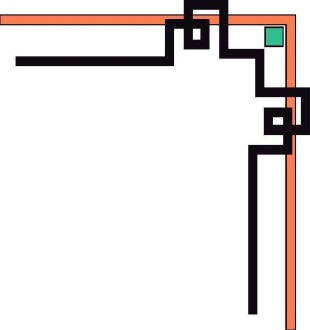
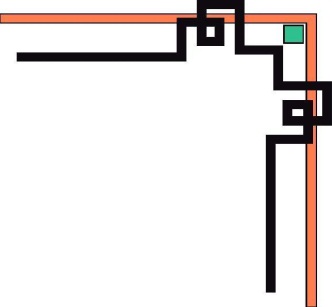
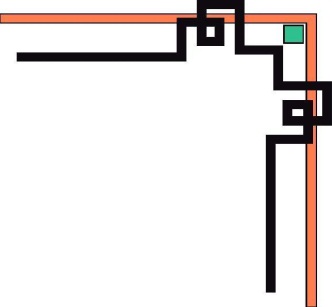
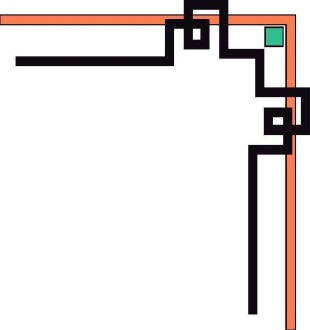
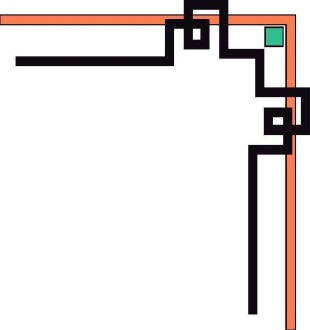
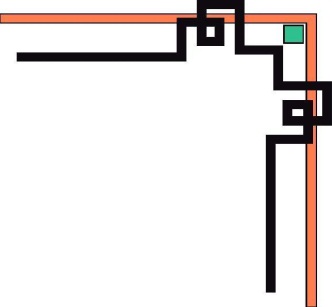
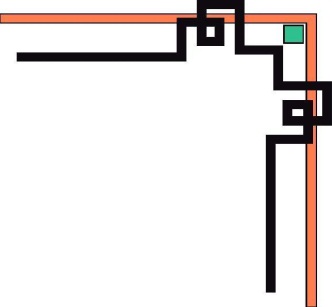
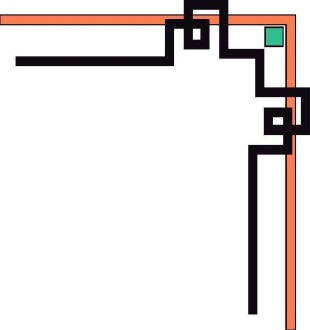
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**HỌC VIỆN QUẢN LÝ GIÁO DỤC**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



**BÁO CÁO BÀI TẬP LỚN**

**Môn:**

**Chủ đề:**

**Sinh viên:**

**Mã số sinh viên:**

**Ngành:**

**Lớp tín chỉ:**

**Hà Nội – 2023**

# LỜI MỞ ĐẦU

Ngày nay, chúng ta đang chứng kiến sự thay đổi lớn lao, kỳ diệu do khoa học và công nghệ mang lại. Cùng với sự phát triển của ngành khoa học kỹ thuật, công nghệ thông tin trên toàn cầu, tin học ngày càng thâm nhập vào đời sống xã hội & con người.Việc ứng dụng tin học vào các ngành sản xuất, khoa học, kỹ thuật cho tới quản lý, sinh hoạt của con người không còn điều mới lạ nữa, mà trở thành công cụ đắc lực cho hoạt động của con người và trong đời sống xã hội. Có thể thấy rằng thời đại hiện nay là thời đại công nghệ thông tin.Sự chính xác cao, khả năng lưu trữ thông tin lớn, tốc độ xử lý thông tin nhanh đã mở ra nhiều ứng dụng vào các ngành nghề, lĩnh vực. Qua đó, việc ứng dụng tin học trong quản lý là một ví dụ điển hình.Với sự phát triển công nghệ thông tin hiện nay của nước ta, hàng loạt các phần mềm bán hàng được ra đời với mục đích trợ giúp chủ doanh nghiệp trong việc kinh doanh online của các doanh nghiệp dễ dàng hơn. Qua đó, các thao tác và mua bán thông qua phần mềm bán hàng online được nhanh, vượt trội và tiện lợi hơn so với mua bán hàng trực tiếp.

Trong phạm vi giới hạn của một chuyên đề nghiên cứu bài tập lớn, chúng em đã nghiên cứu hoạt động bán hàng online thông qua các trang mạng. Hầu hết việc bán hàng đều thông qua phần mềm bán hàng để trợ giúp chủ doanh nghiệp quá trình bán hàng trực tuyến, đáp ứng được đầu ra các thông tin một cách chính xác và tốn ít thời gian nhất có thể. Bên cạnh đó, hệ thống vẫn còn tồn tại những điểm yếu chưa được khắc phục.

Qua việc nghiên cứu, học tập cũng như trao dồi kiến thức của bản thân. Chúng em quyết định tạo và phát triển “ Phần mềm Quản Lý Bán Hàng Điện Thoại” và chọn làm đề tài cho bài tập lớn. Chúng em xin cảm ơn giảng viên ThS Phạm Hùng– người đã trực tiếp giảng dạy và hướng dẫn tận tình cho đề tài chúng em được hoàn thành đúng tiến độ và kế hoạch.

Chúng em xin chân thành cảm ơn các thầy cô, các bạn và các diễn đàn đã giúp đỡ em trong quá trình hoàn thành bài tập lớn lập trình Mobile.

1. Giới thiệu vấn đề nghiên cứu

* Lí do chọn đề tài nghiên cứu

Công nghệ đang thay đổi mọi mặt của cuộc sống, từ các sinh hoạt thường ngày cho đến việc vận hành kinh doanh. Chẳng thế mà các cửa hàng bán lẻ vừa và nhỏ, vốn hay quản lý theo cách truyền thống dùng sổ sách đã dần chuyển sang dùng các trang thương mại điện tử hay phần mềm bán hàng online không chỉ bởi tiện lợi cho khách hàng mua hàng hơn mà còn một số tiện ích khác.

Phần mềm bán hàng online đã và đang trở thành một phần tất yếu của các shop, cửa hàng. Do vừa mang lại sự hiệu quả trong việc kinh doanh bán các mặt hàng mà còn là xu thế hiện đại ngày nay khi con người dần chuyển sang đặt mua các mặt hàng mình thích thông qua chiếc điện thoại. Nên chúng em quyết định chọn đề tài “Phần mềm bán hàng điện thoại Online” để giúp các khách hàng có tiện lợi đặt mua những chiếc điện thoại mới yêu thích để thay thông qua điện thoại.

* Lịch sử nghiên cứu

Hiện nay đa số các App bán hàng đều hợp lý hoá việc thu thập và lưu trữ dữ liệu an toàn bằng hệ thống quản lý cơ sở dữ liệu, hỗ trợ trên nhiều nền tảng .v.v.v như:

* Sàn thương mại điện tử Shoppe
* Các sàn thương mại điện tử có hỗ trợ trên nền tảng Mobile khác như: Lazada, Tiki, Sendo.
* Mục đích của đề tài

Từ công tác nghiệp vụ liên quan tới hoạt động bán hàng, đề tài sẽ tiến hành phân tích và thiết kế một hệ thống quản lý nhằm tin học hóa các chức năng có thể được thực hiện trên điện thoại. Từ đó, xây dựng một chương trình ứng dụng hỗ trợ cho quá trình thực hiện các chức năng như quản lý thông tin, xử lý bán hàng online và báo cáo thống kê định kì.

Như vậy, mục tiêu cuối cùng là xây dựng một phần mềm quản lý bán hàng tối ưu và dễ sử dụng. Đáp ứng nhu cầu xử lý các chức năng nghiệp vụ trong quá trình thực hiện các hoạt động quản lý.

* Đối tượng nghiên cứu đề tài
* *Khách thể nghiên cứu:*

Xây dựng hệ thống bán hàng online trên Android Studio với ngôn ngữ Java.

* *Đối tượng nghiên cứu:*

Các phần mềm quản lý bán hàng mã nguồn mở.

* Phạm vi nghiên cứu đề tài

Xây dựng App bán Điện Thoại cho doanh nghiệp vừa và nhỏ.

* Luận điểm nghiên cứu

Làm thế nào để xây dựng App bán hàng Điện Thoại doanh nghiệp nhỏ?

* Phương pháp nghiên cứu
* **Phương pháp lí thuyết**
* Ngôn ngữ Java
* FireBase
* Phân tích thiết kế hướng đối tượng
* **Phương pháp quan sát nghiên cứu**
* Quan sát thực tiễn các hoạt động của các sàn thương mại điện tử.
* Môi trường nghiên cứu

Môi trường thực hiện nghiên cứu là hệ điều hành Android.

1. Cấu trúc của bài tập lớn

Đề tài bao gồm:

**Lời mở đầu**

**Phần I: Tổng quan về đề tài**

**A: Nhiệm vụ của đề tài**

**B: Cách tiếp cận và hướng giải quyết**

**PHẦN II: Phân tích chức năng**

**A: Xác định tác nhân và thiết kế cơ sở dữ liệu**

**B: Phân tích chức năng hệ thống**

**C: Giao diện chương trình Phân mềm quản lý nhân sự và Sơ đồ luồng dữ liệu DFD**

**Kết luận và Hướng phát triển**

**Tài liệu tham khảo**

# PHẦN I: TỔNG QUAN VỀ ĐỀ TÀI

1. Nhiệm vụ của đề tài

Áp dụng kiến thức về ngôn ngữ lập trình Java, cơ sở dữ liệu và phân tích thiết kế hướng đối tượng để xây dựng App Quản Lý Bán hàng tự động thực hiện các chức năng một số công việc bằng điện thoại có thể thay thế một phần công việc cho con người như:

* *Xây dựng đăng nhập, đăng kí cho người dùng.*
* *Xây dựng chức năng tìm kiếm sản phẩm(điện thoại) và danh mục sản phẩm cho người khách hàng.*
* *Xây dựng chức năng mua hàng,xem giỏ hàng, thanh toán, đánh giá, xem lịch sử mua cho người khách hàng.*
* *Xây dựng phân quyền, quản lý người dùng cho Admin.*
* *Xây dựng quản lý tài khoản cho người dùng.*
* *Xây dựng chức năng quản lý sản phẩm(điện thoại), quản lý đơn hàng, quản lý thông báo, thống kê cho Nhân Viên.*
* *Xây dựng giao diện thân thiện và dễ sử dụng.*

1. Cơ sở lý thuyết và công cụ giải quyết.
   1. **Các khái niệm cơ bản.**
      1. **Khái niệm phụ thuộc dữ liệu và dạng chuẩn.**

Một thuộc tính gọi là phụ thuộc vào các thuộc tính khác khi giá trị của thuộc tính này phụ thuộc vào giá trị của thuộc tính kia. Sự phụ thuộc này có thể là gián tiếp hay trực tiếp.

Một quan hệ bao giờ cũng có một nhóm thuộc tính mà giá trị của chúng quy định giá trị của các thuộc tính khác, nhóm thuộc tính đó gọi là khóa.

Với một quan hệ, tùy vào các phụ thuộc của các thuộc tính vào khóa có trong đó mà ta phân chia các quan hệ đó thành các dạng chuẩn khác nhau. Các dạng chuẩn cơ bản:

* Dạng chuẩn 1NF: Một lược đồ quan hệ R được gọi là ở dạng chuẩn 1NF khi và chỉ khi toàn bộ các miền có mặt trong R đều chỉ chứa các giá trị nguyên tố.
* Dạng chuẩn 2NF: Nếu nó ở dạng chuẩn 1NF và nếu mỗi thuộc tính không khóa của R là phụ thuộc hàm đầy đủ vào khóa chính.
* Dạng chuẩn 3NF: Nếu nó là 2NF và mỗi thuộc tính không khóa của R là không phụ thuộc hàm bắc cầu vào khóa chính.

Các dữ liệu lưu trữ dưới dạng chuẩn 3NF tránh được hiện tượng dư thừa dữ liệu, tạo cho dữ liệu có tính độc lập cao. Các quan hệ nếu chưa ở dạng chuẩn 3NF sẽ được phân rã thành các quan hệ nhỏ hơn có dạng chuẩn 3NF.

Phụ thuộc hàm: Cho R(U) là một lược đồ quan hệ với U = {A1, A2,..., An} là một tập thuộc tính. X và Y là tập con của U. Nói rằng X -> Y (đọc là X xác định Y hoặc Y phụ thuộc hàm vào X) nếu r là một quan hệ xác định trên R(U) sao cho bất kỳ hai bộ t1, t2 thuộc r mà: t1[X] = t2[X] thì t1[Y] = t2[Y].

* + 1. **Khái niệm chỉ dẫn và khóa chỉ dẫn**

Để có thể tìm kiếm thông tin nhanh theo một tiêu chuẩn nào đó chúng ta tạo ra các thông tin chỉ dẫn theo tiêu chuẩn đó. Các thông tin này gọi là khóa chỉ dẫn. Khóa chỉ dẫn có thể là một trường hoặc nhiều trường trong trường hợp này phải chỉ ra thứ tự. Với cách tạo ra khóa theo tiêu chuẩn nào đó ta có thể tìm kiếm dữ liệu nhanh theo tiêu chuẩn đó.

* + 1. **Biểu đồ Use Case**

Một biểu đồ Use Case chỉ ra một số lượng các tác nhân ngoại cảnh và mối liên kết của chúng đối với Use Case mà hệ thống cung cấp. Một Use Case là một lời miêu tả của một chức năng mà hệ thống cung cấp. Lời miêu tả Use Case thường là một văn bản tài liệu, nhưng kèm theo đó cũng có thể là một biểu đồ hoạt động. Các Use Case được miêu tả duy nhất theo hướng nhìn từ ngoài vào của các tác nhân (hành vi của hệ thống theo như sự mong đợi của người sử dụng); không miêu tả chức năng được cung cấp sẽ hoạt động nội bộ bên trong hệ thống ra sao. Các Use Case định nghĩa các yêu cầu về mặt chức năng đối với hệ thống.

* + 1. **Biểu đồ hoạt động**

Một biểu đồ hoạt động chỉ ra một trình tự lần lượt của các hoạt động. Biểu đồ hoạt động thường được sử dụng để miêu tả các hoạt động được thực hiện trong một thủ tục, mặc dù nó cũng có thể được sử dụng để miêu tả các dòng chảy hoạt động khác, ví dụ như trong một Use Case hay trong một trình tự tương tác. Biểu đồ hoạt động bao gồm các trạng thái hành động, chứa đặc tả của một hoạt động cần phải được thực hiện. Một trạng thái hành động sẽ qua đi khi hành động được thực hiện xong (khác với biểu đồ trạng thái: Một trạng thái chỉ chuyển sang trạng thái khác sau khi đã xảy ra một sự kiện rõ ràng!). Dòng điều khiển ở đây chạy giữa các trạng thái hành động liên kết với nhau. Biểu đồ còn có thể chỉ ra các quyết định, các điều kiện, cũng như phần thực thi song song của các trạng thái hành động. Biểu đồ ngoài ra còn có thể chứa các loại đặc tả cho các thông điệp được gửi đi hoặc được nhận về, trong tư cách là thành phần của hành động được thực hiện.

* + 1. **Phần tử mô hình**

Các khái niệm được sử dụng trong các biểu đồ được gọi là các phần tử mô hình. Một phần tử mô hình được định nghĩa với ngữ nghĩa, đó là một định nghĩa về bản chất phần tử hay là một xác định ý nghĩa chính xác xem nó sẽ thể hiện điều gì trong những lời khẳng định rõ ràng. Mỗi phần tử mô hình còn có sự miêu tả trực quan, một ký hiệu hình học được sử dụng để miêu tả phần tử này trong biểu đồ. Một phần tử có thể tồn tại trong nhiều dạng biểu đồ khác nhau, nhưng cũng có những nguyên tắc xác định loại phần tử nào có thể được chỉ ra trong loại biểu đồ nào. Một vài ví dụ cho phần tử mô hình là lớp, đối tượng, trạng thái, nút mạng, gói, thành phần.

Một vài loại quan hệ đáng chú ý:

* Kết nối: nối các phần tử và các thực thể nối
* Khái quát hóa: còn được gọi là tính kế thừa kế, có ý nghĩa rằng một phần tử này có thể là một sự chuyên biệt hóa của một phần tử khác
* Sự phụ thuộc: chỉ ra rằng một phần tử này phụ thuộc trong một phương thức nào đó vào một phần tử khác.
* Kết tập: một dạng của nối kết, trong đó một phần tử này chứa các phần tử khác.

Ngoài ra còn có các phần tử mô hình khác như thông điệp, hành động và khuôn mẫu.

* 1. **Tổng quan về công cụ giải quyết**
     1. **Firebase:**

Firebase là một nền tảng phát triển ứng dụng web và di động được cung cấp bởi Google. Firebase cung cấp các dịch vụ đám mây giúp các nhà phát triển xây dựng, triển khai và quản lý ứng dụng dễ dàng và nhanh chóng.

Firebase cung cấp một loạt các tính năng và dịch vụ như cơ sở dữ liệu thời gian thực, xác thực người dùng, lưu trữ đám mây, máy chủ hàm, tin nhắn đám mây, phân tích và nhiều hơn nữa. Điều này cho phép nhà phát triển tập trung vào việc phát triển các tính năng chính của ứng dụng mà không cần phải quan tâm đến việc phát triển và quản lý hạ tầng.

Firebase được tích hợp một cách dễ dàng với nhiều công nghệ khác nhau như React, Angular, Vue, iOS, Android và các nền tảng web khác. Firebase cũng cung cấp cho nhà phát triển các công cụ phát triển và quản lý ứng dụng dễ sử dụng như Firebase Console, cùng với các tài liệu chi tiết và hỗ trợ tốt.

Tóm lại, Firebase là một nền tảng đám mây hoàn chỉnh, dễ sử dụng và mạnh mẽ cho việc phát triển ứng dụng web và di động.Tiếp theo sau đây, chúng ta hãy cùng xem Firebase có những chức năng cơ bản gì.

* **Một số chức năng chính của Firebase**
* **Firebase Analytics**

Bằng Firebase Analytics, ta có thể phân tích sự tương tác của người dùng với ứng dụng cũng như tình trạng sử dụng ứng dụng đó.

Firebase có sẵn chức năng tạo report. Nhìn report ấy, chúng ta có thể nắm được một cách rõ ràng hoạt động của người dùng.

Bằng việc cài đặt SDK (Software Development Kit, cụ thể hơn với trường hợp này là FirebaseAnalytics.unitypackage), chức năng phân tích các access trở nên khả dụng. Khi đó, ta không chỉ xem được hoạt động của người dùng mà còn có thể biết được thông tin về thuộc tính hoạt động hay hiệu quả quảng cáo, tình trạng trả phí, v.v.

* + **Firebase Hosting**

Firebase Hosting là dịch vụ deploy trang web và web app chỉ bằng những thao tác đơn giản.

Khi tạo một app, nhiều trường hợp nhà phát triển sẽ muốn xử lý một số chức năng trên trang web.

Ví dụ tiêu biểu nhất là trường hợp của Điều khoản dịch vụ (Terms of service) hoặc Chính sách bảo mật (Privacy policy), ta thường sẽ sử dụng trang web có sẵn để hiển thị chúng thay vì tạo riêng một trang trong application.

Khi sử dụng Firebase Hosting, ta có thể sử dụng web page cho ứng dụng iOS và Android của mình.

Với các ưu điểm thao tác nhanh, đơn giản, tính an toàn cao, Firebase hosting giúp các nhà phát triển tạo lập ứng dụng một cách đơn giản hơn so với việc tự tạo hay thuê server thực.

* + **Realtime Database**

Realtime Database của Firebase là một dịch vụ cung cấp khả năng đồng bộ dữ liệu thời gian thực cho các ứng dụng web và di động. Nó được lưu trữ trên đám mây và cung cấp các tính năng như quyền truy cập dữ liệu, đồng bộ hóa thời gian thực, bảo mật dữ liệu và lưu trữ không kết nối.

Realtime Database giúp các nhà phát triển tập trung vào việc xây dựng các tính năng chính của ứng dụng và cải thiện trải nghiệm người dùng.

* + **Firebase Cloud Messaging**

Với Firebase Cloud Messaging, ta có thể gửi nhận tin nhắn miễn phí. Khái niệm "Gửi message" ở đây còn dùng để chỉ việc push thông báo.

Khi tên nhắn mới được gửi tới, người dùng có thể nhận được thông báo. Ví dụ, với trường hợp người dùng sử dụng iPhone, khi có mail hay thông báo từ application, ở góc trên bên phải icon của app đó sẽ xuất hiện dấu tròn nhỏ màu đỏ, hay tiện lợi hơn, trong dấu tròn đỏ ấy sẽ hiển thị số lượng thông báo, tin nhắn mới.

Thêm vào đó, ta cũng có thể thiết lập để tin nhắn chỉ gửi tới những đối tượng mà ta mong muốn, ví dụ như việc gửi thông báo yêu cầu viết feedback sản phẩm cho những khách hàng thường xuyên sử dụng sản phẩm.

* + **Firebase Authentication**

Firebase Authentication là chức năng dùng để xác thực người dùng bằng Password, số điện thoại hoặc tài khoản Google, Facebook hay Twitter, v.v.

Việc xác thực người dùng là một chức năng quan trọng trong phát triển ứng dụng. Tuy nhiên, việc đối ứng với nhiều phương pháp xác thực khác nhau sẽ tốn nhiều thời gian và công sức. Firebase Authentication giúp thực hiện việc chia sẻ ID giữa các ứng dụng, giúp người dùng dễ dàng tiếp cận sản phẩm hơn. Vì thế, nó là một chức năng rất quý.

* + **Firebase Cloud Storage**

Firebase Cloud Storage hỗ trợ việc quản lý, chia sẻ các content người dùng upload lên như ảnh, video; cũng như sử dụng những tài nguyên ấy cho ứng dụng của bạn.

Data được lưu trữ trong Google Cloud Storage buckets, và có thể được access từ server. Đây cũng là một chức năng quan trọng và tiện lợi cho việc phát triển dịch vụ.

* + **Firebase Remote Config**

Firebase Remote Config giúp thực hiện việc thay đổi UI cũng như hoạt động của ứng dụng mà không cần phải update app đó.

Việc này thực hiện bằng cách set giá trị ban đầu (giá trị default) cho giao diện và hoạt động của app, sau đó sử dụng Remote Config REST API hoặc Firebase Console để ghi đè giá trị mới lên setting default.

* + **Firebase Test Lab**

Firebase Test Lab được sử dụng để test ứng dụng trên nền tẳng cloud. Bằng cách sử dụng Test Lab, ta có thể test hoạt động của ứng dụng trên nhiều thiết bị đa dạng, nhờ đó có thể xác minh được app sẽ chạy thế nào trên thiết bị thực của người dùng.

* + **Firebase Console**

Để sử dụng những chức năng được giới thiệu ở trên, ta cần đến Firebase console.

Để đăng nhập vào Firebase console, ta cần có tài khoản Google, vậy bước trước tiên cần được thực hiện là đăng ký một tài khoản Google nếu bạn chưa sở hữu, và đăng nhập với tài khoản đó.

**\*StarUML:**

StarUML là một mô hình nền tảng, là phần mềm hỗ trợ UML (Unified Modeling Language). Nó hỗ trợ các phương pháp tiếp cận MDA (Model Driven Architecture) bằng cách hỗ trợ các khái niệm hồ sơ UML. Tức là StarUML hỗ trợ phân tích và thiết kế hệ thống một điều mà bất cứ dự án nào đều cần có. Ngoài ra dùng StarUML sẽ đảm bảo tối đa hóa năng suất và chất lượng của các dự án phần mềm của bạn. Vì nó cho phép mô hình hóa nên sẽ không phụ thuộc vào người code, ngôn ngữ code hay nền tảng sử dụng. Do mô hình hóa nên rất dễ đọc và dễ hiểu. Một điều mình thích nhất ở StarUML là khả năng tự sinh code từ những mô hình class.

**Cài đặt**

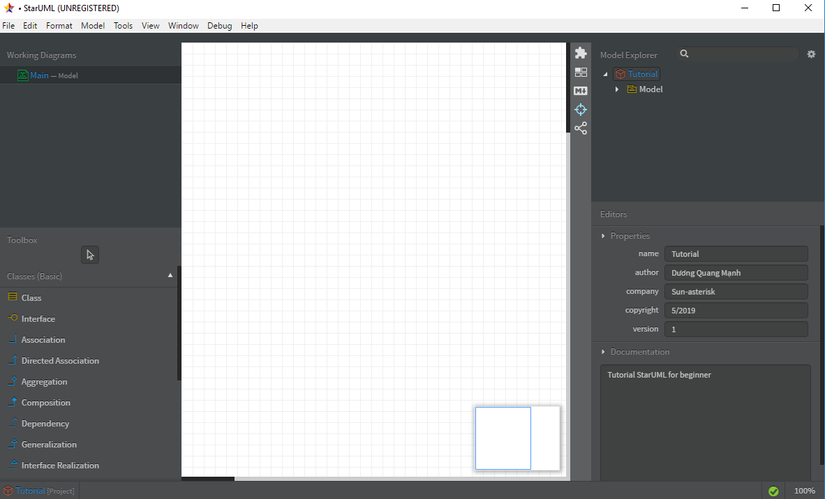
Để sử dụng được StarUML thì trước hết phải cài đặt nó, có thể tải trực tiếp từ trang chủ của [staruml](http://staruml.io/download) Đối với ubuntu sau khi tải về cần cài đặt, chạy các lệnh dưới đây (cần thay phiên bản phù hơp):

wget http://staruml.io/download/releases/StarUML-3.0.1-x86\_64.AppImage

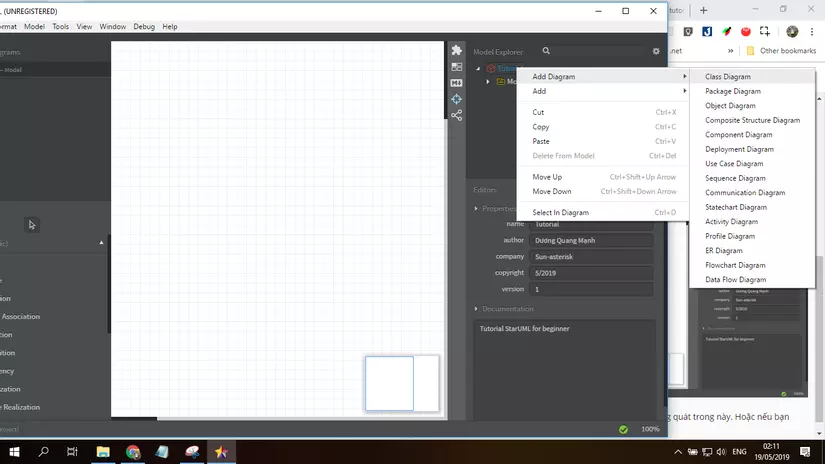
chmod +x StarUML-3.0.1-x86\_64.AppImage

./StarUML-3.0.1-x86\_64.AppImage

**Sử dụng**

Mở StarUML lên, để bắt đầu sử dụng bạn có thể thiết lập các thông số của project theo nhu cầu sử dụng:

Đầu tiên sẽ có một model Main tạo sẵn có thể thiết kế class tổng quát trong này. Hoặc nếu bạn muốn tạo một sơ đồ khác thì cần click chuột phải vào tên Project hoặc Model chọn Add Diagram -> Chọn một diagram muốn tạo



StarUML hỗ trợ đầy đủ các khái niệm hồ sơ UML:

* Class Diagram
* Package Diagram
* Object Diagram
* Composite Structure Diagram
* Component Diagram
* Deployment Diagram
* Use Case Diagram
* Sequence Diagram
* Comunication Diagram
* Statechart Diagram
* Activity Diagram
* Profile Diagram
* ER Diagram (mô hình quan hệ - thực thể; CSDL)
* Flowchart Diagram
* Data Flow Diagram

Với mỗi Diagram khi tạo sẽ cho ra một model mới. Bằng cách kéo thả các thành phần bên phải vào thì sẽ xây dựng được một Diagram hoàn chỉnh. Mọi người có thể tự mình thử, rất hữu ích.

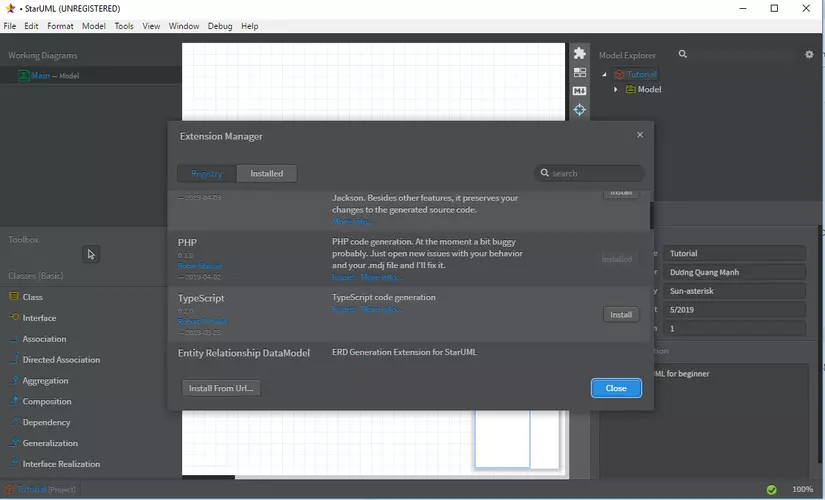
**Note**: các thực thể được định nghĩa ở các Diagram khác nhau cũng có thể được sử dụng lại bằng cách kéo thả từ bên phải vào.

**Tự sinh code bằng StarUML**

Không phụ thuộc vào nền tảng, ngôn ngữ nên StarUML có thể biên dịch ra tất cả các ngôn ngữ (mới thử với PHP). Tức là từ các Diagram ta có thể sinh ra code của một ngôn ngữ một cách tự động. Thật tuyệt khi không cần code chỉ cần thiết kế là code tự sinh! Sau đây mình xin giới thiệu tính năng tuyệt vời này.

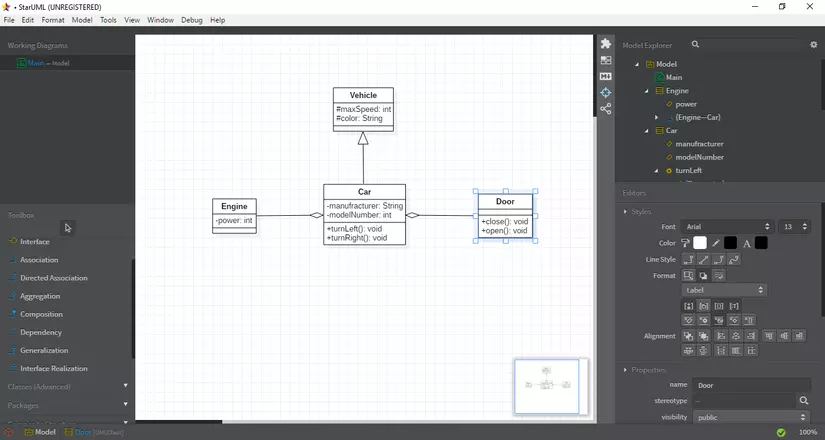
**Cài đặt Extension**

Trước hết để sử dụng ta cần cài đặt Extension của ngôn ngữ muốn sinh code vào StarUML, Click vào biểu tượng hình mảnh ghép phía trên cùng. Sau đó chọn ngôn ngữ muốn thêm:

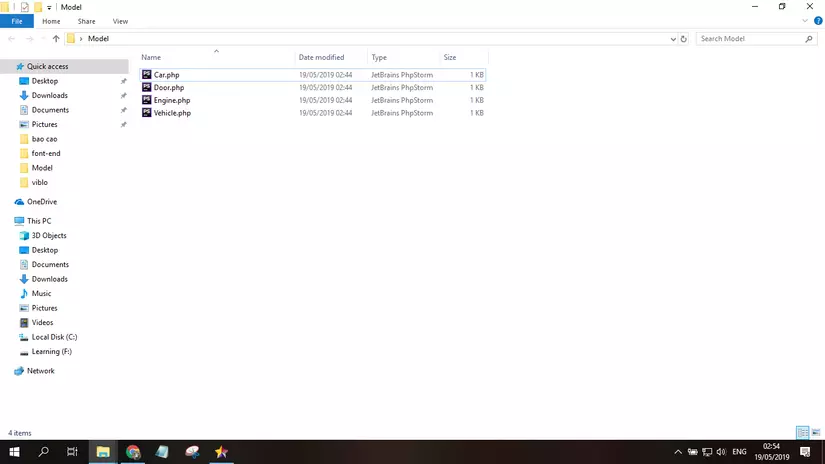


## Sinh code tự động

Sau khi cài xong Extension Chúng ta cần có một Diagram, ở đây mình đã xây dựng một Class Diagram:



cần chỉ ra chính xác các thuộc tính và phương thức với các data type, access modifiers tương ứng, các quan hệ giữa các class. Bây giờ các yếu tố cần thiết đã đầy đủ chúng ta có thể bắt đầu sinh code chỉ tốn trong 2s Chọn Tools-> Ngôn ngữ đã Extension -> Generate Code -> chọn Model -> save



**\*Android Studio**:

- Android Studio là Môi trường phát triển tích hợp (IDE) chính thức để phát triển ứng dụng Android. Nhờ có công cụ cho nhà phát triển và trình soạn thảo mã mạnh mẽ của [IntelliJ IDEA](https://www.jetbrains.com/idea/), Android Studio cung cấp thêm nhiều tính năng giúp bạn nâng cao năng suất khi xây dựng ứng dụng Android, chẳng hạn như:

* Một hệ thống xây dựng linh hoạt dựa trên Gradle
* Một trình mô phỏng nhanh và nhiều tính năng
* Một môi trường hợp nhất nơi bạn có thể phát triển cho mọi thiết bị Android
* Tính năng Áp dụng các thay đổi để đẩy các thay đổi về mã và tài nguyên vào ứng dụng đang chạy mà không cần khởi động lại ứng dụng
* Mã mẫu và tích hợp GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến cũng như nhập mã mẫu
* Đa dạng khung và công cụ thử nghiệm
* Công cụ tìm lỗi mã nguồn (lint) để nắm bắt hiệu suất, khả năng hữu dụng, khả năng tương thích với phiên bản và các vấn đề khác
* Hỗ trợ C++ và NDK
* Tích hợp sẵn tính năng hỗ trợ [Google Cloud Platform](https://cloud.google.com/tools/android-studio/docs/?hl=vi), giúp dễ dàng tích hợp Google Cloud Messaging và App Engine
* Android Studio có nhiều tính năng tuyệt vời giúp nâng cao năng suất khi xây dựng ứng dụng Android. Một số tính năng chính của Android Studio bao gồm:
* [**Áp dụng thay đổi**: Tính năng Áp dụng thay đổi (Apply Changes) của Android Studio cho phép bạn đẩy các thay đổi về mã và tài nguyên vào ứng dụng mà không cần khởi động lại và trong một số trường hợp mà không cần khởi động lại hoạt động hiện tại](https://developer.android.com/studio/features?hl=vi).
* [**Trình soạn thảo mã thông minh**: Trình soạn thảo mã giúp bạn viết mã tốt hơn, hoạt động nhanh hơn và làm việc hiệu quả hơn bằng cách cung cấp tính năng hoàn thành, tái cấu trúc và phân tích mã nâng cao](https://developer.android.com/studio/features?hl=vi).
* **Trình mô phỏng nhanh và giàu tính năng**: Trình mô phỏng Android cài đặt và khởi động ứng dụng của bạn nhanh hơn thiết bị thực tế, đồng thời cho phép bạn tạo nguyên mẫu và thử nghiệm ứng dụng trên nhiều cấu hình thiết bị Android: điện thoại, máy tính bảng, Android Wear, và các thiết bị Android TV. [Bạn cũng có thể mô phỏng nhiều tính năng phần cứng như vị trí GPS, độ trễ mạng, cảm biến chuyển động và dữ liệu nhập đa điểm](https://developer.android.com/studio/features?hl=vi)

[Android Studio cung cấp nhiều công cụ giúp bạn thử nghiệm ứng dụng Android của mình bằng JUnit 4 và các khung thử nghiệm giao diện người dùng giàu chức năng1](https://developer.android.com/studio/test?hl=vi). [Đối với các nhu cầu kiểm thử cơ bản, Android Studio tích hợp các tính năng giúp bạn tạo, chạy và xem kết quả kiểm thử, tất cả đều thuộc môi trường IDE](https://developer.android.com/studio/test?hl=vi). [Khi dùng Android Studio, bạn có thể trỏ và nhấp vào mã nguồn của ứng dụng để tạo và chạy kiểm thử cho các lớp hoặc phương thức cụ thể, sử dụng trình đơn để định cấu hình nhiều thiết bị kiểm thử, đồng thời tương tác với cửa sổ công cụ Ma trận thử nghiệm để trực quan hoá kết quả kiểm thử](https://developer.android.com/studio/test?hl=vi)

# PHẦN II: PHÂN TÍCH THIẾT KẾ PHẦN MỀM

1. **Xác định tác nhân và thiết kế cơ sở dữ liệu**

***Tác nhân 1: DienThoai:***

* Mỗi tác nhân tượng trưng cho một loại sản phẩm (điện thoại).
* Các thuộc tính:
* **id** là nút đại diện sản phẩm với một khóa duy nhất để phân biệt loại sản phẩm này với sản phẩm khác.
* **BinhLuan** là 1 object con của bao gồm các thuộc tính con khác:
* **id** là nút đại diện cho bình luận với một khóa duy nhất để phân biệt bình luận này với bình luận khác.
* Các đối tượng khác: **ngay**, **noiDung**, **sao,** **username.**
* Các đối tượng khác: **chiTiet, daBan, giaTien, linkAnh, soLike, ten.**

***Tác nhân 2: NguoiDung***

* Mỗi tác nhân tượng trưng cho một người dùng.
* Các thuộc tính:
* **id** là nút đại diện người dùng với một khóa duy nhất để phân biệt người dùng này với người dùng khác.
* Các dối tượng khác : **anh**, **diaChi**, **email**, **hoTen**, **password, phanQuyen, sdt.**

***Tác nhân 3: ThongBao***

* Mỗi tác nhân tượng trưng cho một thông báo.
* Các thuộc tính:
* **id** là nút đại diện thông báo với một khóa duy nhất để phân biệt thông báo này với thông báo khác
* Các đối tượng khác: **tieuDe**, **noiDung**.

***Tác nhân 4: ThongKe***

* Mỗi tác nhân tượng trưng cho một tập các đơn hàng của một loại sản phẩm.
* Các thuộc tính:
* **keyDH** là nút đại diện cho đơn hàng của loại sản phẩm đó với một khóa duy nhất để phân biệt đơn hàng này với đơn hàng khác của cùng một loại sản phẩm.
* Các đối tượng khác: **anhSP**, **diaChi**, **giaDT**, **giaSP**, **keyDT**, **nam**, **ngay**, **thang**, **sdt**, **soLuong**, **tenNguoiNhan**, **tenSP**, **trangthai**, **uid**.

***Tác nhân 5: ThongTinDonHang***

* Mỗi tác nhân tượng trưng cho một tập các đơn hàng của một người dùng.
* Các thuộc tính:
* **uid:** là một nút đại diện cho tập các đơn hàng của người dùng với một khóa duy nhất để phân biệt các đơn hàng đã đặt của người dùng này với các đơn hàng đã đặt của người dùng khác.
* **keyDH** là một object con gồm các thuộc tính con khác:
  + **keyDH** là nút đại diện cho đơn hàng của loại sản phẩm đó với một khóa duy nhất để phân biệt đơn hàng này với đơn hàng khác của cùng một loại sản phẩm.
  + Các đối tượng khác: **anhSP**, **diaChi**, **giaDT**, **giaSP**, **keyDT**, **nam**, **ngay**, **thang**, **sdt**

1. **Phân tích chức năng**

Phần mềm cần đảm bảo được những yêu cầu sau:

1. **Yêu cầu chức năng:**
   * Hệ thống cần đảm bảo các chức năng sau:
2. Đăng nhập & Đăng kí

+ Đăng nhập cho người dùng(Khách hàng,Nhân viên)

+ Đăng kí cho khách hàng tiềm năng

1. ***Phân quyền hệ thống***

-Quản lý (Admin):

+ Toàn bộ quyền trong hệ thống.

+ Cập nhập người dùng (nhân viên).

1. ***Cập nhập thông tin***

+ Thông tin cá nhân của nhân viên.

+ Cập nhập thông tin danh mục sản phẩm.

1. Mua sản phẩm

+ Tạo giỏ hàng và mua hàng qua mạng.

+ Thêm chức năng mua hàng online và thanh toán trực tiếp tiền mặt.

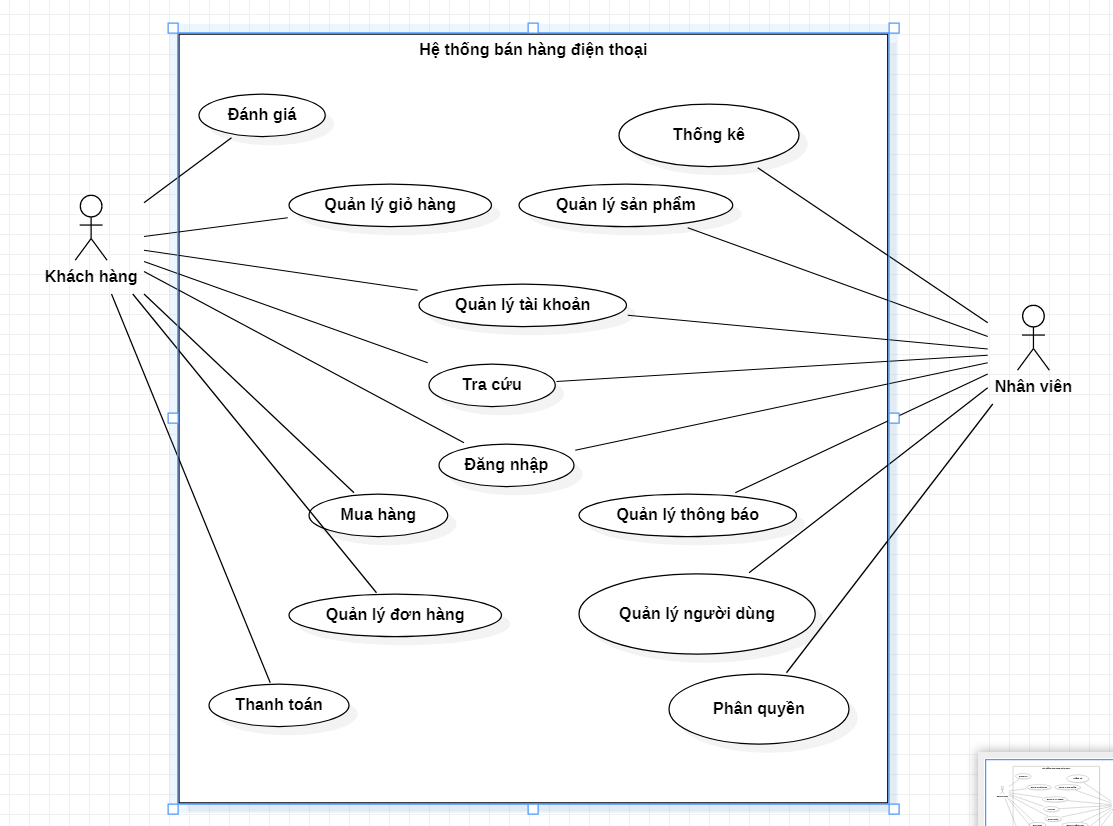
1. ***Tìm kiếm thông tin:***

+ Tìm kiếm thông tin sản phẩm.

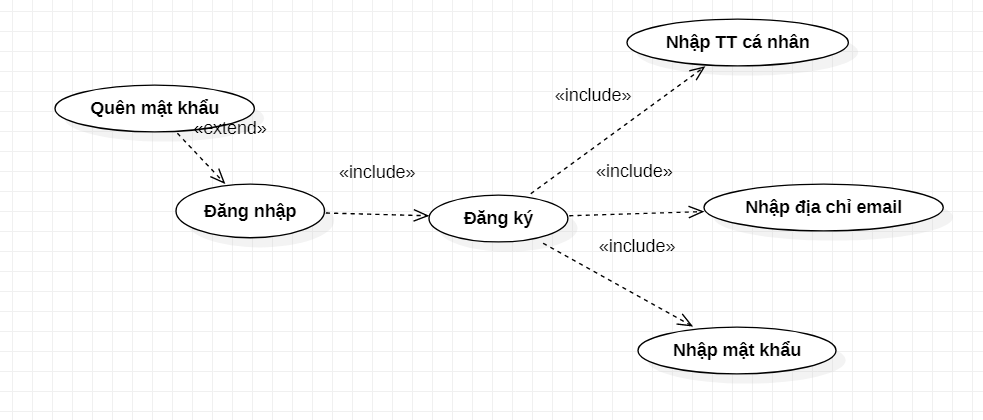
1. Báo cáo thống kê

+ Cập nhập thống kê

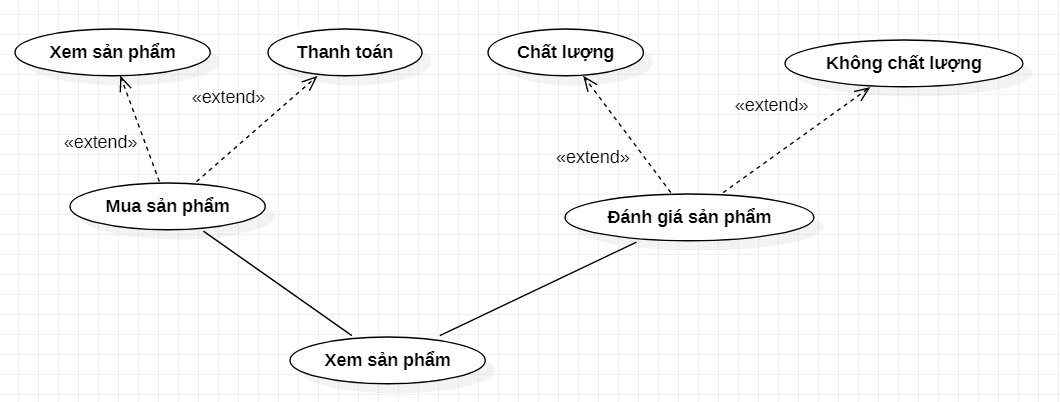
1. **Yêu cầu hệ thống:**
2. ***Phần mềm có giao diện thân thiện với người dùng, thuận tiện và dễ sử dụng***
3. **Mô hình UseCase**
4. **Mô hình usecase tổng quát**

****

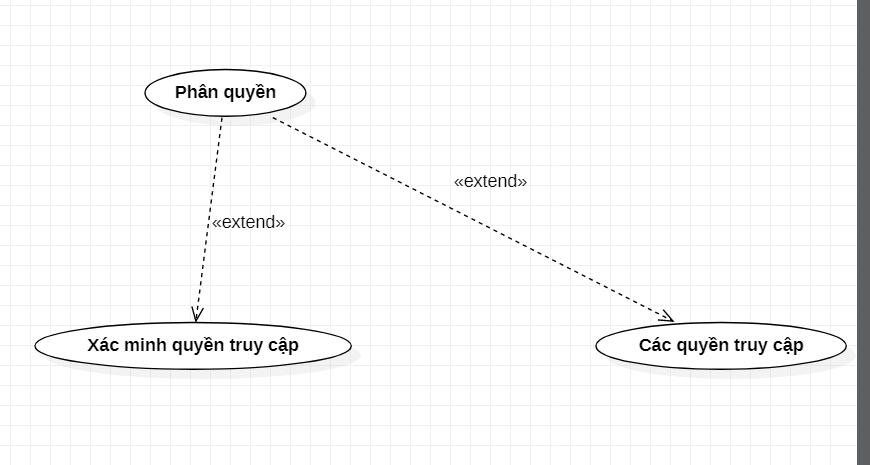
1. **Mô hình usecase phân rã chức năng: Đăng nhập**

****

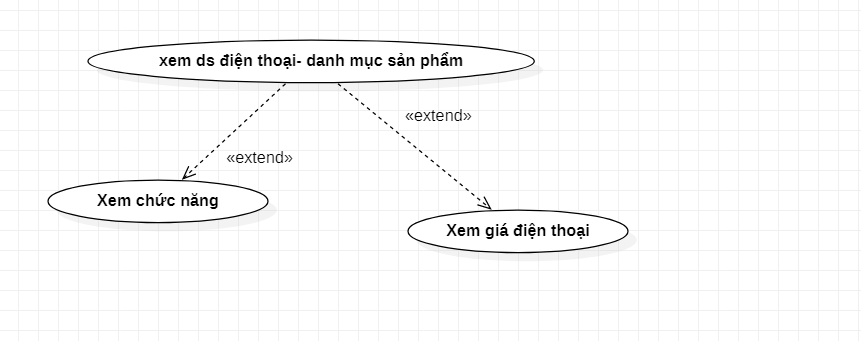
1. **Mô hình usecase phân rã chức năng: Mua sản phẩm & Đánh giá sản phẩm**

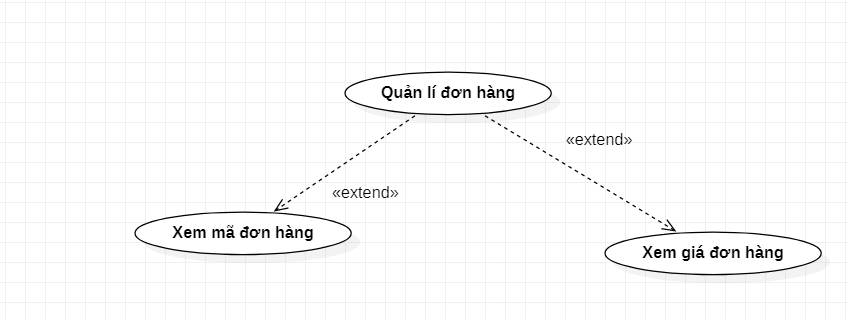
****

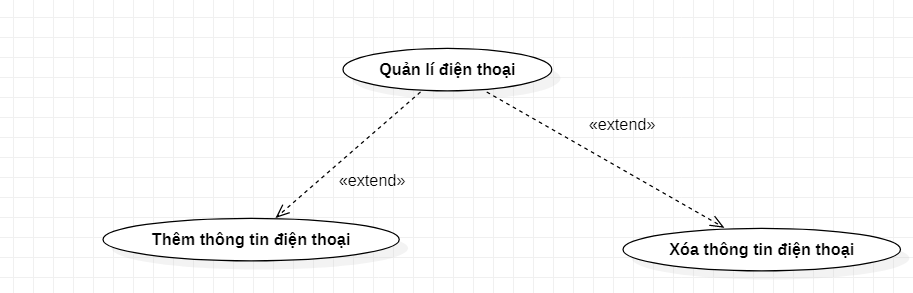
1. **Mô hình usecase phân rã chức năng: Phân quyền**



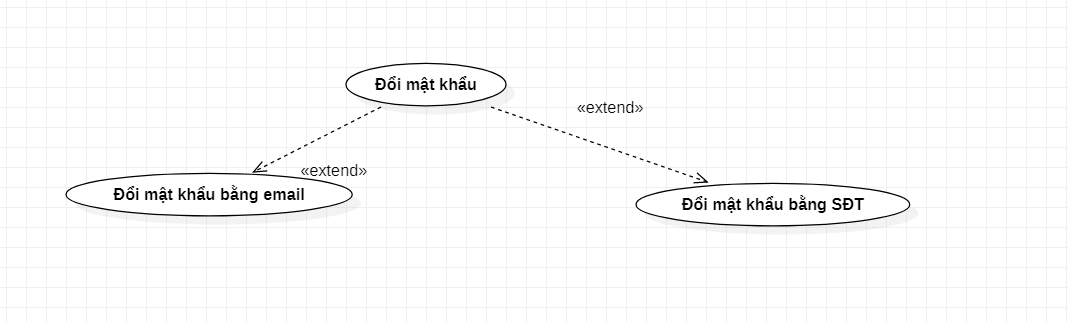
1. **Mô hình usecase phân rã chức năng: Xem danh sách sản phẩm.**

****

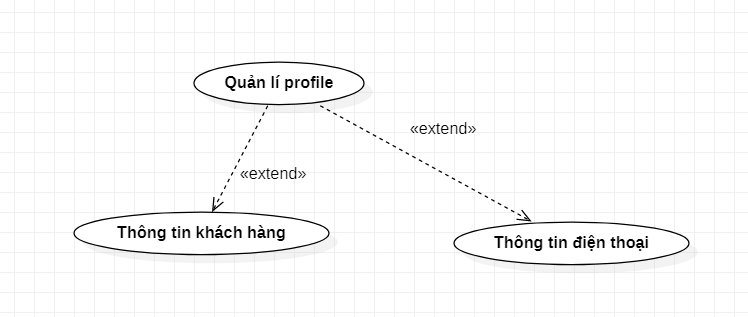
1. **Mô hình usecase phân rã chức năng: Quản lý đơn hàng**
2. **Mô hình usecase phân rã chức năng: Quản lý sản phẩm**

****

1. **Mô hình usecase phân rã chức năng: Đổi mật khẩu**

****

### Mô hình usecase phân rã chức năng: Quản lý profile



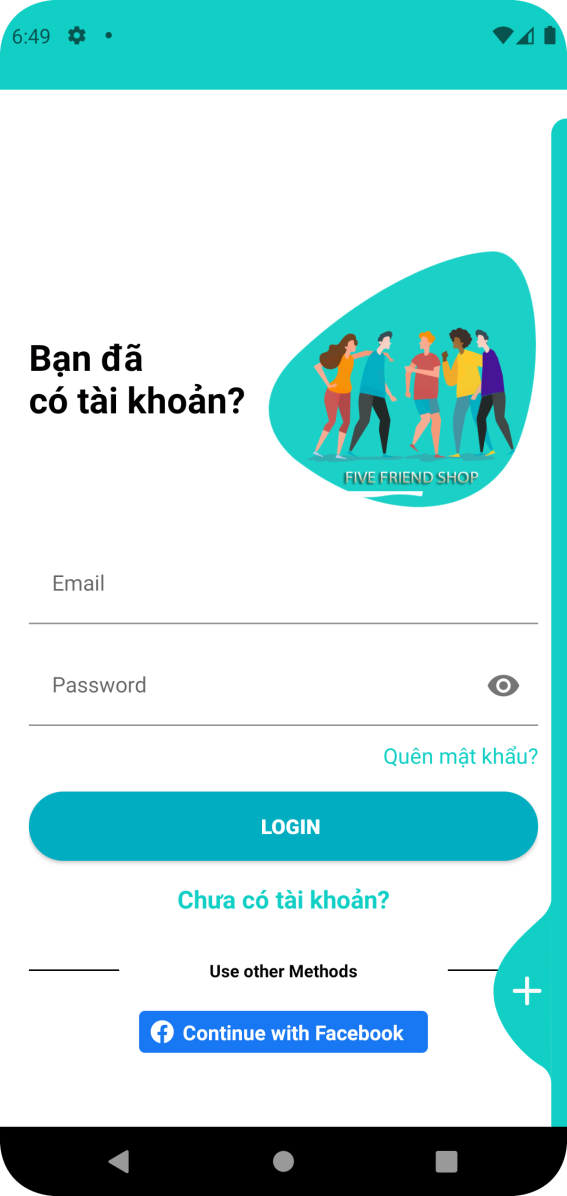
1. **Mô hình usecase phân rã chức năng: Xem lịch sử mua.**

### 

1. **Giao diện chương trình**

**PHẦN MỀM BÁN HÀNG ĐIỆN THOẠI**

1. **Menu chung.**
2. **Đăng nhập**

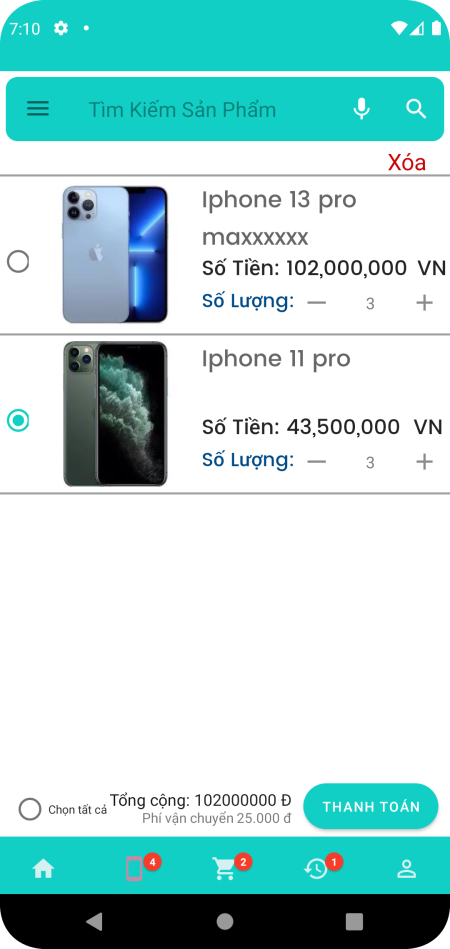
****

Khi người dùng sử dụng app, giao diện này sẽ hiện lên và yêu cầu người dùng đăng nhập để sử dụng các chức năng quan trọng khác của app.

Đăng nhập thành công sẽ vào giao diện Home của App.

Đăng nhập thất bại sẽ thông báo lỗi.

1. **Giỏ hàng**

****

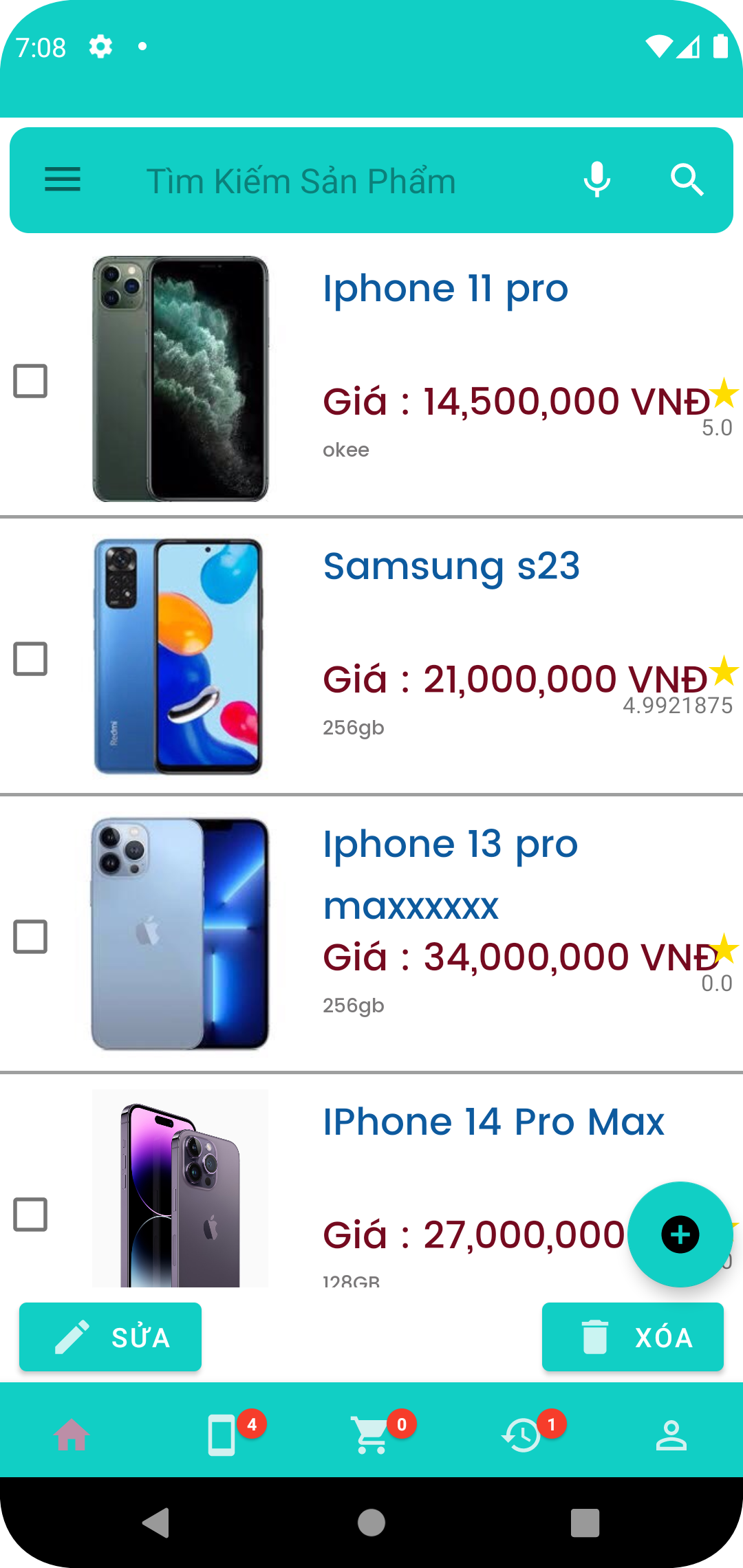
Sau khi người dùng thêm sản phẩm vào giỏ hàng. Sản phẩm sẽ hiện ở trong giỏ hàng. Giỏ hàng sẽ hiển thị số lượng sản phẩm đã được thêm vào trên công cụ.

1. **Chi tiết sản phẩm**



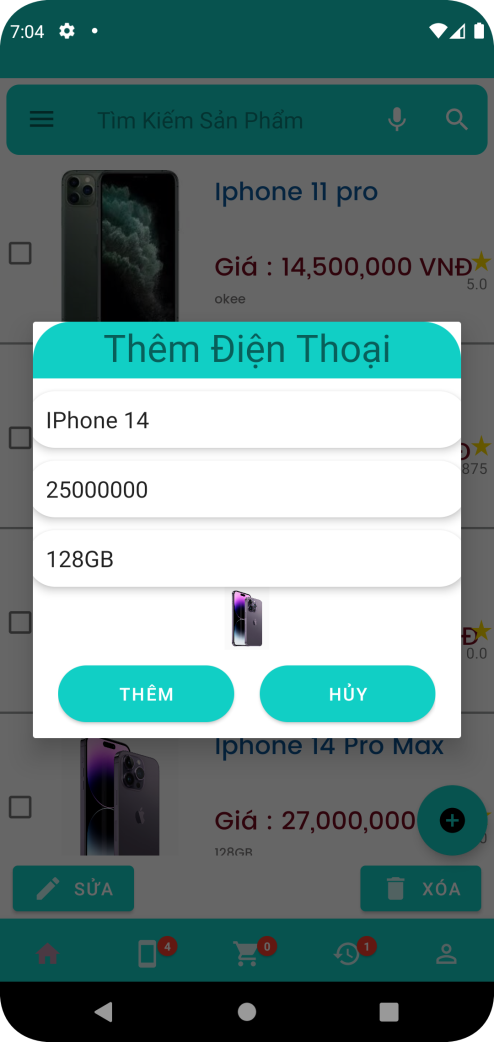
Sau khi người dùng chọn sản phẩm muốn xem chi tiết. Thông tin chi tiết về sản phẩm sẽ được hiện ra trên màn hình và người dùng có thể đặt vào giỏ hàng.

1. Giao diện cập nhập thông tin sản phẩm



Sau khi nhân viên chọn biểu tượng chỉnh sửa thì giao diện sẽ hiện thị để giúp nhân viên có thể điều chỉnh thông tin sản phẩm, thêm sản phẩm hay xóa sản phẩm.

1. Giao diện thêm sản phẩm



Khi nhân viên nhấn nút thêm: giao diện sẽ hiện thị để nhân viên nhập tên, giá tiền và các thông tin chi tiết và thêm ảnh mẫu cho sản phảm.

Khi chọn “Thêm” sản phẩm sẽ được bày bán trên kệ hàng giao diện.

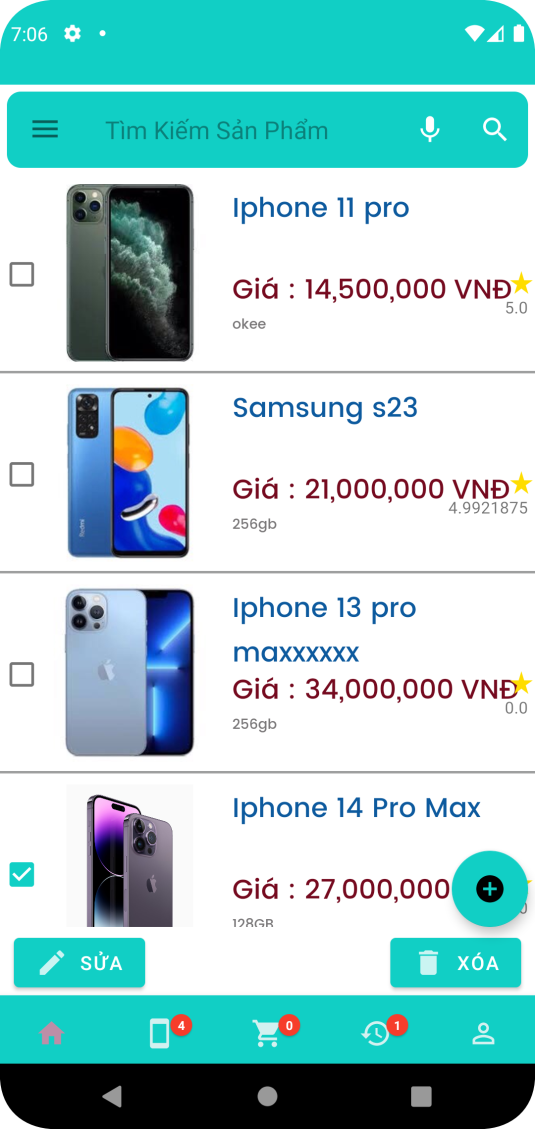
Khi chọn “Hủy” sản phẩm sẽ không được lưu và giao diện chuyển về giao diện chỉnh sửa ban đầu

1. Giao diện sửa sản phẩm

Khi chọn sửa giao diện. Nhân viên cân chọn sản phẩm muốn chỉnh sửa và chọn “Sửa” để tiến hành sửa thông tin sản phẩm.

Khi chọn “Lưu” các thông tin được sửa đổi sẽ được lưu trên hệ thống và bày bán lại lên kệ hàng.

Chọn lưu “Hủy” giao diện sẽ quay về giao diện chỉnh sửa, các thông tin sửa đổi cũng không được lưu trên hệ thống.



1. Giao diện xóa sản phẩm

Sản phẩm để được chọn sau khi chọn chức năng “XÓA” sẽ nhắc lại cho người dùng có chắc chắn muốn xóa không. Nếu chọn “có” sản phẩm sẽ thoát khỏi kệ hàng và xóa dữ liệu trên hệ thống database.

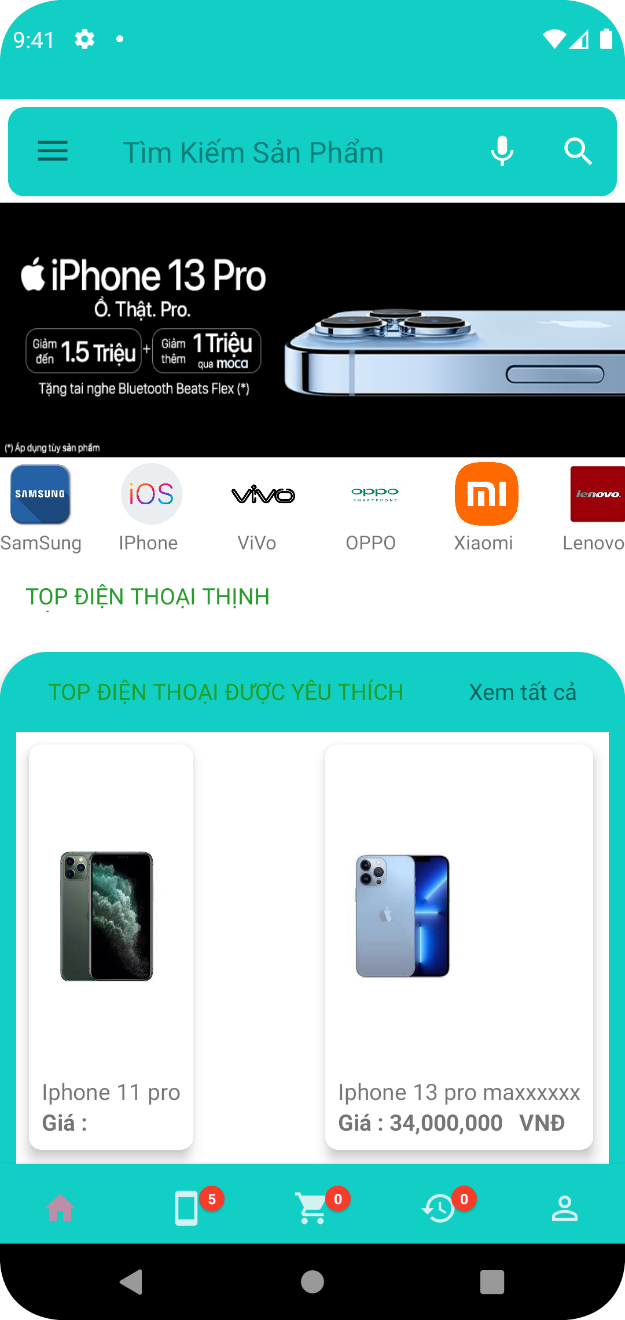
1. Giao diện thống kê

Khi nhân viên chọn mục thống kê, giao diện chức năng này sẽ được hiển thị trên màn hình.

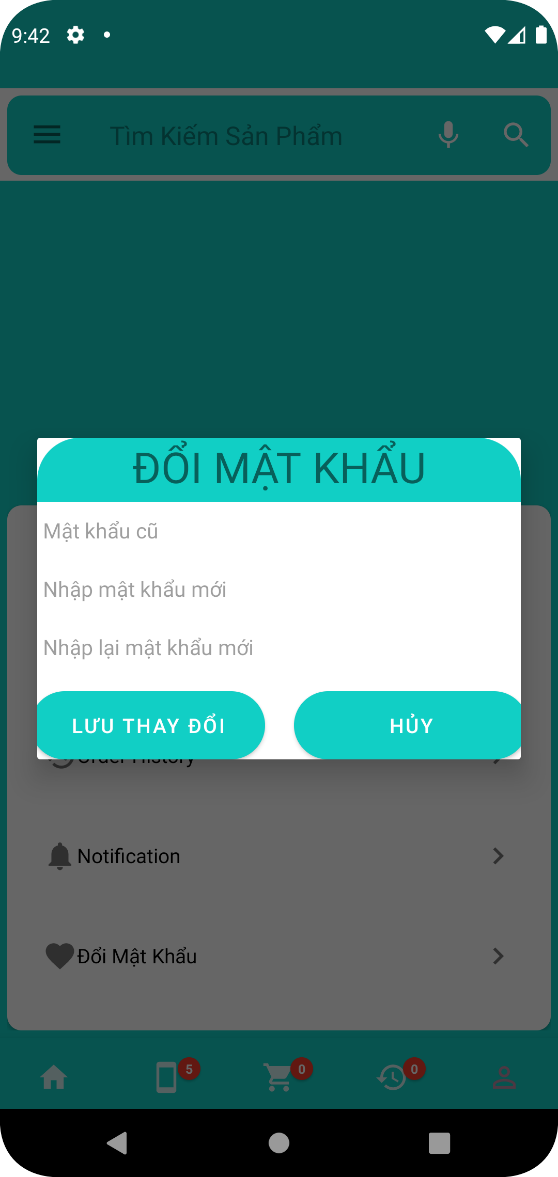


Nhân viên có thể xem thống kê chi tiết: Tổng đơn hàng mua,Tổng thành viên sử dụng App, tổng thành viên, tổng sản phẩm .v.v.v theo tháng, năm.

Giao diện trang chủ



Giao diện đổi mật khẩu



# KẾT LUẬN

Bài tập lớn Csharp :” Phần mềm quản lý nhân sự WordUp” được xây dựng trên Winform của Csharp kết hợp với Hệ quản trị cơ sở dữ liệu 2019.

Tuy nhiên do thời gian hoàn thành đề tài có hạn và kinh nghiệm nhiệm vụ chưa sâu cho nên đề tài của chúng em khó tránh được những thiếu sót về nghiệp vụ và chuyên môn. Chúng em mong nhận được những ý kiến đóng góp của các thầy cô và các bạnd dể chúng em được cải thiện hơn và đi vào thực tế.

1. **Kết quả đạt được**
   * Tạo cơ sở dữ liệu cho quản lý nhân sự
   * Chức năng của phần mềm tương đối hoàn thiện: chức năng cập nhập thông tin,chức năng tìm kiếm,chức năng tính giờ làm việc , tạo và tra cứu báo cáo, tra cứu nhân viên
2. **Kết quả chưa đạt được**
   * Hệ thống tính lương
3. **Hưong phát triển**
   * **Tính số giờ làm việc và số giờ làm việc tăng ca => tính tiền lương**

# TÀI LIỆU THAM KHẢO

**[1]**Vũ Lê Quỳnh Giang, Phạm Quanh Trình**, Hệ quản trị cơ sở dữ liệu SQL Sever.**Nhà Xuất Bản Giáo Dục Việt Nam

[2] Phạm Hữu Khang. ***C# 2005 -***  ***Lập trình Windows Forms***. Nhà xuất bản Minh Khai. 2005

[3] Phạm Hữu Khang. ***C# 2005 – Lập trình cơ sở dữ liệu quyển (4 tập)***. Nhà xuất bản Minh Khai. 2005

[4] Đỗ Thị Thanh Tuyền **Nhập Môn Công Nghệ Phần Mềm.**

[4***] Các diễn dàn tin học trên mạng internet***