thầy Bùi Tiến Đức đã truyền dạy những kiến thức liên quan để em có thể làm ra được ứng dụng đầu tiên trên chiếc máy điện thoại. Trong quá trình giảng dạy thầy luôn tận tình và giúp đỡ em rất nhiều. Giúp em có trải nghiệm thật nhiều trong quá trình làm ra ứng dụng đầu tiên chạy trên chiếc điện thoại.

Trong báo cáo lần này với các kiến thức đạt được, kĩ thuật cũng như những trải nghiệm khi làm ứng dụng, không tránh thiếu những sai sót không đáng có, nên em mong nhận được những lời góp ý và đóng góp từ thầy để hoàn thiện hơn trong những lần báo cáo sau. Em xin chân thành cảm ơn

Em xin gửi những lời chúc tốt đẹp nhất và mong thầy đạt được nhiều sức khỏe, thành công trong công việc.

Trân Trọng.

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1 NĂM HỌC 2023 - 2024** |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

BM-ChT-11

Môn thi: Lập trình trên thiết bị di động Lớp học phần:010107644605

Nhóm sinh viên thực hiện: Trương Hùng Vỹ

1. Tham gia đóng góp:

2. Tham gia đóng góp:

3. Tham gia đóng góp:

4. Tham gia đóng góp:

5. Tham gia đóng góp:

6. Tham gia đóng góp:

7. Tham gia đóng góp:

8. Tham gia đóng góp:

Ngày thi:29/01/2024 Phòng thi:

Đề tài tiểu luận/báo cáo của sinh viên: ỨNG DỤNG BÁN BALO

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo |  | |  |  |
| Nội dung |  | |  |  |
| * Các nội dung thành phần |  | |  |  |
| * Lập luận |  | |  |  |
| * Kết luận |  | |  |  |
| Trình bày |  | |  |  |
| **TỔNG ĐIỂM** |  | |  |  |
|  | | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)* | | | |

**MỤC LỤC**

[*LỜI CẢM ƠN 3*](#_Toc156174928)

[*MỤC LỤC 5*](#_Toc156174929)

[*DANH MỤC HÌNH 7*](#_Toc156174930)

[*KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT 8*](#_Toc156174931)

[*CHƯƠNG I: GIỚI THIỆU 9*](#_Toc156174932)

[*1.Tổng quan về đề tài nghiên cứu 9*](#_Toc156174933)

[*1.1 Khảo sát thực trạng 9*](#_Toc156174934)

[*1.2 Nhiệm vụ đồ án 9*](#_Toc156174935)

[*1.3 Yêu cầu hệ thống 12*](#_Toc156174936)

[*CHƯƠNG II: CÔNG VIỆC LIÊN QUAN 12*](#_Toc156174937)

[*2.Tổng quan về các ứng dụng liên quan 12*](#_Toc156174938)

[*2.1 Quy trình sản phẩm 12*](#_Toc156174939)

[*CHƯƠNG III: CƠ SỞ LÝ THUYẾT 14*](#_Toc156174940)

[*3.Lý thuyết đạt được 14*](#_Toc156174941)

[*3.1 Ngôn ngữ lập trình Java 14*](#_Toc156174942)

[*3.1.1 Java là gì ? 14*](#_Toc156174943)

[*3.1.2 lịch sử phát triển của ngôn ngữ lập trình Java 14*](#_Toc156174944)

[*3.1.3 Android Studio 15*](#_Toc156174945)

[*3.1.4 lịch sử phát triển của android studio 15*](#_Toc156174946)

[*3.1.5 Hệ điều hành Android 16*](#_Toc156174947)

[*3.1.6 Lịch sử hình thành của android 16*](#_Toc156174948)

[*3.1.7 Các phiên bản của hệ điều hành android 17*](#_Toc156174949)

[*3.2 Thiết kế giao diện người dùng (UI/UX) 20*](#_Toc156174950)

[*3.3 SQLITE 22*](#_Toc156174951)

[*CHƯƠNG IIII: HIỆN THỰC SẢN PHẨM 24*](#_Toc156174952)

[*4.Giao diện sản phẩm 24*](#_Toc156174953)

[*4.1 Chức năng từng phần 27*](#_Toc156174954)

[*4.1.1 các thành phần của Android 27*](#_Toc156174955)

[*4.1.2 Vòng đời của một ứng dụng Android 27*](#_Toc156174956)

[*4.1.3 Sử lý xử kiện 28*](#_Toc156174957)

[*4.1.4 trình nghe sử lý sự kiện 29*](#_Toc156174958)

[*4.1.5 Trình xử lý sự kiện 32*](#_Toc156174959)

[*4.1.6 Chế độ cảm ứng 33*](#_Toc156174960)

[*4.1.7 Xử lý tiêu điểm 33*](#_Toc156174961)

[*4.2 Tài nguyên ứng dụng trong android studio 34*](#_Toc156174962)

[*4.2.1 các kiểu tài nguyên 34*](#_Toc156174963)

[*4.2.2 Values 36*](#_Toc156174964)

[*4.2.3 Asset, Share Preferences 37*](#_Toc156174965)

[*4.3 Lưu trữ dữ liệu 38*](#_Toc156174966)

[*4.4 Tương tác với CSDL 39*](#_Toc156174967)

[*4.4.1 Thao Tác Cơ Bản với Sản Phẩm: 39*](#_Toc156174968)

[*4.4.2 Quản Lý Đơn Hàng: 40*](#_Toc156174969)

[*4.4.3 Quản Lý Khách Hàng: 40*](#_Toc156174970)

[*HƯỚNG PHÁT TRIỂN 40*](#_Toc156174971)

[*TÀI LIỆU THAM KHẢO 40*](#_Toc156174972)

# DANH MỤC HÌNH

[*hình 2. 1 : Hình ảnh quy trình đặt hàng 12*](#_Toc156171003)

[*Hình 3. 1 : Ngôn Ngữ lập trình Java 13*](#_Toc156171009)

[*Hình 3. 2 : Android Studio IDE 14*](#_Toc156171010)

[*Hình 3. 3 :Logo Android 15*](#_Toc156171011)

[*Hình 3. 4 : Android 1.0 17*](#_Toc156171012)

[*Hình 3. 5 : Android 1.6 Donut 18*](#_Toc156171013)

[*Hình 3. 6 : Android 2.0-2.1 Ésclair 19*](#_Toc156171014)

[*Hình 3. 7: Android 13 20*](#_Toc156171015)

[*Hình 3. 8 : Các đối tượng viewgroup tạo ra các nhánh bố cục và chứa view 21*](#_Toc156171016)

[*Hình 3. 9 : Layout Editor hiển thị Activity\_main.xml 22*](#_Toc156171017)

[*Hình 3. 10 : SQLITE 23*](#_Toc156171018)

[*Hình 4. 1: Giao diện Trang chủ 24*](#_Toc156171019)

[*Hình 4. 2 : Giao diện Chi tiết sản phẩm 25*](#_Toc156171020)

[*Hình 4. 3: Giao diện giỏ hàng 26*](#_Toc156171021)

[*Hình 4. 4: Vòng đời của Activity 27*](#_Toc156171022)

[*Hình 4. 5: Các kiểu tài nguyên trong Android 35*](#_Toc156171023)

[*Hình 4. 6: Hình ảnh tương tác với CSDL 39*](#_Toc156171024)

# KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

|  |  |
| --- | --- |
| Chữ viết tắt | Ý nghĩa |
| CSDL | Cơ sở dữ liệu |
| XML | Extensible Markup Language |
| BDF | Business function diagram |

# CHƯƠNG I: GIỚI THIỆU

## **1.Tổng quan về đề tài nghiên cứu**

Balo, túi xách, là những vật quan trọng để đựng những vật dụng quan trọng khi di chuyển đi lại. Balo và túi xách chia làm nhiều, loại balo là một loại túi đeo vai hoặc là lưng, thường có dây đeo chéo qua vai và được thiết kể để đeo ở lưng. Balo thường được sử dụng mang sách vở, laptop, hoặc các đồ cá nhân khi đi học hoặc đi làm và du lịch. Đối với túi xách thì nó lại là một loại túi nhỏ gọn có thể cầm hoặc đeo ngang qua vai, thường được sử dụng để đựng các vật dụng nhỏ như điện thoại di động, chìa khóa và một số đồ cơ bản khác. Túi xách thường được sử dụng như một phụ kiện thời trang.

## **1.1 Khảo sát thực trạng**

Theo một cuộc khảo sát từ Circana - theo dõi biên lai bán lẻ trực tuyến tại cửa hàng cho khoảng 150.000 người tiêu dùng - chỉ 39% phụ nữ từ 18 đến 35 tuổi mang túi xách cho các hoạt động khác ngoài đi làm hoặc đi học. Đây là con số khá khiêm tốn so với 60% phụ nữ trên 35 tuổi luôn mang theo túi xách khắp mọi nơi.

Trong khi đó, doanh số bán túi đeo hông, túi đeo thắt lưng và túi đeo ngực tăng 56% ở phụ nữ trẻ, riêng doanh số bán balo hàng ngày tăng 7%. Beth Goldstein, nhà phân tích giày dép và phụ kiện của Circana, cho biết những phụ nữ trẻ tuổi đang tìm kiếm sản phẩm vừa có chức năng tốt, vừa thời trang cũng như hướng đến việc mua những chiếc túi đeo chéo nhỏ gọn thay vì túi xách xa xỉ như trước đây.

Qua đó ở đất nước ta thì túi xách và balo đã xuất hiện từ rất lâu, cả hai được mô tả là các sản phẩm đựng đồ cá nhân và việc sử dụng chúng dần đang trở nên phổ biến trong văn hóa ngày nay. Do đó dần dần xuất hiện nhiều các thương hiệu lớn nhắm tới việc làm balo và túi xách chất lượng để phụ vụ nhu cầu người dùng, với các bản thiết kế đặc biệt, cập nhật xu thế thời trang trên thế giới, đáp ứng nhu cầu người tiêu dùng về giá cả và chất lượng hợp lý

Ngoài về chất liệu và kỹ thuật may theo tiêu chuẩn của người tiêu dùng thì các nhãn hàng thương hiệu lớn còn thiết kế ra những logo đặc biệt của riêng họ thu hút người tiêu dùng. Để giúp người dùng có thể tiếp cận với những mặt hàng chất lượng cao của thương hiệu mình thì các nhãn hàng đã tạo ra những trang web và ứng dụng để đưa sản phẩm của họ lên các trang web hoặc ứng dụng để tiếp cận người dùng dễ dàng mà không cần phải đi đến tận nơi để có thể xem những mặt hàng đó.

## **1.2 Nhiệm vụ đồ án**

Tạo ra ứng dụng tiếp cận với người dùng thông qua các tiêu chí như:

* Tiếp cận kỹ thuật số: Tiếp dụng tiếp thị trực tuyến để quảng bá sản phẩm của mình thông qua quảng cáo trên nhiều nền tảng như TikTok, YouTube, Facebook,…
* Mở rộng kênh bán hàng: Ngoài việc quảng bá sản phẩm trên mạng, chúng ta có thể xem xét việc mở rộng kênh bán hàng thông qua cửa hàng bán lẻ đối tác, sự kiện offline,…
* Xây dựng mối quan hệ khách hàng: Tạo ra chính sách thông tin khách hàng hiệu quả, khuyến mãi hấp dẫn và dịch vụ hậu mãi tốt để giữ chân khách hàng cũ và thu hút khách hàng mới.
* Sáng tạo trong sản phẩm: Phát triển và có kế hoạch để phát triển được sản phẩm mình sắp ra mắt.

Xây dựng thương hiệu:

* Thiết kế độc đáo: Tạo ra sản phẩm tinh tế và độc đáo, có thể nhờ vào việc sáng tạo các mẫu mã, chất liệu hoặc kiểu dáng phù hợp với đặc điểm thương hiệu riêng của mình.
* Nền tảng truyền thông: Xây dựng cốt lõi và giá trị thương hiệu thông qua chiến lược truyền thoong chặt chẽ, bao gồm truyền thông trên mạng xã hội và các chiến dịch quảng cáo.
* Kết nối cộng đồng: Tạo cộng đồng người hâm mộ trung thành thông qu aviệ ctham gia vào các sự kiện, giao lưu trên mạng xã hội.
* Chất lượng sản phẩm: Đảm bảo rằng sản phẩm của chúng ta luôn duy trì chất lượng cao và thỏa mãn nhu cầu của khách hàng, điều này tạo nên sự tin cậy đối với thương hiệu của bạn.
* Tuyên truyền giá trị cốt lõi: Tôn vinh những giá trị cốt lõi của thương hiệu, thông qua việc tuyên truyền từ việc bảo vệ môi trường đến việc phong cách sống năng động.

Tối ưu hóa rải nghiệm của khách hàng:

* Tạo sản phẩm chất lượng: Đảm bảo rằng balo và túi xách của chúng ta được làm từ chất liệu chất lượng, bền bỉ và chống thấm. Hãy chú trọng đến cả những chi tiết nhỏ nhất để đảm bảo sản phẩm hoàn thiện tốt và đáp ứng được nhu cầu và mong muốn của khách hàng.
* Thiết kế thông minh và tiện ích: Tạo ra các sản phẩm có thiết kế thông minh và tiện ích, với nhiều ngăn, túi và các tính năng tiện lợi để khách hàng có thể tổ chức và mang theo đồ dùng cá nhân một cách dễ dàng và tiện lợi.
* Phong cách và thẩm mỹ hợp lý: Đảm bảo rằng balo và túi xách của chúng ta có phong cách hiện đại và thẩm mỹ hợp lý, phù hợp với xu hướng thị trường và sở thích của khách hàng. Hãy thường xuyên cập nhật các mẫu mới và theo dõi các xu hướng thời trang để đáp ứng được sự đa dạng và đòi hỏi của người tiêu dùng.
* Cung cấp các thông tin chi tiết về sản phẩm: Đảm bảo rằng chúng ta nêu rõ thông tin chi tiết về sản phẩm, bao gồm kích thước, chất liệu, khả năng chống thấm, phân loại và cách sử dụng. Bên cạnh đó, cung cấp hình ảnh sản phẩm chất lượng cao và đặc biệt là video giới thiệu để khách hàng có thể xem trước và cảm nhận sản phẩm thật sự.
* Đảm bảo dịch vụ khách hàng tốt: Tạo ra một trải nghiệm dịch vụ khách hàng tốt bằng cách cung cấp hỗ trợ nhanh chóng và chuyên nghiệp. Đáp lại các câu hỏi và thắc mắc của khách hàng một cách nhanh nhất có thể và cung cấp dịch vụ sau bán hàng tốt, bao gồm chính sách đổi trả hoặc bảo hành một cách linh hoạt và thuận tiện cho khách hàng.
* Tạo một trải nghiệm mua hàng trực tuyến dễ dàng: Nếu chúng kinh doanh trực tuyến, đảm bảo rằng trang web của chúng ta được thiết kế tốt, dễ dùng và có tính năng tìm kiếm, xem trước và đánh giá sản phẩm đáng tin cậy. Kết nối với các cổng thanh toán an toàn và cung cấp các phương thức vận chuyển thuận tiện để khách hàng có thể mua hàng một cách dễ dàng và tiện lợi.
* Lắng nghe ý kiến đánh giá của khách hàng: Thường xuyên thu thập ý kiến đánh giá từ khách hàng về sản phẩm và dịch vụ của chúng ta. Điều này giúp chúng ta hiểu rõ hơn về những ưu điểm và kẻ khuyết của mình và điều chỉnh để cải thiện trải nghiệm của khách hàng.

Mở rộng thị trường:

* Nghiên cứu thị trường: Hiểu rõ về thị trường balo và túi xách, bao gồm xu hướng thị trường, nhu cầu và sự đa dạng của khách hàng. Điều này giúp bạn xác định mục tiêu khách hàng và tạo ra các sản phẩm phù hợp với mục tiêu đó.
* Phát triển sản phẩm đa dạng: Mở rộng danh mục sản phẩm của bạn bằng cách thêm vào các loại balo và túi xách mới. Đảm bảo rằng sản phẩm của bạn có sự đa dạng về kiểu dáng, kích thước, màu sắc và chất liệu để đáp ứng nhu cầu của nhiều đối tượng khách hàng.
* Quảng cáo và tiếp thị: Tận dụng các kênh quảng cáo và tiếp thị để giới thiệu và quảng bá thương hiệu balo và túi xách của bạn. Sử dụng các kênh trực tuyến như website, mạng xã hội, email marketing và quảng cáo trực tuyến để tiếp cận một lượng lớn khách hàng tiềm năng. Đồng thời, không nên bỏ qua cách tiếp cận truyền thống như quảng cáo trên các phương tiện truyền thông, triển lãm và sự kiện.
* Xây dựng thương hiệu mạnh mẽ: Tạo ra một thương hiệu riêng biệt và hấp dẫn bằng cách định vị và tạo ra một thông điệp độc đáo cho thương hiệu của bạn. Quan tâm đến việc thiết kế logo, bao bì sản phẩm, website và các tài liệu quảng cáo khác để tạo sự nhận biết và gắn kết với khách hàng.
* Hợp tác và đối tác: Xây dựng mối quan hệ hợp tác với các đối tác, như các cửa hàng bán lẻ, đại lý phân phối hoặc blogger/vlogger thời trang có ảnh hưởng. Điều này giúp tăng cường quy mô phân phối và tăng khả năng tiếp cận khách hàng mục tiêu.
* Chăm sóc khách hàng: Xây dựng một dịch vụ chăm sóc khách hàng tốt để duy trì và phát triển mối quan hệ với khách hàng hiện có. Đây bao gồm việc đáp ứng nhanh chóng các yêu cầu, cung cấp hỗ trợ sau bán hàng và thiết lập các chính sách đổi trả và bảo hành linh hoạt và thuận tiện.
* Mở rộng vào các kênh bán hàng mới: Không chỉ tập trung vào việc bán hàng trực tuyến, hãy xem xét mở rộng vào các kênh bán hàng khác như cửa hàng bán lẻ, sự kiện thời trang, kênh bán hàng đa cấp hoặc thậm chí hợp tác với các nhà bán hàng trực tiếp để gia tăng khả năng tiếp cận khách hàng.
* Mở rộng thương hiệu balo và túi xách đòi hỏi sự tư duy chiến lược và sự đa dạng trong cung cấp sản phẩm và dịch vụ. Xác định đúng đối tượng khách hàng và xây dựng một thương hiệu mạnh mẽ để thu hút và giữ chân khách hàng.

## **1.3 Yêu cầu hệ thống**

Đáp ứng nhu cầu của khách hàng hiện nay, chúng ta cần phải năm bắt được xu hướng của xã hội và thời trang của thế giới hiện nay, thay đổi theo từng ngày.

Phạm vi là ở các thành phó lớn như TP.HCM, Hà nội,… Hoặc các khu vực lân cận.

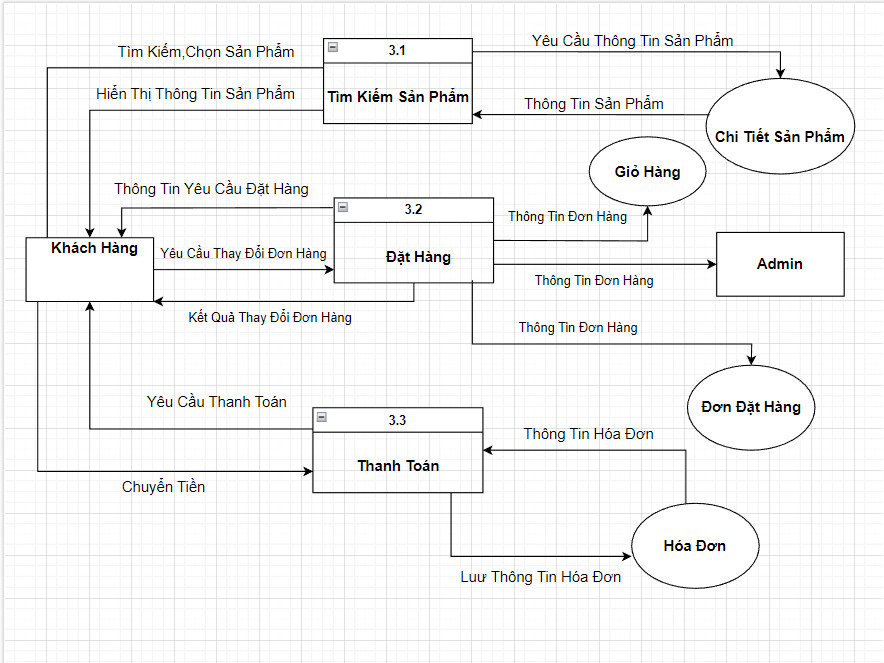
# CHƯƠNG II: CÔNG VIỆC LIÊN QUAN

## **2.Tổng quan về các ứng dụng liên quan**

Về các ứng dụng liên quan có thể kể đến như Shopee, Lazada… Đây là các ứng dụng đặt hàng nổi tiếng tại Việt Nam. Các ứng dụng này được xây dựng và thiết kế dễ dàng đến với tay người dùng có thể tìm kiếm và xem thông tin chi tiết đơn hàng cũng như thương hiệu đang bán sản phẩm đó. Dựa theo theo các ứng dụng đó thì em đã tạo một ứng dụng có khả năng tiếp cận với dùng dễ dàng xem hàng và đặt hàng cũng như xem chi tiết đơn hàng đó.

## **2.1 Quy trình sản phẩm**

Để thiết kế được một ứng dụng hoàn hảo thì chúng ta cần phải có chức năng để tạo nên thành quả tốt. Chức năng BDF có thể đáp ứng được kết quả cần đạt được:



**hình 2. 1 : Hình ảnh quy trình đặt hàng**

Mô tả chức năng:

- Tìm kiếm sản phẩm: giúp cho người mua hàng tìm được sản phẩm mà mình cần mua

- Admin: là nhà cung cấp sản phẩm

- Giỏ hàng: sau khi tìm kiếm được món hàng thích hợp, ngươi dùng sẽ thêm vào giỏ để có thể mua được hàng

- Khách hàng: người mua hàng truy cập dưới tên của mình

- Chi tiết sản phẩm: người dùng có thể xem được chất liệu cũng như giá cả của đồ dùng

- Đặt hàng: sau khi tìm kiếm được sản phẩm mà mình mong muốn người dùng sẽ đặt hàng trên ứng dụng

- Thanh toán: Người bán hàng cần lập hóa đơn cho người mua đồng thời giao hàng đến người mua

- Hóa đơn: sau khi hoàn thành thanh toán người bán cần truy xuất hóa đơn cho người mua

# CHƯƠNG III: CƠ SỞ LÝ THUYẾT

## **3.Lý thuyết đạt được**

Thông qua các buổi được học và thực hành cũng như trải nghiệm qua những buổi ở lớp thì em đã đúc kết những kiến thức cần thiết để tạo nên một ứng dụng giúp người dùng có thể tiếp cận dễ dàng. Công cụ em thực hiện cho đồ án lần này là Android studio và ngôn ngữ lập trình java.

## **3.1 Ngôn ngữ lập trình Java**



**Hình 3. 1 : Ngôn Ngữ lập trình Java**

### **3.1.1 Java là gì ?**

Ngôn ngữ lập trình Java là một ngôn ngữ lập trình hướng đối tượng, nó có thể viết mã cho các ứng dụng web và ứng dụng trên thiết bị di đông android. Ngôn ngữ này là một sự lựa chọn phổ biến của các nhà phát triển trong 2 thập niên. Hiện nay có hàng triệu ứng dụng java đang được sử dụng. Java là một ngôn ngữ đa nền tảng, hướng đối tượng, lấy mạng làm trung tâm và có thể được sử dụng như một nền tảng. Đây là một ngôn ngữ lập trình nhanh, bảo mật, đáng tin cậy, hỗ trợ các framework và các thư viện đa dạng, chỉ cần viết một lần có thể chạy bất kì nơi đâu.

### **3.1.2 lịch sử phát triển của ngôn ngữ lập trình Java**

James Gosling, Mike Sheridan và Patrick Naughton khởi xướng dự án ngôn ngữ Java vào tháng 6 năm 1991. Java ban đầu được thiết kế cho truyền hình tương tác, nhưng nó quá tiên tiến đối với ngành truyền hình cáp kỹ thuật số vào thời điểm đó. Ban đầu ngôn ngữ này được gọi là Oak theo tên một cây sồi bên ngoài văn phòng của Gosling. Sau đó, dự án có tên là Green và cuối cùng được đổi tên thành Java, từ cà phê Java, loại cà phê đến từ Indonesia. Gosling đã thiết kế Java với cú pháp kiểu C/C++ mà các lập trình viên hệ thống và ứng dụng đã quen thuộc.

### **3.1.3 Android Studio**



**Hình 3. 2 : Android Studio IDE**

Android Studio là IDE chính thức được sử dụng trong phát triển ứng dụng Android dựa trên IntelliJ IDEA. Chức năng chính của Android Studio là cung cấp các giao diện giúp người dùng có thể tạo các ứng dụng và xử lý các công cụ file phức tạp sau hậu trường. Ngôn ngữ lập trình được sử dụng trong Android Studio là Java và nó sẽ được cài đặt sẵn trên thiết bị của bạn. Khi sử dụng Android Studio thì bạn chỉ cần viết, chỉnh sửa và lưu trữ chúng trên các dự án của mình và các file nằm trong dự án đó. Đồng thời, Android Studio còn cung cấp quyền truy cập vào Android SDK. Ngoài ra, bạn có thể xem Android Studio là đuôi cho code Java cho phép nó chạy trơn tru trên các thiết bị Android rồi tận dụng được lợi thế của phần cứng gốc. Bạn chỉ cần sử dụng ngôn ngữ lập trình Java để có thể viết chương trình, khi đó Android SDK sẽ có nhiệm vụ kết nối các phần này lại với nhau. Khi đó, Android Studio sẽ kích hoạt để thực hiện chạy code và thông qua trình giả lập hoặc dựa vào bất kỳ phần cứng giúp kết nối với thiết bị. Sau đó, bạn có thể gỡ rối cho chương trình của mình ngay khi nó chạy và nhận phản hồi giúp giải thích các sự cố,...Cho đến hiện nay, Google đã và đang rất nỗ lực để giúp cho Android Studio sẽ trở nên mạnh mẽ và hữu ích hơn. Khi bạn gõ code, nó sẽ giúp bạn cung cấp danh sách gợi ý hoàn thành để giúp người dùng có thể hoàn thiện được dòng code đó. Đây là một trong những chức năng rất hữu ích đề phòng cho trường hợp người dùng không nhớ chính xác cú pháp giúp bạn tiết kiệm thời gian hiệu quả hơn.

### **3.1.4 lịch sử phát triển của android studio**

Android Studio đã được công bố vào năm 2013 tại hội nghị Google I/O và được phát hành vào năm 2014 sau nhiều phiên bản khác nhau. Trước đó, thì các nhà phát triển của Android thường sử dụng các công cụ như Eclipse IDE hoặc một IDE Java chung để hỗ trợ cho nhiều ngôn ngữ lập trình khác. Android Studio giúp cho việc tạo ứng dụng dễ dàng hơn so với các phần mềm chuyên dụng. Với người mới, sẽ có rất nhiều thứ phải học và nhiều thông tin có sẵn. Thậm chí, chúng còn thông qua nhiều kênh chính thức hoặc có thể có lỗi khiến người dùng hoang mang.

### **3.1.5 Hệ điều hành Android**



**Hình 3. 3 :Logo Android**

Android là hệ điều hành trên điện thoại di động (và hiện nay là cả trên một số đầu phát HD, HD Player, TV) phát triển bởi Google và dựa trên nền tảng Linux. Lập trình android là một lập trình ứng dụng di động phổ biến. Trước đây, Android được phát triển bởi công ty liên hợp Android (sau đó được Google mua lại vào năm 2005).Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào ngày 5 tháng 11 năm 2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng, phần mềm và viễn thông nhằm mục đính tạo nên một chuẩn mở cho điện thoại di động trong tương lai.

### **3.1.6 Lịch sử hình thành của android**

- Lịch sử Android bắt đầu vào tháng 10/2003 - trước khi thuật ngữ điện thoại thông minh trở nên phổ biến, và vài năm trước khi Apple công bố iPhone cùng hệ điều hành iOS đầu tiên. Android Inc được thành lập tại Palo Alto, California. Rich Miner, Nick Sears, Chris White và Andy Rubin là 4 người sáng lập của HĐH này. Vào thời điểm đó, Rubin đề cập đến việc Android Inc sẽ phát triển “các thiết bị di động thông minh hơn có khả năng nhận biết rõ hơn về vị trí và sở thích của chủ sở hữu”.

- Rubin đã tiết lộ trong một bài phát biểu năm 2013 tại Tokyo rằng, Android ban đầu được dùng để cải thiện hệ điều hành của máy ảnh kỹ thuật số. Ngay cả khi đó, thị trường máy ảnh kỹ thuật số độc lập bị suy giảm. Vài tháng sau, Android Inc quyết định chuyển hướng sang sử dụng hệ điều hành bên trong điện thoại di động.

- Năm 2005, chương quan trọng tiếp theo trong lịch sử Android bắt đầu khi công ty ban đầu được Google mua lại. Rubin và các thành viên sáng lập khác tiếp tục phát triển hệ điều hành dưới quyền chủ sở hữu mới của họ. Sau đó, họ quyết định sử dụng Linux làm nền tảng cho hệ điều hành Android. Điều đó làm cho nó có thể cung cấp hệ điều hành miễn phí cho các nhà sản xuất di động bên thứ ba. Google và nhóm Android cho rằng công ty có thể thu lợi từ việc cung cấp các dịch vụ khác, bao gồm cả ứng dụng.

Rubin ở lại Google với tư cách là người đứng đầu nhóm Android cho đến năm 2013. Cuối năm 2014, Rubin rời Google hoàn toàn và thành lập một “vườn ươm” doanh nghiệp khởi nghiệp trước khi quay lại kinh doanh điện thoại thông minh với chiếc Essential xấu số vào năm 2017.

- Biểu tượng Android: Irina Blok đã tạo ra logo quen thuộc cho hệ điều hành Android trong khi làm việc cho Google. Nó trông giống như sự kết hợp giữa một con robot và một con bọ màu xanh lá cây. Blok cho biết yêu cầu duy nhất mà nhóm thiết kế của Google đưa ra cho cô ấy là làm cho logo trông giống như một con robot. Blok cũng nói rằng một trong những nguồn cảm hứng của cô ấy cho thiết kế cuối cùng cho biểu tượng Android là các logo nhà vệ sinh quen thuộc đại diện cho “Nam giới” và “Phụ nữ”.

Một điều mà Blok và Google cùng quyết định là biến bản thân robot Android trở thành một dự án mã nguồn mở. Gần như mọi công ty lớn khác sẽ bảo vệ một logo hoặc “linh vật” khỏi những kẻ “đạo nhái”. Tuy nhiên, rất nhiều người đã sửa đổi biểu tượng của Android, vì Google cho phép những thay đổi đó theo giấy phép Ghi công Creative Commons 3.0.

### **3.1.7 Các phiên bản của hệ điều hành android**

**Android 1.0 :** Sự khởi đầu của lịch sử Android



**Hình 3. 4 : Android 1.0**

Năm 2007, Apple ra mắt chiếc iPhone đầu tiên và mở ra một kỷ nguyên mới trong lĩnh vực điện toán di động. Vào thời điểm đó, Google vẫn đang bí mật làm việc trên Android, và vào tháng 11 năm đó, công ty bắt đầu tiết lộ kế hoạch cạnh tranh với Apple cũng như các nền tảng di động khác. Trong một bước ngoặt lớn, Google đã dẫn đầu việc thành lập liên minh Open Handset Alliance. Nó bao gồm các nhà sản xuất điện thoại như HTC và Motorola, các nhà sản xuất chip như Qualcomm và Texas Instruments, và các nhà cung cấp dịch vụ bao gồm nhà mạng T-Mobile.

Android 1.5 Cupcake:

Tên mã công khai chính thức đầu tiên cho Android đã không xuất hiện cho đến khi Google phát hành phiên bản 1.5 Cupcake vào tháng 4/2009. Việc đặt tên các phiên bản Android theo tên món tráng miệng của Google là ý tưởng đến từ giám đốc dự án Ryan Gibson. Tuy nhiên, vẫn chưa rõ vì sao người này lại sử dụng quy ước đặt tên như vậy.

**Android 1.6 Donut:**



**Hình 3. 5 : Android 1.6 Donut**

Google nhanh chóng tung ra Android 1.6 Donut vào tháng 9/2009. Hệ điều hành mới này mang tới hỗ trợ cho các nhà cung cấp dịch vụ sử dụng mạng dựa trên CDMA. Điều này cho phép tất cả các nhà mạng trên thế giới bán điện thoại Android. Các tính năng khác đáng chú ý khác của Android 1.6 Donut bao gồm sự ra đời của Quick Search Box và chuyển đổi nhanh giữa Camera, Camcorder và Gallery để hợp lý hóa trải nghiệm chụp ảnh đa phương tiện. Android 1.6 Donut Donut cũng giới thiệu widget Power Control để quản lý Wi-Fi, Bluetooth, GPS,...

Một trong những chiếc điện thoại đầu tiên được cài đặt sẵn hệ điều hành Android 1.6 Donut khi bán ra là chiếc Dell Streak xấu số. Nó có màn hình 5 inch khổng lồ (vào thời điểm đó) và được mô tả là “điện thoại thông minh/máy tính bảng”. Ngày nay, màn hình 5 inch được coi là tương đối nhỏ đối với một chiếc smartphone.

Android 2.0-2.1 Éclair



**Hình 3. 6 : Android 2.0-2.1 Ésclair**

Vào tháng 10/2009 - tức khoảng một năm sau khi Android 1.0 xuất hiện - Google đã phát hành phiên bản 2.0 của hệ điều hành này, với tên mã chính thức là Eclair. Android 2.0 mang đến khá nhiều tính năng mới cho người dùng và là phiên bản đầu tiên bổ sung hỗ trợ chuyển văn bản thành giọng nói, đồng thời cũng giới thiệu hình nền động, hỗ trợ nhiều tài khoản và điều hướng Google Maps,...

Ngoài ra còn rất nhiều phiên bản khác và hiện nay mới nhất là Android 13



**Hình 3. 7: Android 13**

Theo bài viết này, phiên bản tiếp theo của Android đang được phát hành. Google đã ra mắt Preview dành cho nhà phát triển đầu tiên của Android 13 vào tháng 2 năm 2022 và bản beta công khai đầu tiên dành cho thiết bị Pixel ra mắt vào tháng 4 năm 2022. Chúng tôi có thể mong đợi thấy nhiều bản phát hành beta hơn trước khi phiên bản cuối cùng của hệ điều hành ra mắt, có thể xảy ra sớm nhất là vào tháng 8 năm 2022. Nhân tiện, tên mã nội bộ của Android 13 là “Tiramisu”.

Một số tính năng mới trong bản dựng dành cho nhà phát triển và bản beta ban đầu bao gồm cách tạo chủ đề cho các biểu tượng của điện thoại, hỗ trợ Bluetooth LE Audio và tiện ích Đang phát được cập nhật. Hệ điều hành cũng sẽ bao gồm một chế độ im lặng được cải tiến để loại bỏ mọi thứ. Điều này bao gồm rung và xúc giác nếu ai đó gọi. Bạn cũng có thể tắt thông báo ứng dụng ngay khi cài đặt ứng dụng mới. Tính năng Ghép nối nhanh sẽ tự động nhận biết nếu bạn có thiết bị ở gần và sẽ hỏi bạn có muốn liên kết với thiết bị đó hay không.

## **3.2 Thiết kế giao diện người dùng (UI/UX)**

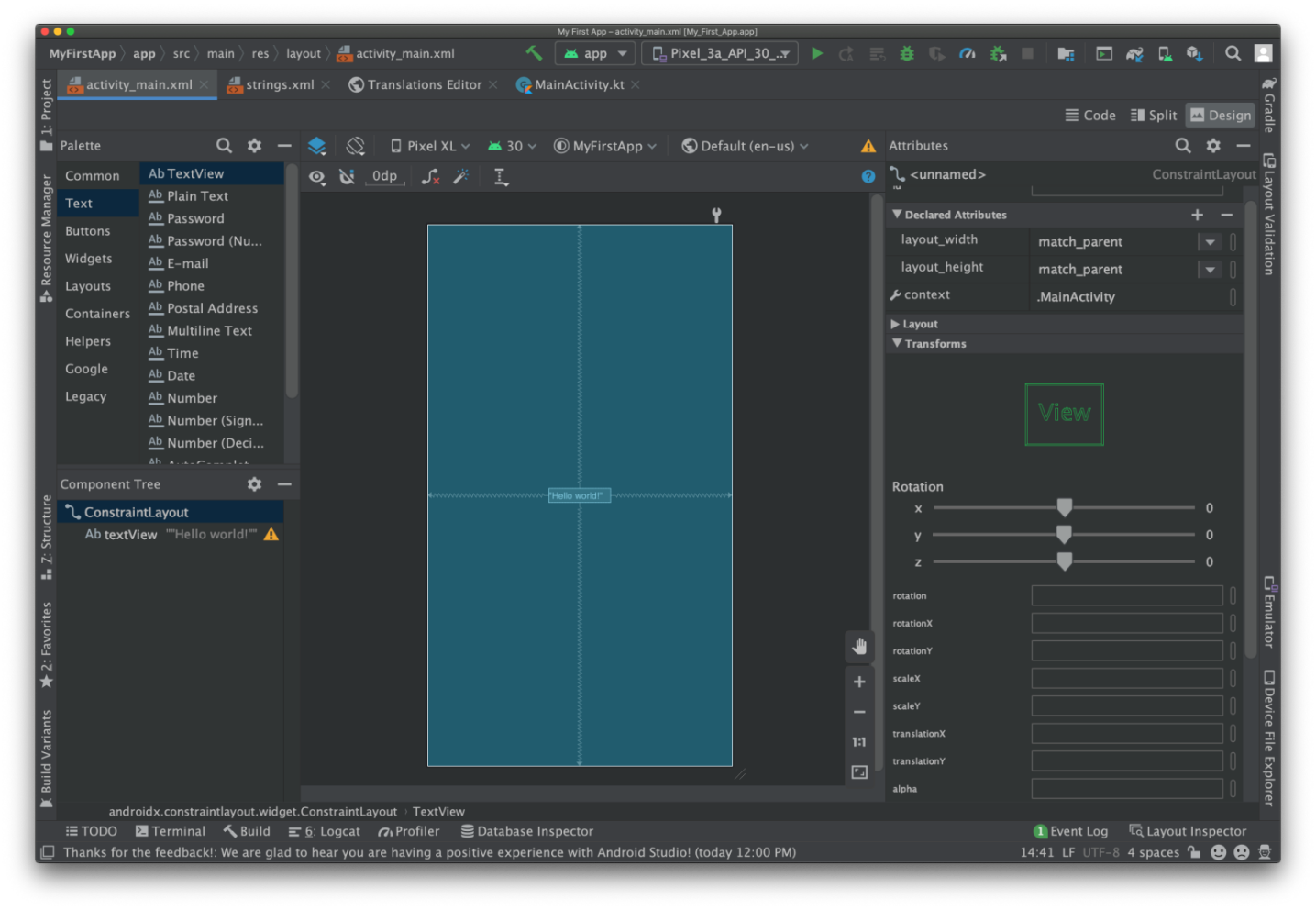


**Hình 3. 8 : Các đối tượng viewgroup tạo ra các nhánh bố cục và chứa view**

Giao diện người dùng (UI) cho ứng dụng Android được xây dựng dưới dạng phân cấp bố cục và tiện ích. Bố cục là các đối tượng ViewGroup, vùng chứa kiểm soát cách định vị các chế độ xem con trên màn hình. Tiện ích là các đối tượng View, các thành phần giao diện người dùng như nút và hộp văn bản.

Android cung cấp từ vựng XML (Ngôn ngữ đánh dấu mở rộng) cho các lớp ViewGroup và View, vì vậy hầu hết giao diện người dùng của bạn được xác định trong các tệp XML. Tuy nhiên, thay vì dạy bạn ghi XML, bài học này sẽ hướng dẫn bạn cách tạo bố cục bằng Layout Editor của Android Studio. Layout Editor ghi XML cho bạn khi bạn kéo và thả các chế độ xem để tạo bố cục.

1. Để bắt đầu, hãy thiết lập không gian làm việc của bạn như sau:
2. Trong cửa sổ Project (Dự án), hãy mở app > res > layout > activity\_main.xml.
3. Để có chỗ cho Layout Editor, hãy ẩn cửa sổ Project (Dự án). Cách làm: Chọn View > Tool Windows > Project (Xem > Cửa sổ công cụ > Dự án) hoặc chỉ cần nhấp vào Project (Dự án) ở bên trái màn hình Android Studio.
4. Nếu trình chỉnh sửa của bạn hiển thị nguồn XML, hãy nhấp vào thẻ Design (Thiết kế) ở trên cùng bên phải của cửa sổ.
5. Nhấp vào (Select Design Surface (Chọn giao diện thiết kế)) rồi chọn Blueprint (Bản vẽ).
6. Nhấp vào (View Options (Các tuỳ chọn chế độ xem)) trong thanh công cụ Layout Editor và đảm bảo rằng bạn đã đánh dấu Show All Constraints (Hiển thị tất cả điều kiện ràng buộc).
7. Đảm bảo bạn đã tắt tính năng Tự động kết nối. Chú giải công cụ trên thanh công cụ sẽ hiển thị (Enable Autoconnection to Parent (Bật tính năng Tự động kết nối với mẹ)) khi tính năng Tự động kết nối đang tắt.
8. Nhấp vào (Default Margins (Lề mặc định)) trên thanh công cụ và chọn 16. Nếu cần, bạn có thể điều chỉnh lề cho mỗi chế độ xem sau này.
9. Nhấp vào (Device for Preview (Thiết bị để xem trước)) trên thanh công cụ và chọn 5.5, 1440 × 2560, 560 dpi (Pixel XL).



**Hình 3. 9 : Layout Editor hiển thị Activity\_main.xml**

## **3.3 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 rang



**Hình 3. 10 : SQLITE**

- Các bước để làm việc với SQLite Database

+ Bạn cần tạo một lớp tiện ích để làm việc với cơ sở dữ liệu SQLite, lớp này nên mở rộng từ lớp SQLiteOpenHelper. Có 2 phương thức quan trọng mà bạn cần phải ghi đè (override) nó là onCreate() và onUpgrade().

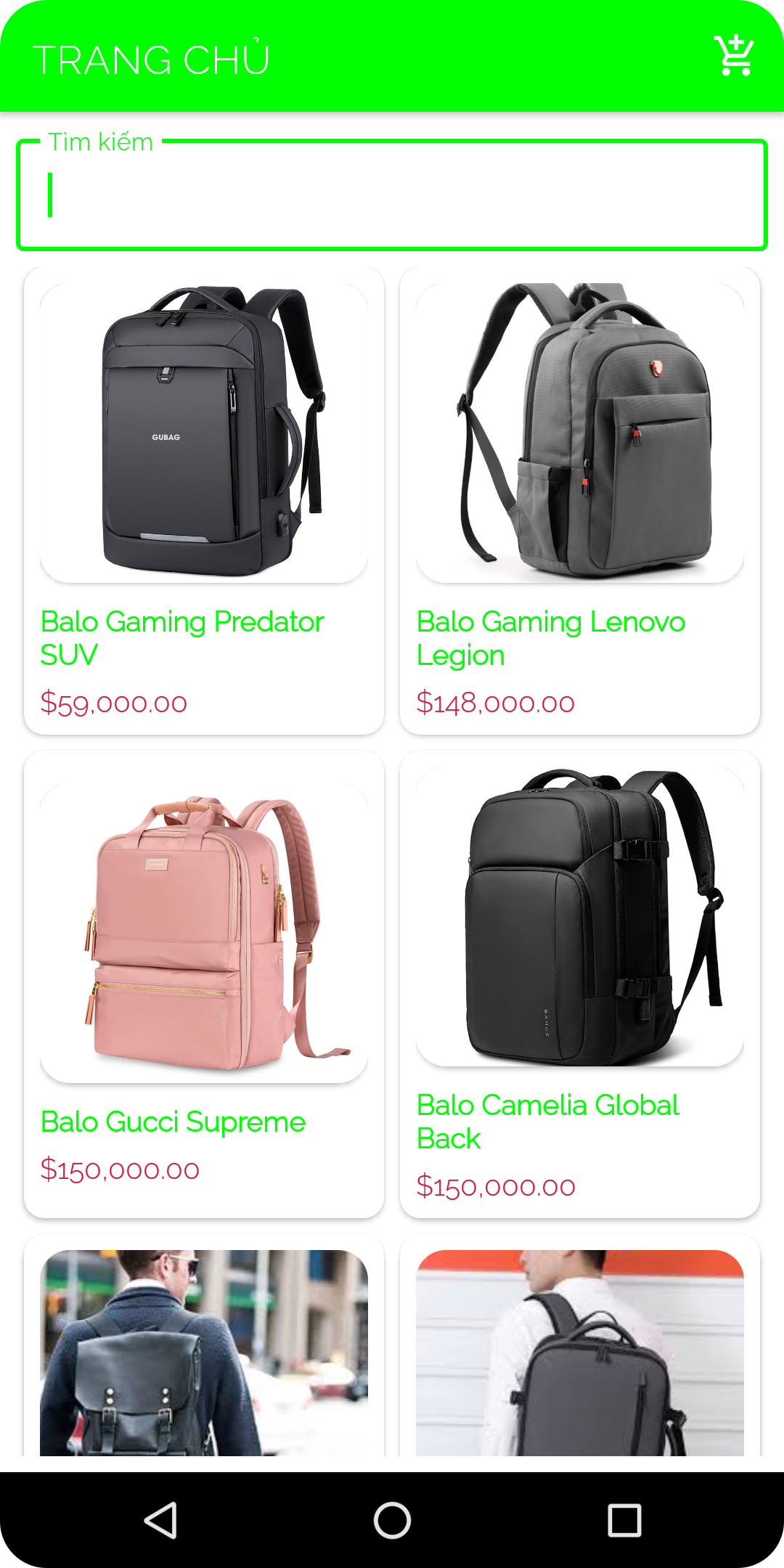
+ Tạo class MyDatabaseHelper mở rộng từ SQLiteOpenHelper.

+ Sau khi mở rộng class của bạn từ SQLiteOpenHelper bạn cần phải ghi đè lên hai phương thức onCreate() và onUpgrage()

* onCreate() - Những là nơi mà chúng ta cần phải viết để tạo bảng. Nó được gọi (called) khi cơ sở dữ liệu được tạo ra.
* onUpgrade() - Phương thức này được gọi khi cơ sở dữ liệu được nâng cấp như thay đổi cấu trúc bảng, thêm giàng buộc cho cơ sở dữ liệu, v..v.

# CHƯƠNG IIII: HIỆN THỰC SẢN PHẨM

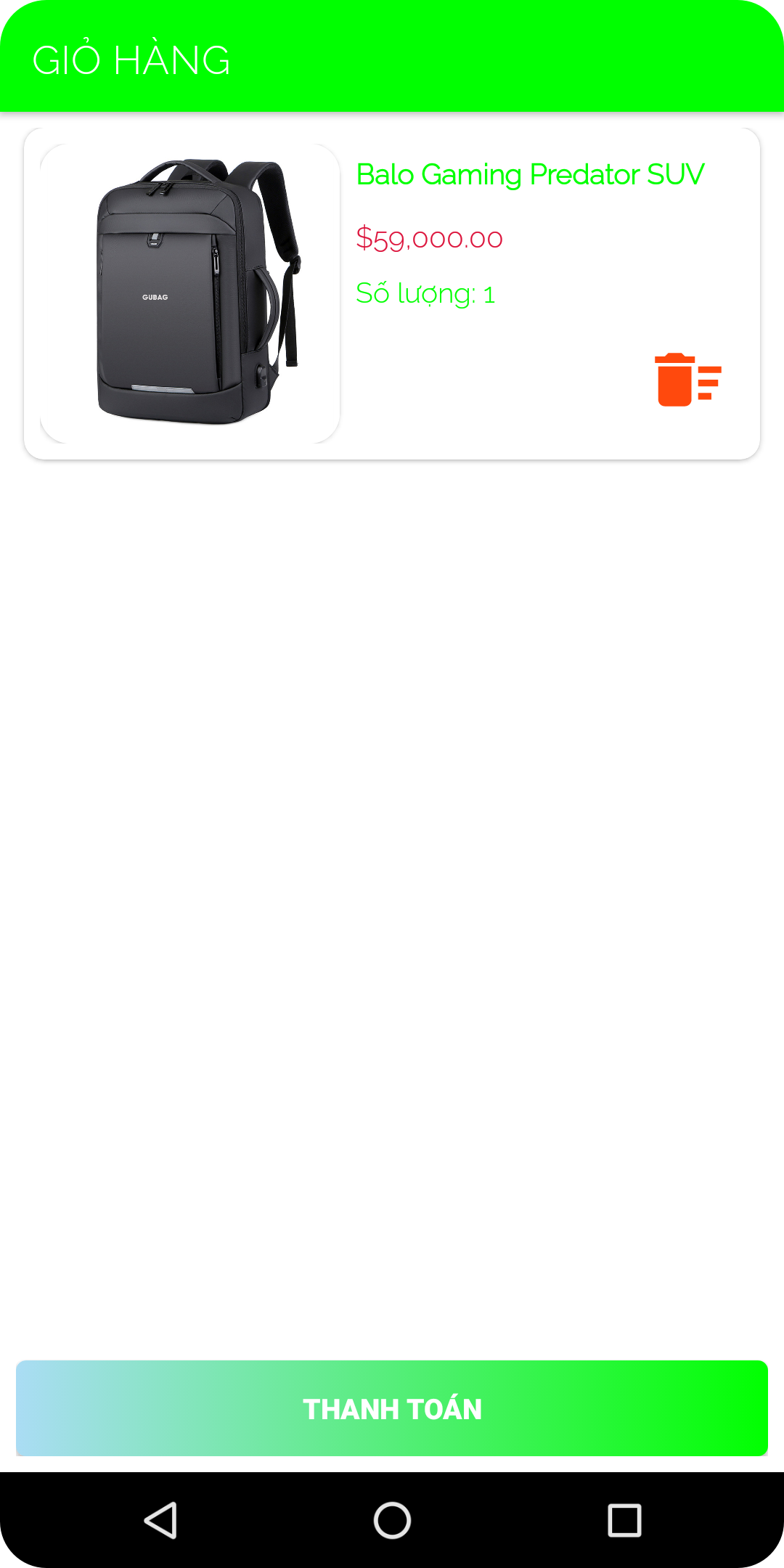
## **4.Giao diện sản phẩm**



**Hình 4. 1: Giao diện Trang chủ**



**Hình 4. 2 : Giao diện Chi tiết sản phẩm**



**Hình 4. 3: Giao diện giỏ hàng**

## **4.1 Chức năng từng phần**

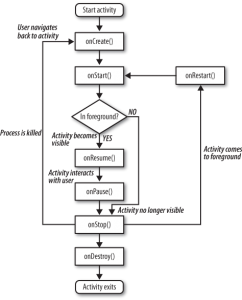
### **4.1.1 các thành phần của Android**

Activity: hiểu một cách đơn giản thì Activity là nền của 1 ứng dụng. Khi khởi động 1 ứng dụng Android nào đó thì bao giờ cũng có 1 main Activity được gọi, hiển thị màn hình giao diện của ứng dụng cho phép người dùng tương tác  
Service: thành phần chạy ẩn trong Android từ lúc chạy cho tới khi thiết bị cầm tay tắt đi. Service sử dụng để update dữ liệu, đưa ra các cảnh báo (Notification) và không bao giờ hiển thị cho người dùng thấy.  
Content Provider: được tạo ra để quản lý và chia sẻ dữ liệu với các hoạt động, dịch vụ khác.  
Intent: nền tảng để truyền tải các thông báo. Intent được sử dụng để gửi các thông báo đi nhằm khởi tạo 1 Activity hay Service để thực hiện công việc bạn mong muốn. VD: khi mở 1 trang web, bạn gửi 1 intent đi để tạo 1 activity mới hiển thị trang web đó.

Broadcast Receiver: thành phần thu nhận các Intent bên ngoài gửi tới. VD: bạn viết 1 chương trình thay thế cho phần gọi điện mặc định của Android, khi đó bạn cần 1 BR để nhận biết các Intent là các cuộc gọi tới.  
Notification: đưa ra các cảnh báo mà không làm cho các Activity phải ngừng hoạt động.

4.1.2 Vòng đời của một ứng dụng Android

- Android có cơ chế quản lý các process theo chế độ ưu tiên. Các process có priority  
thấp sẽ bị Android giải phóng mà không hề cảnh báo nhằm đảm bảo tài nguyên.  
+ Foreground process: là process của ứng dụng hiện thời đang được người dùng  
tương tác.  
+ Visible process: là process của ứng dụng mà activity đang hiển thị đối với người  
dùng (onPaused() của activity được gọi).  
+ Service process: là Service đang running.  
+ Background process: là process của ứng dụng mà các activity của nó ko hiển thị  
với người dùng (onStoped() của activity được gọi).  
+ Empty process: process không có bất cứ 1 thành phần nào active.  
Theo chế độ ưu tiên thì khi cần tài nguyên, Android sẽ tự động kill process, trước tiên là các empty process.



**Hình 4. 4: Vòng đời của Activity**

Activity bao gồm 4 state:  
active (running): Activity đang hiển thị trên màn hình (foreground).  
paused: Activity vẫn hiển thị (visible) nhưng không thể tương tác (lost focus).  
VD: một activity mới xuất hiện hiển thị giao diện đè lên trên activity cũ, nhưng  
giao diện này nhỏ hơn giao diện của activity cũ, do đó ta vẫn thấy được 1 phần  
giao diện của activity cũ nhưng lại không thể tương tác với nó.  
stop: Activity bị thay thế hoàn toàn bởi Activity mới sẽ tiến đến trạng thái stop  
killed: Khi hệ thống bị thiếu bộ nhớ, nó sẽ giải phóng các tiến trình theo nguyên  
tắc ưu tiên. Các Activity ở trạng thái stop hoặc paused cũng có thể bị giải phóng và  
khi nó được hiển thị lại thì các Activity này phải khởi động lại hoàn toàn và phục  
hồi lại trạng thái trước đó.

### **4.1.3 Sử lý xử kiện**

**-** Trên Android, có nhiều cách để chặn các sự kiện từ tương tác của người dùng với ứng dụng. Để xem xét các sự kiện trong phạm vi giao diện người dùng, nên chụp các sự kiện từ đối tượng chế độ hiển thị cụ thể mà người dùng tương tác. Lớp chế độ hiển thị sẽ cung cấp phương tiện để thực hiện điều này.

- Trong số các lớp chế độ hiển thị được dùng để thiết kế bố cục, bạn sẽ thấy một số phương thức gọi lại công khai hữu ích cho các sự kiện giao diện người dùng. Các phương thức này được khung Android gọi lại khi có hành động tương ứng xảy ra trên đối tượng đó. Chẳng hạn, khi chạm vào một chế độ hiển thị (chẳng hạn như Nút), phương thức onTouchEvent() sẽ được gọi trên đối tượng đó. Tuy nhiên, để ngăn chặn điều này, phải mở rộng lớp và ghi đè phương thức. Tuy nhiên, việc mở rộng mọi đối tượng chế độ hiển thị để điều khiển những sự kiện như vậy sẽ không thực tế. Vì vậy, lớp chế độ hiển thị cũng chứa một tập hợp các giao diện lồng nhau cùng với các lệnh gọi lại rất dễ xác định. Các giao diện này, có tên là [trình nghe sự kiện](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#EventListeners), giúp thu hút người dùng tương tác với giao diện người dùng.

- Mặc dù trình nghe sự kiện thường được dùng để theo dõi tương tác của người dùng, sẽ có những lúc cần mở rộng lớp chế độ hiển thị để tạo thành phần tùy chỉnh. Có trường hợp cần mở rộng lớp [Button](https://developer.android.com/reference/android/widget/Button?hl=vi) để thêm hiệu ứng đẹp hơn. Trong trường hợp này, có thể xác định các hành vi sự kiện mặc định cho lớp bằng cách sử dụng [trình xử lý sự kiện](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#EventHandlers) của lớp.

**4.1.4 trình nghe sử lý sự kiện**

- Trình nghe sự kiện là một giao diện trong lớp [View](https://developer.android.com/reference/android/view/View?hl=vi)chứa duy nhất một phương thức gọi lại. Phương thức này sẽ được khung Android bật lên khi chế độ hiển thị đã được trình nghe đăng ký được kích hoạt khi người dùng tương tác với mục trong giao diện người dùng.

- Trong giao diện của trình nghe sự kiện có các phương thức gọi lại sau đây:

+ onClick() **–** [View.OnClickListener](https://developer.android.com/reference/android/view/View.OnClickListener?hl=vi): Lệnh này được gọi khi người dùng nhấn vào mục (ở chế độ cảm ứng), hoặc đặt tiêu điểm vào mục bằng các phím điều hướng hoặc bi xoay rồi nhấn phím "enter" hoặc bi xoay.

Sử dụng thuộc tính onClick trong ViewLayout

- Từ API level 4 (Android 1.6) trở đi, tên của một hàm có thể được gán vào thuộc tính android:onClick trong layout.

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button 1"

android:id="@+id/button1"

android:layout\_below="@+id/textView"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:onClick="handleClick" />

- Sau đó thì hàm đó được viết trong Activity như sau:

class MainActivity:AppCompatActivity() {

fun onCreate(savedInstanceState:Bundle) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

}

fun handleClick(arg0:View) {

val btn = arg0 as Button

val tv = findViewById(R.id.textView) as TextView

tv.setText("You pressed " + btn.getText())

}

}

+ onLongClick() **–** [View.OnLongClickListener](https://developer.android.com/reference/android/view/View.OnLongClickListener?hl=vi): Lệnh này được gọi khi người dùng chạm và giữ mục (ở chế độ cảm ứng), hoặc đặt tiêu điểm vào mục bằng các phím điều hướng hoặc bi xoay rồi nhấn và giữ phím "Enter" hoặc bi xoay (trong một giây).

+ onFocusChange() **–** [View.OnFocusChangeListener](https://developer.android.com/reference/android/view/View.OnFocusChangeListener?hl=vi): Lệnh này được gọi khi người dùng di chuyển lên hoặc ra khỏi mục bằng các phím điều hướng hoặc bi xoay.

+ onKey() **–** [View.OnKeyListener](https://developer.android.com/reference/android/view/View.OnKeyListener?hl=vi): Lệnh này được gọi khi người dùng đặt tiêu điểm vào mục và nhấn hoặc nhả phím cứng trên thiết bị.

+ onTouch() **–** [View.OnTouchListener](https://developer.android.com/reference/android/view/View.OnTouchListener?hl=vi): Lệnh này được gọi khi người dùng thực hiện một thao tác chạm, bao gồm nhấn, nhả hay bất kỳ thao tác chuyển động nào trên màn hình (trong phạm vi của mục đó).

+ onCreateContextMenu() **–** [View.OnCreateContextMenuListener](https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener?hl=vi) Lệnh này được gọi khi một Trình đơn ngữ cảnh đang được tạo (do một lượt "nhấp và giữ" liên tục). Xem giới thiệu về trình đơn ngữ cảnh trong hướng dẫn dành cho lập trình viên [Trình đơn](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#context-menu).

- Các phương thức này là những thành tố duy nhất trong giao diện tương ứng. Để xác định một phương thức và điều khiển các sự kiện, hãy triển khai giao diện lồng nhau trong phần Hoạt động hoặc định nghĩa giao diện đó là một lớp ẩn danh. Sau đó, hãy chuyển một phiên bản triển khai sang phương thức View.set...Listener()tương ứng. (Ví dụ: gọi [setOnClickListener()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#setOnClickListener(android.view.View.OnClickListener))và chuyển phần triển khai [OnClickListener](https://developer.android.com/reference/android/view/View.OnClickListener?hl=vi) cho nó.)

Minh họa:

// Create an anonymous implementation of OnClickListener  
private OnClickListener corkyListener = new OnClickListener() {  
 public void onClick(View v) {  
 // do something when the button is clicked  
 }  
};  
  
protected void onCreate(Bundle savedValues) {  
 ...  
 // Capture our button from layout  
 Button button = (Button)findViewById(R.id.corky);  
 // Register the onClick listener with the implementation above  
 button.setOnClickListener(corkyListener);  
 ...  
}

Việc triển khai trình nghe theo lượt nhấp như một phần trong Hoạt động . Điều này giúp tránh tải thêm lớp và phân bổ đối tượng.

Minh họa:

public class ExampleActivity extends Activity implements OnClickListener {  
 protected void onCreate(Bundle savedValues) {  
 ...  
 Button button = (Button)findViewById(R.id.corky);  
 button.setOnClickListener(this);  
 }  
  
 // Implement the OnClickListener callback  
 public void onClick(View v) {  
 // do something when the button is clicked  
 }  
 ...  
}

- Lưu ý lệnh gọi lại onClick() trong ví dụ trên không có giá trị trả về, nhưng một số phương thức trình xử lý sự kiện khác phải trả về một giá trị boolean. Lý do phụ thuộc vào sự kiện. Dưới đây là lý giải cho điều này:

- [onLongClick()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onLongClick(android.view.View)) – Lệnh này sẽ trả về một giá trị boolean cho biết liệu sự kiện đã được sử dụng hay chưa và sự kiện không nên được thực hiện thêm nữa. Nghĩa là, trả về giá trị true (đúng) để cho biết bạn đã xử lý sự kiện và sự kiện sẽ dừng tại đây; trả về giá trị false (sai) nếu bạn chưa xử lý sự kiện và/hoặc sự kiện sẽ tiếp tục với bất kỳ trình nghe lượt nhấp nào khác.

- [onKey()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onKey(android.view.View,%20int,%20android.view.KeyEvent)) – Lệnh này sẽ trả về một giá trị boolean cho biết liệu sự kiện đã được sử dụng hay chưa và sự kiện không nên được thực hiện thêm nữa. Nghĩa là, trả về giá trị true (đúng) để cho biết bạn đã xử lý sự kiện và sự kiện sẽ dừng tại đây; trả về giá trị false (sai) nếu bạn chưa xử lý sự kiện và/hoặc sự kiện sẽ tiếp tục với bất kỳ trình nghe phím nào khác.

- [onTouch()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onTouch(android.view.View,%20android.view.MotionEvent)) – Lệnh này sẽ trả về một giá trị boolean cho biết trình nghe có sử dụng sự kiện này hay không. Sự kiện này có thể có nhiều hành động nối tiếp nhau. Vì vậy, nếu trả về giá trị false (sai) khi nhận được sự kiện hoạt động không hành động nghĩa là bạn chưa sử dụng sự kiện và cũng không quan tâm đến các hành động tiếp theo sau sự kiện này. Do đó, bạn sẽ không được gọi thêm bất kỳ hành động nào khác trong phạm vi sự kiện, chẳng hạn như một thao tác ngón tay bằng cử chỉ hoặc sự kiện hành động cuối.

- Hãy nhớ các sự kiện chính của phần cứng luôn được chuyển tới Chế độ xem hiện đang được lấy tiêu điểm. Chúng xuất phát từ đỉnh của hệ phân cấp Chế độ xem, rồi di chuyển xuống cho tới khi đạt điểm đến thích hợp. Nếu Chế độ xem (hoặc một Chế độ xem con) hiện có tiêu điểm, bạn có thể thấy sự kiện này di chuyển qua phương thức [dispatchKeyEvent()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#dispatchKeyEvent(android.view.KeyEvent)). Ngoài việc ghi lại các sự kiện quan trọng thông qua Chế độ xem, bạn cũng có thể tiếp nhận tất cả các sự kiện bên trong Hoạt động thông qua [onKeyDown()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onKeyDown(int,%20android.view.KeyEvent)) và [onKeyUp()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onKeyUp(int,%20android.view.KeyEvent))**.**

- Ngoài ra, khi xem xét việc nhập văn bản cho ứng dụng, hãy nhớ nhiều thiết bị chỉ có phương thức nhập phần mềm. Những phương thức đó không bắt buộc phải dựa trên phím bấm; một số có thể sử dụng tính năng nhập bằng giọng nói, chữ viết tay, v.v. Ngay cả khi phương thức nhập hiển thị một giao diện giống bàn phím, phương thức đó thường cũng sẽ không kích hoạt nhóm sự kiện [onKeyDown()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onKeyDown(int,%20android.view.KeyEvent)). Không được tạo một giao diện người dùng yêu cầu chế độ nhấn phím cụ thể trừ khi bạn chỉ muốn giới hạn ứng dụng ở các thiết bị có bàn phím cứng. Cụ thể, đừng dựa vào các phương thức này để xác thực đầu vào khi người dùng nhấn phím trả về; thay vào đó, hãy sử dụng các thao tác như [IME\_ACTION\_DONE](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#IME_ACTION_DONE) để báo cho phương thức nhập biết ứng dụng dự kiến sẽ phản ứng thế nào, nhờ đó, ứng dụng có thể thay đổi giao diện người dùng theo cách phù hợp. Tránh các giả định về cách thức hoạt động của phương thức nhập phần mềm và chỉ cần tin tưởng phương thức đó sẽ cung cấp cho ứng dụng văn bản đã được định dạng.

**4.1.5 Trình xử lý sự kiện**

**-** [onKeyDown(int, KeyEvent)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onKeyDown(int,%20android.view.KeyEvent)) – Được gọi khi một sự kiện nhấn phím mới xảy ra.

- [onKeyUp(int, KeyEvent)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onKeyUp(int,%20android.view.KeyEvent)) – Được gọi khi một sự kiện nhả phím xảy ra.

- [onTrackballEvent(MotionEvent)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onTrackballEvent(android.view.MotionEvent)) – Được gọi khi một sự kiện chuyển động bi xoay xảy ra.

- [onTouchEvent(MotionEvent)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onTouchEvent(android.view.MotionEvent)) – Được gọi khi một sự kiện chuyển động trên màn hình cảm ứng xảy ra.

- [onFocusChanged(boolean, int, Rect)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onFocusChanged(boolean,%20int,%20android.graphics.Rect)) – Được gọi khi chế độ hiển thị nhận hoặc mất tiêu điểm.

- Ngoài ra còn có một số phương thức khác bạn nên biết. Những phương thức này không thuộc lớp Chế độ hiển thị, nhưng có thể tác động trực tiếp đến cách xử lý các sự kiện. Vì vậy, khi quản lý các sự kiện phức tạp hơn bên trong một bố cục, hãy cân nhắc các phương thức sau:

- [Activity.dispatchTouchEvent(MotionEvent)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#dispatchTouchEvent(android.view.MotionEvent))– Cho phép [Activity](https://developer.android.com/reference/android/app/Activity?hl=vi) chặn tất cả các sự kiện chạm trước khi chúng được gửi đến cửa sổ.

- [ViewGroup.onInterceptTouchEvent(MotionEvent)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onInterceptTouchEvent(android.view.MotionEvent)) – Cho phép [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup?hl=vi) theo dõi các sự kiện khi chúng được gửi đến các Chế độ hiển thị con.

- [ViewParent.requestDisallowInterceptTouchEvent(boolean)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#requestDisallowInterceptTouchEvent(boolean)) – Gọi lệnh này theo Chế độ hiển thị gốc để cho thấy chế độ này không nên chặn các sự kiện chạm bằng [onInterceptTouchEvent(MotionEvent)](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onInterceptTouchEvent(android.view.MotionEvent))**.**

### **4.1.6 Chế độ cảm ứng**

- Khi người dùng đang thao tác trên một giao diện người dùng bằng các phím định hướng hoặc bi xoay, cần chú ý các mục thao tác (như các nút) để thấy mục nào sẽ tiếp nhận đầu vào. Tuy nhiên, nếu thiết bị có khả năng cảm ứng và người dùng bắt đầu tương tác với giao diện bằng cách chạm thì không cần đánh dấu các mục hoặc tập trung vào một Chế độ hiển thị cụ thể nữa. Do đó, có một chế độ dành cho tương tác có tên là "chế độ cảm ứng".

- Đối với thiết bị có khả năng cảm ứng, khi người dùng chạm vào màn hình, thiết bị sẽ chuyển sang chế độ cảm ứng. Từ thời điểm này trở đi, chỉ các Chế độ hiển thị có [isFocusableInTouchMode()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#isFocusableInTouchMode()) là true mới có thể đặt tiêu điểm, chẳng hạn như các tiện ích chỉnh sửa văn bản. Các Chế độ hiển thị khác có thể chạm, chẳng hạn như các nút, sẽ không đặt tiêu điểm khi được chạm; khi được nhấn vào, chúng sẽ kích hoạt trình nghe theo nhấp chuột.

- Bất cứ khi nào người dùng nhấn vào phím định hướng hoặc cuộn bằng bi xoay, thiết bị sẽ thoát khỏi chế độ cảm ứng và tìm một chế độ hiển thị để lấy tiêu điểm. Lúc này, người dùng có thể tiếp tục tương tác với giao diện người dùng mà không cần chạm vào màn hình.

- Trạng thái của chế độ cảm ứng được duy trì trên toàn bộ hệ thống (tất cả các cửa sổ và hoạt động). Để truy vấn trạng thái hiện tại, có thể gọi [isInTouchMode()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#isInTouchMode()) để xem thiết bị hiện có đang ở chế độ cảm ứng hay không.

### **4.1.7 Xử lý tiêu điểm**

- Khung này sẽ xử lý chuyển động tiêu điểm thường xuyên để phản hồi khi người dùng nhập liệu. Công việc này bao gồm thay đổi tiêu điểm khi Chế độ hiển bị xóa hoặc bị ẩn, hoặc khi có Chế độ hiển thị mới. Chế độ hiển thị thể hiện trạng thái sẵn sàng lấy tiêu điểm thông qua phương thức [isFocusable()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#isFocusable()). Để cho phép Chế độ hiển thị có thể lấy tiêu điểm, gọi [setFocusable()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#setFocusable(boolean)).

- Trên các thiết bị chạy phiên bản Android 9 (API cấp 28) trở lên, các hoạt động sẽ không chỉ định tiêu điểm ban đầu. Thay vào đó, nếu muốn, phải yêu cầu rõ tiêu điểm ban đầu.

- Chuyển động tiêu điểm dựa trên thuật toán tìm yếu tố lân cận gần nhất theo một hướng nhất định. Trong một số ít trường hợp, thuật toán mặc định có thể không phù hợp với hành vi mong muốn của lập trình viên.

- Thêm một trong các thuộc tính này vào Chế độ xem từ tiêu điểm để lại. Xác định giá trị của thuộc tính là id của Chế độ xem cho tiêu điểm cần cung cấp.

Minh họa:

<LinearLayout  
 android:orientation="vertical"  
 ... >  
 <Button android:id="@+id/top"  
 android:nextFocusUp="@+id/bottom"  
 ... />  
 <Button android:id="@+id/bottom"  
 android:nextFocusDown="@+id/top"  
 ... />  
</LinearLayout>

- Thông thường, trong bố cục dọc này, việc di chuyển lên từ Nút đầu tiên sẽ không hiển thị ở bất cứ đâu hoặc cũng không di chuyển xuống từ Nút thứ hai. Hiện tại, Nút trên cùng đã xác định phần tử dưới cùng là nextFocusUp (và ngược lại), tiêu điểm điều hướng sẽ xoay vòng từ trên xuống dưới và từ dưới lên trên.

- Nếu muốn khai báo một Chế độ xem có thể lấy tiêu điểm trong giao diện người dùng (thường thì không thể) thì hãy thêm thuộc tính XML android:focusable vào Chế độ xem trong bản khai báo bố cục. Đặt giá trị true. Bạn cũng có thể khai báo Chế độ xem có thể lấy tiêu điểm khi ở Chế độ cảm ứng với android:focusableInTouchMode.

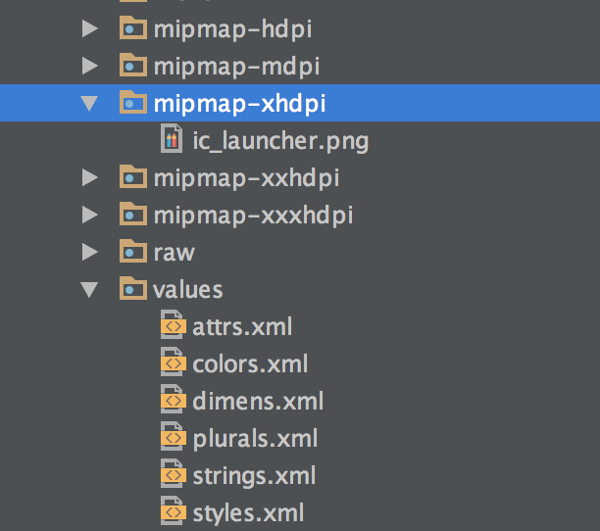
- Để yêu cầu một Chế độ xem cụ thể tập trung, hãy gọi [requestFocus()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#requestFocus())**.**

- Để theo dõi các sự kiện tập trung (được thông báo khi Chế độ xem nhận được hoặc mất tiêu điểm), hãy sử dụng [onFocusChange()](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#onFocusChange(android.view.View,%20boolean)), như đã gợi ý trong phần [Trình xử lý sự kiện](file:///C:\Users\dohun\OneDrive\Documents\Word\TranHanhNguyen_2100002561.docx#EventListeners)**.**

## **4.2 Tài nguyên ứng dụng trong android studio**

### **4.2.1 các kiểu tài nguyên**

- Để sử dụng tài nguyên trong một dự án Android, bạn cần phải sắp xếp chúng vào các thư mục con trong thư mục res. Cấu trúc thư mục của tài nguyên của một dự án Android tiêu chuẩn có thể trông giống như thế này:



**Hình 4. 5: Các kiểu tài nguyên trong Android**

- Có một số giới hạn các tên thư mục được hỗ trợ bởi hệ thống tài nguyên Android và hiểu biết về chúng là điều cần thiết khi bạn tiếp tục phát triển các ứng dụng Android. Hãy cùng tìm hiểu chúng chi tiết hơn.

+ animator: Thư mục tài nguyên animator có thể chứa các tập tin XML định nghĩa [hiệu ứng động](https://developer.android.com/guide/topics/graphics/prop-animation.html) cho các đối tượng View trong ứng dụng của bạn. Thay vì định nghĩa hiệu ứng động lặp đi lặp lại trong code Java của bạn, thì bạn có thể nhập một tập tin XML animator ở mọi nơi mà bạn cần phải sử dụng một hiệu ứng động cụ thể. Nếu bạn quyết định thay đổi hiệu ứng động sau này, thì bạn có thể thay đổi nó ở một nơi trong ứng dụng của bạn chứ không phải là nhiều nơi.

+ color: Dù bạn có thể lưu các giá trị màu hex trong thư mục values, nhưng thư mục color được sử dụng để lưu trữ các tập tin XML mà tương quan với [thông tin trạng thái với các màu sắc cụ thể](https://developer.android.com/guide/topics/resources/color-list-resource.html). Điều này là hữu ích nếu bạn cần thay đổi màu sắc văn bản của một View hoặc nền khi chạm vào để cung cấp phản hồi cho người dùng.

+ drawable: Nếu ứng dụng đòi hỏi tài nguyên hình ảnh cục bộ, chẳng hạn như JPGs và PNGs, thì bạn cần phải đặt chúng trong thư mục drawable. Đây cũng là nơi bạn chứa tập tin XML xác định [hình dạng và vector](https://developer.android.com/guide/topics/resources/drawable-resource.html) hoặc hình ảnh được vẽ ra trong code.

+ mipmap: Tương tự như drawable, thư mục mipmap chấp nhận các hình ảnh bitmap, nhưng nó được sử dụng đặc biệt cho các biểu tượng launcher của ứng dụng. Điều này là cần thiết vì Android sử dụng các hình ảnh một kích thước lớn hơn so với những gì các thiết bị thường sử dụng cho các biểu tượng launcher.

+ layout: [Thư mục layout](https://developer.android.com/guide/topics/resources/layout-resource.html) chứa các tập tin XML xác định giao diện người dùng ứng dụng của bạn.

+ menu: Các tập tin menu XML được sử dụng để xác định action bar và [menu điều hướng và menu con](https://developer.android.com/guide/topics/resources/menu-resource.html)**.** Các nguồn tài nguyên này được đặt trong thư mục tài nguyên menu.

+ raw: Thư mục raw được sử dụng để lưu trữ các tập tin raw tùy ý cho ứng dụng của bạn, chẳng hạn như âm thanh, video và các tập tin văn bản. Điều này cung cấp một cách dễ dàng để truy cập các tập tin, mặc dù nếu bạn cần truy cập vào tên gốc của tập tin hoặc phân cấp thư mục, bạn nên xem xét việc đặt các tập tin raw trong thư mục asset của Android.

+ xml: Thư mục này chứa các tập tin XML tùy ý được sử dụng bởi Android cho các tác vụ khác nhau, chẳng hạn như việc xác định [cấu hình tìm kiếm](https://developer.android.com/guide/topics/search/searchable-config.html) hoặc khả năng bên ngoài, chẳng hạn như việc sử dụng [Android Auto](https://www.android.com/auto/)**.**

### **4.2.2 Values**

- Trong khi danh sách của các thư mục nguồn tài nguyên có sẵn ở trên là khá dài, có thêm một thư mục mà bạn sẽ sử dụng thường xuyên khi là một nhà phát triển Android và đó là thư mục values. Thư mục values có thể chứa nhiều tập tin XML bao gồm các giá trị đơn giản được sử dụng bên trong các ứng dụng của bạn, bao gồm:

+ arrays: Các đối tượng mảng đơn giản có thể được đọc trong ứng dụng của bạn.

+ colors: Đặt tên giá trị hex của màu sắc có thể được sử dụng ở nhiều nơi trong ứng dụng của bạn.

+ dimens: Các kích thước có thể định nghĩa bất cứ điều gì liên quan đến kích thước trong Android, bao gồm cả văn bản và padding.

+ integers: Nếu có các số nguyên cụ thể mà bạn cần sử dụng trong ứng dụng của bạn, bạn có thể lưu trữ chúng trong tập tin tài nguyên integers do đó chúng có thể dễ dàng tìm và thay đổi sau này trong quá trình phát triển của bạn.

+ strings: Chuỗi là một điều cần thiết trong bất kỳ ứng dụng Android nào mà bạn xây dựng. Thay vì phân phát chúng trong code của bạn, bạn có thể giữ các chuỗi trong tập tin giá trị strings.xml.

+ plurals: Tương tự chuỗi, plurals cho phép bạn cung cấp sự thay thế chuỗi khi một con số được truyền vào một hàm. Điều này xử lý các tình huống nơi mà một phép lượng hoá chuỗi sẽ thay đổi dựa trên số lượng các phần mà nó đại diện.

+ styles: Bất kỳ ứng dụng mà bạn xây dựng cần một giao diện nhất quán và điều này có thể đạt được bằng cách sử dụng tập tin styles.xml Sử dụng styles, bạn có thể thay thế giao diện mặc định cho các thành phần và các thuộc tính có thể được liên kết với các đối tượng View khác nhau trong các tập tin layout.

### **4.2.3 Asset, Share Preferences**

Asset : Trong Android, assets là một thư mục trong dự án của bạn được sử dụng để lưu trữ các tệp tài nguyên không thay đổi như hình ảnh, âm thanh, video, tệp văn bản, tệp HTML, v.v. Thư mục assets nằm trong thư mục main trong dự án của bạn.

- Một số điểm quan trọng về thư mục assets :

+ Quyền truy cập: Các tệp trong thư mục assets được coi là tài nguyên không thay đổi, nghĩa là bạn chỉ có thể đọc từ thư mục này. Bạn không thể ghi hoặc cập nhật tệp tin trực tiếp trong assets .

+ Cấu trúc thư mục: Thư mục assets thường không có các cấu trúc phụ chia nhỏ, tất cả các tệp tin và thư mục đều nằm trực tiếp trong thư mục assets .

+ Đọc tệp từ assets : Để truy cập và đọc tệp tin từ assets , bạn có thể sử dụng lớp AssetManager . Bạn có thể mở một tệp tin từ assets bằng cách gọi open() hoặc các phương thức tương tự, sau đó đọc dữ liệu từ InputStream trả về.

Share Preferences: Shared Preferences là đối tượng Android cung cấp cho việc lưu trữ và truy xuất các dữ liệu có kiểu cơ bản như Boolean, string, float, long, and integer trong lập trình các ứng dụng đơn giản.   
- Shared Preference sẽ được lưu lại thông qua việc sử dụng chỉ định khóa cho từng giá trị dữ liệu (theo từng cặp key/value), những giá trị key.value này sẽ được tự động ghi vào một tập tin XML được chứa bên trong thư mục của ứng dụng. Những cặp giá trị này sẽ tồn tại và sẵn sàng cho việc sự dụng (đọc và cập nhật) trong suốt phiên làm việc của ứng dụng và được chia sẻ bên trong các thành phần cúa ứng dụng (như activity, service,…). Các giá trị shared preference này không thể truy xuất và sử dụng từ những ứng dụng khác.

- Dữ liệu của ứng dụng được lưu trữ trong thư mục data/data/ Tên package của ứng dụng.

- Có 2 cách lưu trữ dữ liệu trong Shared Preference:

+ Cách 1: Sử dụng Activity Preferences:

Chúng ta phải gọi getPreferences (int mode) có trong lớp Activity.Chỉ sử dụng khi preference file là cần thiết trong Activity Nó không yêu cầu tên vì nó sẽ là tập tin ưu tiên duy nhất cho Activity.Lập trình viên thường không thích sử dụng tính năng này ngay cả khi họ chỉ cần một tệp tùy chọn trong Activity. Họ thích sử dụng phương thức tùy biến getSharedPreferences (tên String, chế độ int).

+ Cách 2: Sử dụng Custom Preferences:

Sử dụng phương thức getSharedPreferences(String name,int mode) cho Custom Preferences. Được sử dụng trong các trường hợp khi yêu cầu nhiều tập tin tùy Preferences trong Activity. Tên của tập tin preference được truyền cho tham số đầu tiên.

- Mode (chế độ) và kiểu trong Shared Preference:

- Để tạo Activity dựa vào preferences hoặc custom preferences phải truyền chế độ (mode) để cho hệ thống biết về quyền riêng tư của tập tin preference.

- Có 3 loại Mode trong Shared Preference:

+ Context.MODE\_PRIVATE – Giá trị mặc định (Không chấp nhận ứng dụng từ bên ngoài)

+ Context.MODE\_WORLD\_READABLE – Có thể đọc từ ứng dụng khác

+ Context.MODE\_WORLD\_WRITEABLE – Đọc/Ghi từ ứng dụng khác

* MODE\_PRIVATE – Đây là chế độ mặc định. MODE\_PRIVATE được sử dụng để thiết lập chế độ riêng tư, không chấp nhận ứng dụng từ bên ngoài.
* MODE\_WORLD\_READABLE – mode điều khiển cho phép đọc tập tin Preferences từ ứng dụng khác.
* MODE\_WORLD\_WRITEABLE – mode điều khiển cho phép chỉnh sửa tập tin Preferences từ ứng dụng khác.

## **4.3 Lưu trữ dữ liệu**

Các bước lưu dữ liệu trong Shared Preference:

Bước 1: Để lưu dữ liệu trong SharedPreferences chúng ta cần gọi phương thức của lớp SharedPreferences. Lớp SharedPreferences trả về một đối tượng Editor .

Bước 2: Lớp Editor và sử dụng các phương thức put<type> để thêm mới (nếu chưa có) hoặc cập nhật giá trị cho các từ khóa (key) được chỉ định như đoạn code bên dưới.

SharedPreferences.Editor editor = mySharedPreferences.edit();

// Store new primitive types in the shared preferences object.

editor.putBoolean("isTrue", true);

editor.putFloat("floatValue", 1f);

editor.putInt("intValue", 2);

editor.putLong("longValue", 3l);

editor.putString("stringValue", "Not Empty");

Bước 3: Sử dụng phương thức để gọi đồng bộ và trả về kết quả lưu thành công hay không hoặc sử dụng phương thức Apply() nếu chúng ta không quan tâm đến kết quả trả về

- Các bước đọc và lấy dữ liệu trong Shared Preference  
  
Bước 1: Để đọc dữ liệu reference từ đối tượng SharedPreferences chúng ta sử dụng phương thức getPreferences (int mode) hoặc getSharedPreferences (String name,int mode).  
Bước 2: Chúng ta sẽ truy xuất lại các giá trị đã lưu bằng cách dùng các phương thức an toàn (gọi là type-safe) SharedPreferences.get<type> với 2 thông số: thông số đầu là key để lấy dữ liệu, và thông số thứ hai là giá trị mặc định (trong trường hợp key đó chưa được lưu trước đó).

// Retrieve the saved values.

boolean isTrue = mySharedPreferences.getBoolean("isTrue", false);

float lastFloat = mySharedPreferences.getFloat("floatValue", 0f);

I int wholeNumber = mySharedPreferences.getInt("intValue", 1);

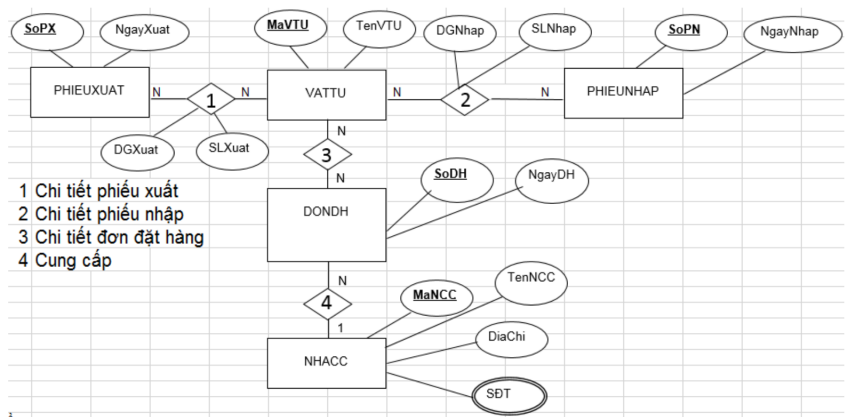
long aNumber = mySharedPreferences.getLong("longValue", 0);

String stringPreference = mySharedPreferences.getString("stringValue", "");

- Xóa dữ liệu hoặc xóa tất cả dữ liệu:

+ Sử dụng phương thức remove(String key) trong SharedPreferences.Editor để xóa. Để xóa tất cả dữ liệu chúng ta gọi phương thức clear(). Sau đó gọi commit() để lưu dữ liệu.

## **4.4 Tương tác với CSDL**



**Hình 4. 6: Hình ảnh tương tác với CSDL**

### **4.4.1 Thao Tác Cơ Bản với Sản Phẩm:**

Thêm Sản Phẩm: Xử lý nghiệp vụ cần tạo và kiểm tra dữ liệu để thêm sản phẩm mới vào CSDL. Điều này bao gồm kiểm tra tính hợp lệ của dữ liệu đầu vào, xác định giá trị mặc định hoặc tự động tạo các giá trị, và sau đó thực hiện câu truy vấn để lưu trữ thông tin sản phẩm mới vào CSDL.

Sửa Đổi Sản Phẩm: Khi người dùng sửa đổi thông tin của sản phẩm, xử lý nghiệp vụ phải kiểm tra và cập nhật dữ liệu tương ứng trong CSDL. Các biện pháp bảo mật và kiểm tra tính nhất quán cũng cần được thực hiện để tránh lỗi dữ liệu.

Xóa Sản Phẩm: Xử lý nghiệp vụ phải xác minh quyền hạn và thực hiện xóa sản phẩm khỏi CSDL. Cần có các biện pháp đề phòng để tránh xóa ngẫu nhiên thông tin quan trọng.

Truy Vấn Sản Phẩm: Khi người dùng yêu cầu xem thông tin sản phẩm, xử lý nghiệp vụ phải thực hiện các câu truy vấn đến CSDL để lấy dữ liệu và trả về cho giao diện

### **4.4.2 Quản Lý Đơn Hàng:**

Tạo Đơn Hàng: Xử lý nghiệp vụ cần tạo và lưu trữ đơn hàng mới vào CSDL, bao gồm thông tin về sản phẩm, khách hàng, giá, và các chi tiết khác.

Cập Nhật Trạng Thái Đơn Hàng: Khi đơn hàng thay đổi trạng thái, xử lý nghiệp vụ cần cập nhật trạng thái tương ứng trong CSDL và thực hiện bất kỳ thao tác khác liên quan.

Xem Đơn Hàng: Xử lý nghiệp vụ sẽ thực hiện truy vấn đến CSDL để lấy thông tin chi tiết về đơn hàng khi được yêu cầu.

### **4.4.3 Quản Lý Khách Hàng:**

Thêm, Sửa, Xóa Khách Hàng: Xử lý nghiệp vụ sẽ quản lý thêm, sửa, và xóa thông tin khách hàng từ CSDL. Điều này đặc biệt quan trọng để duy trì cập nhật thông tin khách hàng.

Liên Kết Đơn Hàng với Khách Hàng: Khi có đơn hàng mới, xử lý nghiệp vụ cần liên kết đơn hàng với thông tin khách hàng tương ứng trong CSDL.

# HƯỚNG PHÁT TRIỂN

- Tốc độ tải trang nhanh quyết định rất lớn vào việc khách hàng có tiếp tục ở lại trải nghiệm trang Web bán hàng của doanh nghiệp hay không. Đây chính là một lưu ý mà dù doanh nghiệp lựa chọn hướng phát triển website như thế nào đều không thể bỏ qua. Nếu tốc độ tải trang quá 3 giây, người dùng sẽ cảm thấy chán nản, khó chịu và lập tức rời đi, thậm chí là không bao giờ quay lại tìm kiếm website đó nữa. Ngoài ra, tốc độ tải trang còn ảnh hưởng rất lớn đến vị trí của website bán hàng trên các công cụ tìm kiếm như Google hoặc Bing, từ đó, tác động đến lưu lượng truy cập tự nhiên cũng như tỷ lệ chuyển đổi khách hàng và tỷ lệ chốt đơn sản phẩm – dịch vụ.

- Doanh nghiệp luôn phải thường xuyên cập nhật website của mình, cả về thông tin, giá bán sản phẩm, các hình ảnh và Video minh họa, các mức giá khuyến mãi hay những ưu đãi theo các dịp lễ – Tết,… Ngoài ra, doanh nghiệp cũng nên thường xuyên cập nhật các nội dung blog (nếu có) để hướng dẫn khách hàng cách sử dụng sản phẩm – dịch vụ hay nhiều thông tin khác trong lĩnh vực mà mình đang kinh doanh. Điều này giúp người dùng cảm thấy sự tận tâm và chuyên nghiệp từ chúng ta để từ đó, tạo niềm tin về thương hiệu trong họ.

- Khi nhận được tương tác từ phía doanh nghiệp, khách hàng sẽ cảm thấy an tâm, tin tưởng và dễ chịu hơn và mong muốn gắn bó lâu dài hơn trong quá trình mua sắm. Từ đó, doanh nghiệp sẽ sở hữu một nguồn khách hàng trung thành, quen thuộc dài lâu, góp phần không nhỏ trong việc tăng trưởng doanh số và doanh thu bền vững.

# TÀI LIỆU THAM KHẢO

**[1]** Slide tham khảo và thông tin liên quan tới Android studio của thầy Bùi Tiến Đức

**[2]** Ngôn ngữ lập trình Java, <https://aws.amazon.com/vi/what-is/java/>

**[3]** Các thành phần của Android, <https://v1study.com/android-cac-thanh-phan-cua-android.html>

**[4]** Lịch sử phát triển hệ điều hành android, <https://cellphones.com.vn/sforum/lich-su-cua-android-su-phat-trien-cua-he-dieu-hanh-di-dong-lon-nhat-tren-the-gioi-p1>

**[5]** Sqlite là gì, <https://viblo.asia/p/sqlite-la-gi-E375zVVR5GW>