

## LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc áp dụng các giải pháp công nghệ vào quản lý và kinh doanh đã trở thành xu hướng tất yếu. Các doanh nghiệp, từ quy mô nhỏ đến lớn, đều nhận thấy tầm quan trọng của việc tự động hóa và số hóa các quy trình quản lý nhằm nâng cao hiệu quả hoạt động và tiết kiệm chi phí. Đối với các quán Billiards, việc quản lý hiệu quả không chỉ giúp cải thiện chất lượng dịch vụ mà còn tối ưu hóa các nguồn lực, từ đó gia tăng lợi nhuận và sự hài lòng của khách hàng.

Xuất phát từ thực tế đó, đề tài "Xây dựng ứng dụng quản lý quán Billiards bằng Android Studio" ra đời với mục tiêu tạo ra một hệ thống quản lý chuyên nghiệp và hiện đại trên nền tảng di động. Ứng dụng này sẽ giúp các chủ quán Billiards dễ dàng quản lý các hoạt động hàng ngày như đặt bàn, thanh toán, quản lý nhân viên và theo dõi doanh thu ngay trên thiết bị di động. Việc sử dụng nền tảng Android Studio cho phép phát triển một ứng dụng với giao diện thân thiện, dễ sử dụng và tích hợp các tính năng tiên tiến đáp ứng nhu cầu thực tế của người dùng.

Thông qua việc áp dụng các công nghệ phát triển ứng dụng Android hiện đại, như Android Studio, Java/Kotlin và SQLite, hệ thống quản lý quán Billiards sẽ được xây dựng với giao diện người dùng tối ưu và hiệu suất cao. Đề tài này không chỉ tập trung vào việc triển khai các chức năng cơ bản mà còn hướng tới việc tối ưu hóa trải nghiệm người dùng và đảm bảo tính bảo mật cho dữ liệu của quán.

Với hy vọng đóng góp vào sự phát triển và hiện đại hóa của ngành dịch vụ giải trí, đề tài "Xây dựng ứng dụng quản lý quán Billiards bằng Android Studio" sẽ là một bước tiến quan trọng, mang lại nhiều lợi ích thiết thực cho các chủ quán và khách hàng. Chúng tôi tin rằng, với sự đầu tư nghiêm túc và nỗ lực không ngừng, ứng dụng này sẽ trở thành một công cụ quản lý hiệu quả, đồng hành cùng sự phát triển bền vững của các quán Billiards trong tương lai.

## BẢNG PHÂN CÔNG CÔNG VIỆC

Stt	Công việc	Người phụ trách	người hỗ trợ	kết quả dự án	Thời gian (Tuần)				Ghi chú
					T1	T2	T3	T4	
1	Lý thuyết	Nguyễn Thị Loan	Nguyễn Tuấn Anh	Hoàn thành	x				
2	mô tả bài toán	Phạm Ngọc Anh	Phạm Văn Bắc	Hoàn thành		x			
3	cài đặt bài toán	Hoàng Văn Thống	Phạm Ngọc Anh	Hoàn thành		x			
4	kiểm thử	Phạm Văn Bắc	Hoàng Văn Thống	Hoàn thành			x		
5	chế bản	Nguyễn Tuấn Anh	Nguyễn Thị Loan	Hoàn thành				x	
6	thiết kế trình bày	Phạm Văn Bắc	Nguyễn Thị Loan	Hoàn thành				x	

## MỤC LỤC

DANH MỤC SƠ ĐỒ, BẢNG BIỂU .....	3
CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI VÀ CÔNG CỤ PHÁT TRIỂN ỨNG DỤNG ANDROID .....	5
1.1. Tổng quan về đề tài .....	5
1.1.1. Lý do chọn đề tài .....	5
1.1.2. Đặt vấn đề .....	6
1.1.3. Khảo sát .....	6
1.1.4. Giải pháp .....	7
1.2. Công cụ phát triển ứng dụng Android .....	8
1.2.1. Giới thiệu về Android Studio .....	8
1.2.2. Ngôn ngữ lập trình cho Android.....	10
1.3. Công cụ phát triển ứng dụng Android.....	12
1.3.1. Giới thiệu về OOP với Java .....	12
1.3.2. Giới thiệu cấu trúc File Manifest trong Android.....	13
1.3.3. Giới thiệu về Intent trong Android Studio.....	15
1.3.4. Giới thiệu về Activity trong Android Studio .....	18
1.3.5. Cơ sở dữ liệu trong Android .....	21
1.2.4. Các công nghệ bổ sung .....	22
CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	24
2.1. Phân tích yêu cầu.....	24
2.1.1. Xác định yêu cầu chức năng.....	24
2.1.2. Xác định yêu cầu phi chức năng.....	27
2.2. Mô hình phân rã chức năng .....	28
2.2.1. Biểu đồ Usecase.....	29
2.2.2. Biểu đồ tuần tự.....	32

2.2.3. Biểu đồ lớp.....	36
2.2.4. Biểu đồ hoạt động.....	37
2.3. Mô hình cơ sở dữ liệu.....	40
2.4.1 Thiết kế giao diện .....	44
<b>CHƯƠNG III: TRIỂN KHAI VÀ PHÁT TRIỂN ỨNG DỤNG.....</b>	<b>53</b>
3.1. Mô tả giao diện của chương trình đã xây dựng.....	53
3.1.1. Cấu trúc mã nguồn.....	53
3.1.2. Giao diện đăng nhập .....	56
3.1.3. Giao diện đăng ký.....	57
3.1.4. Giao diện quản lý billiards.....	58
3.1.5. Giao diện quản lý bàn .....	59
3.1.6. Giao diện trang quản lý sản phẩm .....	61
3.1.7. Quản lý khách hàng .....	62
3.1.8. Quản lý hóa đơn.....	63
3.1.9. Thống kê .....	64
3.2. Triển khai ứng dụng .....	65
3.2.1. Danh sách các test case Unit Test.....	65
3.2.2. Thực hiện Function Unit test case .....	66
3.2.3. Kiểm thử login .....	76
3.3. Kết quả thực hiện chương trình.....	78
3.3.1. Kết quả kiểm thử Unit test case .....	78
3.3.2. Kết quả kiểm thử login .....	79
3.4. Triển khai ứng dụng .....	82
<b>KẾT LUẬN.....</b>	<b>83</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>84</b>

## DANH MỤC SƠ ĐỒ, BẢNG BIỂU

Sơ đồ 2.1: Mô hình phân rã chức năng.....	28
Sơ đồ 2.2: Usecase Tổng quát .....	29
Sơ đồ 2.3: Đăng nhập .....	29
Sơ đồ 2.4: Usecase Đăng xuất.....	30
Sơ đồ 2.5: Usecase Quản lý sản phẩm .....	30
Sơ đồ 2.6: Usecase Quản lý bàn .....	31
Sơ đồ 2.7: Usecase Quản lý hóa đơn.....	31
Biểu đồ 2.1: Biểu đồ tuần tự đăng nhập .....	32
Biểu đồ 2.2: Biểu đồ tuần tự đăng xuất .....	32
Biểu đồ 2.3: Biểu đồ tuần tự quản lý sản phẩm.....	33
Biểu đồ 2.4: Biểu đồ tuần tự quản lý bàn .....	34
Biểu đồ 2.5: Biểu đồ tuần tự quản lý hóa đơn .....	35
Biểu đồ 2.6: Biểu đồ lớp.....	36
Biểu đồ 2.7: Biểu đồ hoạt động đăng nhập .....	37
Biểu đồ 2.8: Biểu đồ hoạt động đăng xuất .....	37
Biểu đồ 2.9: Biểu đồ hoạt động quản lý sản phẩm.....	38
Biểu đồ 2.10: Biểu đồ hoạt động quản lý bàn .....	38
Biểu đồ 2.11: Biểu đồ hoạt động quản lý hóa đơn .....	39
Hình 2.4.1: Thiết kế giao diện đăng nhập .....	44
Hình 2.4.2: Thiết kế giao diện đăng ký .....	44
Hình 2.4.3: Thiết kế giao diện trang chủ .....	45
Hình 2.4.4: Thiết kế giao diện quản lý bàn .....	46
Hình 2.4.5: Thiết kế giao diện quản lý sản phẩm.....	47
Hình 2.4.6: Thiết kế giao diện quản lý khách hàng.....	48
Hình 2.4.7: Thiết kế giao diện quản lý hóa đơn .....	49
Hình 2.4.8: Thiết kế giao diện quản lý thống kê .....	50
Hình 2.4.9: Thiết kế giao diện menu tùy chọn .....	51
Hình 2.4.10: Thiết kế giao diện order bàn, sản phẩm .....	52
Hình 3.1: Cấu trúc mã nguồn .....	53

Hình 3.2: Giao diện đăng nhập .....	56
Hình 3.3: Giao diện đăng ký .....	57
Hình 3.4: Giao diện quản lý billiards của user .....	58
Hình 3.5: Giao diện quản lý bàn .....	59
Hình 3.6: Quản lý sản phẩm .....	61
Hình 3.7: Quản lý khách hàng .....	62
Hình 3.8: Quản lý hóa đơn .....	63
Hình 3.9: Giao diện thống kê .....	64
Hình 3.10: Kết quả test [Login-1] .....	79
Hình 3.11: Kết quả test [Login-2] .....	80
Hình 3.12: Kết quả test [Login-3] .....	81
 Bảng 2.1: Bảng dữ liệu account .....	40
Bảng 2.2: Bảng dữ liệu member .....	41
Bảng 2.3: Bảng dữ liệu hóa đơn .....	41
Bảng 2.4: Bảng dữ liệu sản phẩm .....	42
Bảng 2.5: Bảng dữ liệu chi tiết hóa đơn .....	42
Bảng 2.6: Bảng dữ liệu thời gian chơi .....	43
Bảng 2.7: Bảng dữ liệu bàn .....	43
 Bảng 3.1: Danh sách các test case cho unit test .....	65
Bảng 3.2: Unit test Function đăng ký .....	66
Bảng 3.3: Unit test Function đăng nhập .....	67
Bảng 3.4: Unit test Function thêm khách hàng .....	68
Bảng 3.5: Unit test Function sửa khách hàng .....	69
Bảng 3.6: Unit test Function tìm kiếm khách hàng .....	70
Bảng 3.7: Unit test Function thêm bàn .....	71
Bảng 3.8: Unit test Function tìm kiếm bàn .....	72
Bảng 3.9: Unit test Function thêm sản phẩm .....	73
Bảng 3.10: Unit test Function sửa sản phẩm .....	74
Bảng 3.11: Unit test Function tìm kiếm sản phẩm .....	75
Bảng 3.12: Tổng hợp kết quả kiểm thử Unit test .....	78

# CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI VÀ CÔNG CỤ PHÁT TRIỂN ỨNG DỤNG ANDROID

## 1.1. Tổng quan về đề tài

Trong bối cảnh kinh doanh hiện nay, các quán Billiards đang phát triển mạnh mẽ, thu hút đông đảo khách hàng nhờ vào sự thú vị và giải trí của bộ môn này. Tuy nhiên, việc quản lý một quán Billiards không hề đơn giản, đòi hỏi phải kiểm soát các hoạt động như quản lý bàn chơi, thời gian sử dụng bàn, thanh toán cho khách hàng, quản lý nhân viên, và lưu trữ thông tin khách hàng.

Hiện nay, nhiều quán vẫn áp dụng phương pháp quản lý thủ công, gây nhiều khó khăn và sai sót trong việc theo dõi doanh thu, kiểm soát thời gian chơi của khách, cũng như làm mất thời gian khi thanh toán. Vì vậy, nhu cầu về một ứng dụng quản lý quán Billiards chuyên nghiệp, hiệu quả trở nên cấp thiết.



Ứng dụng này sẽ được xây dựng trên nền tảng Android Studio, mang lại sự thuận tiện cho người dùng di động, giúp quản lý quán một cách linh hoạt và nhanh chóng, đồng thời giảm thiểu các sai sót trong quá trình quản lý.

### 1.1.1. Lý do chọn đề tài

- Nhu cầu thực tế: Quản lý quán Billiards thủ công gây nhiều bất tiện và sai sót, do đó cần một giải pháp công nghệ để nâng cao hiệu quả quản lý.
- Áp dụng công nghệ: Sử dụng ứng dụng di động trên nền tảng Android giúp tự động hóa quy trình quản lý, dễ sử dụng và phù hợp với xu hướng hiện đại.
- Tiết kiệm chi phí: Ứng dụng giúp giảm thiểu sai sót, tiết kiệm thời gian, tối ưu hóa chi phí nhân sự và vận hành.
- Khả năng mở rộng: Ứng dụng dễ dàng nâng cấp, tích hợp thêm tính năng như thanh toán điện tử, quản lý khách hàng trung thành.

- Phát triển kỹ năng: Đề tài giúp nhóm phát triển nâng cao kỹ năng lập trình Android, làm việc nhóm và quản lý dự án.

### **1.1.2. Đặt vấn đề**

Hiện nay, hầu hết các quán billiards vẫn sử dụng phương pháp quản lý thủ công, điều này dẫn đến nhiều hạn chế như:

- Khó khăn trong việc quản lý và theo dõi hoạt động kinh doanh: Các thông tin về doanh thu, chi phí, số lượng khách hàng thường không được ghi chép và thống kê chính xác.
- Thời gian chờ đợi lâu: Khách hàng phải chờ đợi lâu để được sắp xếp bàn chơi, dẫn đến sự không hài lòng và giảm trải nghiệm.
- Quản lý nhân viên thiếu hiệu quả: Việc phân công công việc và giám sát nhân viên gặp nhiều khó khăn, dẫn đến tình trạng làm việc kém hiệu quả.
- Khó khăn trong việc thanh toán: Quy trình thanh toán thủ công có thể gây ra sai sót và mất thời gian cho cả nhân viên và khách hàng



### **1.1.3. Khảo sát**

Qua khảo sát các quán Billiards tại khu vực, có thể thấy rằng:

- Quản lý bàn chơi: Thường ghi chú bằng tay hoặc qua máy tính cá nhân, nhưng vẫn còn nhiều trường hợp không kiểm soát được thời gian sử dụng của khách dẫn đến tính tiền sai.
- Quản lý thanh toán: Phần lớn vẫn sử dụng máy tính tiền hoặc ghi tay, dễ gây sai sót, tính nhầm chi phí. Các quán nhỏ hơn có thể chưa có hệ thống hóa đơn chuyên nghiệp.
- Quản lý nhân viên: Một số quán có khó khăn trong việc theo dõi giờ làm của nhân viên và phân công nhiệm vụ cho từng người.

- Quản lý khách hàng: Việc chăm sóc khách hàng trung thành, lưu trữ thông tin và khuyến mãi vẫn chưa được quan tâm đúng mức.

Từ thực tế này, có thể thấy việc ứng dụng công nghệ vào quản lý sẽ giúp tiết kiệm thời gian, nâng cao hiệu quả hoạt động, giảm thiểu rủi ro và nâng cao trải nghiệm của khách hàng.

#### **1.1.4. Giải pháp**

Xây dựng một ứng dụng di động trên nền tảng Android Studio giúp quản lý toàn diện các hoạt động của quán Billiards. Cụ thể:

- Quản lý bàn chơi: Ứng dụng sẽ cung cấp tính năng theo dõi số lượng bàn trống, quản lý thời gian chơi của từng bàn. Khi khách bắt đầu chơi, nhân viên chỉ cần chọn bàn và kích hoạt bộ đếm thời gian. Khi kết thúc, hệ thống tự động tính toán chi phí dựa trên thời gian sử dụng.
- Quản lý thanh toán: Ứng dụng sẽ tích hợp công cụ tính toán tự động, kết nối với các dịch vụ thanh toán điện tử, giúp việc thanh toán diễn ra nhanh chóng và chính xác. Cung cấp hóa đơn điện tử trực tiếp cho khách hàng thông qua ứng dụng.
- Quản lý nhân viên: Quản lý lịch làm việc của nhân viên, theo dõi thời gian làm việc, phân chia công việc dễ dàng thông qua ứng dụng.



- Quản lý khách hàng: Lưu trữ thông tin khách hàng trung thành, tích hợp hệ thống tích điểm, khuyến mãi và các chương trình ưu đãi.
- Giao diện thân thiện: Thiết kế giao diện dễ sử dụng, phù hợp với nhu cầu của nhân viên và chủ quán, đảm bảo quá trình thao tác nhanh chóng, tiết kiệm thời gian.

Ứng dụng sẽ mang lại sự chuyên nghiệp trong công tác quản lý, cải thiện chất lượng dịch vụ và tạo ra một môi trường kinh doanh hiện đại cho các quán Billiards. Nền tảng Android giúp ứng dụng dễ dàng triển khai trên các thiết bị di động, thuận tiện cho chủ quán và nhân viên trong việc sử dụng hằng ngày.

## 1.2. Công cụ phát triển ứng dụng Android

### 1.2.1. Giới thiệu về Android Studio

#### ⊕ Định nghĩa

Android Studio là môi trường phát triển tích hợp (IDE) chính thức được Google phát triển và duy trì cho việc xây dựng ứng dụng Android. Android Studio được xây dựng dựa trên IntelliJ IDEA, một IDE mạnh mẽ cho Java, với các cải tiến và tích hợp thêm các công cụ cụ thể dành cho phát triển ứng dụng Android.

- Môi trường phát triển tích hợp (IDE): Android Studio cung cấp một nền tảng tích hợp hoàn chỉnh với các công cụ hỗ trợ mọi bước của quy trình phát triển ứng dụng, từ viết mã đến kiểm thử và triển khai.
- Đặc điểm chính: Giao diện người dùng trực quan, tích hợp các công cụ mạnh mẽ cho việc thiết kế giao diện người dùng (UI), gỡ lỗi, và tối ưu hóa hiệu suất ứng dụng.



#### ⊕ Tính năng nổi bật

- Trình biên tập mã nguồn: Android Studio cung cấp một trình biên tập mã nguồn mạnh mẽ với tính năng hoàn thành mã tự động (code completion), kiểm tra cú pháp, và hỗ trợ các ngôn ngữ lập trình như Java, Kotlin, và XML.
- Công cụ gỡ lỗi: Bao gồm một loạt công cụ gỡ lỗi, như trình gỡ lỗi đồ họa, phân tích hiệu suất, và công cụ phân tích bộ nhớ, giúp phát hiện và sửa lỗi trong mã nguồn ứng dụng.
- Thiết kế giao diện: Công cụ Layout Editor cho phép tạo và chỉnh sửa giao diện người dùng bằng cách kéo và thả các thành phần, cùng với tính năng Constraint Layout giúp xây dựng giao diện tương thích với nhiều kích thước màn hình khác nhau.

- Tích hợp hệ thống kiểm thử: Android Studio hỗ trợ kiểm thử đơn vị (Unit Testing) và kiểm thử giao diện người dùng (UI Testing) thông qua các công cụ như JUnit và Espresso.

## Cài đặt và cấu hình

Yêu cầu phần cứng: Android Studio yêu cầu ít nhất 8 GB RAM (khuyến nghị 16 GB) và 4 GB dung lượng ổ đĩa trống. Tùy vào kích thước dự án và số lượng công cụ hỗ trợ cài đặt, dung lượng ổ đĩa cần thiết có thể lớn hơn.

Yêu cầu phần mềm: Android Studio hỗ trợ các hệ điều hành Windows 10 trở lên, macOS 10.14 hoặc mới hơn, và các phiên bản phổ biến của Linux. JDK (Java Development Kit) 11 hoặc mới hơn là cần thiết.

### ➤ Trên Windows:

- Tải xuống tệp cài đặt Android Studio từ trang web chính thức của Google.
- Chạy tệp .exe để bắt đầu quá trình cài đặt.
- Làm theo hướng dẫn trên màn hình để hoàn tất cài đặt, bao gồm việc chọn các thành phần bổ sung như Android SDK và Android Virtual Device (AVD).

### ➤ Trên macOS:

- Tải xuống tệp .dmg từ trang web của Android Studio.
- Mở tệp .dmg và kéo Android Studio vào thư mục Applications.
- Mở Android Studio từ thư mục Applications và làm theo hướng dẫn để hoàn tất cấu hình ban đầu.

### ➤ Trên Linux:

- Tải xuống tệp .zip từ trang web của Android Studio.
- Giải nén tệp .zip vào thư mục mong muốn.
- Mở terminal, chuyển đến thư mục Android Studio và chạy tệp studio.sh để bắt đầu cài đặt.

### **1.2.2. Ngôn ngữ lập trình cho Android**

#### **Java**

Java là một ngôn ngữ lập trình hướng đối tượng phổ biến, được phát triển bởi Sun Microsystems (hiện thuộc Oracle) vào năm 1995. Java nổi bật với khả năng "Write Once, Run Anywhere" (WORA), nghĩa là một ứng dụng Java có thể chạy trên bất kỳ nền tảng nào hỗ trợ Java mà không cần chỉnh sửa mã nguồn. Java thường được sử dụng cho các ứng dụng di động, đặc biệt là Android, các hệ thống doanh nghiệp và ứng dụng web.

➤ *Ưu điểm của Java:*

- Tính độc lập nền tảng: Java có thể chạy trên nhiều hệ điều hành khác nhau nhờ cơ chế máy ảo Java (JVM).
- Bảo mật tốt: Java cung cấp các tính năng bảo mật mạnh mẽ, phù hợp cho các ứng dụng doanh nghiệp.
- Cộng đồng lớn và hỗ trợ mạnh mẽ: Java có một cộng đồng lập trình viên đông đảo, tài liệu phong phú và nhiều thư viện hỗ trợ.
- Hiệu năng tốt: Dù không nhanh như C++, Java vẫn có hiệu suất ổn định và thường được tối ưu hóa trong quá trình biên dịch JIT (Just-In-Time).
- Hướng đối tượng hoàn chỉnh: Hỗ trợ các khái niệm hướng đối tượng đầy đủ, giúp phát triển phần mềm một cách hệ thống và dễ bảo trì.



➤ *Nhược điểm của Java:*

- Cú pháp dài dòng: So với các ngôn ngữ lập trình hiện đại khác như Kotlin, cú pháp của Java thường dài dòng và phức tạp hơn.
- Hiệu suất thấp hơn một số ngôn ngữ biên dịch trực tiếp: Dù JVM giúp Java hoạt động độc lập nền tảng, hiệu suất của Java không nhanh bằng các ngôn ngữ biên dịch trực tiếp như C/C++.
- Sử dụng bộ nhớ cao: Java tiêu thụ nhiều bộ nhớ hơn so với một số ngôn ngữ khác, do cơ chế quản lý bộ nhớ tự động bằng garbage collection.

#### **Kotlin**

Kotlin là một ngôn ngữ lập trình hiện đại, được phát triển bởi JetBrains và chính thức ra mắt vào năm 2011. Nó được thiết kế để tương thích với Java nhưng cải thiện các hạn chế của Java. Kotlin đã được Google công nhận là ngôn ngữ chính thức cho phát triển ứng dụng Android vào năm 2017. Kotlin có cú pháp ngắn gọn, hiện đại và hỗ trợ nhiều tính năng lập trình chức năng.

➤ Ưu điểm của Kotlin:

- Cú pháp ngắn gọn và dễ hiểu: Kotlin giảm thiểu các đoạn mã lặp lại và dài dòng, giúp code trở nên dễ đọc và dễ bảo trì hơn so với Java.
- Tương thích với Java: Kotlin hoàn toàn tương thích với Java, cho phép các dự án Java có thể dễ dàng chuyển đổi và sử dụng thư viện từ hệ sinh thái Java.
- An toàn về null: Kotlin giảm thiểu lỗi NullPointerException thông qua cơ chế null-safety, giúp tăng độ an toàn khi phát triển.
- Lập trình hàm: Kotlin hỗ trợ lập trình hàm (functional programming), mang lại tính linh hoạt và hiệu quả cao trong phát triển phần mềm.
- Hiệu suất tốt hơn Java trong một số trường hợp: Kotlin biên dịch thành mã bytecode JVM, giúp tối ưu hóa tốt hơn trong nhiều tình huống, đặc biệt khi xử lý đồng thời hoặc tính toán phức tạp.



➤ Nhược điểm của Kotlin:

- Thời gian biên dịch lâu hơn: So với Java, Kotlin thường có thời gian biên dịch dài hơn, đặc biệt là với các dự án lớn.
- Cộng đồng nhỏ hơn Java: Mặc dù Kotlin đang phát triển nhanh chóng, nhưng cộng đồng lập trình viên và tài liệu hỗ trợ vẫn chưa phong phú bằng Java.
- Độ phức tạp đối với người mới: Kotlin có nhiều tính năng hiện đại, đôi khi gây khó khăn cho những lập trình viên mới tiếp cận, đặc biệt là những người quen với Java.

## 1.3. Công cụ phát triển ứng dụng Android

### 1.3.1. Giới thiệu về OOP với Java

**Lập trình hướng đối tượng (OOP)** là một cách tiếp cận lập trình mà trong đó, các đối tượng và lớp đóng vai trò chính. Java là một ngôn ngữ lập trình hướng đối tượng mạnh mẽ, hỗ trợ đầy đủ các nguyên tắc cơ bản của OOP. Dưới đây là các khái niệm chính trong OOP với Java:

#### 1) Lớp (Class):

- **Mô tả:** Lớp là một mẫu hoặc bản thiết kế để tạo ra đối tượng. Nó định nghĩa các thuộc tính (biến) và phương thức (hàm) mà các đối tượng của lớp sẽ có.
- **Ví dụ:** Một lớp Car có thể có các thuộc tính như color, model, và các phương thức như start(), stop().



#### 2) Đối tượng (Object):

- **Mô tả:** Đối tượng là một thực thể cụ thể được tạo ra từ lớp. Mỗi đối tượng có trạng thái riêng biệt và hành vi cụ thể.
- **Ví dụ:** Một đối tượng myCar của lớp Car có thể có màu đỏ và mô hình Toyota.

#### 3) Tính kế thừa (Inheritance):

- **Mô tả:** Là khả năng cho phép một lớp kế thừa các thuộc tính và phương thức từ một lớp khác, giúp tái sử dụng mã và mở rộng chức năng của lớp.
- **Ví dụ:** Một lớp ElectricCar có thể kế thừa từ lớp Car và thêm các thuộc tính và phương thức riêng, như batteryLevel và charge().

4) Tính đóng gói (Encapsulation):

- **Mô tả:** Là việc đóng gói dữ liệu và phương thức vào một lớp và chỉ cung cấp các phương thức công khai để truy cập và thay đổi dữ liệu đó. Điều này giúp bảo vệ dữ liệu khỏi việc thay đổi không mong muốn từ bên ngoài lớp.
- **Ví dụ:** Một lớp Person có thể có thuộc tính name và age được đóng gói và chỉ có thể được truy cập thông qua các phương thức getName() và setName().

5) Tính đa hình (Polymorphism):

- **Mô tả:** Là khả năng của một phương thức để hành động khác nhau tùy thuộc vào đối tượng mà nó được gọi trên đó. Điều này bao gồm việc ghi đè phương thức (method overriding) và phương thức trùng lặp (method overloading).
- **Ví dụ:** Phương thức draw() có thể được ghi đè trong các lớp con Circle, Rectangle để vẽ các hình dạng khác nhau, mặc dù chúng đều gọi phương thức draw().

6) Tính trừu tượng (Abstraction):

- **Mô tả:** Là việc ẩn đi các chi tiết cụ thể và chỉ hiện ra các thông tin quan trọng, giúp giảm độ phức tạp và tập trung vào những gì cần thiết. Trong Java, điều này được thực hiện thông qua lớp trừu tượng và giao diện.
- **Ví dụ:** Một lớp trừu tượng Shape có thể định nghĩa các phương thức trừu tượng như draw() và area(), mà các lớp con như Circle, Square phải thực hiện các phương thức này.

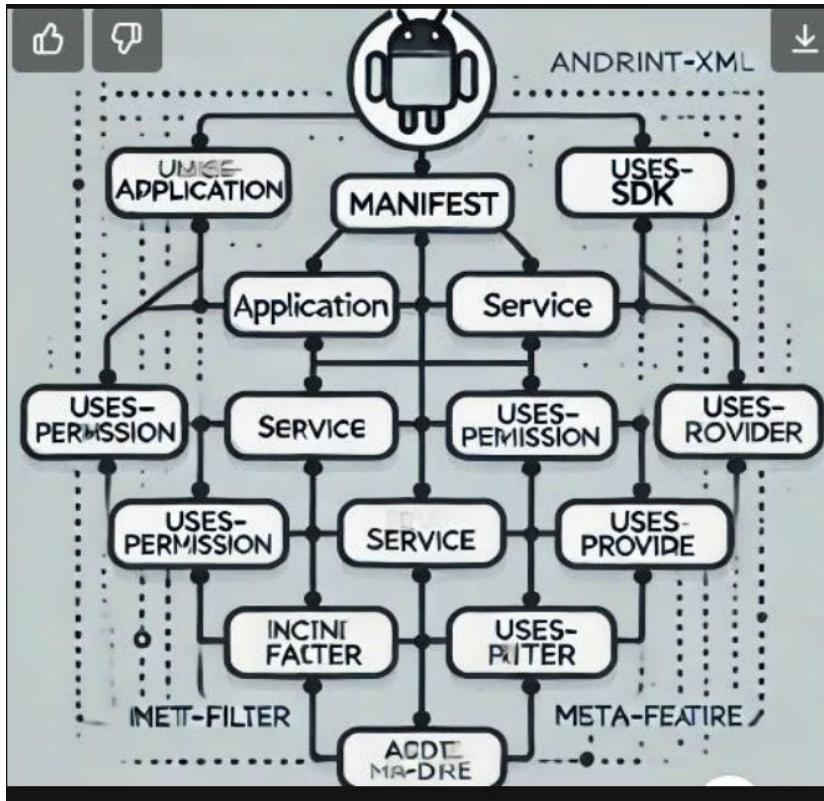
Những nguyên tắc này không chỉ giúp tổ chức mã nguồn một cách hiệu quả mà còn giúp giảm thiểu lỗi, tăng cường khả năng bảo trì và mở rộng của ứng dụng, đồng thời thúc đẩy việc viết mã sạch và dễ hiểu.

### 1.3.2. Giới thiệu cấu trúc File Manifest trong Android

File **AndroidManifest.xml** là một trong những tệp tin quan trọng nhất trong một dự án Android, đóng vai trò như một bản đồ cung cấp thông tin về ứng dụng của bạn cho hệ điều hành Android. File này chứa các thông tin cần thiết giúp Android quản lý và khởi chạy ứng dụng một cách chính xác. Dưới đây là cấu trúc và các thành phần chính của **AndroidManifest.xml**:

### 1) Thẻ gốc <manifest>

Thẻ **<manifest>** là thẻ bao bọc toàn bộ nội dung của file **AndroidManifest.xml**. Thẻ này định nghĩa thông tin chung về ứng dụng, chẳng hạn như tên gói (package) - đây là định danh duy nhất của ứng dụng trên hệ thống Android. Ngoài ra, trong thẻ này còn chứa các thông tin về phiên bản của ứng dụng, bao gồm **mã phiên bản** và **tên phiên bản**.



### 2) Thẻ <application>

Thẻ **<application>** bao gồm các thông tin mô tả về ứng dụng và chứa các thành phần quan trọng như **Activity**, **Service**, **BroadcastReceiver**, và **ContentProvider**. Các thuộc tính của thẻ này có thể bao gồm nhãn (tên ứng dụng hiển thị trên màn hình), biểu tượng (icon của ứng dụng), chủ đề (theme), khả năng sao lưu dữ liệu, và hỗ trợ cho giao diện từ phải sang trái (RTL).

### 3) Khai báo các thành phần trong thẻ <application>

#### *a) Activity*

**Activity** là thành phần giao diện người dùng chính của một ứng dụng. Mỗi Activity cần được khai báo trong file **AndroidManifest.xml**. Đối với Activity chính (thường là màn hình chính khi mở ứng dụng), bạn sẽ cần xác định hành động (action) để hệ thống biết đây là Activity khởi đầu khi người dùng mở ứng dụng.

#### *b) Service*

**Service** là các thành phần chạy ngầm (background) không có giao diện người dùng. Chúng thực hiện các tác vụ như xử lý dữ liệu, đồng bộ hóa hoặc phát nhạc mà không yêu

cầu người dùng tương tác trực tiếp. Tất cả các Service đều cần được khai báo trong file manifest.

*c) BroadcastReceiver*

**BroadcastReceiver** là thành phần cho phép ứng dụng lắng nghe và phản hồi các sự kiện hệ thống hoặc các thông điệp được phát ra từ các ứng dụng khác hoặc từ hệ điều hành. Những sự kiện này có thể bao gồm tin nhắn đến, kết nối mạng thay đổi, hoặc trạng thái pin.

*d) ContentProvider*

**ContentProvider** giúp quản lý và chia sẻ dữ liệu giữa các ứng dụng. Thông qua ContentProvider, các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác một cách an toàn và bảo mật.

*4) Permissions (Quyền)*

Một phần quan trọng trong file **AndroidManifest.xml** là khai báo các quyền mà ứng dụng cần để hoạt động. Các quyền này có thể bao gồm quyền truy cập vào Internet, sử dụng camera, vị trí, gửi tin nhắn, hoặc đọc danh bạ. Người dùng cần cấp quyền này cho ứng dụng khi cài đặt hoặc sử dụng ứng dụng lần đầu.

*5) Yêu cầu phần cứng và tính năng*

Trong **AndroidManifest.xml**, bạn có thể khai báo các yêu cầu về phần cứng mà ứng dụng cần để hoạt động đúng cách, ví dụ như yêu cầu có camera, GPS, hoặc cảm biến nào đó. Ngoài ra, bạn cũng có thể chỉ định phiên bản Android tối thiểu mà ứng dụng hỗ trợ, cũng như phiên bản Android mà ứng dụng hướng tới để tối ưu hóa.

*6) Metadata (Siêu dữ liệu)*

Siêu dữ liệu là các thông tin bổ sung mà bạn có thể truyền đến ứng dụng hoặc các thư viện bên ngoài. Các dữ liệu này có thể bao gồm khóa API hoặc các thông tin cấu hình cần thiết cho một số dịch vụ hoặc thư viện cụ thể mà ứng dụng sử dụng.

### **1.3.3. Giới thiệu về Intent trong Android Studio**

Trong phát triển ứng dụng Android, **Intent** là một khái niệm quan trọng cho phép các thành phần trong ứng dụng giao tiếp với nhau, dù là trong cùng một ứng dụng hay giữa các ứng dụng khác nhau. Nó có vai trò giúp các thành phần của hệ thống Android biết được ứng dụng muốn thực hiện hành động nào và dữ liệu nào liên quan đến hành động đó.

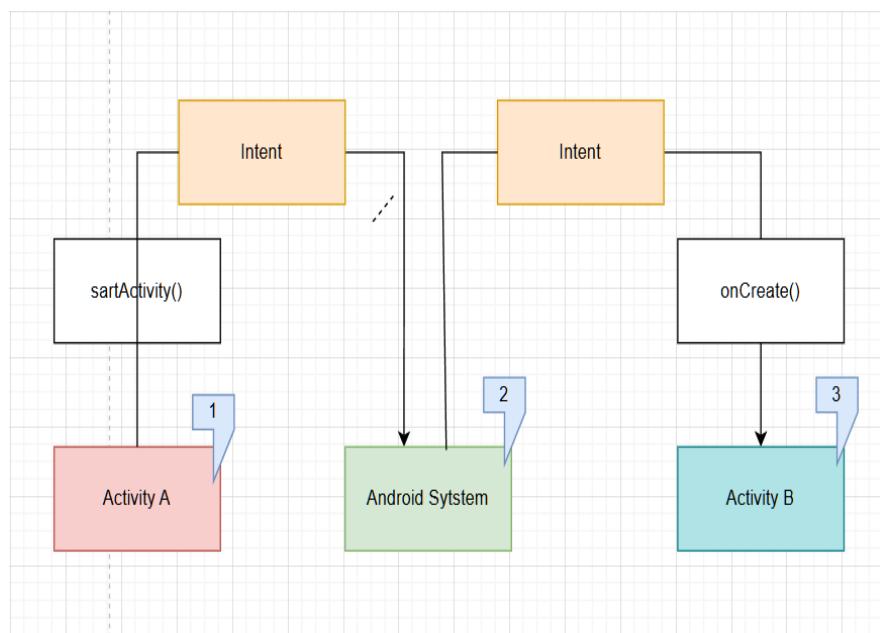
Intent có thể được coi là "ý định" của một ứng dụng khi thực hiện một hành động nào đó, chẳng hạn như khởi chạy một **Activity**, **Service**, phát một **Broadcast**, hoặc truyền tải dữ liệu giữa các thành phần. Intent không chỉ giúp tương tác giữa các thành phần nội bộ của ứng dụng mà còn có thể giao tiếp với các ứng dụng bên ngoài.

## Phân loại Intent

Intent trong Android được chia thành hai loại chính:

### ***Explicit Intent (Intent tường minh):***

Đây là loại Intent được sử dụng khi bạn muốn chỉ rõ thành phần đích cần khởi chạy trong ứng dụng của mình, chẳng hạn như một **Activity** hoặc **Service**. Intent tường minh cho phép ứng dụng trực tiếp chuyển điều khiển đến một thành phần cụ thể, thường được sử dụng khi các thành phần đều nằm trong cùng một ứng dụng. Bạn sẽ chỉ định rõ lớp của thành phần cần khởi chạy trong Intent này.



### ***Implicit Intent (Intent ngầm định):***

Loại Intent này được sử dụng khi bạn không chỉ rõ tên của thành phần cần khởi chạy mà thay vào đó bạn mô tả hành động mà bạn muốn thực hiện. Hệ thống Android sẽ dựa trên mô tả này để tìm kiếm ứng dụng hoặc thành phần nào có khả năng xử lý yêu cầu. Implicit Intent thường được dùng để thực hiện các tác vụ như mở một URL, gửi email, hoặc gọi điện thoại, và có thể được xử lý bởi nhiều ứng dụng khác nhau trên thiết bị.

## Cấu Trúc Của Intent

Intent bao gồm một số thành phần chính giúp xác định hành động và dữ liệu cần thực hiện. Các thành phần chính bao gồm:

- **Action (Hành động):** Đây là chuỗi mô tả hành động mà ứng dụng muốn thực hiện, chẳng hạn như hiển thị nội dung, gửi dữ liệu, hoặc thực hiện một cuộc gọi. Mỗi hành động có một mã định danh đặc biệt và được xác định trước bởi hệ thống Android.
- **Data (Dữ liệu):** Dữ liệu đi kèm với một Intent thường được đại diện dưới dạng **URI** (Uniform Resource Identifier) và chỉ định nội dung mà hành động sẽ áp dụng. Ví dụ, khi một ứng dụng yêu cầu mở một trang web, dữ liệu trong Intent sẽ chứa URL của trang web đó.
- **Category (Danh mục):** Một Intent có thể bao gồm một hoặc nhiều danh mục bổ sung giúp mô tả thêm về hành động mà nó sẽ thực hiện. Hệ thống Android sử dụng các danh mục này để xác định cách xử lý hành động mà Intent yêu cầu.
- **Extras (Dữ liệu bổ sung):** Đây là các cặp khóa-giá trị chứa dữ liệu bổ sung mà bạn muốn truyền cùng với Intent. Dữ liệu này giúp cung cấp thêm thông tin cho thành phần nhận Intent để thực hiện hành động theo cách cụ thể.
- **Flags (Cờ):** Cờ cho phép bạn kiểm soát cách hệ thống quản lý hoạt động của các thành phần khi Intent được khởi chạy. Ví dụ, bạn có thể yêu cầu hệ thống mở một **Activity** mới hoặc sử dụng lại một **Activity** đã tồn tại trong ngăn xếp.

## Các Ứng Dụng Chính Của Intent

- **Chuyển đổi giữa các Activity:** Intent là phương tiện chính để điều hướng giữa các màn hình trong ứng dụng Android. Khi người dùng cần chuyển từ một màn hình này sang màn hình khác, Intent sẽ giúp thực hiện quá trình chuyển đổi này một cách mượt mà.
- **Khởi động Service:** Intent cũng được sử dụng để khởi động các dịch vụ nền (background services) nhằm thực hiện các tác vụ chạy ngầm mà không tương tác trực tiếp với người dùng, chẳng hạn như tải xuống dữ liệu hoặc phát nhạc.
- **Phát thông báo Broadcast:** Intent có thể phát thông báo đến toàn bộ hệ thống hoặc một nhóm ứng dụng cụ thể. Các **BroadcastReceiver** sẽ lắng nghe các Intent này để thực hiện các hành động tương ứng. Đây là cách hệ thống Android truyền tải các sự

kiện hệ thống hoặc các thay đổi trạng thái như pin yếu, kết nối mạng thay đổi, hay tin nhắn SMS đến ứng dụng.

- **Chia sẻ và truyền dữ liệu:** Intent được sử dụng để truyền tải dữ liệu giữa các Activity hoặc Service trong ứng dụng. Điều này giúp các thành phần của ứng dụng có thể chia sẻ thông tin dễ dàng, chẳng hạn như kết quả từ một màn hình tìm kiếm đến một màn hình kết quả.

## **Tính Linh Hoạt Của Intent**

**Intent** trong Android không chỉ giúp giao tiếp giữa các thành phần trong cùng một ứng dụng mà còn có khả năng giao tiếp giữa các ứng dụng khác nhau. Ví dụ, khi bạn muốn mở một trang web, Intent có thể yêu cầu hệ thống mở một ứng dụng trình duyệt. Nếu có nhiều ứng dụng có khả năng xử lý yêu cầu, hệ thống sẽ cho người dùng lựa chọn ứng dụng nào họ muốn sử dụng để thực hiện tác vụ.

### **1.3.4. Giới thiệu về Activity trong Android Studio**

Activity là một trong những thành phần quan trọng nhất của ứng dụng Android. Nó đại diện cho một màn hình giao diện người dùng mà người dùng có thể tương tác. Mỗi activity thường chứa các thành phần giao diện như nút (Button), trường văn bản (TextView), hộp thoại (Dialog), và nhiều hơn nữa.

#### **1). Đặc điểm chính của Activity**

*Đại diện cho một màn hình giao diện:* Mỗi activity tương ứng với một màn hình. Khi người dùng mở ứng dụng, họ thường bắt đầu tương tác với một activity cụ thể.

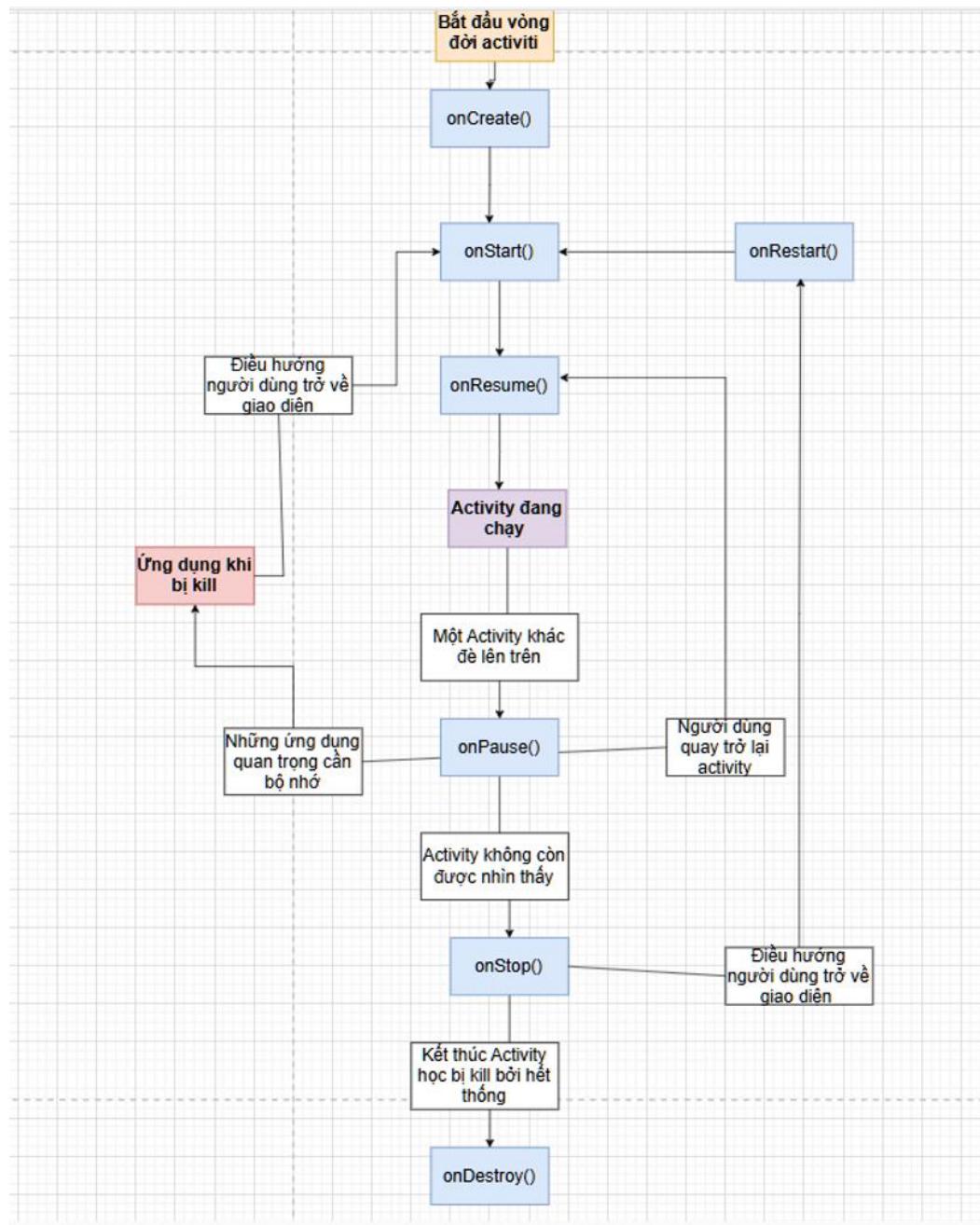
*Vòng đời của Activity:* Mỗi activity trong Android có một vòng đời (lifecycle) riêng. Nó bao gồm các trạng thái từ lúc khởi tạo, hoạt động, dừng lại và cuối cùng bị hủy. Các phương thức vòng đời như onCreate(), onStart(), onResume(), onPause(), onStop(), và onDestroy() giúp bạn quản lý và điều chỉnh hành vi của activity tại các thời điểm khác nhau.

*Giao tiếp với người dùng:* Activity chịu trách nhiệm cung cấp giao diện mà người dùng tương tác. Các tương tác này có thể bao gồm nhấn nút, nhập văn bản, hoặc điều hướng giữa các activity.

#### **2). Các phương thức vòng đời Activity trong Android**

Vòng đời của Activity trong Android là một chuỗi các phương thức mà hệ điều hành Android gọi để thông báo cho ứng dụng về các thay đổi trong trạng thái của Activity. Việc

năm vững vòng đời của Activity là rất quan trọng để quản lý tài nguyên và đảm bảo trải nghiệm người dùng mượt mà. Dưới đây là các phương thức chính trong vòng đời của Activity:



### [1].onCreate(Bundle savedInstanceState):

**Mô tả:** Được gọi khi Activity lần đầu tiên được tạo ra. Đây là nơi khởi tạo các thành phần của giao diện người dùng, thiết lập cấu hình ban đầu, và khôi phục trạng thái trước đó (nếu có).

**Ví dụ sử dụng:** Khởi tạo các Views, đặt các sự kiện (event listeners), và khôi phục trạng thái từ savedInstanceState.

#### [2].onStart():

- *Mô tả:* Được gọi ngay sau onCreate(). Activity đang trở nên có thể nhìn thấy nhưng chưa ở trạng thái tương tác với người dùng.
- *Ví dụ sử dụng:* Bắt đầu các hoạt động cần thiết mà Activity không cần khi Activity đang bị ẩn, như việc đăng ký các dịch vụ.

#### [3].onResume():

- *Mô tả:* Được gọi khi Activity bắt đầu tương tác với người dùng. Đây là thời điểm Activity trở nên tương tác và nổi bật trên màn hình.
- *Ví dụ sử dụng:* Khôi phục các thay đổi trong dữ liệu và tiếp tục hoạt động mà cần đến tương tác của người dùng, như bắt đầu phát video hoặc âm thanh.

#### [4].onPause():

- *Mô tả:* Được gọi khi Activity không còn tương tác trực tiếp với người dùng nhưng vẫn hiển thị trên màn hình. Đây là cơ hội để thực hiện các hành động cần thiết trước khi Activity bị dừng hoặc tạm ngưng.
- *Ví dụ sử dụng:* Lưu trạng thái hiện tại, dừng các hoạt động có thể tiếp tục sau đó, như tạm dừng phát video hoặc lưu lại các dữ liệu thay đổi.

#### [5].onStop():

- *Mô tả:* Được gọi khi Activity không còn hiển thị trên màn hình. Đây là thời điểm để giải phóng tài nguyên không cần thiết và lưu trạng thái dài hạn.
- *Ví dụ sử dụng:* Giải phóng các tài nguyên nặng như cơ sở dữ liệu hoặc dịch vụ mạng không cần thiết, lưu trữ trạng thái cho lần khởi động tiếp theo.

#### [6].onRestart():

- *Mô tả:* Được gọi khi Activity trở lại trạng thái hoạt động sau khi đã dừng. Đây là một phần của quá trình chuyển từ trạng thái dừng (Stopped) sang trạng thái hoạt động (Started).
- *Ví dụ sử dụng:* Khôi phục hoặc làm mới dữ liệu trước khi Activity tiếp tục tương tác với người dùng.

### [7].onDestroy():

- *Mô tả:* Được gọi khi Activity bị hủy, thường là khi người dùng rời khỏi ứng dụng hoặc khi hệ điều hành cần giải phóng tài nguyên.
- *Ví dụ sử dụng:* Thực hiện các công việc dọn dẹp, như giải phóng tài nguyên còn lại, đóng các kết nối mạng hoặc lưu lại trạng thái cuối cùng.

### 3) Cách tạo một Activity

- Khi tạo một activity mới trong Android Studio:
- Nhấn chuột phải vào thư mục java trong Android Studio.
- Chọn New -> Activity.
- Chọn loại activity phù hợp (ví dụ: Empty Activity, Basic Activity).
- Android Studio sẽ tự động tạo file Java hoặc Kotlin cho activity cùng với file giao diện XML tương ứng.

#### 1.3.5. Cơ sở dữ liệu trong Android

##### SQLite

SQLite là một cơ sở dữ liệu nhẹ, tích hợp sẵn trong Android, không yêu cầu một server cơ sở dữ liệu riêng biệt. SQLite cho phép lưu trữ dữ liệu ứng dụng một cách hiệu quả và nhanh chóng .



Vai trò: SQLite được sử dụng để lưu trữ các dữ liệu quan trọng như thông tin người dùng, dữ liệu giao dịch, và cấu hình ứng dụng. Nó cho phép thực hiện các thao tác cơ bản như thêm, sửa, xóa, và truy vấn dữ liệu.

Các hoạt động cơ bản: Thực hiện các thao tác cơ bản như thêm, sửa, xóa và truy vấn dữ liệu sử dụng các lệnh SQL cơ bản: Sử dụng các lệnh SQL để thực hiện các thao tác như thêm (INSERT), sửa (UPDATE), xóa (DELETE), và truy vấn (SELECT) dữ liệu. Sử dụng lớp SQLiteOpenHelper để quản lý vòng đời của cơ sở dữ liệu.

Sử dụng SQLiteOpenHelper: Để quản lý cơ sở dữ liệu và xử lý các phiên bản cơ sở dữ liệu.

##### Room Database

Room là một thư viện của Android giúp làm việc với cơ sở dữ liệu SQLite dễ dàng hơn bằng cách cung cấp một lớp trừu tượng và các tính năng hữu ích như hỗ trợ các truy vấn SQL và ánh xạ dữ liệu tự động.

Hỗ trợ truy vấn SQL: Room hỗ trợ viết các truy vấn SQL trực tiếp và ánh xạ kết quả vào các đối tượng Java hoặc Kotlin, giúp đơn giản hóa việc tương tác với cơ sở dữ liệu.

Các lớp Entity, DAO, Database:

- Entity: Định nghĩa các bảng cơ sở dữ liệu dưới dạng các lớp dữ liệu với các thuộc tính đại diện cho các cột trong bảng.
- DAO (Data Access Object): Cung cấp các phương thức để thực hiện các thao tác trên cơ sở dữ liệu như truy vấn, thêm, sửa, và xóa dữ liệu.
- Database: Cung cấp một lớp cơ sở dữ liệu chính, quản lý vòng đời của cơ sở dữ liệu và cung cấp các đối tượng DAO.
- Tích hợp Room: Thêm các phụ thuộc Room vào tệp build.gradle của dự án, tạo các lớp Entity, DAO, và Database, và sử dụng Room để thực hiện các thao tác cơ sở dữ liệu. Cài đặt và cấu hình Room để dễ dàng quản lý cơ sở dữ liệu và tương tác với dữ liệu.

#### 1.2.4. Các công nghệ bổ sung

##### Firebase

Firebase là nền tảng phát triển ứng dụng của Google cung cấp một loạt các dịch vụ backend như xác thực người dùng, lưu trữ dữ liệu, và thông báo đẩy, giúp tăng tốc phát triển và triển khai ứng dụng.

Các dịch vụ chính:

- Xác thực người dùng: Cung cấp các dịch vụ xác thực mạnh mẽ, hỗ trợ nhiều phương thức như đăng nhập bằng email và mật khẩu, xác thực qua mạng xã hội, và xác thực qua số điện thoại.
- Lưu trữ dữ liệu: Firebase Realtime Database và Firestore cho phép lưu trữ và đồng bộ hóa dữ liệu trong thời gian thực giữa các thiết bị.



- Thông báo đẩy: Firebase Cloud Messaging (FCM) cung cấp dịch vụ thông báo đẩy để gửi thông báo đến người dùng trên các thiết bị di động và web.

## **API và Web Services**

API (Application Programming Interface) là tập hợp các giao diện lập trình cho phép các ứng dụng giao tiếp với nhau. Trong phát triển ứng dụng di động, API thường được sử dụng để kết nối và lấy dữ liệu từ các dịch vụ web và hệ thống bên ngoài.

### Kết nối với Web Services

- Sử dụng API: Kết nối ứng dụng với các dịch vụ web thông qua các yêu cầu HTTP, cho phép gửi và nhận dữ liệu từ các máy chủ từ xa.
- Xử lý dữ liệu từ xa: Xử lý dữ liệu trả về từ các dịch vụ web, bao gồm phân tích JSON hoặc XML và cập nhật giao diện người dùng hoặc cơ sở dữ liệu ứng dụng.

### Thực hiện yêu cầu mạng:

- Gửi yêu cầu HTTP: Sử dụng các thư viện như Retrofit hoặc Volley để gửi các yêu cầu HTTP và xử lý phản hồi từ server. Các thư viện này giúp đơn giản hóa việc thực hiện các yêu cầu mạng và xử lý dữ liệu trả về.
- Xử lý phản hồi: Phân tích dữ liệu JSON hoặc XML trả về từ server và thực hiện các hành động cần thiết, chẳng hạn như cập nhật giao diện người dùng hoặc lưu trữ dữ liệu vào cơ sở dữ liệu.

## CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Phân tích yêu cầu

#### 2.1.1. Xác định yêu cầu chức năng

##### Chức năng Đăng ký

Chức năng đăng ký cho phép nhân viên tạo tài khoản mới để truy cập vào hệ thống quản lý bàn của quán Billiards. Các yêu cầu cụ thể cho chức năng này bao gồm:

- *Thông tin cần thiết:*
  - Nhân viên sẽ cần nhập các thông tin cá nhân như:
    - Tài Khoản
    - Mật khẩu
- *Kiểm tra tính hợp lệ:*
  - Hệ thống sẽ kiểm tra tính hợp lệ của thông tin đã nhập.
  - Mật khẩu cần có độ phức tạp tối thiểu, ví dụ: ít nhất 6 ký tự bao gồm cả chữ cái và số.
- *Xác nhận đăng ký:*
  - Sau khi hoàn thành thông tin, hệ thống sẽ gửi một thông báo xác nhận đăng ký thành công đến nhân viên.

##### Chức năng Đăng nhập

Chức năng đăng nhập cho phép nhân viên truy cập vào hệ thống bằng tài khoản đã đăng ký. Các yêu cầu cụ thể cho chức năng này bao gồm:

- *Thông tin đăng nhập:*
  - Nhân viên sẽ cần nhập:
    - Tên người dùng
    - Mật khẩu
- *Xác thực thông tin:*
  - Hệ thống sẽ xác thực thông tin đăng nhập và chỉ cho phép truy cập khi thông tin chính xác.
  - Nếu thông tin không chính xác, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại.

- *Khóa tài khoản:*
  - Nếu đăng nhập sai quá số lần quy định, tài khoản có thể bị khóa tạm thời để bảo vệ an toàn.
- *Tùy chọn lấy lại mật khẩu:*
  - Nếu nhân viên quên mật khẩu, họ có thể yêu cầu lấy lại mật khẩu thông qua email hoặc số điện thoại đã đăng ký.

 Chức năng quản lý bàn:

Chức năng quản lý bàn cho phép nhân viên quản Billiards theo dõi và quản lý tình trạng các bàn chơi một cách hiệu quả. Các yêu cầu cụ thể cho chức năng này bao gồm:

- Danh sách bàn: Hiển thị danh sách tất cả các bàn trong quán cùng với thông tin chi tiết như trạng thái (trống, đang sử dụng, bảo trì) và số lượng người tối đa cho mỗi bàn.
- Cập nhật trạng thái bàn: Nhân viên có thể cập nhật trạng thái của bàn, ví dụ: chuyển bàn từ trạng thái "trống" sang "đang sử dụng" khi có khách hàng ngồi chơi hoặc chuyển sang "bảo trì" khi bàn cần sửa chữa.
- Thông tin chi tiết bàn: Cho phép nhân viên xem thông tin chi tiết về từng bàn, bao gồm vị trí, loại bàn (billiards, snooker, etc.) và thời gian sử dụng.
- Theo dõi thời gian sử dụng: Ghi nhận thời gian mà khách hàng đã sử dụng bàn để tính phí chính xác và dễ dàng quản lý các dịch vụ bổ sung khác.

 Chức năng quản lý sản phẩm:

Chức năng quản lý sản phẩm giúp quán Billiards theo dõi và quản lý các sản phẩm dịch vụ mà quán cung cấp. Các yêu cầu cụ thể cho chức năng này bao gồm:

- Thông tin sản phẩm: Quản lý thông tin về các sản phẩm và dịch vụ như bàn Billiards, đồ uống, đồ ăn, và các dịch vụ giải trí khác.
- Quản lý giá cả: Thiết lập và cập nhật giá cho từng sản phẩm, đảm bảo giá cả luôn chính xác và phù hợp với thị trường.
- Theo dõi tồn kho: Theo dõi số lượng sản phẩm trong kho để đảm bảo luôn có đủ hàng phục vụ khách hàng, đồng thời đưa ra cảnh báo khi hàng hóa sắp hết.

#### Chức năng quản lý khách hàng:

Chức năng quản lý khách hàng cho phép quán Billiards theo dõi và quản lý thông tin khách hàng một cách hiệu quả. Các yêu cầu cụ thể cho chức năng này bao gồm:

- Thông tin khách hàng: Lưu trữ thông tin cá nhân của khách hàng như họ tên, số điện thoại, địa chỉ email và lịch sử giao dịch.
- Quản lý khách hàng thân thiết: Tạo danh sách khách hàng thân thiết để cung cấp các ưu đãi, chương trình khuyến mãi và quản lý điểm thưởng.
- Theo dõi lịch sử sử dụng bàn: Ghi nhận lịch sử sử dụng bàn của khách hàng để cải thiện dịch vụ và cá nhân hóa trải nghiệm cho từng khách hàng.
- Đánh giá và phản hồi: Cho phép khách hàng đánh giá trải nghiệm của họ và gửi phản hồi, giúp quán nâng cao chất lượng dịch vụ.

#### Chức năng quản lý hóa đơn

Chức năng quản lý hóa đơn cho phép quán Billiards theo dõi và quản lý tất cả các giao dịch thanh toán. Các yêu cầu cụ thể cho chức năng này bao gồm:

- Tạo hóa đơn: Hệ thống tự động tạo hóa đơn cho mỗi giao dịch thanh toán, bao gồm thông tin chi tiết như tên khách hàng, danh sách sản phẩm và dịch vụ đã sử dụng, thời gian sử dụng bàn, tổng chi phí, và phương thức thanh toán.
- Lưu trữ hóa đơn: Hóa đơn sẽ được lưu trữ trong hệ thống để dễ dàng truy cập và tra cứu sau này, bao gồm khả năng lưu trữ cả hóa đơn điện tử (PDF).
- Xem và in hóa đơn: Nhân viên có thể dễ dàng xem lại hóa đơn đã tạo và có khả năng in hóa đơn để đưa cho khách hàng.
- Quản lý lịch sử hóa đơn: Cung cấp chức năng để tra cứu lịch sử hóa đơn dựa trên các tiêu chí như ngày tháng, tên khách hàng hoặc trạng thái thanh toán (đã thanh toán, chưa thanh toán).
- Chỉnh sửa và hủy hóa đơn: Nhân viên có thể chỉnh sửa thông tin hóa đơn (trong trường hợp có sai sót) hoặc hủy hóa đơn nếu cần thiết, đồng thời ghi nhận lý do hủy.

- Báo cáo doanh thu: Cung cấp chức năng tạo báo cáo doanh thu từ hóa đơn theo từng khoảng thời gian (hàng ngày, hàng tuần, hàng tháng), giúp chủ quán dễ dàng theo dõi và phân tích tình hình kinh doanh.
- Thông kê hóa đơn: Cung cấp thông tin thống kê về số lượng hóa đơn, tổng doanh thu, số hóa đơn chưa thanh toán và các chỉ số liên quan khác để hỗ trợ ra quyết định kinh doanh.

### **2.1.2. Xác định yêu cầu phi chức năng**

 Hiệu suất:

Yêu cầu về hiệu suất của ứng dụng bao gồm:

- Thời gian phản hồi: Ứng dụng phải đảm bảo thời gian phản hồi dưới 2 giây cho các thao tác cơ bản như quản lý bàn, thanh toán và xem thông tin.
- Khả năng xử lý đồng thời: Ứng dụng phải có khả năng xử lý tối thiểu 100 người dùng đồng thời mà không bị giảm hiệu suất.

 Bảo mật:

Các yêu cầu bảo mật của ứng dụng bao gồm:

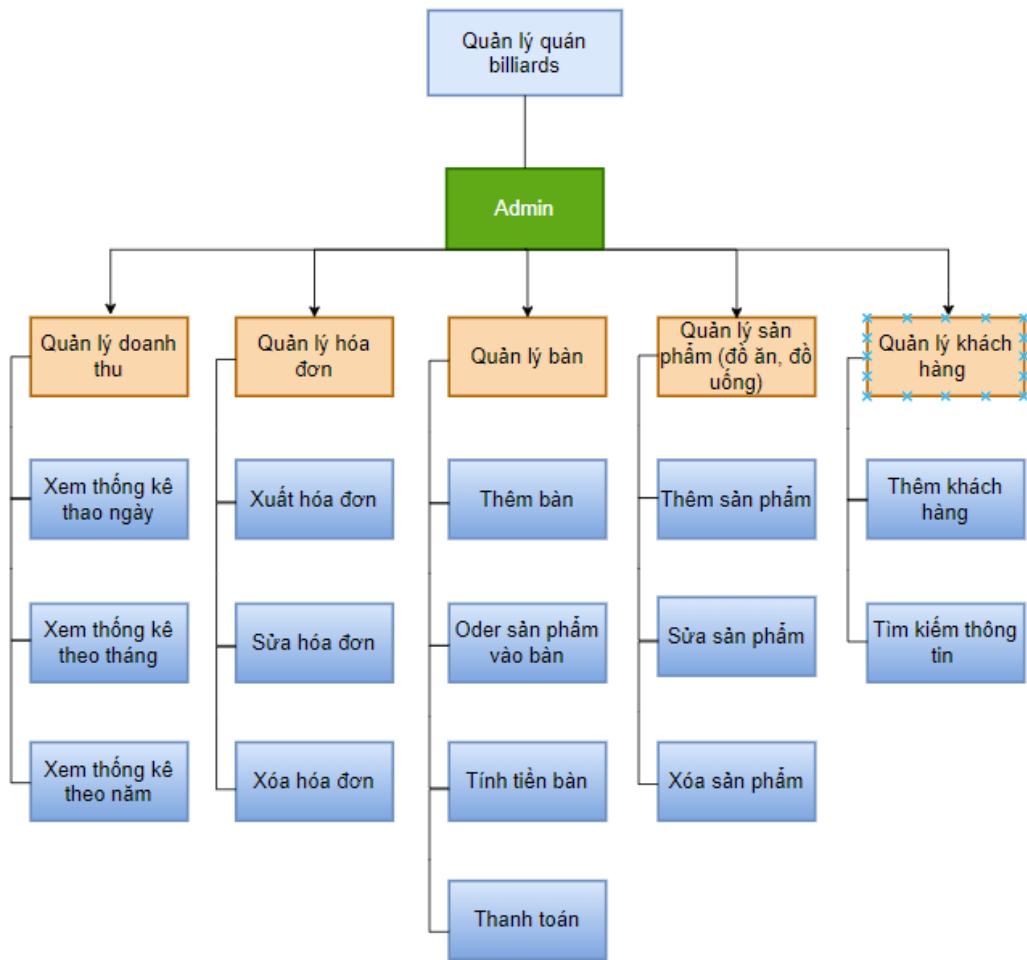
- Bảo vệ dữ liệu người dùng: Mọi thông tin cá nhân và lịch sử giao dịch của khách hàng phải được mã hóa và bảo vệ.
- Thông tin thanh toán: Đảm bảo tất cả thông tin thanh toán được xử lý qua giao thức bảo mật (HTTPS) và tuân thủ các tiêu chuẩn bảo mật dữ liệu (PCI-DSS).
- Quản lý quyền truy cập: Các tài khoản nhân viên phải có mức độ truy cập khác nhau tùy thuộc vào vai trò, và có thẻ yêu cầu xác thực hai yếu tố (2FA) để tăng cường bảo mật.

 Tính khả dụng:

Yêu cầu về tính khả dụng của ứng dụng bao gồm:

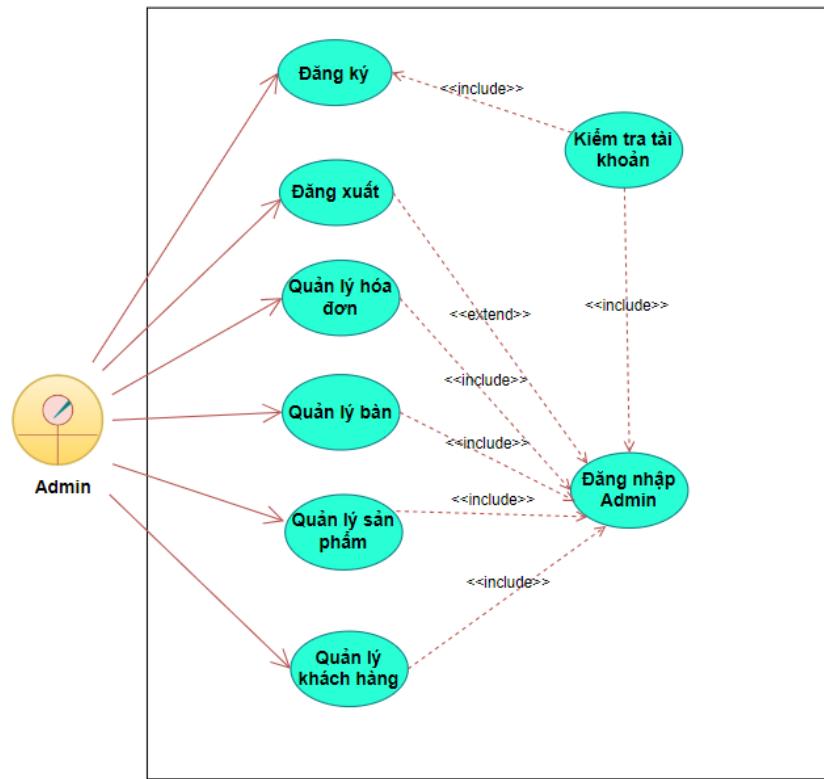
- Hoạt động liên tục: Ứng dụng cần đảm bảo thời gian hoạt động tối thiểu 99.5% trong tháng, có kế hoạch bảo trì và khắc phục sự cố kịp thời.
- Khả năng phục hồi: Ứng dụng cần có khả năng phục hồi nhanh chóng sau khi gặp sự cố, với các phương án sao lưu và phục hồi dữ liệu được thực hiện định kỳ.

## 2.2. Mô hình phân rã chức năng



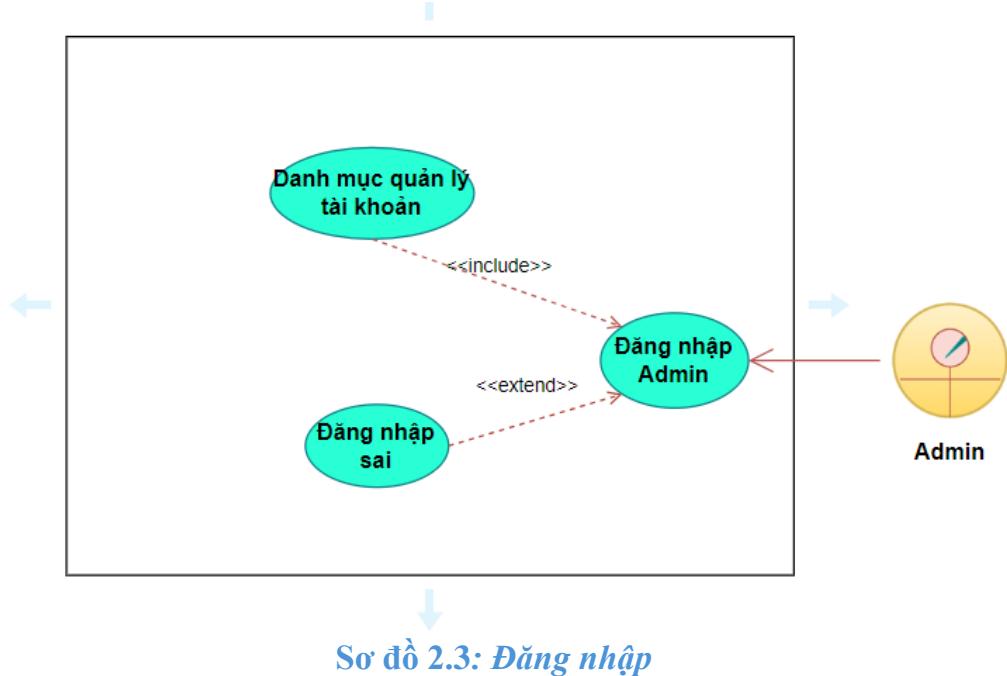
Sơ đồ 2.1: Mô hình phân rã chức năng

### 2.2.1. Biểu đồ Usecase



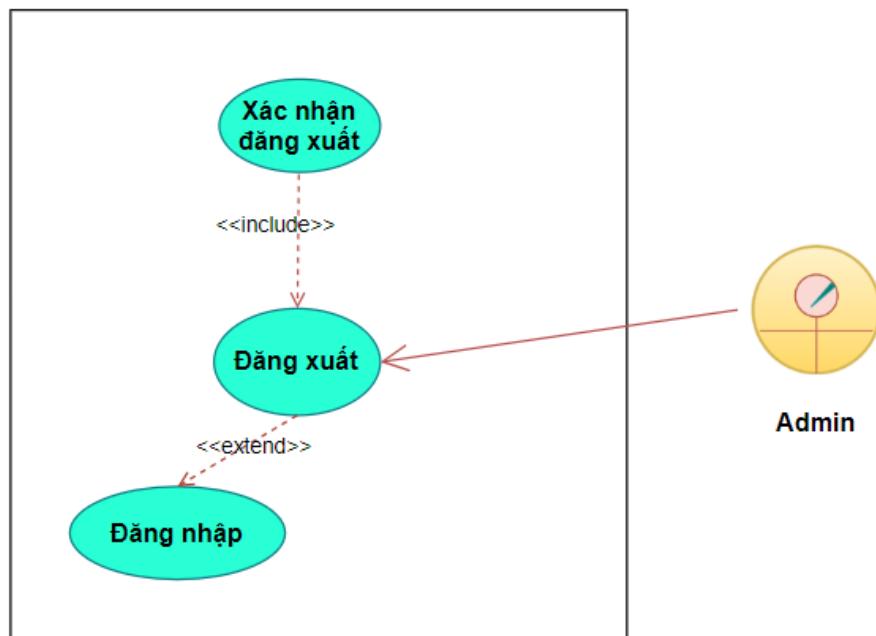
Sơ đồ 2.2: Usecase Tổng quát

✚ Đăng nhập



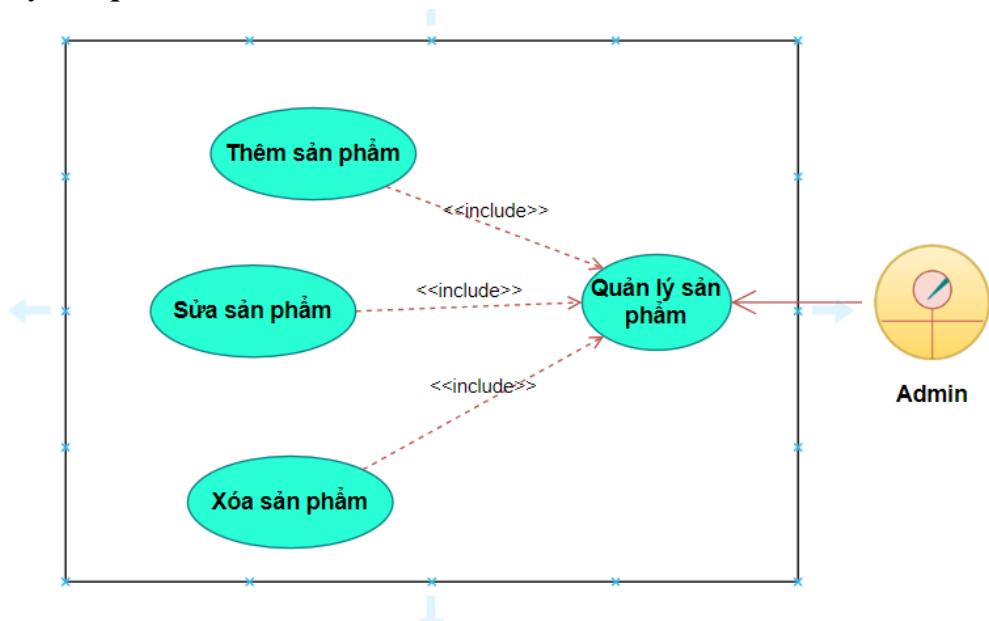
Sơ đồ 2.3: Đăng nhập

## ✚ Đăng xuất



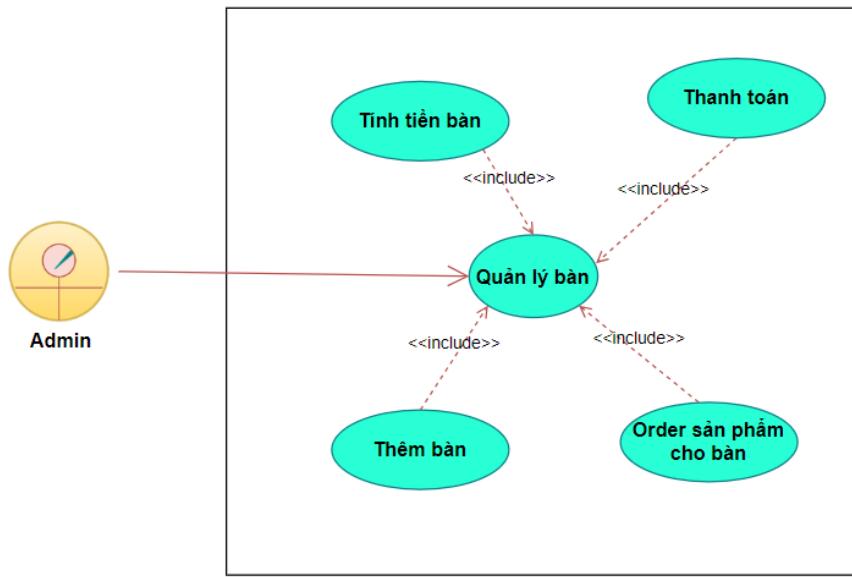
Sơ đồ 2.4: Usecase Đăng xuất

## ✚ Quản lý sản phẩm



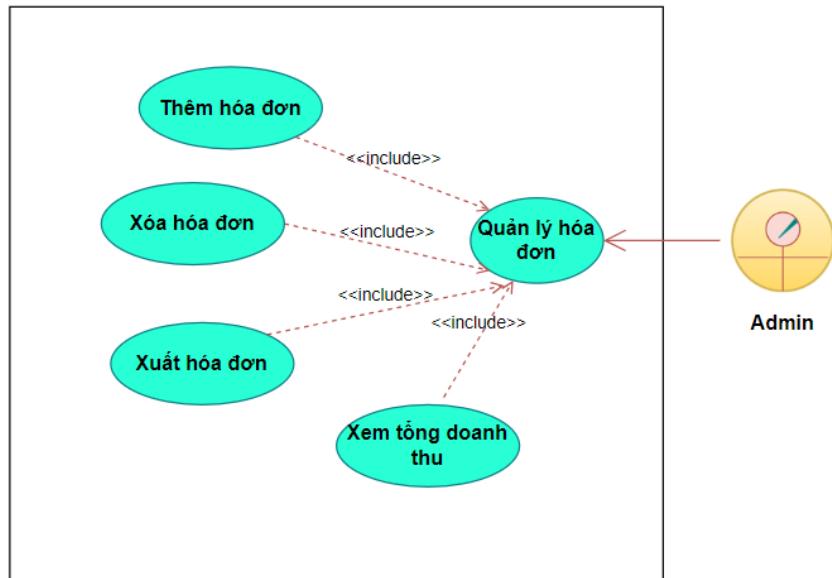
Sơ đồ 2.5: Usecase Quản lý sản phẩm

## Quản lý bàn



Sơ đồ 2.6: Usecase Quản lý bàn

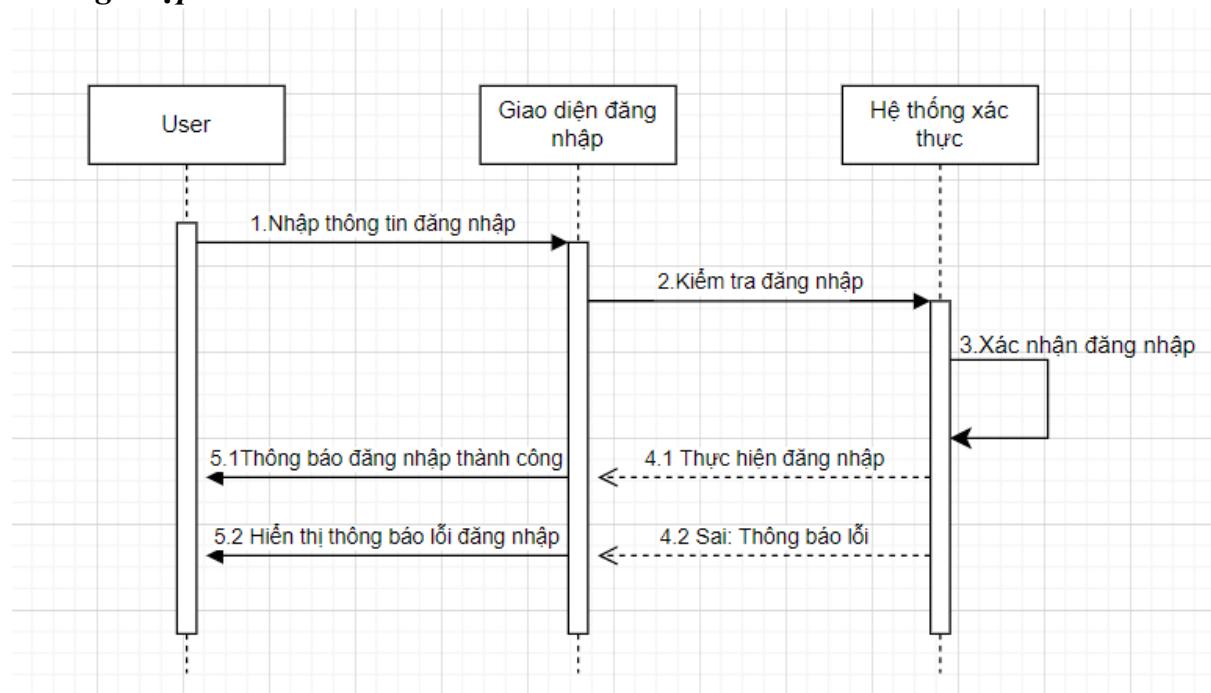
## Quản lý hóa đơn



Sơ đồ 2.7: Usecase Quản lý hóa đơn

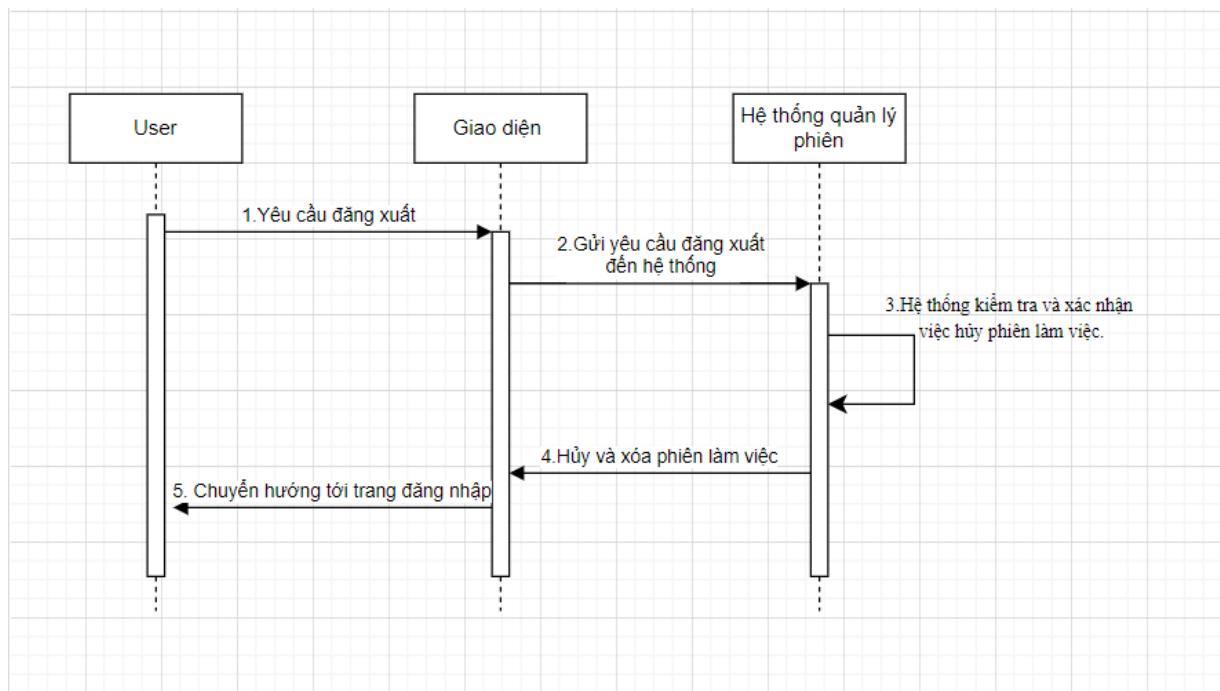
## 2.2.2. Biểu đồ tuần tự

### Đăng nhập



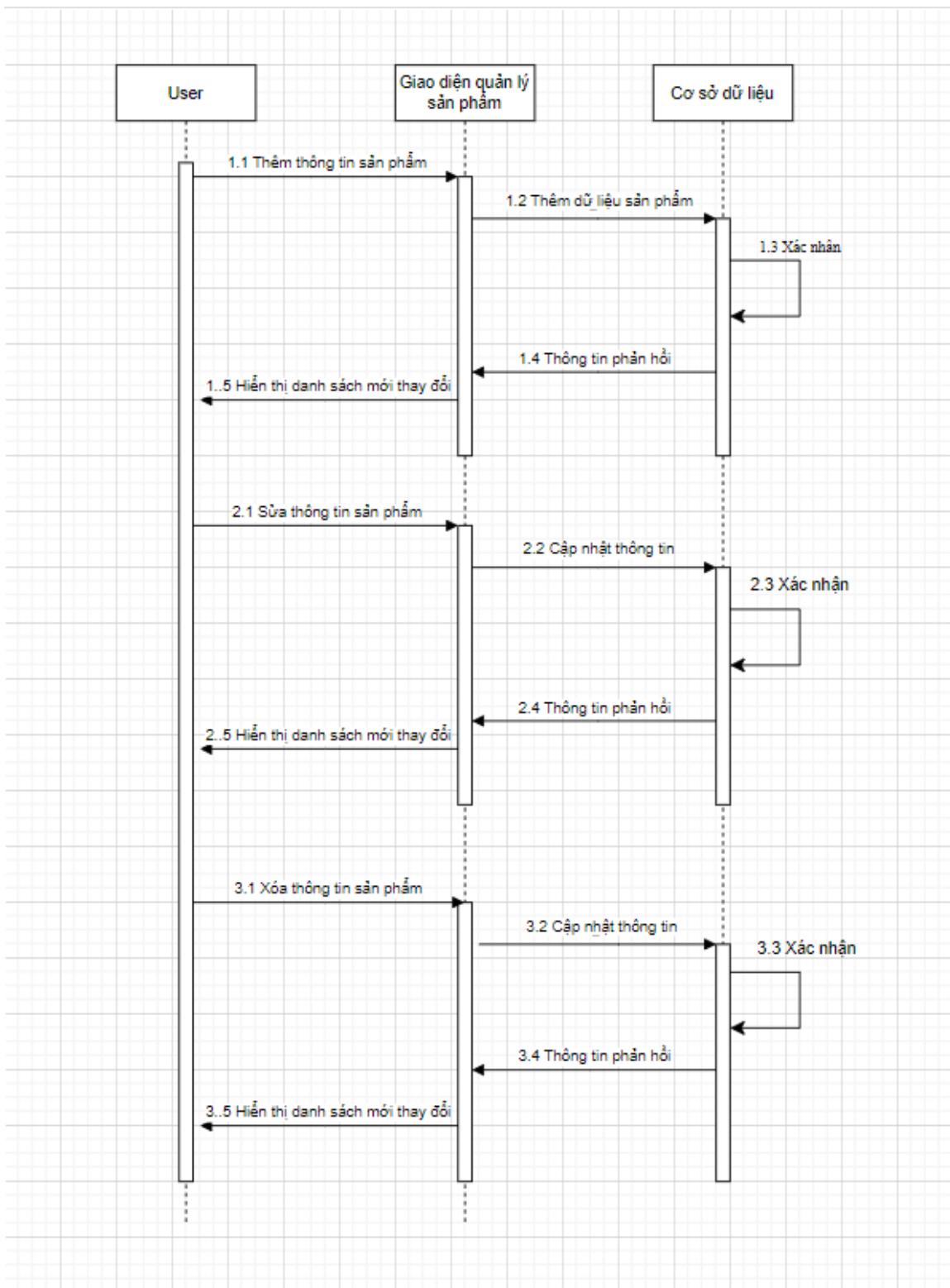
Biểu đồ 2.1: Biểu đồ tuần tự đăng nhập

### Đăng xuất



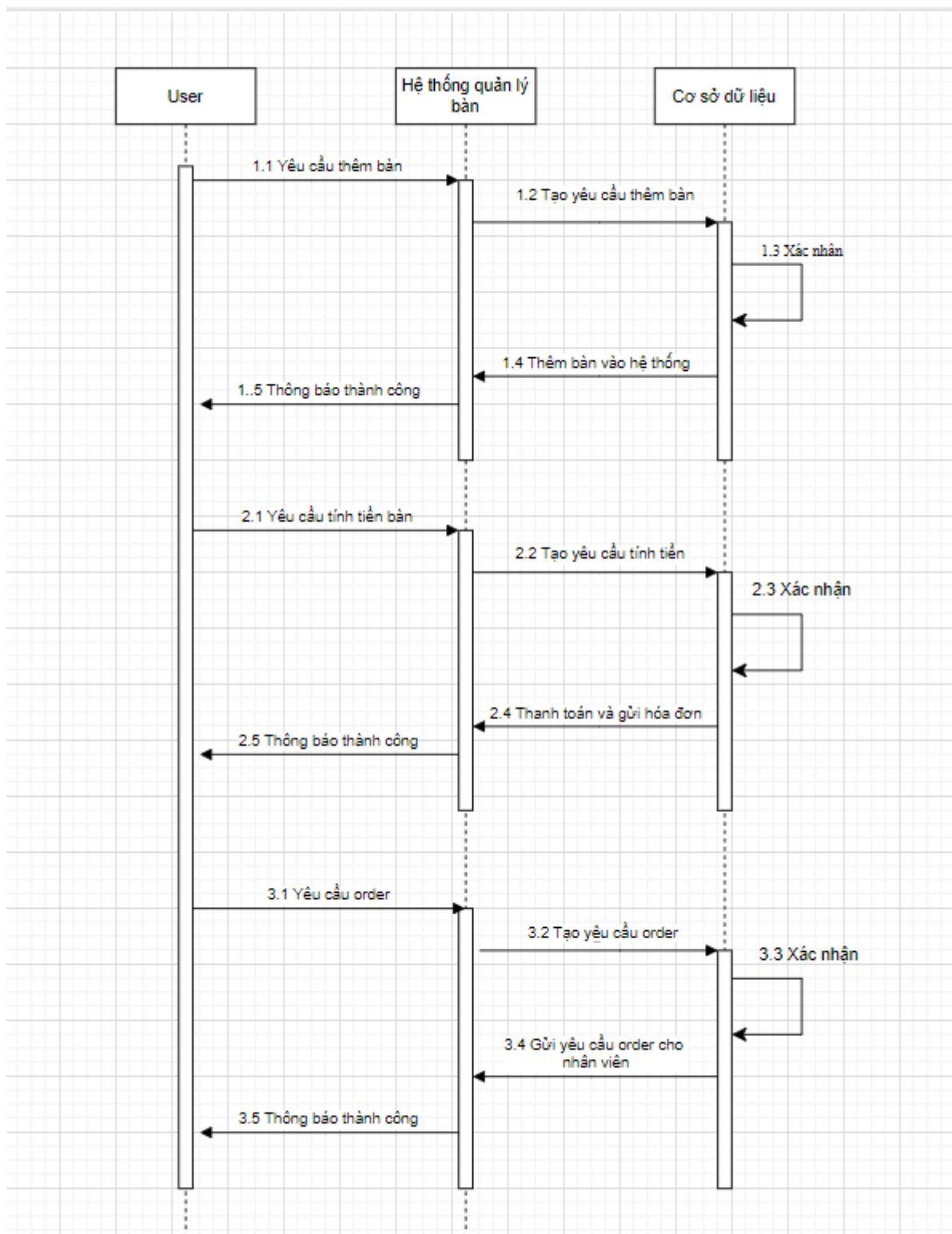
Biểu đồ 2.2: Biểu đồ tuần tự đăng xuất

## Quản lý sản phẩm



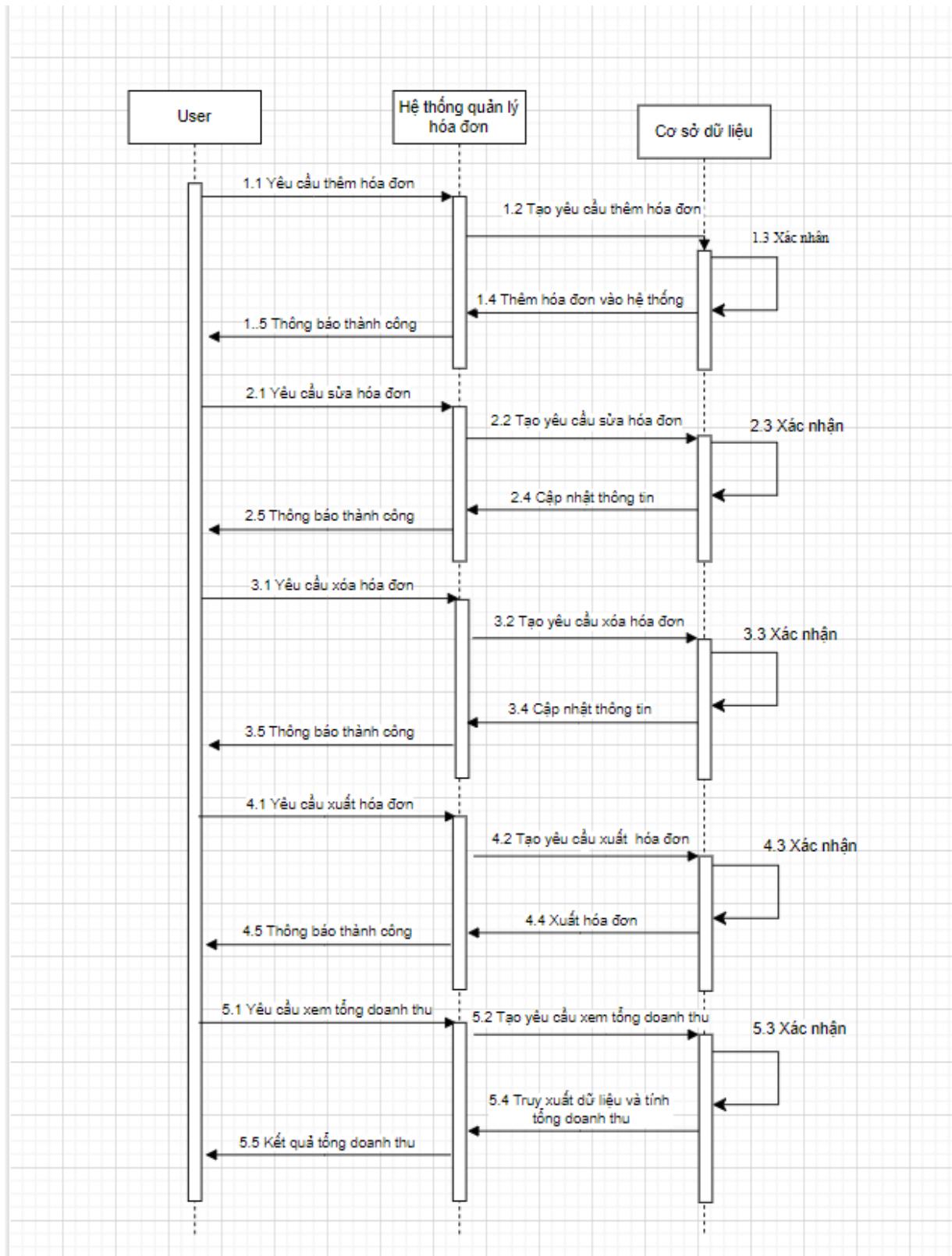
*Biểu đồ 2.3: Biểu đồ tuần tự quản lý sản phẩm*

## Quản lý bàn



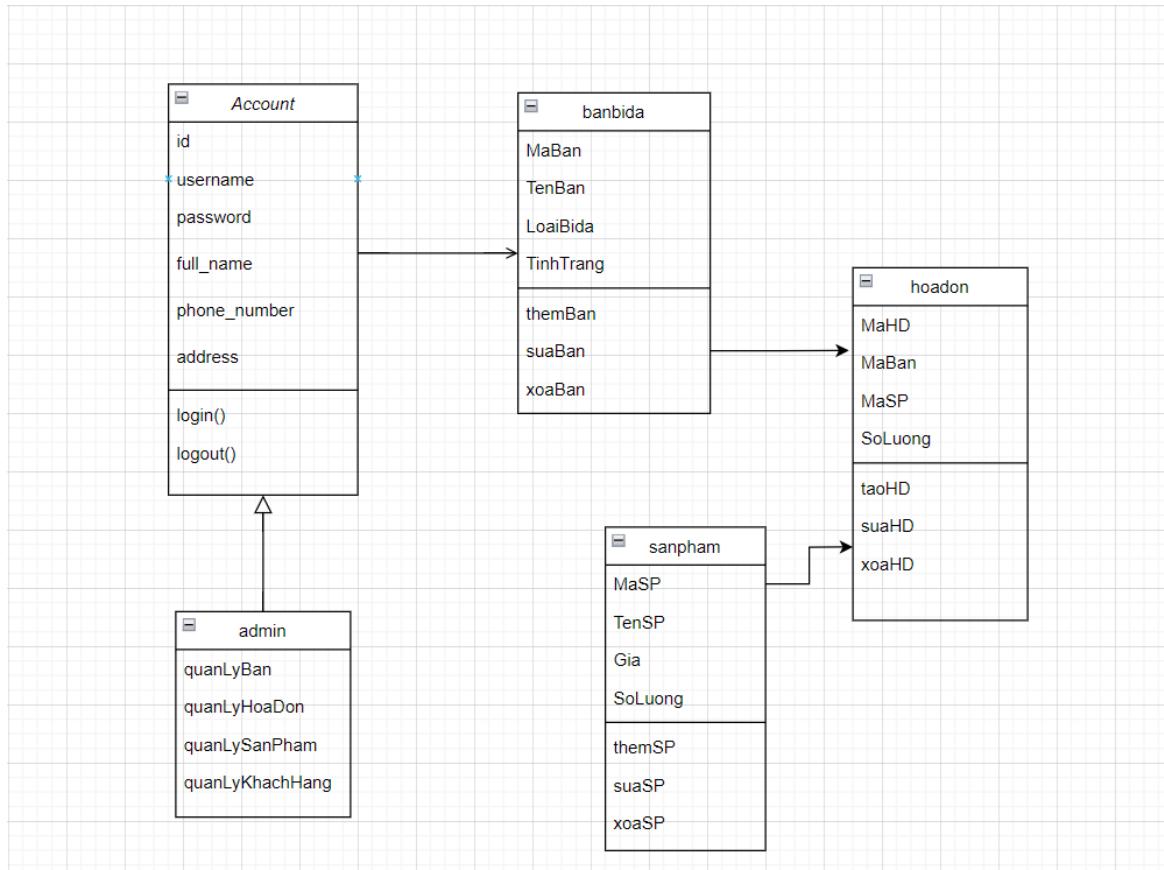
Biểu đồ 2.4: Biểu đồ tuần tự quản lý bàn

## Quản lý hóa đơn



Biểu đồ 2.5: Biểu đồ tuần tự quản lý hóa đơn

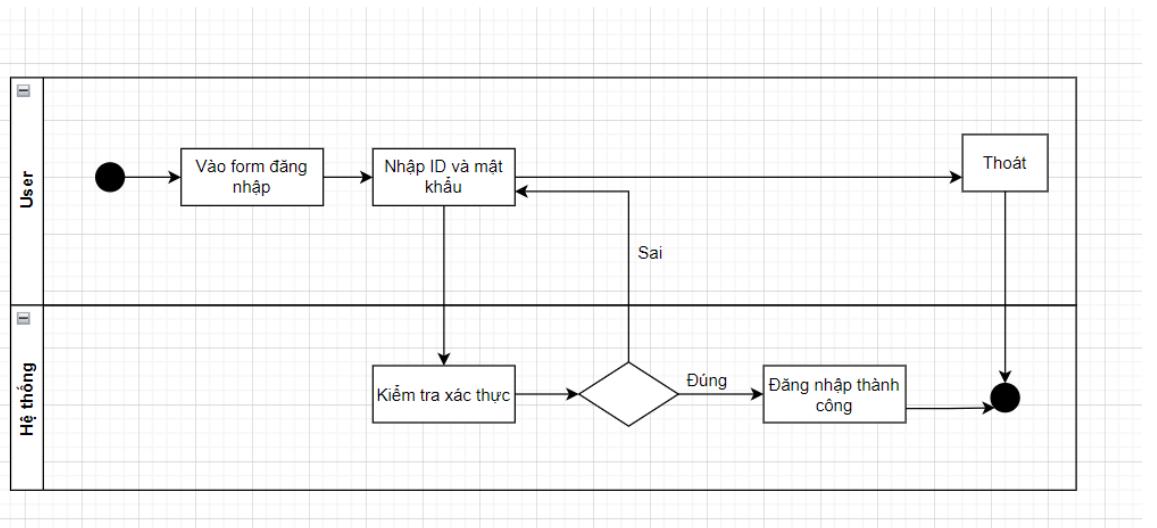
### 2.2.3. Biểu đồ lớp



Biểu đồ 2.6: Biểu đồ lớp

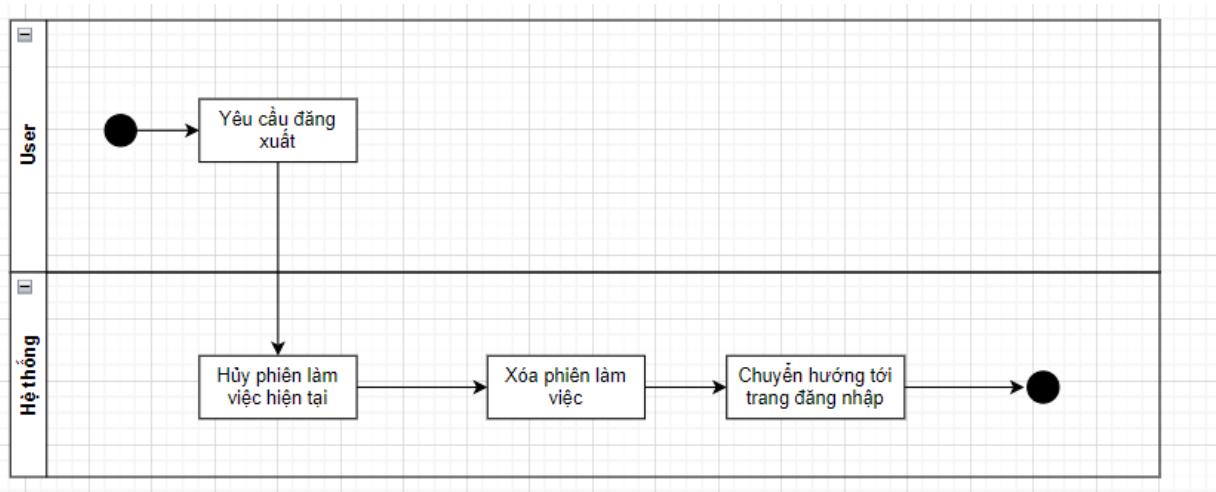
#### 2.2.4. Biểu đồ hoạt động

##### Đăng nhập



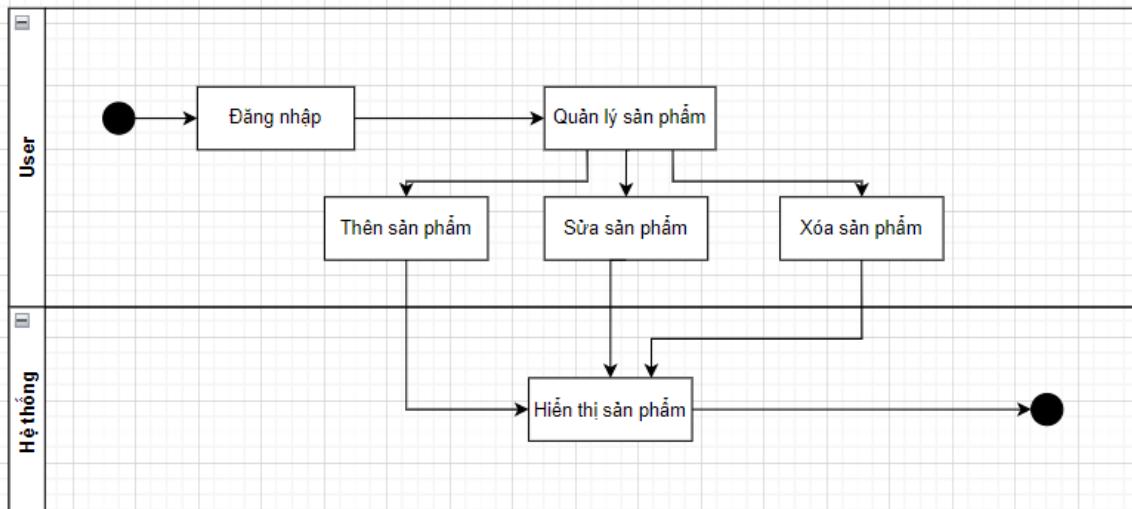
Biểu đồ 2.7: Biểu đồ hoạt động đăng nhập

##### Đăng xuất



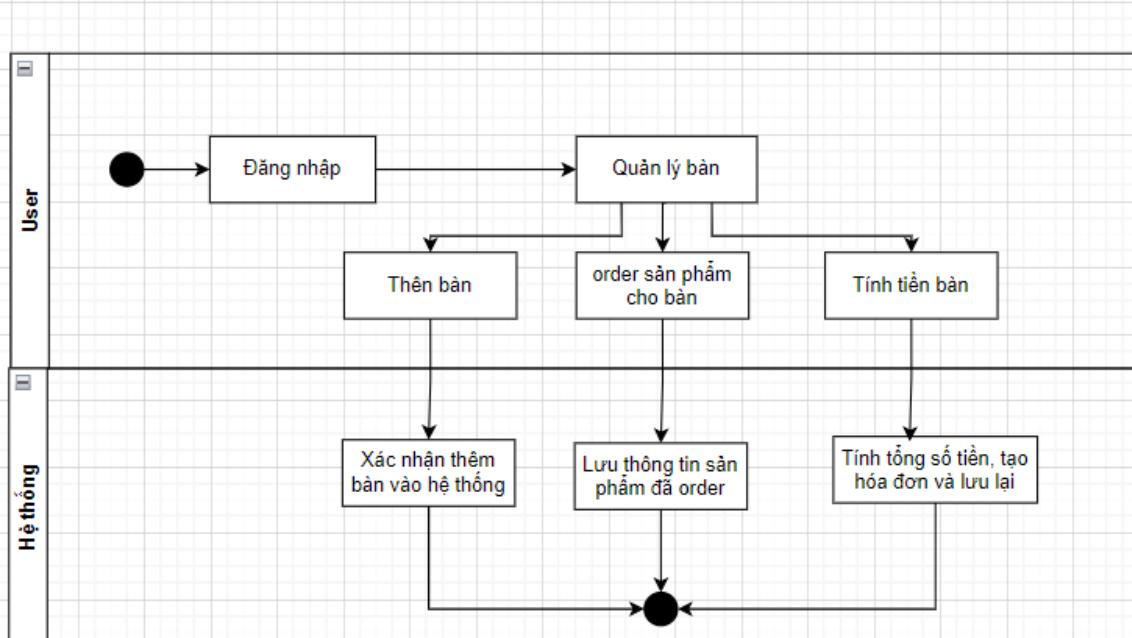
Biểu đồ 2.8: Biểu đồ hoạt động đăng xuất

## ✚ Quản lý sản phẩm



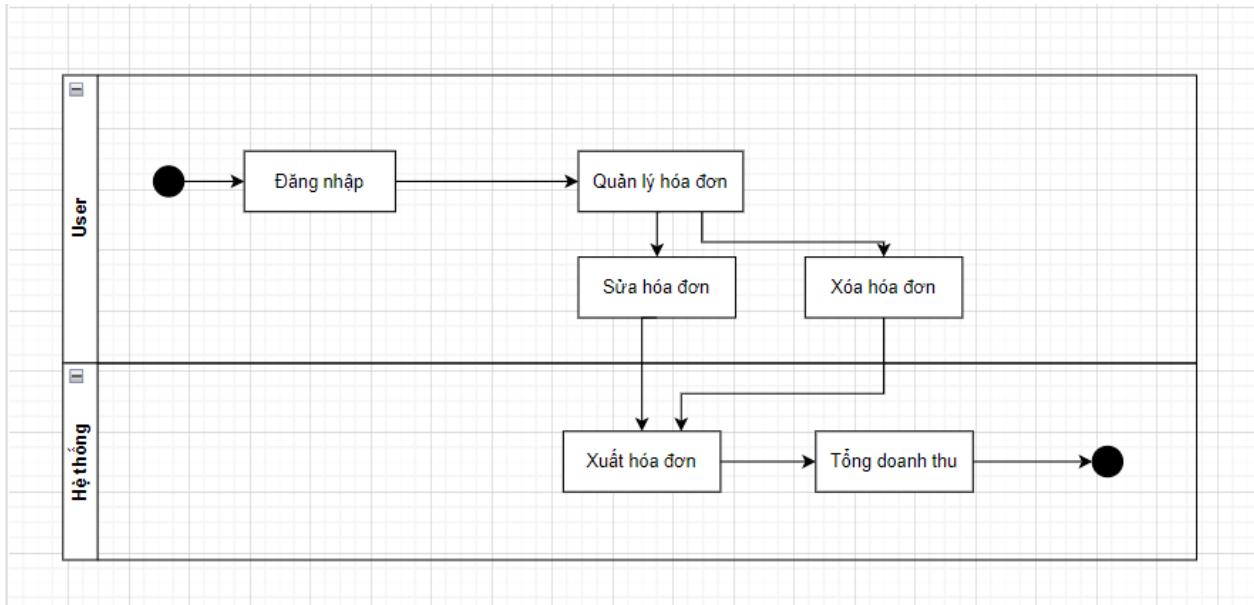
*Biểu đồ 2.9: Biểu đồ hoạt động quản lý sản phẩm*

## ✚ Quản lý bàn



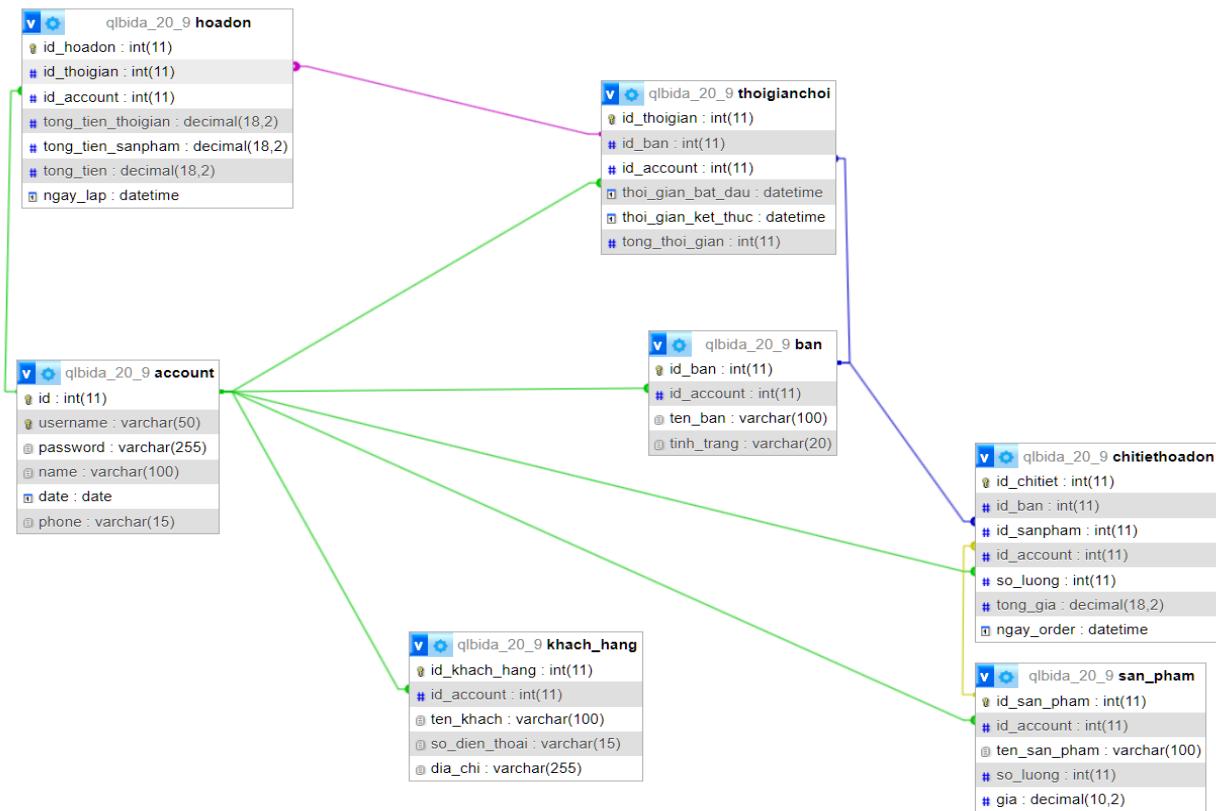
*Biểu đồ 2.10: Biểu đồ hoạt động quản lý bàn*

## Quản lý hóa đơn



Biểu đồ 2.11: Biểu đồ hoạt động quản lý hóa đơn

## 2.3. Mô hình cơ sở dữ liệu



*Sơ đồ 1: Mô hình thực thể liên kết*

### Mô tả bảng dữ liệu

#### Bảng account

STT	Tên thuộc tính	Kiểu	Khóa chính Khóa phụ (PK/FK)	Mô tả
1	id	Int(5)	PK	Mã tài khoản
2	username	Varchar(20)		Tên đăng nhập
3	password	Varchar(20)		Mật khẩu
4	name	Varchar(30)		Họ tên
5	phone	Varchar(15)		Số điện thoại

*Bảng 2.1: Bảng dữ liệu account*

➤ Bảng khach hang

<b>STT</b>	<b>Tên thuộc tính</b>	<b>Kiểu</b>	<b>Khóa chính Khóa phụ (PK/FK)</b>	<b>Mô tả</b>
<b>1</b>	id_khachhang	Int(11)	PK	Mã thành viên
<b>2</b>	id_account	Int(11)	FK	Id account
<b>2</b>	ten_khach	Varchar(^100)		Tên khách hàng
<b>3</b>	So_dien_thoai	Varchar(15)		Số điện thoại
<b>4</b>	dia_chi	Varchar(255)		Địa chỉ

*Bảng 2.2: Bảng dữ liệu member*

➤ Bảng hoadon

<b>STT</b>	<b>Tên thuộc tính</b>	<b>Kiểu</b>	<b>Khóa chính Khóa phụ (PK/FK)</b>	<b>Mô tả</b>
<b>1</b>	id_hoadon	Int(11)	PK	Mã hóa đơn
<b>2</b>	id_account	Int(11)	FK	Mã tài khoản
<b>3</b>	id_thoigian	Int(11)	FK	Mã thời gian
<b>4</b>	tong_tien_thoigian	Datetime		Tổng tiền thời gian chơi
<b>5</b>	tong_tien_sanpham	Datetime		Tổng tiền sản phẩm đã order
<b>6</b>	tong_tien	Decimal(18,2)		Thành tiền thanh toán
<b>7</b>	ngay_lap	datetime		Ngày tạo

*Bảng 2.3: Bảng dữ liệu hóa đơn*

➤ Bảng sản phẩm

STT	Tên thuộc tính	Kiểu	Khóa chính Khóa phụ (PK/FK)	Mô tả
1	id_sanpham	Int(11)	PK	Mã sản phẩm
2	id_account	Int(11)	FK	Mã tài khoản
3	ten_san_pham	Varchar(100)		Tên sản phẩm
4	so_luong	Int(11)		Số lượng
5	gia	Decimal(10,2)		Giá sản phẩm

*Bảng 2.4: Bảng dữ liệu sản phẩm*

➤ Bảng chitiethoadon

STT	Tên thuộc tính	Kiểu	Khóa chính Khóa phụ (PK/FK)	Mô tả
1	id_chitiet	Int(11)	PK	Mã chi tiết
2	id_ban	Int(11)	FK	Mã bàn
3	id_sanpham	Int(11)	FK	Mã sản phẩm
4	id_account	int(11)	FK	Mã tài khoản
5	so_luong	Int(11)		Số lượng
6	tong_gia	Decimal(18,2)		Tổng giá
7	ngay_order	Datetime		Ngày order

*Bảng 2.5: Bảng dữ liệu chi tiết hóa đơn*

➤ Bảng thoigianchoi

STT	Tên thuộc tính	Kiểu	Khóa chính Khóa phụ (PK/FK)	Mô tả
1	id_thoigian	Int(11)	PK	Mã thời gian
2	id_ban	Int(11)	FK	Mã bàn
3	id_account	Int(11)	FK	Mã tài khoản
2	thoi_gian_bat_dau	Datetime		Thời gian bắt đầu
3	thoi_gian_ket_thuc	Datetime		Thời gian kết thúc
4	tong_thoi_gian	Decimal(18,2)		Tổng thời gian chơi

*Bảng 2.6: Bảng dữ liệu thời gian chơi*

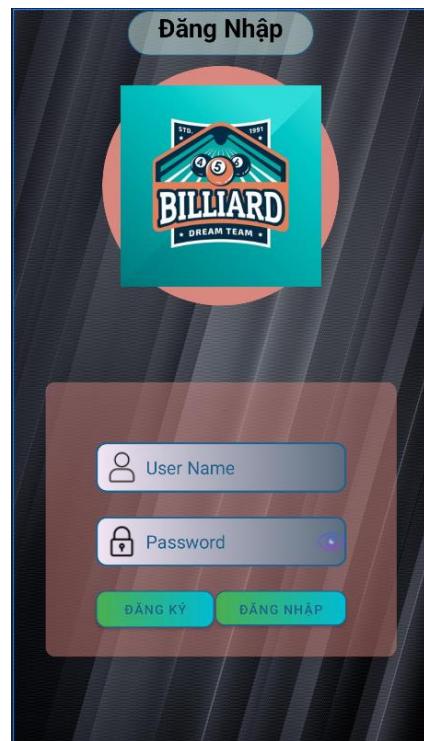
➤ Bảng bàn

STT	Tên thuộc tính	Kiểu	Khóa chính Khóa phụ (PK/FK)	Mô tả
1	Id_ban	Int(11)	PK	Mã bàn
2	Id_account	Int(11)	FK	Mã tài khoản
3	Ten_ban	Varchar(100)		Tên bàn
4	Tinh_trang	Varchar(20)		Tình trạng

*Bảng 2.7: Bảng dữ liệu bàn*

## 2.4.1 Thiết kế giao diện

### 2.4.1. Thiết kế giao diện đăng nhập



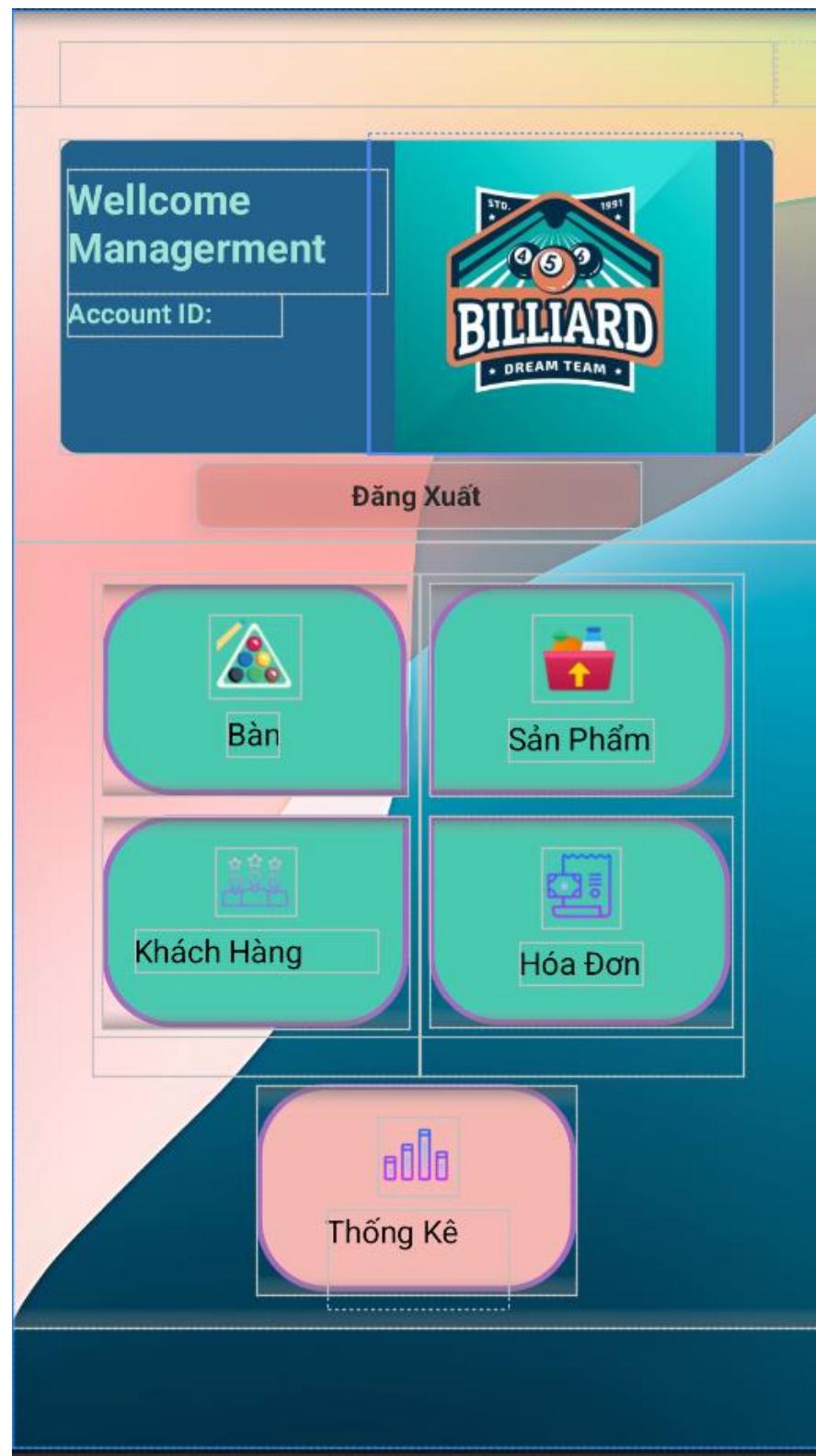
Hình 2.4.1: Thiết kế giao diện đăng nhập

### 2.4.1. Thiết kế giao diện đăng ký



Hình 2.4.2: Thiết kế giao diện đăng ký

#### 2.4.2. Thiết kế giao diện trang chủ



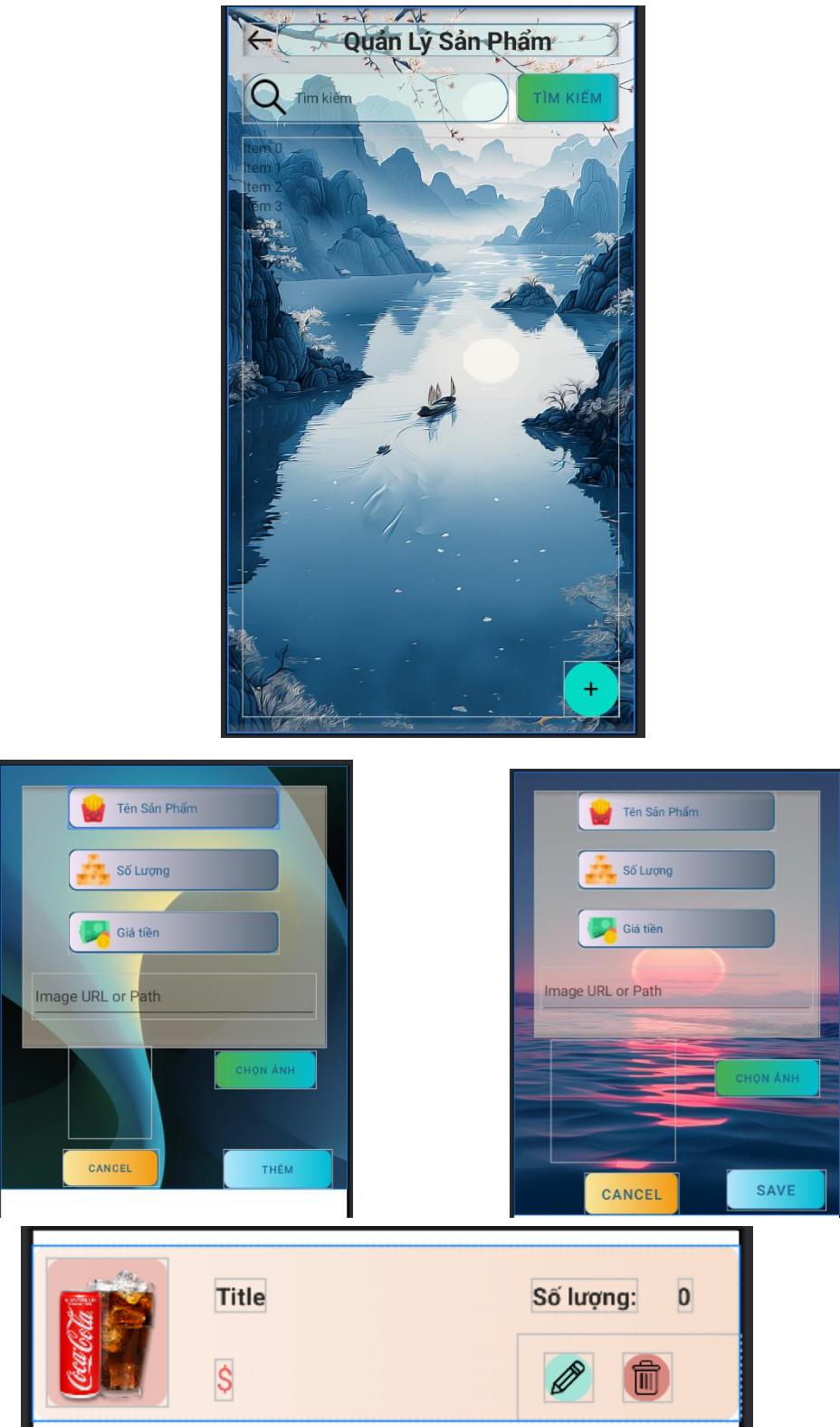
Hình 2.4.3: Thiết kế giao diện trang chủ

### 2.4.3. Thiết kế giao diện quản lý bàn



Hình 2.4.4: Thiết kế giao diện quản lý bàn

#### 2.4.4. Thiết kế giao diện quản lý sản phẩm



Hình 2.4.5: Thiết kế giao diện quản lý sản phẩm

#### 2.4.5. Thiết kế giao diện quản lý khách hàng



This form allows users to enter customer details. It includes three fields: 'Tên Khách hàng' (Customer Name) with a person icon, 'Số điện thoại' (Phone Number) with a phone icon, and 'Địa chỉ' (Address) with a location pin icon. At the bottom are two buttons: 'CANCEL' (yellow) and 'THÊM' (green).

This form shows customer details. It lists 'Name' with a person icon, 'Cash' with a phone icon, and 'Ha Noi' with a location pin icon. To the right of the 'Name' field are edit and delete icons. The entire form is set against a light orange background.

Hình 2.4.6: Thiết kế giao diện quản lý khách hàng

#### 2.4.6. Thiết kế giao diện quản lý hóa đơn



ID Hóa Đơn	\$-
Thời gian	\$-
Tiền Bàn	\$-
Tiền Hàng	\$-
<b>Tổng Tiền</b>	<b>\$-</b>

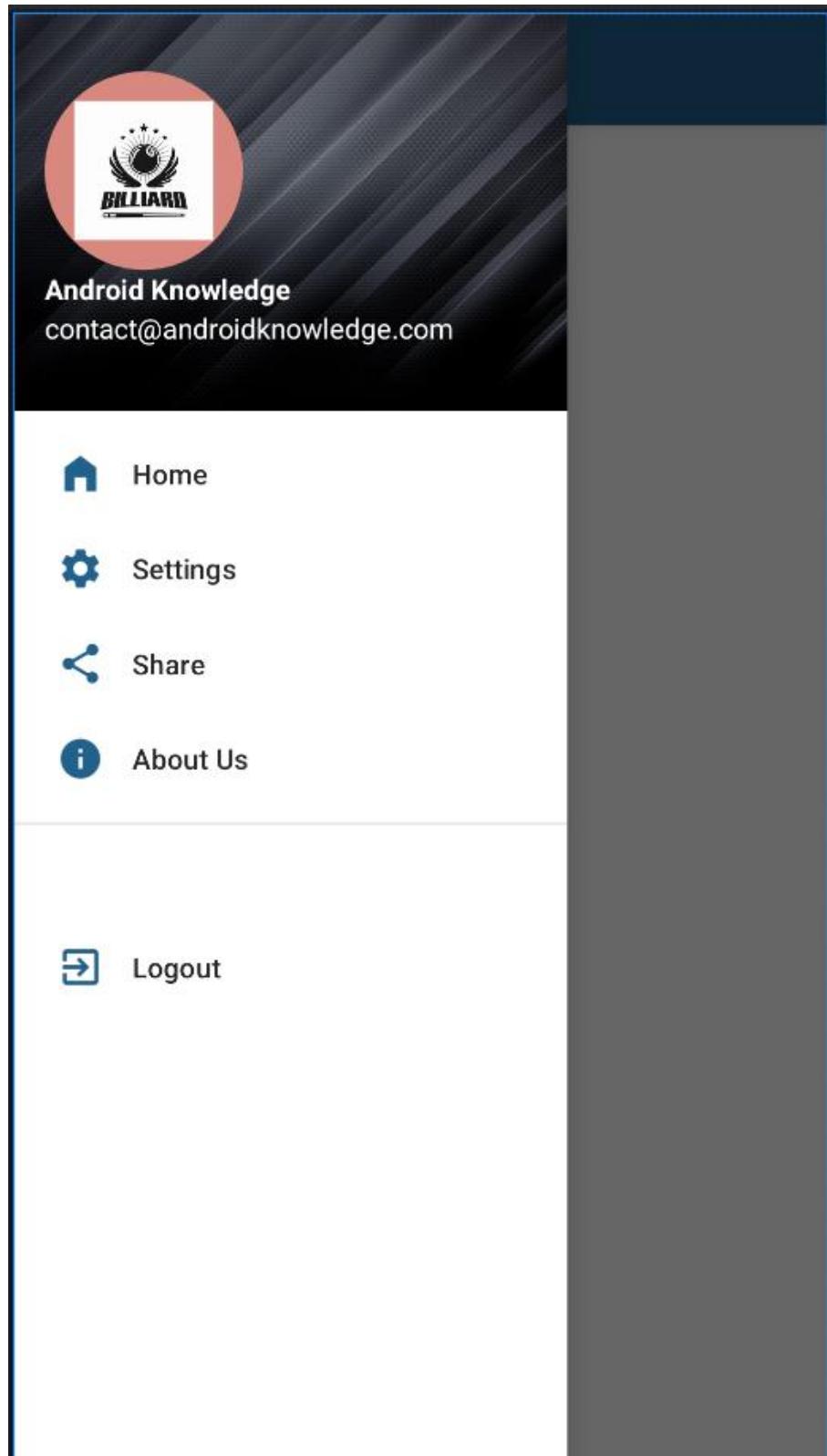
Hình 2.4.7: Thiết kế giao diện quản lý hóa đơn

#### 2.4.7. Thiết kế giao diện quản lý thống kê

Thời gian	\$-
Tiền Bàn	\$-
Tiền Hàng	\$-
<b>Tổng Tiền</b>	<b>\$-</b>

Hình 2.4.8: Thiết kế giao diện quản lý thống kê

#### 2.4.8. Thiết kế giao diện menu tùy chọn



Hình 2.4.9: Thiết kế giao diện menu tùy chọn

#### 2.4.9. Thiết kế giao diện order bàn, sản phẩm

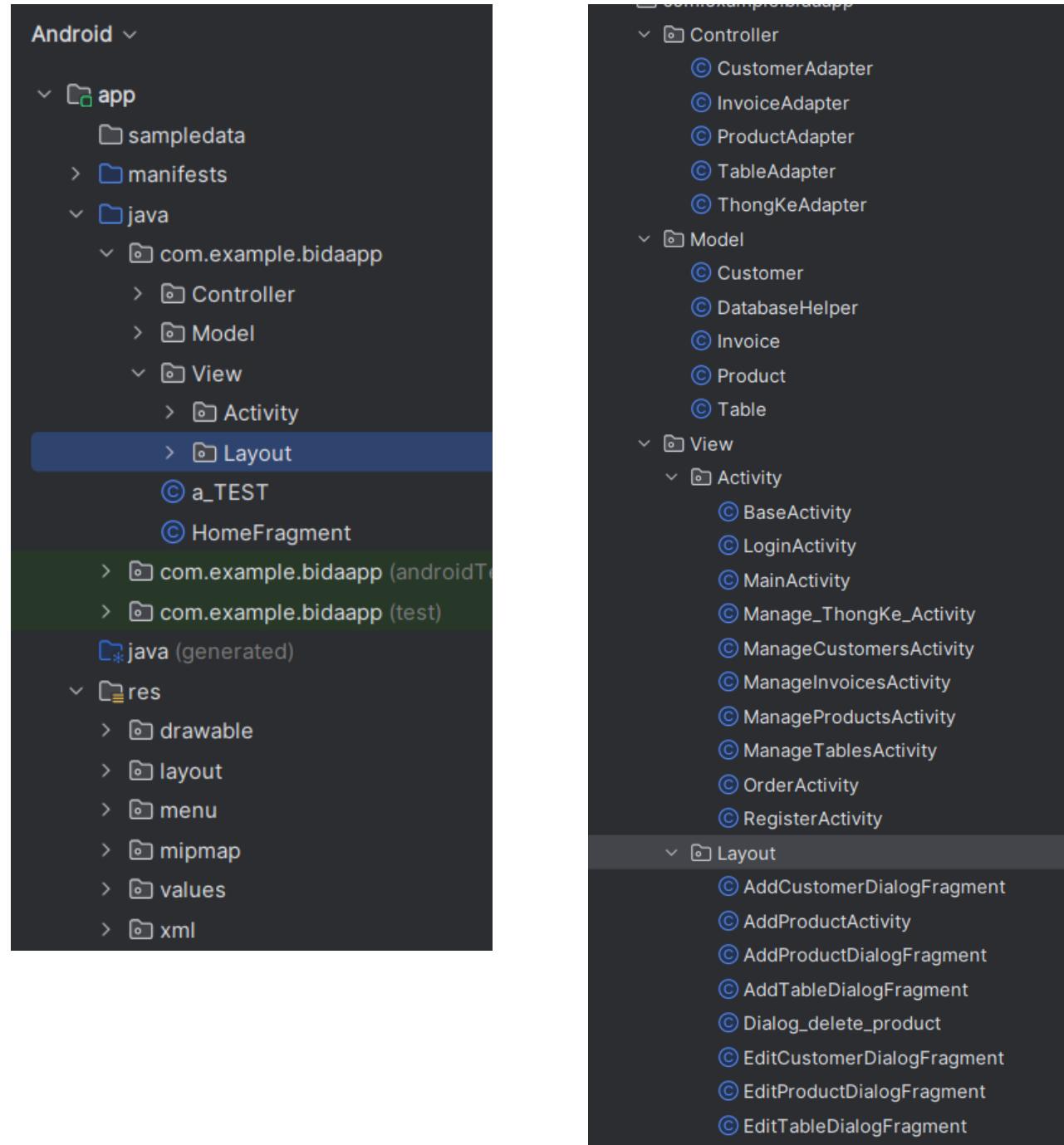
Tên sản phẩm	Giá	Số lượng
Item 1 Sub Item 1		
Item 2 Sub Item 2		
Item 3 Sub Item 3		
Item 4 Sub Item 4		
Item 5 Sub Item 5		
Item 6 Sub Item 6		
Item 7 Sub Item 7		
Item 8 Sub Item 8		

Hình 2.4.10: Thiết kế giao diện order bàn, sản phẩm

# CHƯƠNG III: TRIỀN KHAI VÀ PHÁT TRIỀN ỨNG DỤNG

## 3.1. Mô tả giao diện của chương trình đã xây dựng

### 3.1.1. Cấu trúc mã nguồn



Hình 3.1: Cấu trúc mã nguồn

## Mô tả mã nguồn:

MVC (Model-View-Controller) là một mô hình kiến trúc phần mềm được sử dụng phổ biến để phát triển ứng dụng, bao gồm ba thành phần chính:

### ✓ *Model (M):*

- Model chứa dữ liệu và logic nghiệp vụ của ứng dụng. Nó chịu trách nhiệm lưu trữ, quản lý và xử lý dữ liệu.
- Trong Android, Model có thể là các lớp quản lý cơ sở dữ liệu (SQLite, Room), hoặc các đối tượng đơn giản (POJO) để quản lý dữ liệu.

### ✓ *View (V):*

- View hiển thị giao diện người dùng và nhận các tương tác từ người dùng (nhấn nút, nhập dữ liệu).
- Trong Android, View là các thành phần giao diện người dùng như Button, TextView, ImageView, được định nghĩa trong các file XML trong thư mục layout.

### ✓ *Controller (C):*

- Controller kết nối giữa Model và View. Nó xử lý logic điều hướng và phản ứng với các tương tác của người dùng, sau đó cập nhật View và Model.
- Trong Android, các Activity, Fragment, và ViewModel có thể đóng vai trò như Controller, chịu trách nhiệm điều khiển ứng dụng.

### ✓ *Drawable:*

- Thư mục drawable chứa các tài nguyên hình ảnh (image assets) như .png, .jpg, .svg, hoặc các tài nguyên đồ họa khác (vector drawable).
- Các hình ảnh này được sử dụng để hiển thị trong giao diện của ứng dụng như biểu tượng, nền hoặc hình minh họa.

### ✓ *Layout:*

- Thư mục layout chứa các file XML mô tả cấu trúc và bố cục giao diện của ứng dụng. Các file này định nghĩa cách các thành phần UI như nút bấm, văn bản, hình ảnh được sắp xếp và hiển thị trên màn hình.

- Ví dụ: file activity\_main.xml định nghĩa giao diện chính của ứng dụng.

✓ **Menu:**

- Thư mục menu chứa các file XML định nghĩa các menu của ứng dụng, như các tùy chọn hoặc menu hành động (ActionBar, Context Menu).
- Ví dụ: file menu\_main.xml có thể chứa định nghĩa cho một menu với các tùy chọn khác nhau khi người dùng bấm vào biểu tượng menu trên ActionBar.

✓ **Mipmap:**

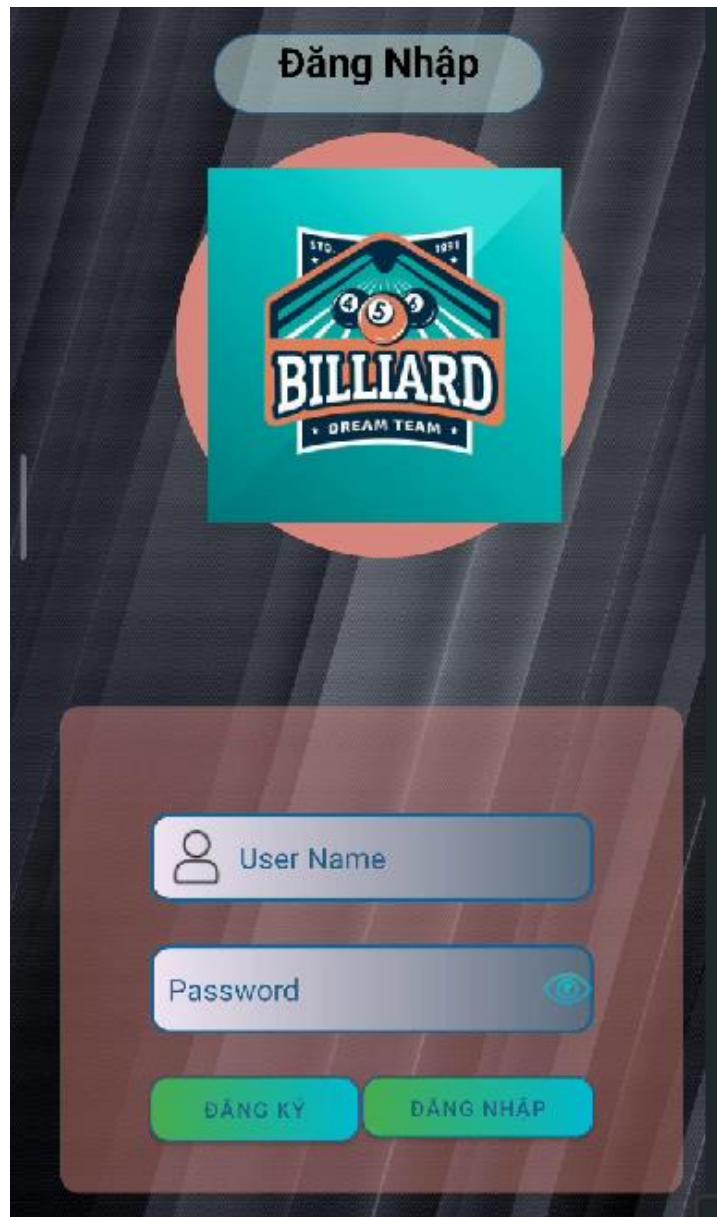
- Thư mục mipmap chứa các tài nguyên hình ảnh với nhiều độ phân giải khác nhau, thường được sử dụng cho biểu tượng ứng dụng (app icon).
- Điểm khác biệt với thư mục drawable là mipmap tối ưu hóa cho các tài nguyên cần hiển thị ở nhiều độ phân giải màn hình khác nhau.
  - Values:
- Thư mục values chứa các file XML định nghĩa các giá trị dùng chung trong ứng dụng, như màu sắc (colors.xml), chuỗi ký tự (strings.xml), kích thước (dimens.xml), và các kiểu giao diện (styles.xml).
- Thư mục này giúp quản lý các tài nguyên có thể tái sử dụng và thay đổi dễ dàng mà không cần sửa đổi trực tiếp trong mã nguồn.

✓ **XML:**

- Thư mục xml chứa các file cấu hình tùy chỉnh cho ứng dụng như thiết lập mạng, cài đặt dữ liệu, hay bất kỳ cấu hình XML nào mà ứng dụng cần (ví dụ như preference.xml cho việc cấu hình cài đặt người dùng).

Những thư mục này giúp quản lý tài nguyên trong ứng dụng Android dễ dàng và có cấu trúc hơn, hỗ trợ cho việc phát triển giao diện và xử lý logic nghiệp vụ một cách tách biệt rõ ràng, phù hợp với mô hình MVC.

### 3.1.2. Giao diện đăng nhập



*Hình 3.2: Giao diện đăng nhập*

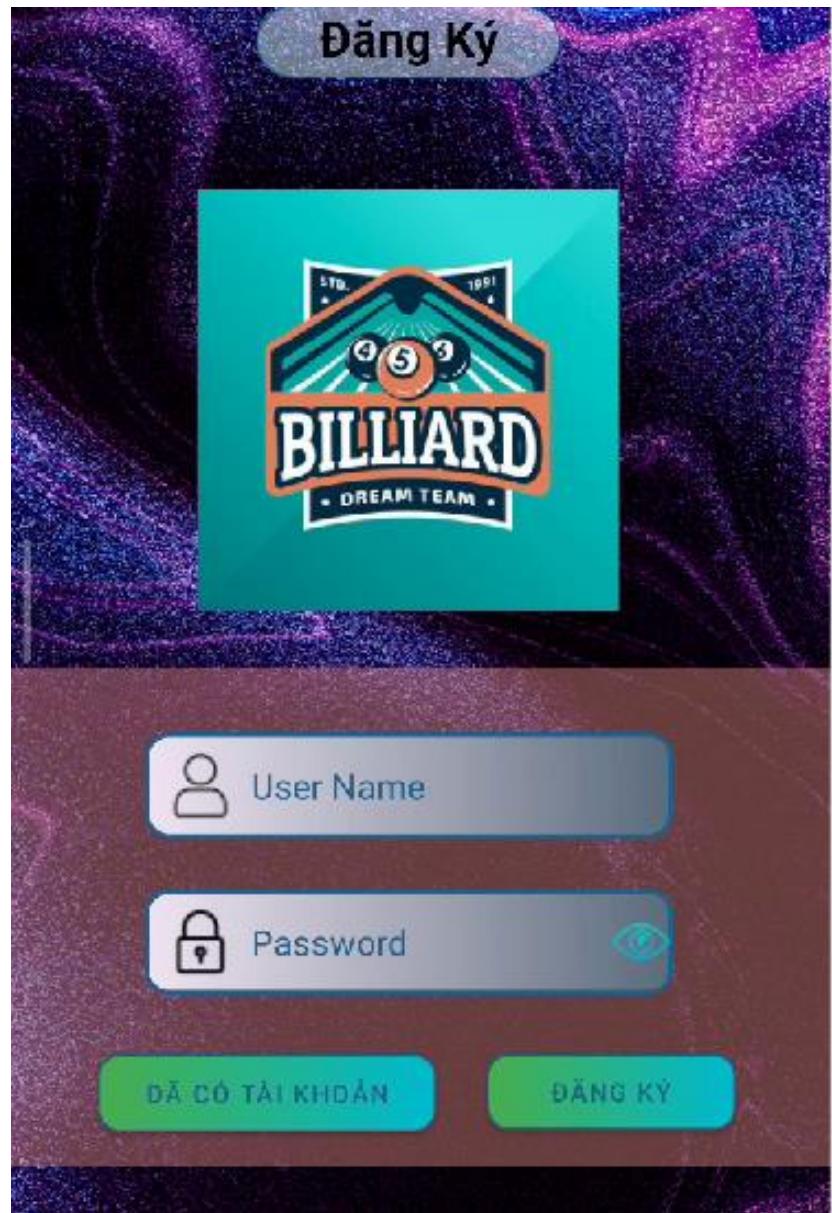
**Mô tả:**

Khi bạn tải app từ các cửa hàng của Google Play, App Store, Amazon Store, khi bạn mở app sẽ đến được trang đăng nhập, bạn phải đăng ký một tài khoản để có thể đăng nhập được.

Đăng ký xong bạn quay trở lại đăng nhập, nhập username, password mà bạn đã đăng ký để đăng nhập.

Sau khi nhập đúng thông tin bạn nhấn nút “Đăng nhập” sẽ dẫn đến trang quản lý.

### 3.1.3. Giao diện đăng ký



Hình 3.3: Giao diện đăng ký

#### Mô tả:

- Điền các thông tin liên quan và nhấn nút đăng ký để tạo tài khoản mới. Nếu như bạn đã có tài khoản, mật khẩu rồi thì nhấn “Đã có tài khoản” bạn sẽ quay về trang đăng nhập.

### 3.1.4. Giao diện quản lý billiards



Hình 3.4: Giao diện quản lý billiards của user

#### ◆ Mô tả:

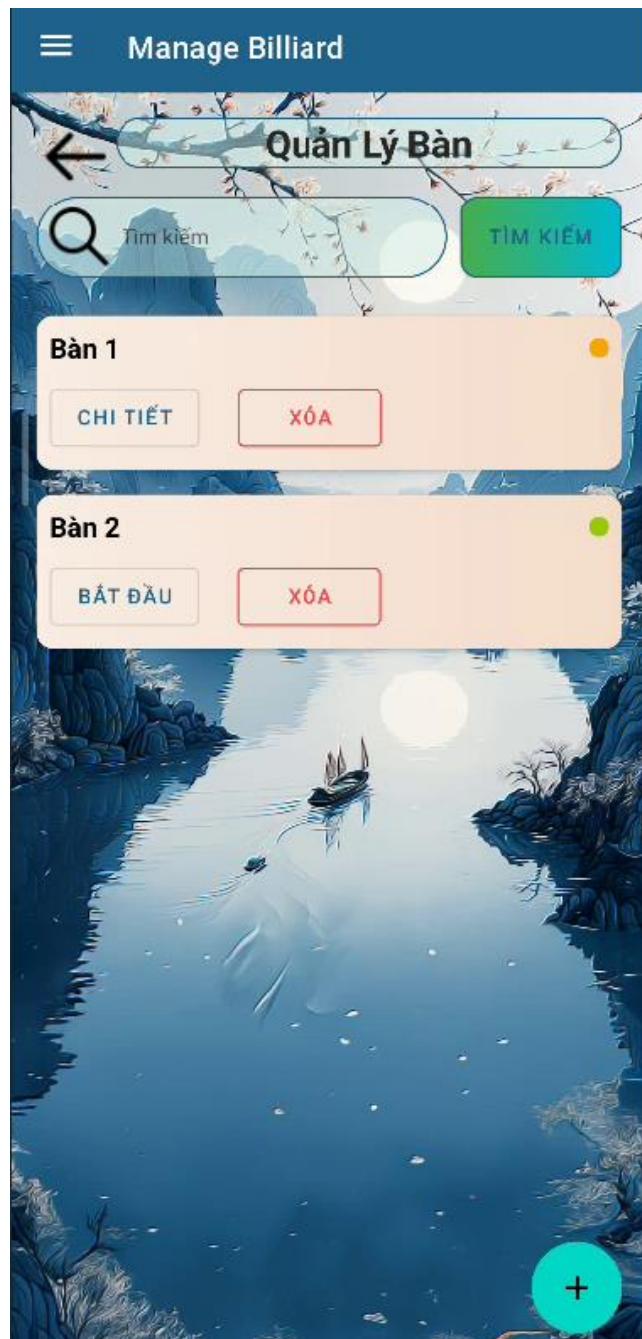
Khi bạn nhấn vào “Đăng nhập” bạn sẽ đến giao diện hiển thị toàn bộ thông tin của trang quản lý bao gồm các chức năng quản lý như: quản lý bàn, sản phẩm, hóa đơn, khách hàng.

Bên cạnh đó mỗi người sẽ hiển thị một id khác nhau để đảm bảo mỗi người sẽ có một giao diện quản lý thuộc về riêng mình.

Khi bạn nhấn các chức năng sẽ được dẫn đến trang quản lý của các chức năng đó.

Nút “Đăng xuất” khi nhấn sẽ hiện ra thông báo “Bạn có chắc muốn đăng xuất?” và bạn có hai lựa chọn là “Hủy” và “Đăng Xuất”. Khi bạn nhấn nút “Hủy” bạn sẽ ở lại trang quản lý, khi bạn nhấn “Đăng Xuất” bạn sẽ quay trở lại màn hình đăng nhập.

### 3.1.5. Giao diện quản lý bàn



Hình 3.5: Giao diện quản lý bàn

## Mô tả

Khi bạn nhấn vào “Bàn” bạn sẽ đến giao diện hiển thị toàn bộ thông tin của trang quản lý bao gồm: Thanh tìm kiếm và nút thêm bàn ở góc phải màn hình. Ngoài ra khi bạn đã tạo được bàn mới thì sẽ hiển thị bàn mà bạn đã thêm và một số nút như bắt đầu, xem chi tiết và xóa bàn.

*Thêm bàn:* khi nhấn vào sẽ hiện lên các thông tin để bạn điều vào ô trống. Khi nhấn nút “thêm” hệ thống sẽ thông báo “Bạn đã thêm bàn thành công”.

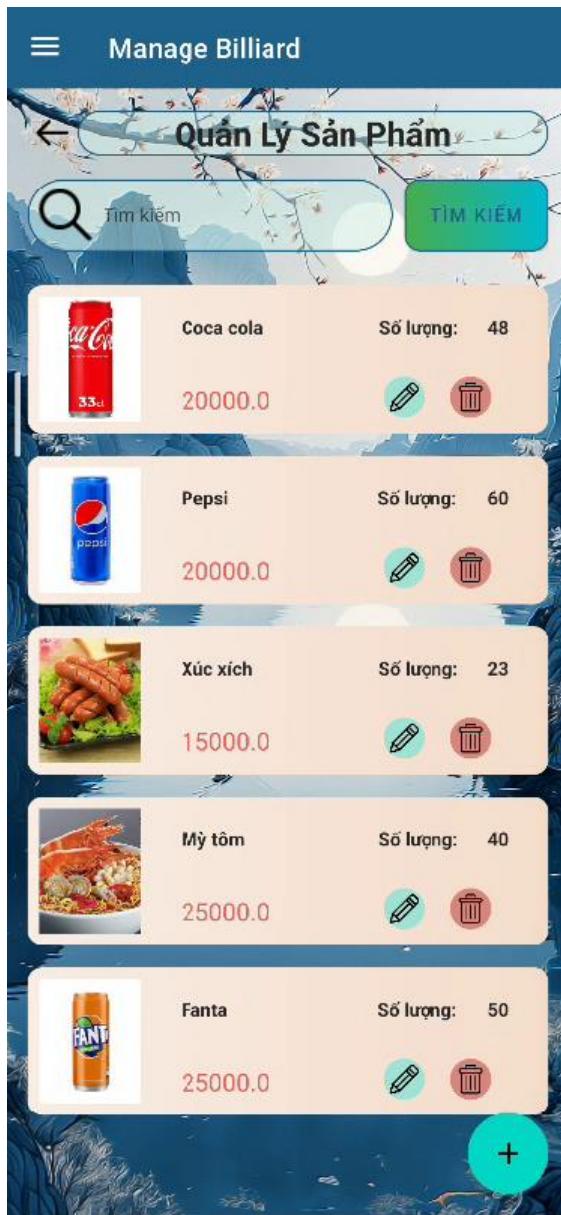
*Tìm kiếm:* Nhập tên bàn cần tìm vào và nhấn nút “tìm kiếm”

*Bắt đầu:* Nhấn khi bàn đó đang ở tình trạng “Trống” để chuyển sang tình trạng bàn “Đang sử dụng”. Khi đó bàn sẽ bắt đầu tính giờ chơi và thêm các dịch vụ khác(nước uống, gậy thuê, ...).

*Chi tiết:* Khi bạn nhấn vào sẽ hiển thị chi tiết hóa đơn bao gồm thời gian bắt đầu, danh sách sản phẩm mà bạn đã order. Khi nhấn “Kết thúc” xong bàn sẽ trở về tình trạng “Trống”, và xuất hóa đơn vào trang “Quản lý hóa đơn”.

*Xóa:* Khi nhấn bàn sẽ bị xóa khỏi trang quản lý billiards.

### 3.1.6. Giao diện trang quản lý sản phẩm



Hình 3.6: Quản lý sản phẩm

#### ✚ Mô tả:

Khi bạn nhấp vào “Sản phẩm” bạn sẽ đến giao diện hiển thị toàn bộ thông tin của trang quản lý bao gồm: nút tìm kiếm và nút thêm(Khi mới đăng ký tài khoản thì sẽ không có sản phẩm nào).

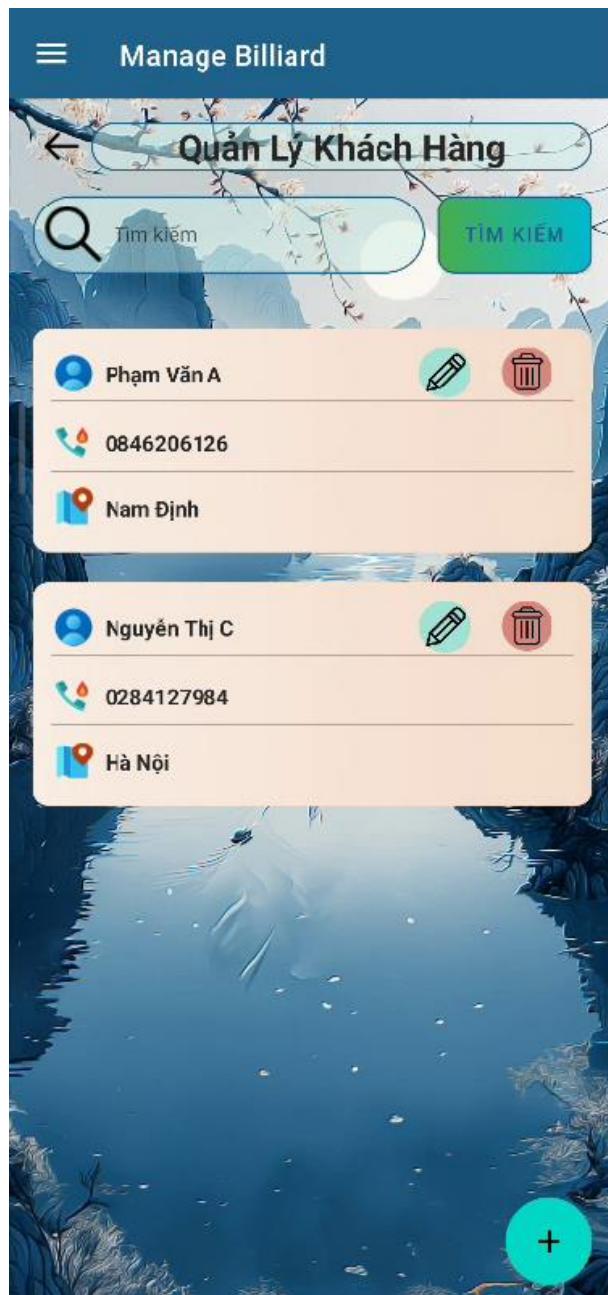
Thêm: khi nhấp vào sẽ hiện lên các thông tin để bạn điều vào ô trống. Khi nhấn nút “thêm sản phẩm” hệ thống sẽ thông báo “Bạn đã thêm sản phẩm thành công”.

Tìm kiếm: Nhập tên sản phẩm cần tìm vào và nhấn nút “tìm kiếm”.

Sửa: Nếu thông tin của sản phẩm bị thiếu hoặc sai về tên, giá hoặc số lượng. Sau khi sửa xong sẽ update trực tiếp lên sản phẩm trong trang quản lý luôn.

Xóa: Khi nhấn xóa, sản phẩm sẽ bị xóa khỏi danh sách sản phẩm.

### 3.1.7. Quản lý khách hàng



Hình 3.7: Quản lý khách hàng

## Mô tả

Khi bạn nhấp vào “Khách hàng” bạn sẽ đến giao diện hiển thị toàn bộ thông tin của trang quản lý bao gồm: nút tìm kiếm và nút thêm(Khi mới đăng ký tài khoản thì sẽ không có sản phẩm nào).

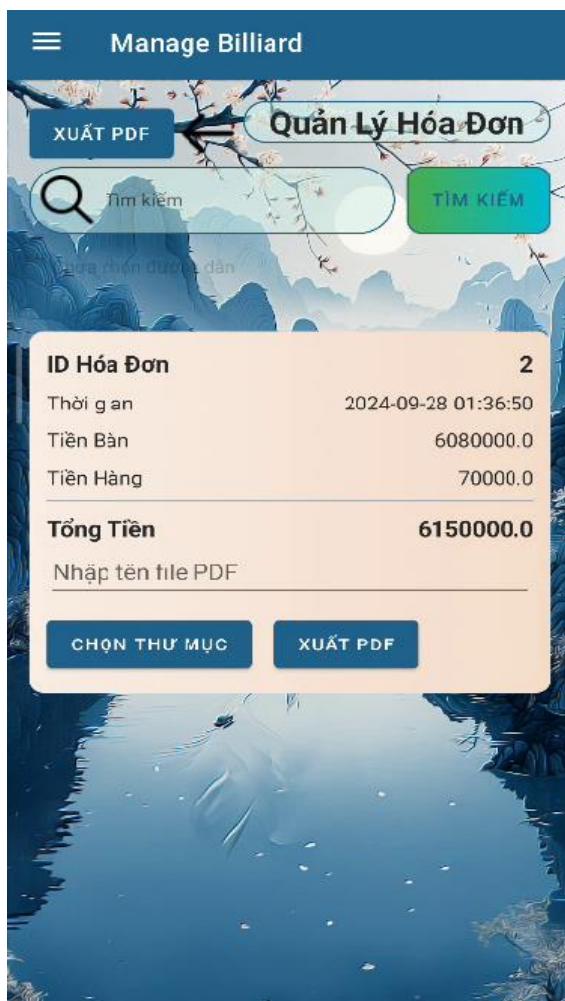
Thêm: khi nhấp vào sẽ hiện lên các thông tin để bạn điều vào ô trống. Khi nhấp nút “thêm” hệ thống sẽ thông báo “Bạn đã thêm khách hàng thành công”.

Tìm kiếm: Nhập tên khách hàng cần tìm vào và nhấp nút “tìm kiếm”.

Sửa: Nếu thông tin của khách hàng bị thiếu hoặc sai về tên, địa chỉ. Sau khi sửa xong sẽ update trực tiếp lên danh sách khách hàng trong trang quản lý luôn.

Xóa: Khi nhấp xóa, sản phẩm sẽ bị xóa khỏi danh sách sản phẩm.

### 3.1.8. Quản lý hóa đơn



*Hình 3.8: Quản lý hóa đơn*

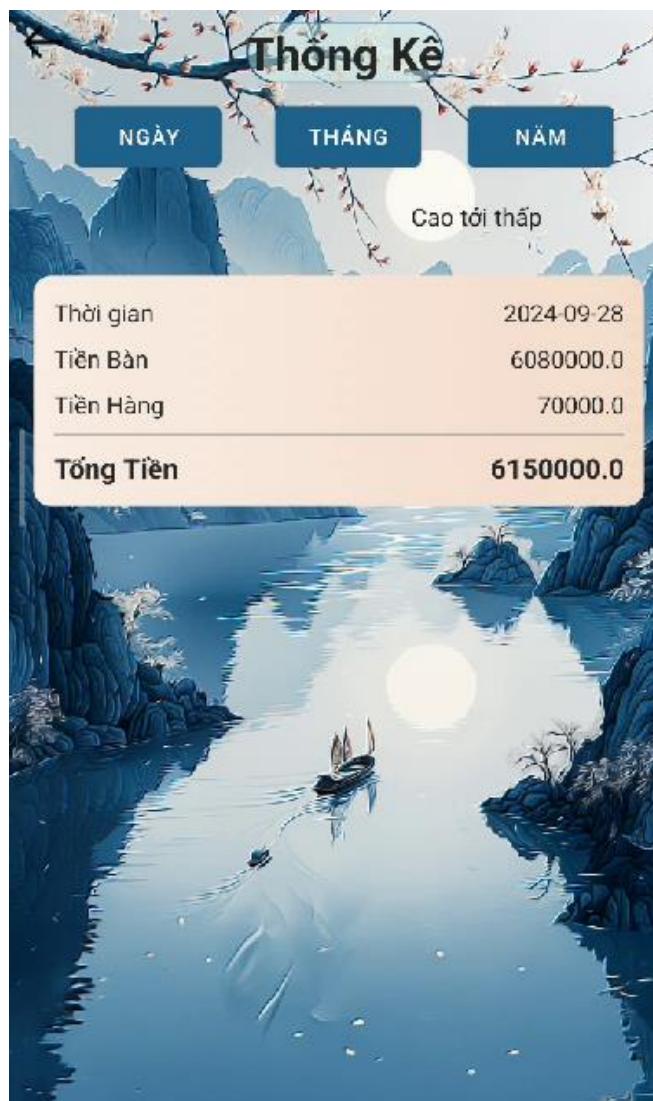
#### ⊕ Mô tả

Khi nhấn kết thúc ở “Quản lý bàn” thì hóa đơn sẽ tự động cập nhật lên trang “Quản lý hóa đơn”.

Tìm kiếm: Nhập id của hóa đơn cần tìm và nhấn nút “Tìm kiếm”.

Bạn muốn xuất hóa đơn ra điện thoại hoặc máy tính bằng file PDF trước tiên bạn phải đặt tên cho file, tiếp đến bạn chọn thư mục để lưu sau đó nhấn nút “Xuất PDF” là hóa đơn sẽ được in ra ở thư mục bạn đã chọn.

#### 3.1.9. Thống kê



**Hình 3.9: Giao diện thống kê**

⊕ Mô tả: Thống kê giúp chúng ta tìm kiếm hóa đơn theo ngày, tháng, năm.

## 3.2. Triển khai ứng dụng

### 3.2.1. Danh sách các test case Unit Test

#### UNIT TEST CASE LIST

Project Name	Xây dựng ứng dụng quản lý quán Billiards bằng Android Studio
Project Code	UT
Normal number of Test cases/KLOC	100
Test Environment Setup Description	1. Server 2. Database 3. Web Browser

No	Requirement Name	Class Name	Function Name	Function Code(Optional)	Sheet Name	Pre-Condition
1		Đăng ký	Đăng ký	F_dang_ky	<a href="#"><u>F_dang_ky</u></a>	
2		Đăng nhập	Đăng nhập	F_dang_nhap	<a href="#"><u>F_dang_nhap</u></a>	
3		thêm bàn	thêm bàn	F_them_ban	<a href="#"><u>F_them_ban</u></a>	
4		tìm kiếm bàn	tìm kiếm bàn	F_sua_ban	<a href="#"><u>F_tim_kiem_ban</u></a>	
5		thêm khách hàng	thêm khách hàng	F_them_khach_hang	<a href="#"><u>F_them_khach_hang</u></a>	
6		sửa khách hàng	sửa khách hàng	F_sua_khach_hang	<a href="#"><u>F_sua_khach_hang</u></a>	
7		tìm kiếm khách hàng	tìm kiếm khách hàng	F_tim_kiem_khach_hang	<a href="#"><u>F_tim_kiem_khach_hang</u></a>	
8		thêm sản phẩm	thêm sản phẩm	F_them_san_pham	<a href="#"><u>F_them_san_pham</u></a>	
9		sửa sản phẩm	sửa sản phẩm	F_sua_san_pham	<a href="#"><u>F_sua_san_pham</u></a>	
10		tìm kiếm sản phẩm	tìm kiếm sản phẩm	F_tim_kiem_san_pham	<a href="#"><u>F_tim_kiem_san_pham</u></a>	

Bảng 3.1: Danh sách các test case cho unit test

### 3.2.2. Thực hiện Function Unit test case

Function Code		<i>F_dang_Ky</i>	Function Name		Đăng ký											
Created By		Phạm Văn Bắc	Executed By		Phạm Văn Bắc											
Test requirement		Chức năng Đăng ký hoạt động bình thường														
Passed		Failed		Untested		N/A/B		Total Test								
7		0		0		3	4	0	7							
Condition	Precondition	Can connect with server														
	Input1	tk: "null" mk: "admin"	0													
	Input2	tk: "admin" mk: "null"	0													
	Input3	tk: "asdfghjh" mk: "admin"	0													
	Input4	tk: "admin" mk: "asdfghjh"	0													
	Input5	tk: "admin" mk: "admin123"	0													
	Input6	tk: "admin" mk: "admi1"	0													
	Input7	tk: "admin" mk: "admin123"	0													
Confirm	Return	TRUE	0													
		FALSE	0	0	0	0										
	Exception	Lỗi xác thực	0	0												
	Log message	"Tên tài khoản hoặc mật khẩu không đúng kiểu dữ liệu!"	0	0	0											
		"Vui lòng nhập mật khẩu"	0													
		Mật khẩu phải từ 6 ký tự	0													
Result	Type(N : Normal, A : Abnormal, B : Boundary)	A A A A N N N	P P P P P P P													
	Passed/Failed	0 0 0 0 0 0 0	4 4 4 4 4 4 4													
	Executed Date	/ / / / / / /	0 0 0 0 0 0 0													
	Defect ID	1 1 1 1 1 1 1														

Bảng 3.2: Unit test Function đăng ký

Function Code		<i>F_dang_Nhap</i>	Function Name		Đăng nhập				
Created By		Phạm Văn Bắc	Executed By		Phạm Văn Bắc				
Test		Chức năng Đăng nhập hoạt động bình thường							
Passed		Failed	Untested		N/A/B	Total Test			
5		0	0		1 4 0	5			
Condition	Precondition	Can connect with server							
	Input1	tk: "null" mk: "adminnv"	O						
	Input2	tk: "adminnv" mk: "null"	O						
	Input3	tk: "asdfghjh" mk: "adminnv"	O						
	Input4	tk: "adminnv" mk: "asdfghjh"	O						
	Input5	tk: "adminnv" mk: "adminnv"	O						
	Return	TRUE	O						
		FALSE	O	O	O	O			
	Exception	Lỗi xác thực	O O						
	Log message	"Tên đăng nhập hoặc mật khẩu không đúng!"	O O						
Result	"Tên đăng nhập hoặc mật khẩu bạn không được để trống!"	O O							
	"Đăng nhập thành công trang !"				O				
	Type(N : Normal, A : Abnormal, B : Boundary)	A A A A N							
	Passed/Failed	P P P P P							
		0 0 0 0 0							
		4 4 4 4 4							
	Executed Date	/ / / / /							
		0 0 0 0 0							
		1 1 1 1 1							
Defect ID									

Bảng 3.3: Unit test Function đăng nhập

<b>Function Code</b>	<i>F_them_khach_hang</i>	<b>Function Name</b>	Thêm Khách Hàng					
<b>Created By</b>	Phạm Văn Bắc	<b>Executed By</b>						
<b>Test requirement</b>	<Brief description about requirements which are tested in this function>							
<b>Passed</b>		<b>Failed</b>	<b>Untested</b>		<b>N/A/B</b>		<b>Total Test Cases</b>	
8		0	0	0	4	3	1	8

			UTCID01	UTCID02	UTCID03	UTCID04	UTCID05	UTCID06	UTCID07	UTCID08
<b>Condition</b>		<b>Precondition</b>								
		Can connect with server								
<b>Train Carriage : input</b>										
		"rỗng"	<input type="radio"/>							
		tên đăng nhập:"bac0511"		<input type="radio"/>						
		"nguyenloan"			<input type="radio"/>					
		Mật khẩu: "bac@123"				<input type="radio"/>				
		Họ tên:"Phạm Văn Bắc"					<input type="radio"/>			
		Số điện thoại:"0983476216"						<input type="radio"/>		
		Địa chỉ: "Nam Định"							<input type="radio"/>	
		Địa chỉ: "Hà Nội"								<input type="radio"/>
<b>Confirm</b>	<b>Return</b>									
			TRUE	<input type="radio"/>			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
			FALSE	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>				<input type="radio"/>
	<b>Exception</b>									
<b>Log message</b>										
		"vui lòng nhập thông tin"	<input type="radio"/>							
		"Thêm thành công."		<input type="radio"/>			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		"sai kiểu dữ liệu"			<input type="radio"/>					
		"thiếu @"								<input type="radio"/>
		"lỗi dữ liệu"		<input type="radio"/>						
<b>Result</b>	Type(N : Normal, A : Abnormal, B : Boundary)	A	N	B	A	N	N	N	N	A
	Passed/Failed	P	P	P	P	P	P	P	P	P

*Bảng 3.4: Unit test Function thêm khách hàng*

Function	<i>F_sua_khach_hang</i>	Function Name	<i>Sửa khách hàng</i>						
Created By	Phạm Văn Bắc	Executed By							
Test	binh thường								
Passed	Failed	Untested		N/A/B		Total			
8	0	0		3	3	2	8		
		UTCIDO1	UTCIDO2	UTCIDO3	UTCIDO4	UTCIDO5	UTCIDO6	UTCIDO7	UTCIDO8
Condition	Precondition								
	Can connect with server								
	Input1 "null"	0							
	Input2 tên đăng nhập" phambac05		0						
	Input3 "05112003"			0					
	Input4 Họ tên:"Phạm Văn Bắc"				0				
	Input5 "0"					0			
	Input6 số điện thoại:"0961537128"						0		
	Input6 "-1"							0	
	Input7 địa chỉ: "Ninh bình @#"								0
Confirmation	Return	TRUE	0	0	0				
		FALSE	0	0	0	0	0	0	0
Log message									
	" không được để trống"	0							
	" sai kiểu dữ liệu"		0						
	"sửa Thành công"	0		0	0				
	"vượt quá ký tự cho phép"								
	Vui lòng nhập một số nguyên dương						0		
	" không đúng kiểu dữ liệu"				0			0	
Result	Type(N : Normal, A : Abnormal, B : Boundary)	A	N	B	N	A	N	B	A
	Passed/Failed	P	P	P	P	P	P	P	P
	Executed Date								
	Defect ID								

*Bảng 3.5: Unit test Function sửa khách hàng*

<b>Function Code</b>	<i>F_tim_kiem_khach_hang</i>	<b>Function Name</b>	<i>Tìm kiếm khách hàng</i>				
<b>Created By</b>	<i>Phạm Văn Bắc</i>	<b>Executed By</b>	<i>Phạm Văn Bắc</i>				
<b>Test requirement</b>	<i>Chức năng Đăng nhập hoạt động bình thường</i>						
<b>Passed</b>	Failed		<b>Untested</b>	<b>N/A/B</b>		<b>Total Test</b>	
4	1		0	2	2	1	5
			UTCIDO	UTCIDO	UTCIDO	UTCIDO	UTCIDO
<b>Condition</b>	<b>Precondition</b>						
	Can connect with server						
<b>Input1</b>	"null"						
<b>Input2</b>	"phambac0511"						
<b>Input3</b>	"nguyenloan"						
<b>Input4</b>	"400"						
<b>Input5</b>	"-1"						
<b>Confirmation</b>	<b>Return</b>						
	TRUE						
	FALSE						
	0						
<b>Exception</b>							
<b>Log message</b>							
	"vui lòng điền vào ô tìm kiếm"						
	0						
	"Không có dữ liệu"						
<b>Result</b>	Type(N : Normal, A : Abnormal, B : Bound)	A	N	N	A	B	
	Passed/Failed	P	P	P	P	F	
		6	6	6	6	6	
	Executed Date	/	/	/	/	/	
		0	0	0	0	0	
	<b>Defect ID</b>						

*Bảng 3.6: Unit test Function tìm kiếm khách hàng*

<b>Function Code</b>	<i>F_them_ban</i>	<b>Function Name</b>	<i>Thêm bàn</i>		
<b>Created By</b>	Phạm Văn Bắc	<b>Executed By</b>	Phạm Văn Bắc		
<b>Test requirement</b>	<Brief description about requirements which are tested in this function>				
Passed	Failed	Untested	N/A/B	Total Test	
0	0	4	3   1	0	4

			UTCID01	UTCID02	UTCID03	UTCID04
<b>Condition</b>	<b>Precondition</b>					
	Can connect with server					
	<b>Train Carriage : input</b>					
	"rỗng"	0				
	bàn: "Bàn 1"		0			
	"Bàn 1"			0		
	"00"				0	
<b>Confirm</b>	<b>Return</b>					
	TRUE	0		0		
	FALSE	0		0		
	<b>Exception</b>					
	<b>Log message</b>					
	"vui lòng nhập thông tin"	0				
	"Thêm thành công."		0		0	
	"lỗi trùng tên"			0		
<b>Result</b>	Type(N : Normal, A : Abnormal, B : Boundary)	A	N	N	N	
	Passed/Failed	P	P	P	P	

Bảng 3.7: Unit test Function thêm bàn

Function Code	<i>F_tim_kiem_ban</i> <th>Function Name</th> <td data-cs="4" data-kind="parent"><i>Tim kiem ban</i></td> <td data-kind="ghost"></td> <td data-kind="ghost"></td> <td data-kind="ghost"></td> <td></td>	Function Name	<i>Tim kiem ban</i>				
Created By	Phạm Văn Bắc	Executed By	Phạm Văn Bắc				
Test requirement	<Brief description about requirements which are tested in this function>						
Passed	Failed	Untested	N/A/B		Total Test Cases		
0	0	6	3	1	2	6	
			UTCID01	UTCID02	UTCID03	UTCID04	UTCID04
<b>Condition</b>	<b>Precondition</b>						
	Can connect with server						
	<b>Input1</b>	null	<input checked="" type="radio"/>				
	<b>Input2</b>	Bàn 1		<input checked="" type="radio"/>			
	<b>Input3</b>	0			<input checked="" type="radio"/>		
	<b>Input4</b>	Bàn 2				<input checked="" type="radio"/>	
	<b>Input5</b>	Bàn 1				<input checked="" type="radio"/>	
	<b>Return</b>						
	TRUE			<input checked="" type="radio"/>	<input checked="" type="radio"/>		
	FALSE		<input checked="" type="radio"/>		<input checked="" type="radio"/>	<input checked="" type="radio"/>	
<b>Confirmation</b>	<b>Log message</b>						
	"Tên không được để trống"		<input checked="" type="radio"/>				
	"Tên sai kiểu dữ liệu"				<input checked="" type="radio"/>		
	sửa Thành công			<input checked="" type="radio"/>	<input checked="" type="radio"/>		
	Trùng tên					<input checked="" type="radio"/>	
<b>Result</b>	Type(N : Normal, A : Abnormal)	A	N	B	N	B	
	Passed/Failed	P	P	P	P	P	
	Executed Date						
	Defect ID						

*Bảng 3.8: Unit test Function tìm kiếm bàn*

Function Code	<i>F_them_san_pham</i>	Function Name	Thêm Sản Phẩm					
Created By	Phạm Văn BẮC	Executed By	Phạm Văn					
Test requirement	<Brief description about requirements which are tested in this function>							
Passed	Failed	Untested	N/A/B		Total Test Cases			
10	0	0	2	2	6	10		
			UTCID01	UTCID02	UTCID03	UTCID04	UTCID05	UTCID06
			UTCID07	UTCID08	UTCID09	UTCID10	UTCID11	UTCID12
Condition	Precondition							
	Can connect with server							
	Train Carriage : input							
	"tên sản phẩm"							
	"rỗng" 0							
	tên_sản_phẩm	"0"	0					
		"nước ngọt "	0					
		"nước ngọt @"	0					
		"-1"		0				
		"500"			0			
Confirm	giá_sản_phẩm	5000				0		
		0				0		
	số_lượng	100				0		
		-1					0	
	Return							
	TRUE		0			0	0	
	FALSE		0	0	0	0	0	0
	Exception							
	Log message							
	"vui lòng nhập thông tin"		0					
	"Thêm thành công."		0			0	0	
	"sai kiểu dữ liệu"		0	0	0	0		
	"Giá trị phải lớn hơn 0"					0		
	"chỉ được phép nhập từ 0 - 9"						0	
Result	Type(N : Normal, A : Abnormal, B : Boundary)	A	B	N	B	B	B	N
	Passed/Failed	P	P	P	P	P	P	P

Bảng 3.9: Unit test Function thêm sản phẩm

<b>Function Code</b>	<i>F_sua_san_pham</i>	<b>Function Name</b>	<i>Sửa sản phẩm</i>						
<b>Created By</b>	<i>Phạm Văn Bắc</i>	<b>Executed By</b>	<i>Phạm Văn Bắc</i>						
<b>Test</b>	<i>bình thường</i>								
<b>Passed</b>		<b>Failed</b>	<b>Untested</b>			<b>N/A/B</b>		<b>Total</b>	
6		1		0		2	4	1	7
			<b>UTCID01</b>	<b>UTCID02</b>	<b>UTCID03</b>	<b>UTCID04</b>	<b>UTCID05</b>	<b>UTCID07</b>	
<b>Condition</b>	<b>Precondition</b>								
	Can connect with server								
<b>Input1</b>	null	0							
<b>Input2</b>	tên sản phẩm: Coca cola		0						
<b>Input3</b>	Pepsi			0					
<b>Input4</b>	giá sản phẩm: 25000				0				
<b>Input5</b>	-1					0			
<b>Input6</b>	số lượng: 100						0		
<b>Input6</b>	100đ							0	
	<b>Return</b>								
	TRUE		0		0		0		
	FALSE	0		0		0		0	
<b>Confirm</b>									
	<b>Log message</b>								
	"tên sản phẩm không được để trống"	0							
	"Tên sản phẩm sai kiểu dữ liệu"		0						
	sửa Thành công		0		0		0		
	giá sản phẩm vượt quá ký tự cho phép								
	Vui lòng nhập một số nguyên dương					0			
	Giá sản phẩm không đúng kiểu dữ liệu							0	
<b>Result</b>	Type(N : Normal, A : Abnormal, B : Bound)	A	A	B	A	N	A	N	
	Passed/Failed	P	P	P	P	P	P	F	
	Executed Date								
	Defect ID								

*Bảng 3.10: Unit test Function sửa sản phẩm*

<b>Function Code</b>	<i>F_tim_kiem_san_pham</i>	<b>Function Name</b>	<i>Tìm kiếm sản phẩm</i>					
<b>Created By</b>	<i>Phạm Văn BẮC</i>	<b>Executed By</b>	<i>Phạm Văn BẮC</i>					
<b>Test requirement</b>	<i>Chức năng Đăng nhập hoạt động bình thường</i>							
Passed	Failed	Untested			N/A/B		Total Test Cases	
5	0	0		2	2	1	5	
		UTCID01	UTCID02	UTCID03	UTCID04	UTCID05		
<b>Condition</b>	<b>Precondition</b>							
	Can connect with server							
	<b>Input1</b>							
	"null"	<b>0</b>						
	<b>Input2</b>							
	"Fanta"		<b>0</b>					
	<b>Input3</b>							
	"Nước ngọt 1"			<b>0</b>				
	<b>Input4</b>					<b>0</b>		
	"400"						<b>0</b>	
<b>Confirm</b>	<b>Input5</b>						<b>0</b>	
	"-1"							
	<b>Return</b>							
	TRUE		<b>0</b>					
	FALSE	<b>0</b>		<b>0</b>	<b>0</b>	<b>0</b>		
<b>Exception</b>								
<b>Result</b>	<b>Log message</b>							
	"vui lòng điền vào ô tìm kiếm"	<b>0</b>						
	"Không có dữ liệu "			<b>0</b>	<b>0</b>	<b>0</b>		
	Type(N : Normal, A : Abnormal, B : Boundary)	A	N	N	A	B		
	Passed/Failed	P	P	P	P	P		
	Executed Date	0 6 /	0 6 /	0 6 /	0 6 /	0 6 /		
	Defect ID							

*Bảng 3.11: Unit test Function tìm kiếm sản phẩm*

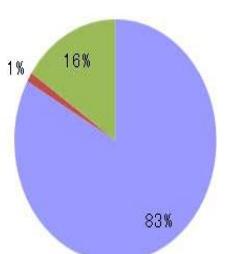
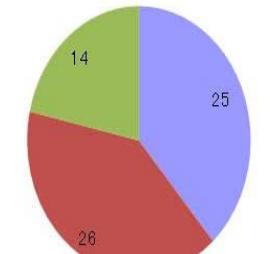
### 3.2.3. Kiểm thử login

ID	Test Case Description	Pre-condition	Test Steps	Expected Output	Post-condition
	<b>Check validation</b>				
[Login-1]	<b>Không nhập username</b>		1. Nhập thông tin: username: bỏ trống password: Admin@1 2. Click button Login	Hệ thống hiển thị thông báo lỗi:"Tài khoản phải từ 3 ký tự, bao gồm chữ và số"	Hệ thống hiển thị thông báo "Tài khoản phải từ 3 ký tự, bao gồm chữ và số"
[Login-2]	<b>Không nhập mật khẩu</b>		1. Nhập thông tin: username: admin1 password: bỏ trống 2. Click button Login	Hệ thống hiển thị thông báo lỗi:"Mật khẩu phải có ít nhất 6 ký tự, bao gồm chữ hoa, số và ký tự đặc biệt"	Hệ thống hiển thị thông báo lỗi:" Mật khẩu phải có ít nhất 6 ký tự, bao gồm chữ hoa, số và ký tự đặc biệt"
	<b>Check Business</b>				

[Login-3]	Kiểm tra login khi nhập đúng thông tin tài khoản	Hệ thống tồn tại account: username: 1 password: 1	1. Nhập thông tin: username: admin1 password: Admin@1 2. Click button Login	Login vào hệ thống thành công	Login vào hệ thống thành công
[Login-4]	Đăng nhập, đăng xuất nhiều lần trên cùng một phiên chạy chương trình	Hệ thống tồn tại account: username: admin1 password: admin1	1. Nhập thông tin: username: admin1 password: admin1 2. Click button Login 3. đăng xuất 4. lặp lại bước 1,2	Hệ thống đăng nhập bình thường	Đăng xuất khỏi hệ thống thành công

### 3.3. Kết quả thực hiện chương trình

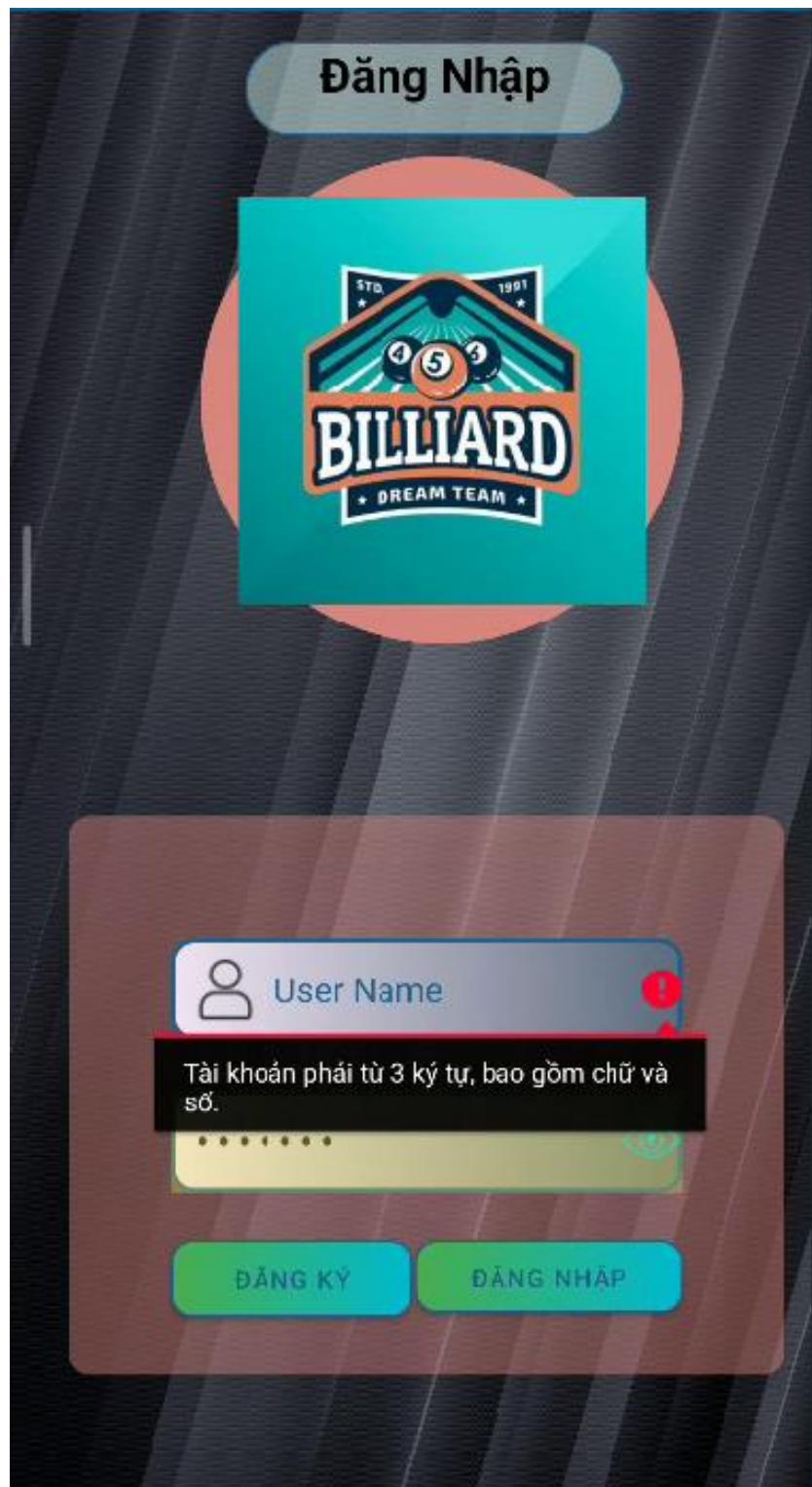
#### 3.3.1. Kết quả kiểm thử Unit test case

UNIT TEST REPORT																									
Project Name		Xây dựng ứng dụng quản lý quán Billiards bằng Android Studio		Creator																					
Project Code		UT		Reviewer/Approver																					
Document Code		UT_Test_Report_vx.x		Issue Date		<Date when this test report is created>																			
Notes		<List modules included in this release> ex: Release 1 includes 2 modules: Module1 and Module2																							
No	Requirement Name	Function code	Passed	Failed	Untested	N	A	B	Total Test Cases																
1	Phạm Văn Bắc	F_dang_ky	7	0	0	3	4	0	7																
2	Phạm Văn Bắc	F_dang_nhap	5	0	0	1	4	0	0																
3	Phạm Văn Bắc	F_them_ban	4	0	0	3	1	0	4																
4	Phạm Văn Bắc	F_tim_kiem_ban	6	0	0	3	1	2	6																
5	Phạm Văn Bắc	F_tim_kiem_khach_hang	5	0	4	2	2	1	5																
6	Phạm Văn Bắc	F_them_khach_hang	8	0	8	4	3	1	8																
7	Phạm Văn Bắc	F_sua_khach_hang	8	0	0	3	3	2	8																
8	Phạm Văn Bắc	F_them_san_pham	10	0	0	2	2	6	10																
9	Phạm Văn Bắc	F_sua_san_pham	6	1	0	2	4	1	7																
10	Phạm Văn Bắc	F_tim_kiem_san_pham	5	0	0	2	2	1	5																
<b>Sub total</b>		64	1	12	25	26	14		60																
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Passed Percent</p>  <table border="1"> <tr><td>Passed</td><td>88%</td></tr> <tr><td>Failed</td><td>1%</td></tr> <tr><td>Untested</td><td>16%</td></tr> </table> </div> <div style="width: 45%;"> <p>Test coverage</p> <table border="1"> <tr><td>Test coverage</td><td>108.33 %</td></tr> <tr><td>Test successful coverage</td><td>106.67 %</td></tr> <tr><td>Normal case</td><td>41.67 %</td></tr> <tr><td>Abnormal case</td><td>43.33 %</td></tr> <tr><td>Boundary case</td><td>23.33 %</td></tr> </table> </div> </div>										Passed	88%	Failed	1%	Untested	16%	Test coverage	108.33 %	Test successful coverage	106.67 %	Normal case	41.67 %	Abnormal case	43.33 %	Boundary case	23.33 %
Passed	88%																								
Failed	1%																								
Untested	16%																								
Test coverage	108.33 %																								
Test successful coverage	106.67 %																								
Normal case	41.67 %																								
Abnormal case	43.33 %																								
Boundary case	23.33 %																								
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Test Type</p>  <table border="1"> <tr><td>N</td><td>25</td></tr> <tr><td>A</td><td>26</td></tr> <tr><td>B</td><td>14</td></tr> </table> </div> </div>										N	25	A	26	B	14										
N	25																								
A	26																								
B	14																								

Bảng 3.12: Tổng hợp kết quả kiểm thử Unit test

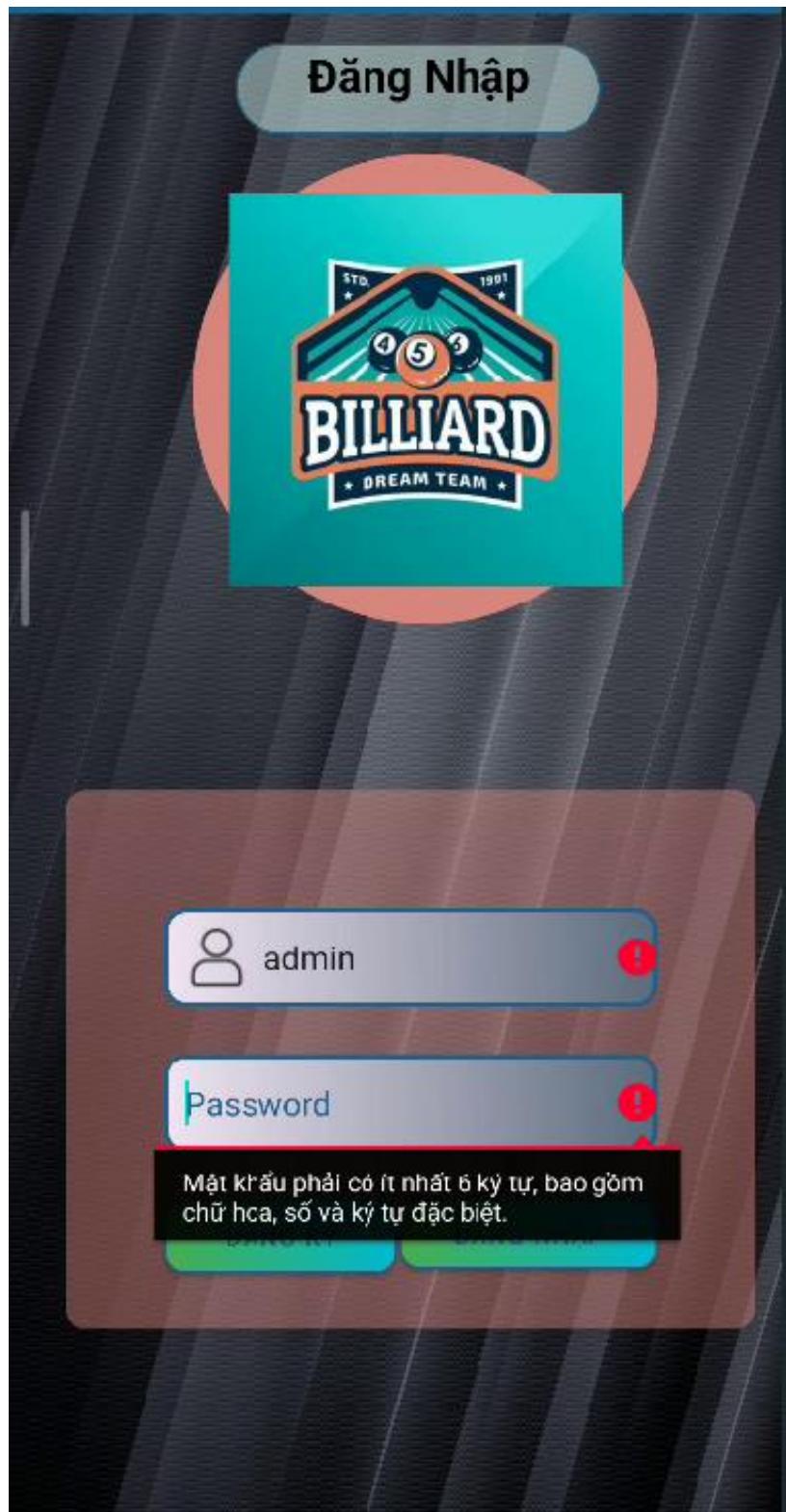
### 3.3.2. Kết quả kiểm thử login

✚ Kết quả [Login-1]



Hình 3.10: Kết quả test [Login-1]

✚ Kết quả [Login-2]



Hình 3.11: Kết quả test [Login-2]

✚ Kết quả [Login-3]



Hình 3.12: Kết quả test [Login-3]

### **3.4. Triển khai ứng dụng**

Ứng dụng đã được triển khai trên các nền tảng cụ thể như sau:

1) Amazon Appstore:

[https://www.amazon.com/dp/B0DHSF1C1D/ref=apps\\_sf\\_sta](https://www.amazon.com/dp/B0DHSF1C1D/ref=apps_sf_sta)

2) Github

<https://github.com/Kero-commits/Manage-Billiards/tree/master/app>

3) Youtube (test app)

<https://youtu.be/zzEiqqqyTkg?si=fX4VqyyURPk0JK3B>

4) Canva (Trình bày)

[https://www.canva.com/design/DAGIXK\\_PueA/ThhmrGBqSHHUj0eDrssJ4A/edit?utm\\_content=DAGIXK\\_PueA&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGIXK_PueA/ThhmrGBqSHHUj0eDrssJ4A/edit?utm_content=DAGIXK_PueA&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

## KẾT LUẬN

Trong quá trình thực hiện đồ án với chủ đề "Xây dựng ứng dụng quản lý quán Billiards bằng Android Studio", nhóm chúng tôi đã hoàn thành và triển khai một ứng dụng quản lý quán Billiards trực tuyến với các chức năng cơ bản và nâng cao. Ứng dụng này được thiết kế nhằm đáp ứng nhu cầu quản lý hiệu quả cho cả quản trị viên và nhân viên quán Billiards.

Về phần quản trị viên, ứng dụng cung cấp các công cụ để quản lý bàn Billiards, nhân viên, ca làm việc, hóa đơn, khách hàng và sản phẩm. Điều này cho phép quản trị viên dễ dàng theo dõi và điều chỉnh các hoạt động trong quán. Đối với nhân viên, ứng dụng hỗ trợ đăng nhập, quản lý bàn Billiards, quản lý khách hàng, đặt bàn và xuất hóa đơn, giúp họ làm việc hiệu quả hơn.

Chúng tôi đã áp dụng công nghệ hiện đại với Android Studio để phát triển ứng dụng, kết hợp với các công nghệ cơ sở dữ liệu và thư viện front-end để cải thiện trải nghiệm người dùng. Các tiêu chuẩn bảo mật và tối ưu hóa hiệu suất được chú trọng để đảm bảo ứng dụng hoạt động một cách an toàn và hiệu quả.

Dự án đã hoàn thành với sự tích hợp nhiều tính năng cần thiết cho cả quản trị viên và nhân viên, nâng cao chất lượng quản lý và sử dụng dịch vụ. Tuy nhiên, vẫn còn một số vấn đề cần cải thiện. Cụ thể, giao diện người dùng cần được tiếp tục cải thiện để tăng tính thẩm mỹ và nâng cao trải nghiệm sử dụng. Hơn nữa, mặc dù đã thực hiện các biện pháp tối ưu hóa, ứng dụng vẫn cần tiếp tục kiểm tra và cải thiện hiệu suất để đáp ứng tốt hơn với số lượng người dùng lớn.

Để nâng cao chất lượng và mở rộng phạm vi sử dụng của ứng dụng trong tương lai, chúng tôi đề xuất một số hướng phát triển. Cần phát triển thêm các tính năng mới như hỗ trợ đa ngôn ngữ, tích hợp thanh toán trực tuyến và tính năng gợi ý dịch vụ thông minh. Đồng thời, việc cải thiện giao diện người dùng và tiếp tục tối ưu hóa hiệu suất cũng là những điểm quan trọng cần được chú trọng.

Chúng tôi hy vọng rằng những kiến thức và kinh nghiệm thu được từ dự án này sẽ là nền tảng vững chắc cho sự nghiệp của chúng tôi trong tương lai. Chúng tôi cũng mong rằng ứng dụng này sẽ tiếp tục được phát triển và cải tiến, đóng góp vào sự phát triển của lĩnh vực quản lý quán Billiards tại Việt Nam.

## TÀI LIỆU THAM KHẢO

- [1]. Trần Xuân Thanh (2024), *Bài giảng môn: phát triển ứng dụng cho thiết bị di động*, Trường Đại học Công Nghệ Đông Á
- [2]. Google. "Android Developers Documentation", <https://developer.android.com/docs> [tham khảo ngày 20 tháng 08 năm 2024].
- [3]. Google. "Android Studio User Guide", <https://developer.android.com/studio/intro> [tham khảo ngày 21 tháng 08 năm 2024].
- [4]. Mark Murphy (2019). \*Hello, Android: Introducing Google's Mobile Development Platform\*, CommonsWare.
- [5]. Chet Haase và Romain Guy (2018). \*Android Programming: The Big Nerd Ranch Guide\*, Big Nerd Ranch Guides.
- [6]. Vogella. "Android Development Tutorial", <https://www.vogella.com/tutorials/Android/article.html> [tham khảo ngày 22 tháng 08 năm 2024].
- [7]. Dan Bader "Mastering Android Development with Kotlin", <https://www.danbader.dev/mastering-android-development-with-kotlin/> [tham khảo ngày 24 tháng 08 năm 2024].
- [8]. SQLite: "SQLite Documentation", <https://www.sqlite.org/docs.html> [tham khảo ngày 25 tháng 08 năm 2024].