|  |
| --- |
| **TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**  **KHOA CÔNG NGHỆ THÔNG TIN**  ---🙟🕮🙝---    **BÀI TẬP LỚN**  LẬP TRÌNH THIẾT BỊ DI ĐỘNG  Đề tài : XÂY DỰNG ỨNG DỤNG DI ĐỘNG  BÁN HÀNG ĐIỆN TỬ  GVHD: **ThS. Lê Viết Tuấn**  SVTH : **Hồ Thị Yến – DH15TH03**  MSSV: **1551010154**  ***Thành phố Hồ Chí Minh, năm 2018*** |

Lời cảm ơn

🏵🕮🏵

Kiến thức là một kho tàng vô giá, là một hành trang vững chắc nhất để vào đời, do đó người truyền đạt kiến thức là những người rất đáng được tôn vinh. Chính vì vậy em xin giành trọn trang đầu tiên của bài báo cáo này kính xin gửi đến các quý thầy cô trường Đại học Mở TP.HCM lòng biết ơn chân thành và sâu sắc.

Em xin chân thành cảm ơn toàn thể Ban giám hiệu trường Đại học Mở TP.HCM, cảm ơn các thầy cô khoa Công nghệ thông tin, những người đã dẫn dắt chúng em trong suốt những năm học tại trường. Đặc biệt là lời cảm ơn chân thành đến cô Thầy Lê Viết Tuấn đã tận tâm hướng dẫn, giúp đỡ em hoàn thành bài tập lớn này.

Dưới sự chỉ bảo tận tâm của thầy đã truyền đạt cho em những kiến thức thật vững vàng để hòa nhập cùng với công việc, cuộc sống. Đó là những kỷ vật vô giá đối với chúng em.

Tuy nhiên, do vốn kiến thức có hạn nên không tránh được những sai sót, vì vậy em rất mong nhận được sự chỉ bảo và góp ý của thầy.

Một lần nữa, em xin gửi đến quý thầy lời cảm ơn chân thành nhất, kính chúc quý thầy cô dồi dào sức khỏe và thành công trong công việc.

TP. Hồ Chí Minh, ngày 15 tháng 04 năm 2019

Sinh viên thực hiện

Hồ Thị Yến

Nhận xét của giáo viên

MỤC LỤC

[CHƯƠNG 1: TỔNG QUAN 5](#_Toc371277653)

[1.1. Giới thiệu 5](#_Toc371277654)

[1.2. Mục đích làm đề tài 5](#_Toc371277655)

[1.3. Phương pháp nghiên cứu 5](#_Toc371277656)

[1.4. Phạm vi đề tài 6](#_Toc371277655)

[1.5. Những đóng góp của đề tài 6](#_Toc371277656)

[1.6. Bố cục của báo cáo 6](#_Toc371277656)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 8](#_Toc371277657)

[2.1 Lập trình Android 8](#_Toc371277658)

[2.1.1. Giới thiệu sơ lược về Android 8](#_Toc371277659)

[2.1.2. Cấu trúc của Google Android 10](#_Toc371277660)

[2.1.3. Các thành phần cơ bản của một ứng dụng Android 11](#_Toc371277661)

[2.1.4. Google Maps 19](#_Toc371277662)

[2.1.5. Chu trình sống của một ứng dụng Android 19](#_Toc371277662)

[2.2 Web Service 20](#_Toc371277663)

[2.2.1. Khái niệm 20](#_Toc371277664)

[2.2.2. Cấu trúc Web Service 20](#_Toc371277665)

[2.2.3. SOAP 21](#_Toc371277666)

[2.2.4. WSDL 22](#_Toc371277667)

[2.2.5. UDDI 22](#_Toc371277668)

[2.2.6. Mô hình hoạt động của Web Service 23](#_Toc371277668)

[2.3 PHP 24](#_Toc371277663)

[2.3.1. Giới thiệu 24](#_Toc371277664)

[2.3.2. PHP cơ bản 24](#_Toc371277665)

[CHƯƠNG 3: CƠ SỞ DỮ LIỆU 28](#_Toc371277669)

[3.1. Cơ sở dữ liệu của ứng dụng 28](#_Toc371277670)

[3.1.1. Mô hình thực thể kết hợp 28](#_Toc371277672)

[3.1.2. Mô hình quan hệ 28](#_Toc371277673)

[3.1.3. Bảng dữ liệu 29](#_Toc371277674)

[3.2. Xây dựng các chức năng trong ứng dụng bán hàng 31](#_Toc371277670)

[3.2.1. Chức năng đăng nhập 31](#_Toc371277672)

[3.2.2. Chức năng hiển thị danh sách sản phẩm lên thiết bị 33](#_Toc371277673)

[3.2.3. Chức năng đặt hàng 34](#_Toc371277674)

[CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG 35](#_Toc371277675)

[4.1. Giới thiệu App bán hàng điện tử 35](#_Toc371277676)

[4.2. Các trang của App 35](#_Toc371277677)

[4.2.1. Giao diện người dùng 34](#_Toc371277678)

[4.2.2. Giao diện quản trị 45](#_Toc371277679)

[CHƯƠNG V: TỔNG KẾT 50](#_Toc371277680)

**Chương 1: TỔNG QUAN**

**1.1 Giới thiệu**

Ngày nay, ngành dịch vụ bán lẻ tại nước ta đang phát triển mạnh, các trung tâm bán lẻ xuất hiện ở khắp mọi nơi và cạnh tranh nhau khốc liệt. Để tạo nên một trung tâm bán kẻ thành công ngoài việc làm thế nào để có giá bán hợp lý, hậu mãi tốt thì quy trình thanh toán sản phẩm phải nhanh gọn, tác phong làm việc của nhân viên phải chuyên nghiệp nhằm tạo nên bộ mặt hiện đại cho trung tâm. Hầu như hiện nay các trung tâm bán lẻ vẫn còn hoạt động khá thủ công, quy trình thanh toán còn phụ thuộc nhiều vào đội ngũ nhân viên. Vì vậy, việc xây dựng chương trình hỗ trợ bán hàng dựa trên nền tảng di động sẽ làm quy trình bán hàng đơn giản và quy trình bán hàng chuyên nghiệp hơn.

Việc áp dụng công nghệ thông tin vào các trung tâm bán lẻ đã được áp dụng ở nhiều quốc gia phát triển. Tất cả quy trình thanh toán cũng như sao lưu dữ liệu gần như tự động, tránh được các sai sót do nhân viên gặp phải, ngoài ra còn giảm tải được công việc cho nhân viên bán hàng.

Trong giới hạn của đề tài này, ta sẽ tập trung nghiên cứu việc sử dụng nền tảng di động để tạo nên ứng dụng hỗ trợ bán hàng cho một trung tâm bán lẻ với các chức năng cơ bản như tìm kiếm sản phẩm, thiết lập hóa đơn và lưu trữ dữ liệu…

**1.2 Mục đích làm đề tài**

Hiện nay trong các trung tâm bán lẻ, khi khách hàng chọn mua một món hàng, các nhân viên phải lập hóa đơn bằng tay tại quầy thanh toán, trong các trung tâm bán lẻ lớn có nhiều khu vực khác nhau, khách hàng phải di chuyển từ khu vực tham quan sản phẩm đến khu vực thanh toán. Để tránh rắc rối này, ứng dụng hỗ trợ bán hàng sẽ chạy trên các thiết bị di động như Smartphone, máy tính bảng,…và kết nối đến một server lưu trữ dữ liệu để thực hiện thanh toán, sao lưu và in hóa đơn. Việc phục vụ ngay tại chỗ sẽ giúp khách hàng thoải mái nhất.

Ngoài ra vì quá trình thanh toán được thực hiện tự động trên hệ thống thông tin nên thời gian thanh toán sản phẩm được giảm đáng kể, giúp khách hàng có thiện cảm tốt với trung tâm bán hàng.

Trong đề tài này ta sẽ nghiên cứu cách áp dụng mô hình lập trình mạng xây dựng mô hình giao tiếp giữa máy chủ lưu trữ và các thiết bị di động.

* 1. **Phương pháp nghiên cứu**

Để có thể xây dựng được ứng dụng, ta cần tìm hiểu lập trình trên nền tảng hệ điều hành Android, cụ thể là: Tìm hiểu cấu trúc và chức năng các thành phần trong chương trình Android; tìm hiểu các lớp, các Control (điều khiển) và cách sử dụng ; các hỗ trợ của hệ điều hành Android cho lập trình viên ; cách cài đặt, chức năng các gói SDK ; các thư viện,…

Trong đề tài này, vấn đề quan trọng là việc xây dựng kết nối từ các thiết bị di động đến máy chủ. Để thực hiện vấn đề này, ta có nhiều kỹ thuật để thực hiện trong đó mô hình giao tiếp theo hướng dịch vụ (Server Oriented Architecture - SOA) hay còn gọi là mô hình Client-Server là giải pháp hiệu quả. Ta sẽ đi tìm hiểu phương pháp giao tiếp, truyền tải dữ liệu, các thông điệp giữa các thiết bị trong mô hình này.

Ngoài ra ta còn phải tìm hiểu cách lưu trữ dữ liệu trên thiết bị di động và máy chủ. Mỗi thiết bị có những phương pháp lưu trữ khác nhau, ta cần tìm ra phương pháp hiệu quả nhất dựa vào độ lớn và độ phức tạp cũng như mức độ quan trọng của dữ liệu.

* 1. **Phạm vi đề tài**

Ứng dụng hỗ trợ bán hàng trên thiết bị di động được xây dựng để áp dụng vào các trung tâm mua sắm lớn có nhiều tầng, nhiều khu vực, với nhiều mặt hàng khác nhau để tận dụng ưu điểm của thiết bị di động và hạ tầng hệ thống mạng của trung tâm.

* 1. **Những đóng góp của đề tài**

Đề tài này đã áp dụng kỹ thuật lập trình mạng và lập trình trên thiết bị di động để khai thác sự phát triển của hạ tầng mạng hiện nay và tính cơ động mà các thiết bị di động mang lại, đề tài sẽ giúp các trung tâm bán hàng giải quyết các vấn đề về nhân lực, cung cấp cho khách hàng một môi trường mua sắm hiện đại. Với những lợi thế và công nghệ hiện tại thì khả năng áp dụng đề tài trong môi trương thực tế là rất cao.

Ngoài ra, ứng dụng hỗ trợ bán hàng trên các thiết bị di động còn được xây dựng theo các chức năng riêng biệt, dễ dàng tùy chỉnh lại để phù hợp với từng mô hình bán hàng trong thực tế.

* 1. **Bố cục của báo cáo**

Chương 1: Tổng quan – Giới thiệu về đề tài, thực trạng hiện nay và khả năng ứng dụng của đề tài.

Chương 2: Cơ sở lý thuyết:

* Phần 1: Lập trình Android – Khái quát về các khái niệm, thành phần cơ bản và các thành phần có liên quan đến đề tài.
* Phần 2: Web Service – Khái niệm cơ bản về Web Service và các chuẩn để giao tiếp giữa Client và Server.

Chương 3: Cơ sở dữ liệu để thực hiện ứng dụng.

Chương 4: Xây dựng ứng dụng – Giải thích các chức năng ứng dụng, cách hoạt động và demo các hình ảnh của ứng dụng.

Chương 5: Tổng kết – Ưu nhược điểm và hướng phát triển của ứng dụng.

**Chương 2: Cơ sở lý thuyết**

**2.1. Lập trình Android**

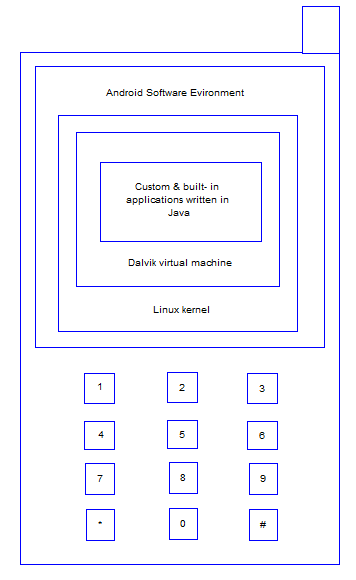
**2.1.1 Giới thiệu sơ lược về Android**

Android là một sản phẩm của tổ chức *Open Handset Alliance* và được *Google* mua lại và tiếp tục phát triển với tiềm lực làm một cuộc cách mạng trên thị trường điện thoại di động thế giới. Tổ chức *Open Handset Alliance* là một tổ chức gồm xấp xỉ 30 tổ chức nhỏ hơn với phương châm mang đến những thứ “tốt hơn” và “mở” cho thị trường điện thoại di đông: “Android được xây dựng với mục tiêu rõ ràng là mở, hoàn chỉnh và là một nền tảng miễn phí được tạo ra chuyên biệt cho điện thoại”. Tính đến thời điểm hiện tại, Android là hệ điều hành cho thiết bị di động phổ biến nhất trên thế giới.

Android là một phần mềm được xây dựng cho thiết bị di động với hạt nhân (kernel) là hệ điều hành Linux được viết bằng C và C++ nhưng các ứng dụng người dùng hay ngay cả các ứng dụng được tích hợp sẵn đều được viết bằng ngôn ngữ lập trình Java. Với đặc điểm là mã nguồn mở, các thành phần trong Android dễ dàng được cộng đồng các nhà phát triển cung cấp và do đó giúp cho Android có khả năng tùy biến cao, chạy được trên nhiều thiết bị phần cứng. Có thể nói ngắn gọn, Android chỉ là phần nhân, còn phần vỏ, tức phần giao diện được các nhà phát triển viết để phù hợp với thiết bị phần cứng và phong thái riêng của từng hãng. Tuy nhiên, việc này cũng có nhược điểm mà hiện nay nhiều công ty phát triển cho Android hay than phiền đó là “sự phân mảng”. Nhưng Google đang cố gắng chuẩn hóa các phiên bản sau này, cộng thêm với sự phát triển vượt bậc của phần cứng thì trong tương lai, sự phân mảng không còn gây quá nhiều khó khăn cho các nhà phát triển.

Android với lời hứa sẽ mang lại cho tất cả mọi người và với việc hỗ trợ rất tốt cho nhiều thiết bị phần cứng chứ không chỉ cho những thiết bị đời mới hay chỉ những điện thoại thông minh (smartphone) đắt tiền như các hệ diều hành cạnh tranh là IOS hay Windows Phone (đương nhiên Android sẽ chạy tốt hơn trên các thiết bị cấu hình cao hơn). Với việc hỗ trợ cho nhiều thiết bị phần cứng một cách linh hoạt và quan trọng hơn hết là miễn phí thì Android luôn là lựa chọn hàng đầu cho các nhà sản xuất thiết bị di động. Còn so với các điện thoại chức năng (feature-phone), điện thoại chỉ dành cho các tác vụ cơ bản như nghe, gọi, nhắn tin thì Android có lợi thế hơn với kho ứng dụng khổng lồ nên có thể thực hiện được nhiều tác vụ hơn. Và khi mà giá cả phần cứng ngày càng rẻ thì các smartphone như Android sẽ là dĩ nhiên, các điện thoại chức năng sẽ chỉ còn là quá khứ. Rõ ràng, thị trường cho Android là hết sức màu mỡ, rộng lớn. Tính đến cuối năm 2017, Android chiếm 72% thị phần thiết bị di động và là hệ điều hành cho thiết bị di động phổ biến nhất trên thế giới và vẫn đang tiếp tục bành trướng.

Sự ra mắt cả 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 phân cứng , phần mềm và viễn thông nhằm mục đích tạo nên một chuẩn mở cho điện thoại di động trong tương lai



Hình 2.1: Nền tảng của Android. Nhân (Kernal) là hệ điều hành Linux, còn các

ứng dụng người dùng và cả ứng dụng tích hợp sẵn được viết bằng Java.

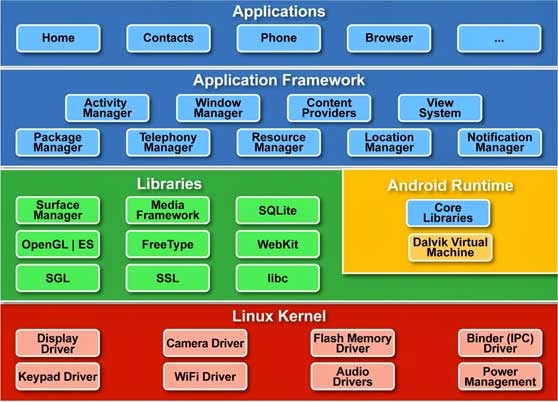
*🟅 Tham khảo Manning – Unlocking Android.*

**2.1.2 Cấu trúc của Google Android**

Có 5 tầng cơ bản trong hệ điều hành Android: Application Framework,

Android Runtime, Native Libraries, Linux Kernel. Mỗi tầng làm viêc đều nhờ sự giúp đỡ của tầng dưới.

* Tầng ứng dụng (Application): bao gồm tất cả các ứng dụng có trong thiết bị chạy Android như: phone (gọi điện), contact (danh bạ), game (trò chơi), browser (trình duyệt), … và một số ứng dụng chạy ngầm. Người dùng có quyền gỡ bỏ hay cài đặt các ứng dụng tùy thích ở tầng này.
* Tầng khung của ứng dụng (Application Framework): tầng này Google xây dựng cho các nhà phát triển xây dựng ứng dụng của họ trên Android, bằng cách gọi các APIs có sẵn mà Google đã viết để sử dụng các tính năng của phần cứng mà không cần hiểu cấu trúc bên dưới.
* Tầng thư viện gốc (Native Libraries): bao gồm một số thư viện C/C++ được sử dụng bởi các thành phần khác nhau của hệ thống Android, một số thư viện cơ bản như: System Clibrary, SQLite, Media Libraries, 3D Libraries,…
* Tầng thực thi (Runtime): mỗi ứng dụng Android chạy trên một tiến trình riêng của máy ảo Dalvik (virtual machine). Dalvik được đề chạy nhiều máy ảo cùng một lúc một cách hiệu quả trên thiết bị di động.
* Tầng nhân của Linux (Linux Kernel): đây là nhân của hệ điều hành Android, mọi xử lý hệ thống đều phải thông qua tầng này. Linux Kernel cung cấp các tiến trình điều khiển thiết bị phần cứng như : Camera, USB, bluetooth, …Kernel hoạt động như một lớp trừu tượng giữa phần cứng và phần mềm còn lại của hệ thống



Hình 2.2: Các tầng cơ bản của Google Android.

**2.1.3 Các thành phần cơ bản của một ứng dụng Android**

**2.1.3.1 Application components**

Application components được xem như những “trụ cột” chính để xây dựng nên một ứng dụng Android. Mỗi component là một ngõ để hệ thống có thể vào ứng dụng. Không phải tất cả các component đều là ngõ chính cho người dùng và một số phụ thuộc vào nhau, nhưng mỗi cái tồn tại và đóng vai trò riêng của nó, giúp định hình hành vi tổng thể của một ứng dụng.

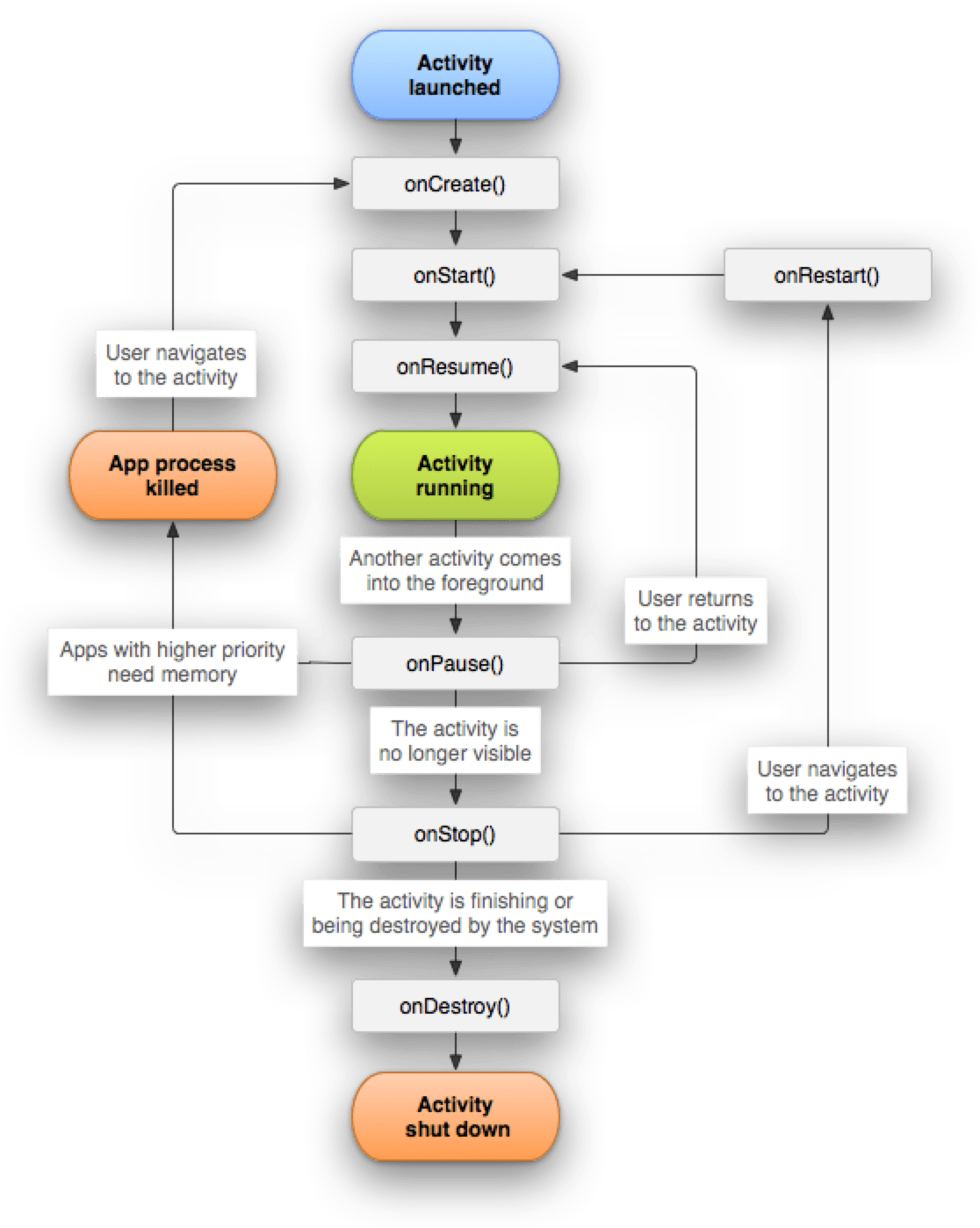
Có 4 loại Application components. Mỗi loại phục vụ một mục đích khác nhau và có vòng đời khác nhau.

**2.1.3.2 Activity**

Đây là thành phần quan trọng nhất, chúng ta làm việc với nó nhiều nhất. Một ứng dụng Android có thể có hoặc không có giao diện người dùng, nhưng nếu có giao diện người dùng thì ứng dụng đó sẽ có một hoặc nhiều Activity. Activity tương ứng với khái niệm Form (cửa sổ) trong hệ điều hành Windows. Activity làm nhiệm vụ hiển thị giao diện và đáp ứng đến hệ thống các sự kiện mà người dùng khởi tạo. Một Activity là một cửa sổ làm việc, thường mỗi cửa sổ này lấp đầy màn hình, những nội dung trực quan ở cửa sổ Activity được thể hiện bởi các Wiget (hay còn được gọi là View) như*: Button, TextView, RadioButton, ListView, Spinner,…*

Vòng đời (Lifecycle) của một Activity bao gồm 3 vòng chính:

* Toàn bộ thời gian sống (Entire Lifetime): từ phương thức *onCreate()* đến *onDestroy().*
* *onCreate():* được gọi khi lần đầu Activity được tạo. Đây là phương thức được dùng để khởi tạo các thành phần, giá trị các biến trong Activity. Khi gọi phương thức *onCreate(),* Android sẽ truyền một đối tượng chứa trạng thái hoạt động nào đó đã được lưu của Activity khi được gọi trược đó.
* *onDestroy():* nới Activity có thể thực hiện các công việc trước khi bị hủy.
* Thời gian hiển thị (Visible Lifetime): từ phương thức *onStart()* đến *onStop().*
* *onStart():* được gọi trước khi Activity xuất hiện trên màn hình. Khi phương thức này hoàn thành, nếu Activity được hiển thị trên màn hình, điều khiển sẽ được chuyển cho *onResume(),* nếu Activity không hiển thị trên màn hình vì lí do nào đó, điều khiển này sẽ được chuyển cho *onStop().*
* *onStop():* được gọi khi Activity không còn được hiển thị nữa, và chuẩn bị được hủy.
* *onRestart()*: được gọi sau khi Activity đã bị stop trước đó, trước khi nó được bắt đầu được khởi động chạy trở lại.
* Thời gian chạy (Foreground Lifetime): từ phương thức *onResume()* đến *onPause().*
* *onResume():* được gọi sau *onStart()* nếu Activity được hiển thị trên màn hình, *onResume()* cũng được gọi khi Activity nhường quyền lại cho một Activity khác, khi Activity đó kết thúc, lúc đó sẽ được hiển thị lại trên màn hình.
* *onPause():* được gọi khi hệ điều hành ưu tiên Activity khác được hiển thị. Ở thời điểm này, Activity bị dừng không xuất hiện trên màn hình.



Hình 2.3 Vòng đời của một Activity

**2.1.3.3 Intent**

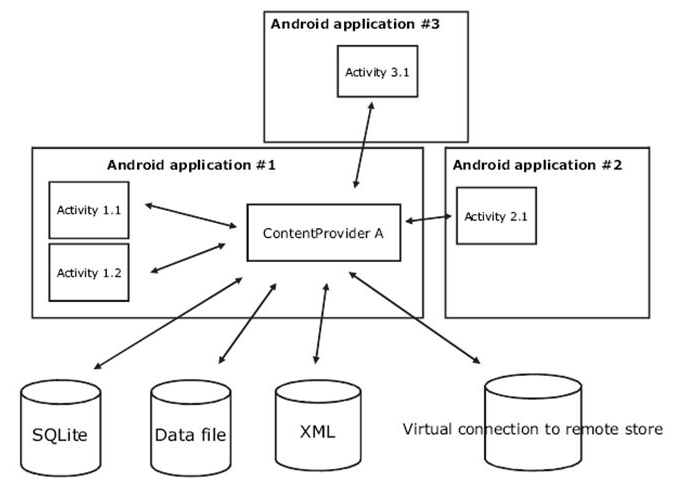
Được dùng để truyền tải các thông báo giữa các Activity hay Service để hiển thị giao diện người dùng hay còn dùng để điều khiển sự kiện. Ví dụ, khi mở một trang Web, Intent gửi một thông báo để tạo một Activity mới hiển thị trang Web. Đây là thành phần trung tâm của một ứng dụng Android.

**2.1.3.4 Services**

Một service là một component chạy bên dưới để thực thi một thao tác lâu và kéo dài, hoặc thực hiện một remote process. Một service không cung cấp giao diện. Ví dụ một service chơi nhạc ở bên dưới khi người dùng đang ở trong một ứng dụng khác, hoặc nó có thể lấy dữ liệu từ network mà không block tương tác của người dùng với một Activity. Một component khác như một Activity có thể khởi động service và để nó chạy hoặc gắn vào để tương tác với nó.

**2.1.3.5 Content Provider**

Một content provider quản lý tập hợp dữ liệu có thể chia sẻ giữa các ứng dụng. Bạn có thể lưu dữ liệu trên một file system, một SQLite database, trên Web hay trên bất kỳ persistent storage location nào mà bạn có thể truy xuất. Với content provider, những ứng dụng khác có thể truy vấn hoặc thay đổi dữ liệu (nếu content provider cho phép). Ví dụ, hệ thống Android với những quyền phù hợp có thể truy vấn những phần của content provider như contacts để đọc và ghi thông tin về một người nào đó. Content provider cũng hữu dụng khi đọc và ghi dữ liệu private với ưng sdungj và không được chia sẻ.



Hình 2.4: Các Activity truy xuất dữ liệu thông qua Content Provider

**2.1.3.6 Broadcast receivers**

Broadcast receiver là một component trả lời những broadcast announcements rộng khắp ứng dụng. Nhiều broadcast bắt nguồn từ hệ thống như broadcast thông báo rằng màn hình bị tắt, pin yếu, hay ảnh được chụp. Các ứng dụng có thể tạo broadcast, ví dụ để cho ứng dụng biết một số dữ liệu được download về thiết bị và có sẵn để sử dụng. Mặc dù broadcast receiver không hiện một giao diện nhưng chúng có thể tạo notification để hiển thị lên status bar, thông báo cho người dùng khi một broadcast xảy ra. Nói chung, broadcast receiver chỉ là một “gateway” đối với những component khác và làm những công việc nhỏ. Ví dụ như khởi tạo một dịch vụ để thực hiện một số công việc dựa trên sự kiện.

Một khía cạnh duy nhất của thiết kế hệ thống Android là bất kỳ ứng dụng nào cũng có thể start một component của ứng dụng khác. Ví dụ, nếu bạn muốn user chụp ảnh với camera của thiết bị, ứng dụng của bạn có thể dùng chức năng chụp ảnh của ứng dụng camera để làm điều đó, không cần phải kết hợp hay link đến code của ứng dụng camera. Đơn giản là start một Activity trong camera application để chụp ảnh, khi hoàn tất thì ảnh sẽ được trả về cho ứng dụng của bạn.

Khi hệ thông start component, nó sẽ start process dành cho ứng dụng đó (nếu nó chưa chạy) và thể hiện những lớp cần thiết cho component này. Ví dụ, nếu ứng dụng của bạn start activity trong ứng dụng camera để chụp ảnh, activity đó sẽ chạy đó sẽ chạy trong process thuộc về ứng dụng camera, không phải là process trong ứng dụng của bạn. Do đó, không giống những ứng dụng trên hầu hết hệ thống, ứng dụng Android không có điểm bắt đầu chính (nghĩa là kkhoong có hàm main()).

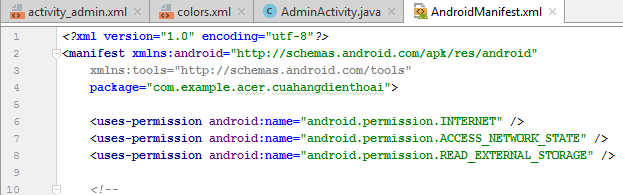
Vì hệ thống chạy mỗi ứng dụng trên một process riêng biệt với quyền truy xuất file giới hạn với ứng dụng khác nên ứng dụng của bạn không thể trực tiếp kích hoạt một component từ ứng dụng khác. Tuy nhiên, hệ thống Android có thể, đẻ kích hoạt một component trong ứng dụng khác, bạn phải gửi thông điệp cho hệ thống đó để chỉ định mục đích (intent) để start component đó. Sau đó hệ thống sẽ kích hoạt component đó cho bạn.

**2.1.3.7 The manifest file**

Trước khi hệ thống Android có thể start component của ứng dụng, hệ thống phải biết component đang tồn tại bằng cách đọc AndroidManifest.xml (manifest). Ứng dụng của bạn phải khai báo tất cả component trong file này.

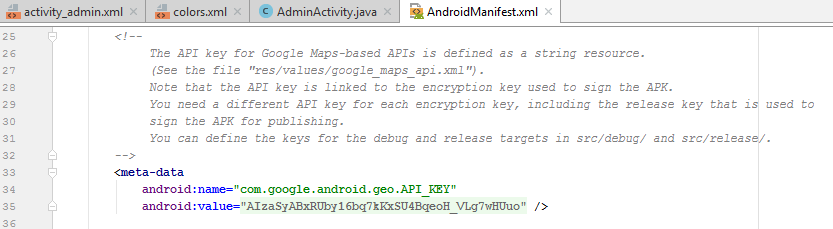
Ngoài việc khai báo component, manifest file làm một số việc như sau:

* Nhận dạng user permission mà ứng dụng yêu cầu, ví dụ như Internet Access hay quyền đọc đối với user’s contact



2.5 AndroidManifest.xml

* Khai báo các đặc tính software và hardware được yêu cầu bởi ứng dụng, ví dụ như camera, bluetooth hay multitouch.
* Khai báo API level tối thiểu được yêu cầu bởi ứng dụng, dựa trên APIs mà ứng dụng dùng.
* Những API libraries mà ứng dụng cần được liên kết đến, ví dụ như Google Maps library.



2.6 Google Maps – AndroidManifest.xml

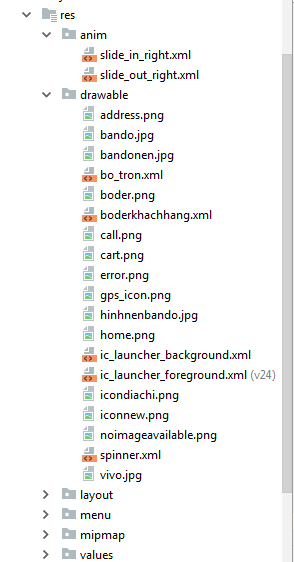
**2.1.3.8 Application resources**

Một ứng dụng Android bao gồm nhiều thứ, không chỉ có code, nó yêu cầu những tài nguyên tách biệt với source code, ví dụ như hình ảnh, audio file hay bất cứ thứ gì liên quan đến việc hiển thị của ứng dụng.

Đối với mỗi resource include trong project, SDK build tools sẽ định nghĩa một ID duy nhất để có thể tham chiếu đến resource đó trong code ứng dụng hoặc từ những resource khác được định nghĩa trong XML

Một trong những khía cạnh quan trọng của việc cung cấp resource tách biệt với code là bạn có thể cung cấp một resource thay thế cho những cấu hình thiết bị khác nhau.

Android hỗ trợ nhiều qualifiers cho những resource thay thế. Qualifier là một chuỗi ngắn được gắn trong tên của thư mục resources để xác định resource nào đượ dùng để phù hợp với cấu hình của thiết bị



2.7 Thư mục resource

**2.1.4 Google Maps**

Một trong những dịch vụ phổ biến nhất của Google, chỉ sau tìm kiếm, đó là Google Maps. Không có gì bất ngờ, hầu hết các thiết bị Android đều tích hợp sẵn Google Maps. Đối với lập trình viên Android, cách đơn giản nhất để tạo một bản đồ cho ứng dụng là tích hợp Google Maps vào trong ứng dụng Android.Việc tích hợp này tương đối đơn giản, chỉ cần viết một lớp thừa kế từ lớp MapActivity, thêm một số phương thức bắt buộc và lấy mã khóa API là ứng dụng của bạn đã có một bản đồ.

**2.1.5 Chu trình sống của một ứng dụng Android**

Android có cơ chế quản lý các tiến trình theo chế độ ưu tiên. Trong Android người dùng không thể tắt cá ứng dụng đi hoàn toàn (trừ ki dùng thêm ứng dụng hỗ trợ bên ngoài). Khi bọ nhớ bị chiếm dụng đến một ngưỡng nhất định, các tiến trình có độ ưu tiên thấp sẽ bị Android giải phóng mà không cảnh báo trước nhằm đảm bảo tài nguyên hệ thống. Cũng chính vì vậy mà biểu tượng của hệ điều hành Android là một con Robot, ý nói hệ điều hành sẽ tự động nhận biết và thực hiện công việc quản lý một cách hiệu quả nhất, còn người dùng chỉ cần sử dụng đơn thuần. Các tiến trình Android với độ ưu tiên từ cao đến thấp như sau:

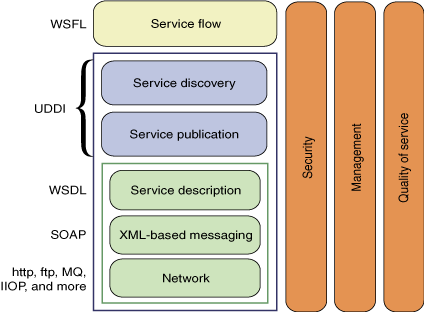
1. **Foreground Process:** là tiến trình của ứng dụng hiện thời đang được người dùng tương tác.
2. **Visible Process:** là tiến trình của ứng dụng mà activity đang hiển thị với người dùng.
3. **Service Process:** là tiến trình Service đang thực thi.
4. **Background Process:** là tiến trình của ứng dụng mà các activity của nó hiển thị với người dùng.
5. **Empty Process:** tiến trình không có bất cứ thành phần nào đang hoạt động.

**2.2 Web Service**

**2.2.1 Khái niệm**

Web Service (dịch vụ web) là một hiện thực của Service sử dụng Web Protocol (phương thức Web) để giao tiếp Web Service là công nghệ cho phép Client(máy khách) truy xuất để thực hiện mọi tác vụ như một ứng dụng web. Về bản chất, Web Service dựa trên XML và HTTP, trong đó XML làm nhiệm vụ mã hóa và giải mã dữ liệu và dùng SOAP để truyền tải. Web Service không phụ thuộc vào platform(nền tảng) nào, do đó bạn có thể dùng Web Service để truyền tải dữ liệu giữa các ứng dụng hay giữa các platform.

**2.2.2 Cấu trúc Web Server**



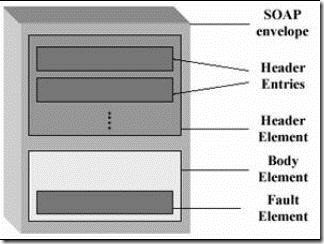
Hình 2.8: Cấu trúc một Web Service.

**2.2.3 SOAP – Simple Object Access Protocol**

SOAP - Simple Object Access Protocol (phương thức truy cập đối tường đơn giản), một tiêu chuẩn của W3C, là một phương thức giao tiếp dùng trong Web Service được xây dựng trên XML. SOAP được sử dụng để đặc tả và trao đổi thông tin và các cấu trúc dữ liệu cũng như các kiểu dữ liệu giữa các thành phần trong hệ thống.

Vì dựa trên XML nên SOAP là phương thức không phụ thuộc platform cũng như bất kỳ ngôn ngữ lập trình nào. Sử dụng SOAP, ứng dụng có thể yêu cầu thực thi phương thức trên máy tính ở xa mà không cần quan tâm đến chi tiết về platform cũng như các phần mềm trên máy tính đó.

Cấu trúc một thông điệp SOAp.



Hình 2.9: Cấu trúc một thông điệp SOAP

* **Envelop – Phần tử gốc:** Phần tử bao trùm nội dung message, khai báo văn bản XML như là một thông điệp SOAP.
* **Header – Phần tử đầu trang**: chưa các thông tin tiêu đề cho trang, phần tử không bắt buộc khai báo trong văn bản. Những đầu mục còn có thể mang những dữ liệu chứng thực, những chữ ký hóa và thông tin mã hóa, hoặc những cách đặt giao tác.
* **Body – Phần tử khai báo nội dung chính trong thông điệp**: Chứa các thông tin yêu cầu và phản hồi.
* **Fault –** **Phân tử phát sinh lỗi**: Cung cấp thông tin lỗi xảy ra trong quá trình xử lý thông điệp.

**2.2.4 WSDL – Web Service Description Language**

WSDL – Web Service Description Language (ngôn ngữ mô tả dịch vụ Web) là ngôn ngữ được sử dụng để mô tả đầy dủ về Web Service theo chuẩn XML như các phương thức, kiểu dữ liệu,… dựa trên XML Schema.

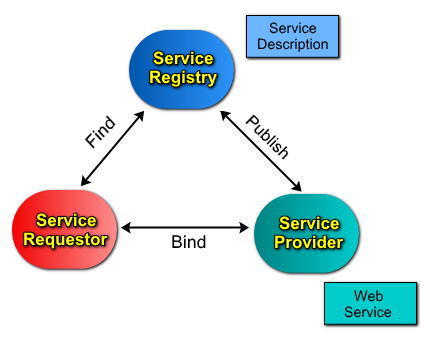
**2.2.5 UDDI – Universal Description Discover and Intergration**

UDDI – Universal Description Discover and Intergration được dùng để xác định các thông tin về các Web Service được cung cấp bởi các nhà cung cấp dịch vụ, cho phép Client truy tìm và nhận thông tin được yêu cầu khi sử dụng dịch vụ Web.

**2.2.6 Mô hình hoạt động của Web Service**

Mô hình hoạt động của Web Service bao gồm 3 phần:

* **Service Provider** (nhà cung cấp dịch vụ): xây dựng và cài đặt đối tượng dịch vụ lên một Web Service và cung cấp bản mô tả dịch vụ để hướng dẫn mọi người cách khai thác, ngôn ngữ sử dụng cho các bản mô tả dịch vụ với cấu trúc cú pháp của XML.
* **Service Requester** (nhà khai thác dịch vụ): xem xét các bản mô tả dịch vụ của nhà cung cấp, dựa vào đó xây dựng các lớp trung gian truy xuất đối tượng dịch vụ, thiết kế giao diện cho ứng dụng.
* **End-User** (người dùng cuối): truy cập trang web của nhà khai thác dịch vụ, sử dụng một cách gián tiếp dịch vụ web của các nhà cung cấp.



Hình 3.1: Mô hình giao tiếp giữa Nhà cung cấp dịch vụ và Nhà khai thức dịch vụ trong Web Service.

**2.3. PHP**

**2.3.1 Giới thiệu**

PHP (Hypertext Preprocessor) là ngôn ngữ lập trình mã nguồn mở chủ yếu được sử dụng để phát triển các ứng dụng web và có thể dễ dàng nhúng vào trang HTML. Vì được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp đơn giản nên PHP đã trở thành ngôn ngữ lập trình web phổ biến nhất hiện nay.

Ứng dụng web được viết bởi ngôn ngữ PHP có thể hoạt động trên hầu hết các loại máy chủ, các hệ điều hành (Windows, Linux…) kết hợp với nhiều hệ quản trị cơ sở dữ liệu khác nhau như MySQL, Microsoft SQL Server…

Ngôn ngữ PHP lấy dữ liệu từ cơ sở dữ liệu trên máy chủ và trả về mã HTML cho trình duyệt hiển thị. Vì vậy, ngôn ngữ PHP có tính linh hoạt cao, cụ thể là khi muốn thay đổi nội dung của trang web, người lập trình chỉ cần chỉnh sửa nội dung trong cơ sở dữ liệu mà không cần chỉnh sửa các đoạn mã HTML của giao diện web. Mặt khác, việc hiển thị một khối lượng lớn dữ liệu trên các trang web là điều không dễ dàng khi thực hiện bằng ngôn ngữ HTML vì người lập trình phải viết rất nhiều trang nhưng chỉ thay đổi nội dung, không thay đổi giao diện. Tuy nhiên, với việc kết hợp ngôn ngữ PHP trong lập trình web, việc hiển thị dữ liệu thay đổi theo nội dung sẽ được thực hiện bằng một vài lệnh hết sức đơn giản trong PHP.

Tuy đa số các trang web hiện nay đều được viết bằng ngôn ngữ PHP nhưng PHP còn được sử dụng để viết các chương trình chạy trên nền web. Ngày nay, với sự phát triển của mạng Internet, các doanh nghiệp thường lựa chọn những ứng dụng có thể hoạt động trên diện rộng, cập nhật thông tin dễ dàng, nhanh chóng. Do đó, các ứng dụng chạy trên nền web thường được các doanh nghiệp ưu tiên sử dụng. Ngôn ngữ PHP được tối ưu hóa cho các ứng dụng này giúp người lập trình tạo ra các ứng dụng với tốc độ thực thi nhanh, nhỏ gọn, tiện lợi.

**2.3.2 PHP cơ bản**

Khai báo:

* Để sử dụng ngôn ngữ PHP, người lập trình cần khai báo những đoạn mã nào là các câu lệnh PHP. Nếu không khai báo, trình duyệt sẽ hiểu các đoạn mã đó là mã HTML và cho hiển thị trực tiếp, không qua xử lý.
* Các câu lệnh PHP sẽ được đặt trong cặp dấu <? ?> hoặc <?php ?>.
* Lệnh echo trong PHP dùng để thể hiện giá trị của một biến hay hiện một chuỗi ra màn hình.
* Ví dụ: Câu lệnh <? echo “Hello PHP”; ?> sẽ cho kết quả hiển thị ra màn hình dòng chữ **Hello PHP**.

Biến:

* Biến dùng để chứa các giá trị trong quá trình tính toán.
* Khai báo: $TenBien=<giatri>;

Các phép toán:

* Số học: +, -, \*, /, % (chia lấy phần dư)
* So sánh: ==, >=, <=, >, <, !=
* Ghép chuỗi: .
* Logic: !, &&, ||
* Gán: =

Lệnh if:

if (điều kiện){

Khối lệnh 1;

}

else{

Khối lệnh 2;

}

Khối lệnh có thể là mã lệnh PHP hoặc mã lệnh HTML. Nếu là mã lệnh HTML cần đóng cặp dấu <? ?> trước khi viết mã lệnh HTML.

Vòng lặp for:

Là lệnh thực thi khối lệnh với số lần lặp nhất định.

for (<biến đếm>; <điều kiện dừng>; <tăng/giảm biến đếm>) {

Khối lệnh;

}

Vòng lặp while:

while (<điều kiện>) {

Khối lệnh;

}

Là lệnh yêu cầu thực thi khối lệnh với số lần không xác định. Điều kiện lặp được kiểm tra trước, do đó mã lệnh có thể không thực hiện lần lặp nào.

Vòng lặp do…while:

do {

Khối lệnh;

} while (<điều kiện>);

Là lệnh yêu cầu thực thi khối lệnh với số lần không xác định. Điều kiện lặp được kiểm tra sau, do đó mã lệnh luôn được thực hiện ít nhất một lần.

Include file:

* Lệnh include hoặc require dùng để nhúng mã lệnh PHP hoặc HTML từ trang khác vào trang hiện hành.
* Cú pháp:
* include “TenFile”; hoặc include\_once “TenFile;
* require “TenFile”; hoặc require\_once “TenFile;
* Lệnh include\_once hoặc require\_once được sử dụng để đảm bảo chỉ nhúng mã lệnh một lần trong trang hiện hành.
* Lệnh include sẽ bỏ qua các mã lệnh được nhúng vào nếu các mã lệnh đó không tồn tại hoặc có lỗi xảy ra. Mặt khác, lệnh require sẽ báo lỗi tại các câu lệnh đó và ngừng việc thực thi.

Truyền dữ liệu: Trong PHP, việc truyền dữ liệu từ trang này sang trang khác được thực hiện thông qua các đối tượng $\_GET và $\_POST:

* $\_GET: lấy dữ liệu từ đối tượng được truyền thông qua thanh địa chỉ. Khi truyền dữ liệu bởi $\_GET, dữ liệu được chứa trong các đối tượng sẽ được hiển thị hoàn toàn trên thanh địa chỉ của trình duyệt. Vì thế, $\_GET thường được dùng để truyền các dữ liệu công khai, không yêu cầu bảo mật.
* $\_POST: lấy dữ liệu từ đối tượng được truyền thông qua form có phương thức là POST. Các dữ liệu này sẽ được truyền ngầm thông qua các đối tượng được người dùng gõ vào hoặc do người lập trình khai báo trong form. Những dữ liệu này thường yêu cầu tính bảo mật như tên đăng nhập, mật khẩu…

Cookie:

* Cookie là biến chứa dữ liệu được lưu trữ trên máy của người dùng. Nếu không chỉ định thời gian khi tạo cookie thì cookie sẽ được lưu trong bộ nhớ và sẽ mất đi khi người dùng đóng trình duyệt.
* Cần tạo cookie trước khi sử dụng qua cú pháp:

setcookie(“TenCookie”, giá trị, [, Thời điểm hết hạn]);

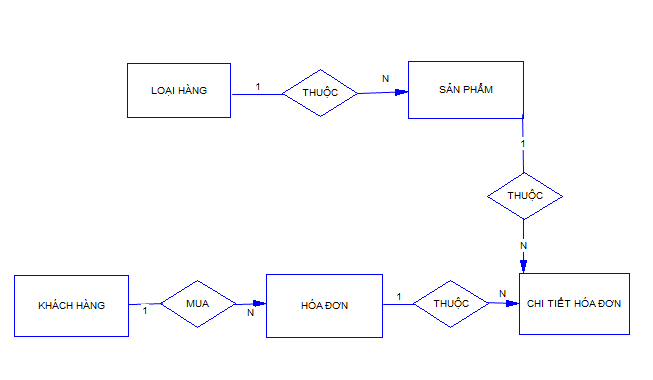
Session:

* Biến SESSION được lưu trên máy chủ và lưu theo các phiên làm việc. Điều đó có nghĩa là khi người dùng đóng trình duyệt, dữ liệu trong biến SESSION sẽ bị mất.
* Trước khi sử dụng biến SESSION phải khai báo lệnh <? session\_start(); ?> ở dòng đầu tiên của trang đó. Cú pháp để lấy dữ liệu trong biến SESSION: $\_SESSION[‘Ten’].

**Chương 3: Cơ sở dữ liệu**

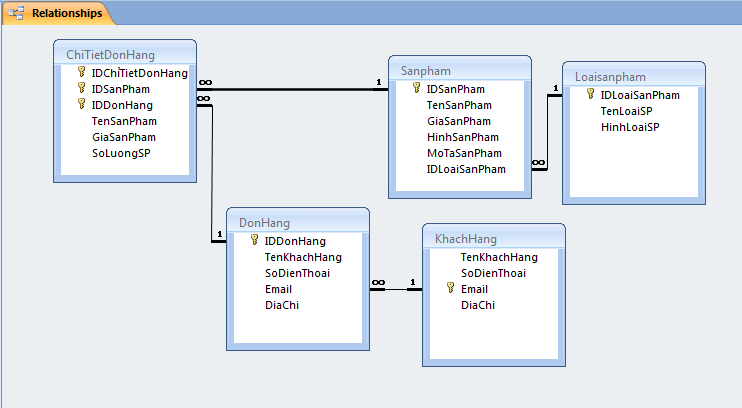
**3.1 Cơ sở dữ liệu của ứng dụng**

**3.1.1 Mô hình thực thể kết hợp**



Hình 3.2: Mô hình thực thể kết hợp của CSDL bán hàng

**3.1.2 Mô hình quan hệ**

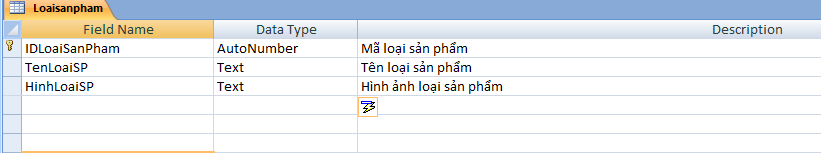


Hình 3.3: Mô hình quan hệ của CSDL bán hàng

**3.1.3 Bảng dữ liệu**

Các bảng được sử dụng trong cơ sở dữ liệu:

* Bảng loại sản phẩm:



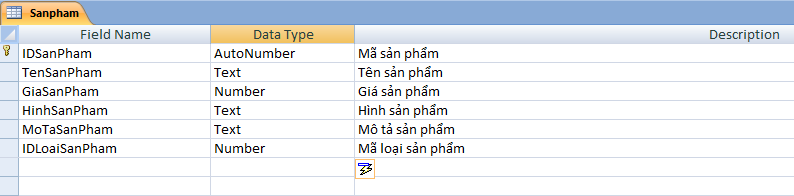
Ý nghĩa các trường:

IDLoaiSanPham: là khóa chính, chỉ ra mã của loại hàng đó, mã tự động tăng.

TenLoaiSP: tên của loại sản phẩm

HinhLoaiSP: đường dẫn đến hình ảnh của loại sản phẩm

* Bảng sản phẩm



Ý nghĩa các trường:

IDSanPham: là khóa chính, chỉ ra mã của sản phẩm, mã tự động tăng.

TenSanPham: tên của mỗi sản phẩm.

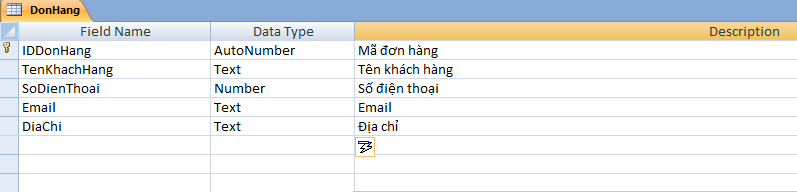
GiaSanPham: giá của sản phẩm.

HinhSanPham: đường dẫn đến hình ảnh của sản phẩm.

MoTaSanPham: mô tả chi tiết sản phẩm.

IDLoaiSanPham: là khóa ngoại, mã loại hàng tham chiếu đến bảng loại sản phẩm.

* Bảng đơn hàng



Ý nghĩa các trường:

IDDonHang: là khóa chính, chỉ ra mã của đơn hàng, mã tự động tăng.

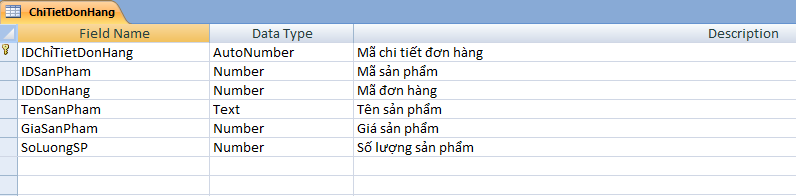
TenKhachHang: tên của khách hàng.

SoDienThoai: số điện thoại của khách hàng.

Email: là khóa ngoại, email của khách hàng tham chiếu đến bảng khách hàng.

DiaChi: địa chỉ của khách hàng.

* Bảng chi tiết đơn hàng



Ý nghĩa các trường:

IDChiTietDonHang: là khóa chính chỉ ra mã của cho tiết đơn hàng, mã tự động tăng.

IDSanPham: là khóa ngoại, mã sản phẩm tham chiếu đến bảng sản phẩm.

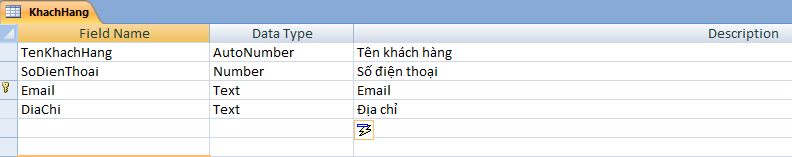
IDDonHang: là khóa ngoại, mã đơn hàng tham chiếu đến bảng đơn hàng.

TenSanPham: tên của sản phẩm.

GiaSanPham: giá của sản phẩm.

SoLuongSP: số lượng của sản phẩm.

* Bảng khách hàng



Ý nghĩa các trường:

TenKhachHang: tên của khách hàng.

SoDienThoai: số điện thoại của khách hàng.

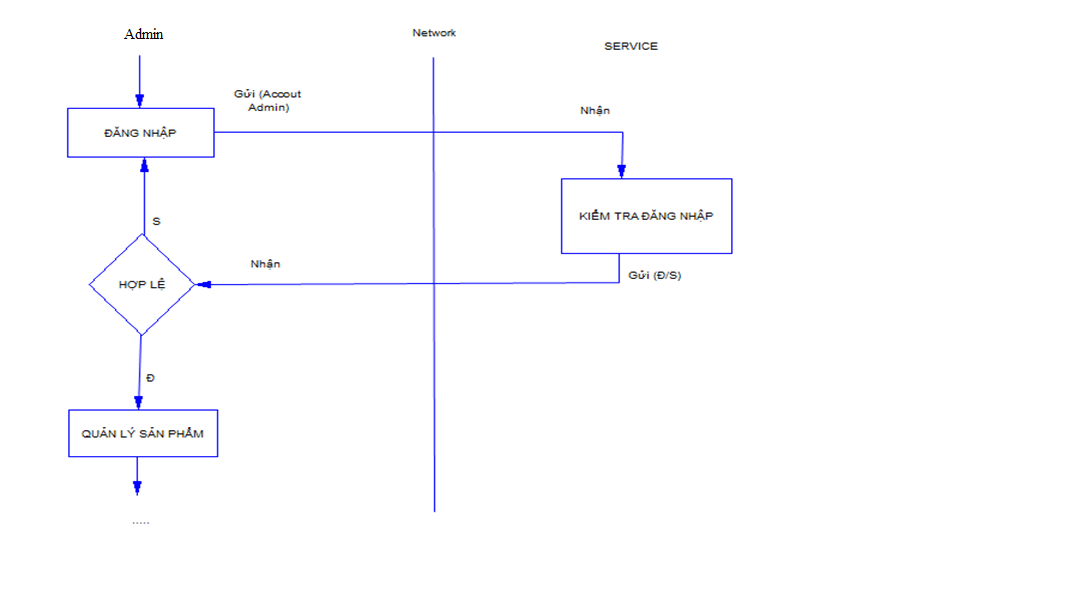
Email: là khóa chính, chỉ ra email của khách hàng.

DiaChi: địa chỉ của khách hàng.

**3.2 Xây dựng các chức năng trong ứng dụng bán hàng**

**3.2.1 Chức năng đăng nhập**

**3.2.1.1 Mô hình hoạt động**

****

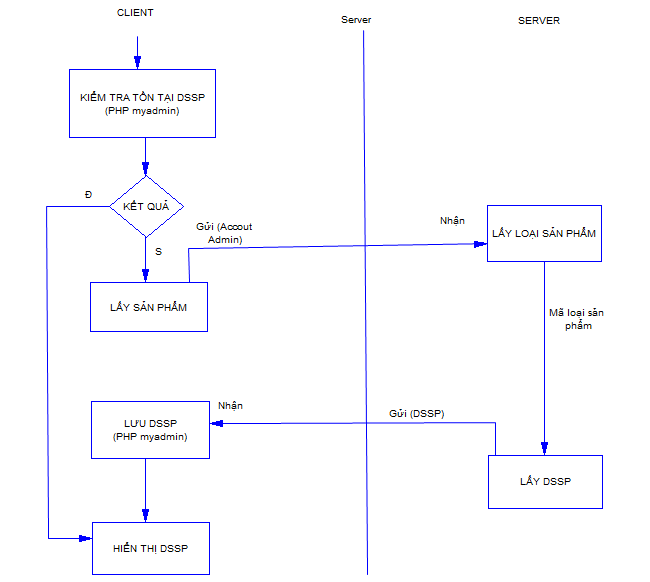
Hình 3.4: Sơ đồ hoạt động của trang đăng nhập.

**3.2.1.2 Hiện thực**

Sau khi đã chuẩn bị một cơ sở dữ liệu như đã trình bày ở mục 3.1, bây giờ chúng ta sẽ xây dựng chức năng đăng nhập của admin để admin có thể đăng nhập vào hệ thống. Chức năng đăng nhập dựa trên nguyên tắc yêu cầu – trả lời. Phía Server ta cần hàm kiểm tra đăng nhập nhận vào hai biến là tài khoản và mật khẩu của admin, nếu kết quả kiểm tra hợp lệ sẽ gửi trả biến True về Client và ngược lại là biến False. Phía Client gọi hàm đăng nhập và hàm đợi kết quả từ phía Server, nếu kết quả hợp lệ ta sẽ tiến hành thực hiện chức năng quản lý sản phẩm, nếu không hợp lệ thì sẽ chuyển về trang đăng nhập từ đầu.

**3.2.2 Chức năng hiển thị danh sách sản phẩm lên thiết bị**

**3.2.2.1 Mô hình hoạt động**

****

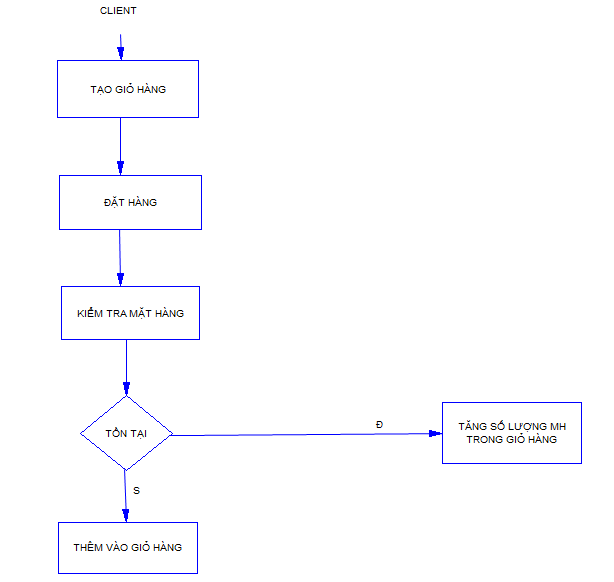
Hình 3.4: Mô hình hoạt động của trang sản phẩm.

**3.2.2.2 Hiện thực**

Để thực hiện hiển thị danh sách sản phẩm lên thiết bị di động, ta cần thực hiện bước tải danh sách sản phẩm từ phía server, và hiển thị danh sách đó lên màn hình. Phương thức tổng hợp danh sách sản phẩm và gửi về Client của Server sẽ thực hiện kết nối cơ sở dữ liệu.Khi có các loại hàng thì Server sẽ thực hiện truy vấn lấy danh sách sản phẩm dựa vào mã loại hàng để gửi về phía Client.

**3.2.3 Chức năng đặt hàng**

**3.2.3.1 Mô hình hoạt động**

****

Hình 3.5: Mô hình hoạt động của chức năng đạt hàng

**3.3.2 Hiện thực**

Để thực hiện chức năng đặt hàng, khách hàng chọn sản phẩm trong danh sách sản phẩm cho vào giỏ hàng, nếu sản phẩm đó có trong giỏ hàng rồi thì số lượng sản phẩm đó tăng lên, nếu chưa thì sản phẩm sẽ được đưa vào giỏ với số lượng chọn ban đầu.

**Chương 4: Xây dựng ứng dụng**

**4.1 Giới thiệu app bán hàng điện tử**

App có tên ***Holy Store***, là app bán thiết bị điện tử với giao diện thân thiện, đẹp mắt, đơn giản hóa mọi chi tiết để dễ tiếp cận với người dùng.

icon.PNG

Hình 3.6: Icon của app

Các chức năng chính của app gồm:

Admin:

* Đăng nhập
* Xem đơn hàng
* Xem chi tiết đơn hàng
* Thêm sản phẩm

User:

* Xem sản phẩm mới nhất
* Xem sản phẩm theo loại
* Xem chi tiết sản phẩm
* Có giỏ hàng chứa sản phẩm của khách hàng khi chọn mua
* Điền vào phiếu thông tin khách hàng khi xác nhận thanh toán

**4.2 Các trang của app**

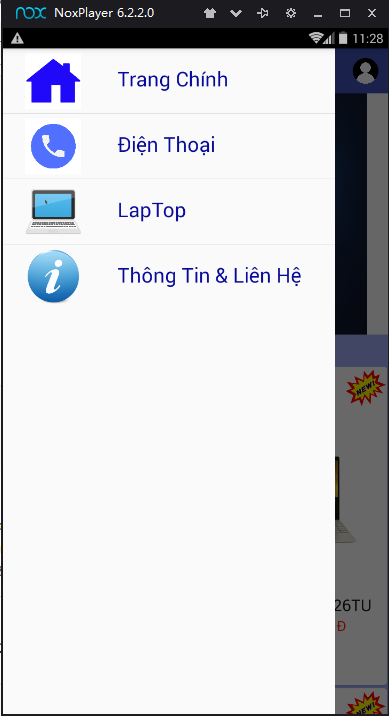
**4.2.1 Giao diện người dùng**

**4.2.1.1. Trang chính**



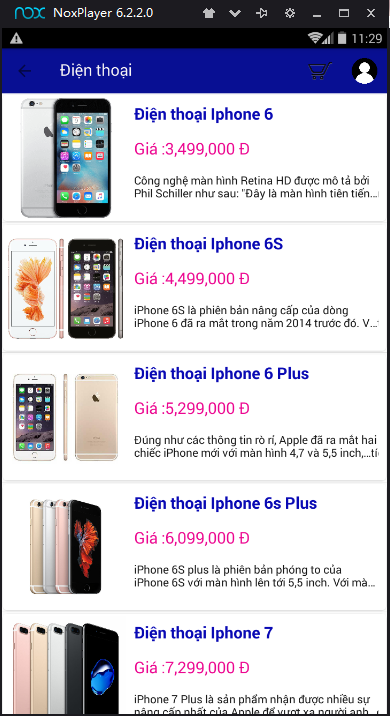
Hình 3.7: Giao diện trang chính người dùng

**4.2.1.2. Menu**



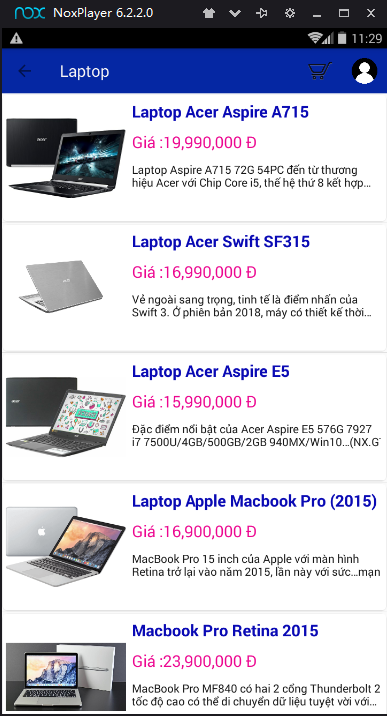
Hình 3.8: Giao diện menu người dùng

**4.2.1.3. Trang điện thoại**



Hình 3.9: Giao diện hiển thị danh sách điện thoại

**4.2.1.4. Trang laptop**



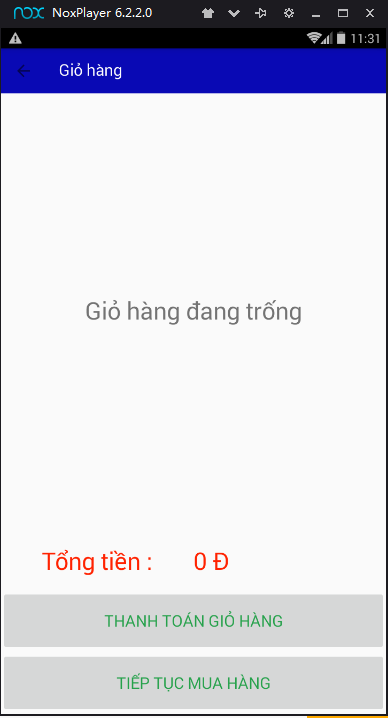
Hình 4.1: Giao diện hiển thị danh sách laptop

**4.2.1.5. Trang chi tiết sản phẩm**

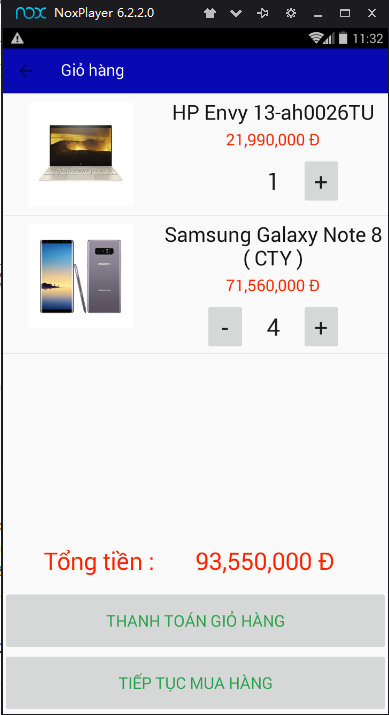


Hình 4.2: Giao diện chi tiết sản phẩm

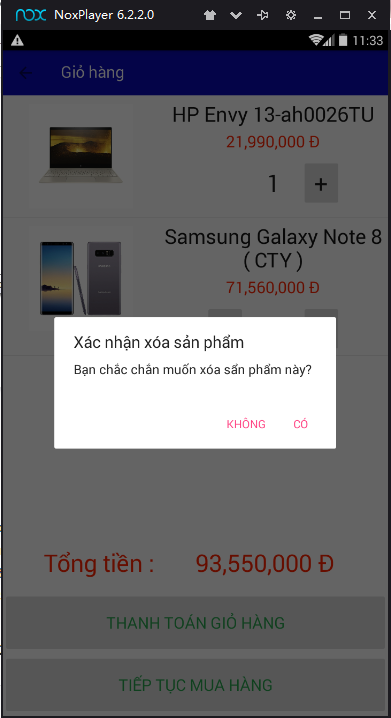
**4.2.1.6. Trang giỏ hàng**



Hình 4.3: Giao diện giỏ hàng đang trống

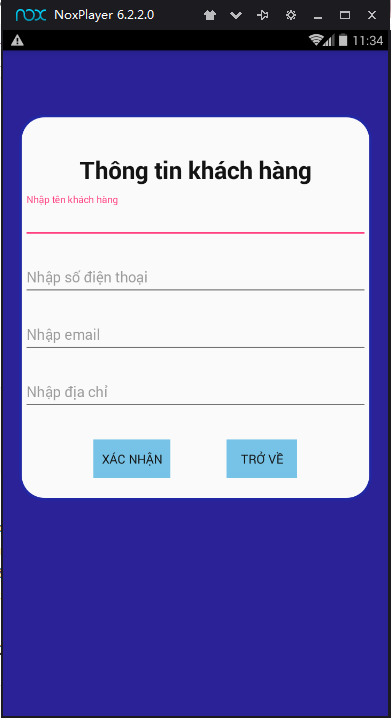


Hình 4.4: Giao diện giỏ hàng có hàng



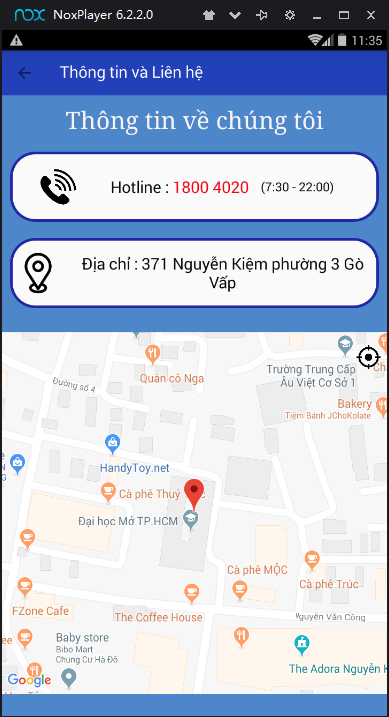
Hình 4.5: Giao diện xóa sản phẩm trong giỏ hàng

**4.2.1.7 Trang giao diện điền thông tin khách hàng**



Hình 4.6: Giao diện điền thông tin khách hàng

**4.2.1.8 Giao diện thông tin liên hệ**



Hình 4.7 Giao diện thông tin liên hệ

**4.2.2 Giao diện quản trị**

**4.2.2.1 Giao diện đăng nhập admin**



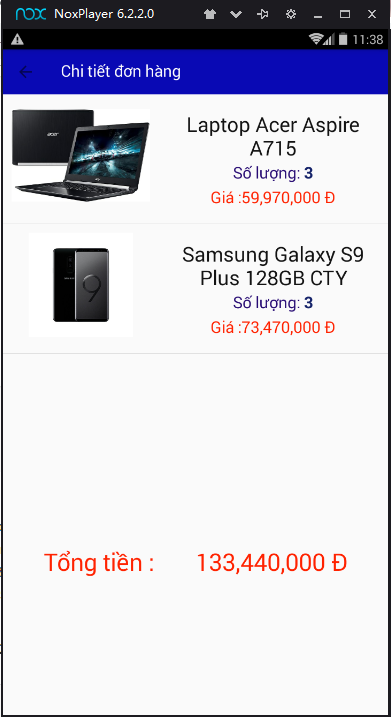
Hình 4.8: Giao diện đăng nhập Admin

**4.2.2.2 Giao diện xem đơn hàng**



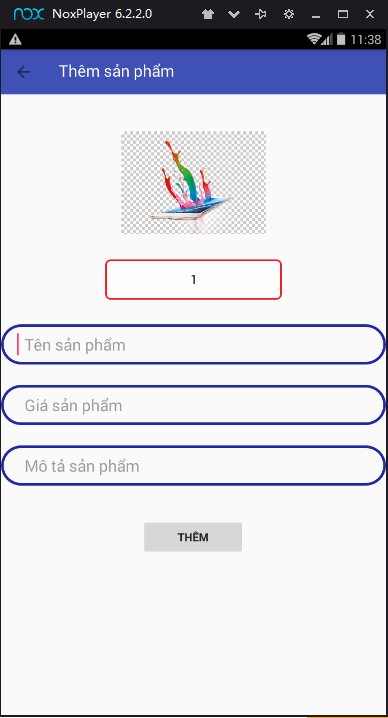
Hình 4.9 Giao diện xem đơn hàng

**4.2.2.3 Giao diện chi tiết đơn hàng**



Hình 5.1 Giao diện hiển thị chi tiết đơn hàng

**4.2.2.4 Giao diện thêm sản phẩm**



Hình 5.2 Giao diện thêm sản phẩm

**Chương 5: Tổng kết**

* 1. **Ưu điểm**

Ứng dụng giúp công việc bán hàng tại các cửa hàng điện tử trở nên hiện đại, tăng năng xuất làm việc. Giúp khâu bán hàng và thanh toán trở trên nhanh chóng.

* 1. **Hạn chế**

Ứng dụng chỉ chạy được trên thiết bị Android, chưa hiện thực trên các nền hệ điều hành khác.

Vì thời gian hạn hẹp để thực hiện nên việc sắp xếp các trang của ứng dụng chưa được hợp lý, quản lý cơ sở dữ liệu chưa được tối ưu, các chức năng chưa chặc chẽ,..

* 1. **Hướng phát triển**

Từ những hạn chế của đề tài, em sẽ phát triển thành một ứng dụng quản lý chuyên dụng, với các chức năng phức tạp.

Thực hiện nghiên cứu trên các cửa hàng để mang lại sự thoải mái khi sử dụng ứng dụng.

**TÀI LIỆU THAM KHẢO**

1. Android Deverloper, <https://developer.android.com/>
2. Stack Overflow, <https://stackoverflow.com/>
3. PHP Manual, <http://php.net>
4. Retrofit trong Android, <https://teamvietdev.com/tim-hieu-retrofit-trong-android/>
5. W3Schools, <https://www.w3schools.com/graphics/google_maps_intro.asp>
6. API Google Map, <https://www.wpgmaps.com/documentation/creating-a-google-maps-api-key/>