

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC GIA ĐỊNH  
KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN**  
**XÂY DỰNG ỨNG DỤNG**  
**QUẢN LÝ CÔNG VIỆC CƠ BẢN**  
**TRÊN NỀN TẢNG ANDROID**  
**MÔN: LẬP TRÌNH**  
**CHO THIẾT BỊ DI ĐỘNG**

**Ngành: Công nghệ thông tin**

**Chuyên ngành: Kỹ thuật phần mềm**

**Giảng viên hướng dẫn: ThS. Tạ Chí Qui Nhơn**

**Sinh viên thực hiện: Nguyễn Gia Bảo**

**MSSV: 22150450**

**Lớp: 221407**

**TP. Hồ Chí Minh, tháng 11 năm 2024**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG**

1. **Sinh viên:** Nguyễn Gia Bảo
2. **Tên đề tài:** Xây dựng ứng dụng quản lý công việc cơ bản trên nền tảng android
3. **Nhận xét:**

**a) Những kết quả đạt được:**

.....

.....

.....

.....

.....

**b) Những hạn chế:**

.....

.....

.....

.....

.....

4. **Điểm đánh giá** (theo thang điểm 10, làm tròn đến 0.5):

Sinh viên:.....

Điểm số: ..... Điểm chữ: .....

TP. HCM, ngày ... tháng ... năm 20.....

**Giảng viên chấm thi**

(Ký và ghi rõ họ tên)

## LỜI CẢM ƠN

Em xin chân thành cảm ơn sâu sắc và được bày tỏ lòng biết ơn đến với Quý Thầy Cô của khoa Công Nghệ Thông Tin Trường Đại học Gia Định và đặc biệt đối với cô **ThS. Tạ Chí Qui Nhơn** đã tận tình hướng dẫn, đồng hành, động viên và chỉ bảo giúp tôi hoàn thành tốt bài báo cáo này cũng như truyền đạt những kiến thức bổ ích và rất quan trọng đối với quá trình đi làm của chúng em sau này.

Trong thời gian nghiên cứu đề tài tiểu luận vừa qua, em đã có nhiều cố gắng trong suốt quá trình thực hiện tiểu luận, tích cực trao đổi thông tin, sưu tầm, tham khảo tài liệu và học hỏi thêm kiến thức từ các Thầy Cô và các học viên khác, không chỉ nhận được những kiến thức đầy bổ ích về chuyên môn mà còn ở những lĩnh vực khác. Những trải nghiệm quý báu đó không chỉ giúp chúng em hoàn thành tốt bài báo cáo mà còn là hành trang quan trọng theo tôi trong suốt thời gian học tập và giúp tôi có thể tự tin bước vào đời làm việc sau này.

Do kiến thức và kinh nghiệm còn hạn chế cho nên bài báo cáo của em còn nhiều thiếu sót, kính mong được sự đánh giá, góp ý của thầy.

Cuối cùng, xin kính chúc cô luôn luôn khỏe mạnh, vui vẻ và đạt được nhiều thành công cao trong công tác giảng dạy. Chúc trường Đại học Gia Định sẽ luôn là nền tảng vững chắc cho nhiều thế hệ sinh viên tiếp bước trên con đường học tập.

**Xin chân thành cảm ơn!**

Nguyễn Gia Bảo

## MỤC LỤC

<b>CHƯƠNG 1: PHÂN TÍCH YÊU CẦU VÀ ĐẶC TẢ HỆ THỐNG.....</b>	<b>1</b>
1.1 TỔNG QUÁT ĐỀ TÀI .....	1
1.2 MÔ TẢ HỆ THỐNG.....	2
1.3 TỔNG QUAN LÝ THUYẾT .....	2
1.4 YÊU CẦU CHỨC NĂNG.....	5
1.5 YÊU CẦU PHI CHỨC NĂNG .....	5
1.6 SƠ ĐỒ USE CASE .....	6
<b>CHƯƠNG 2: THIẾT KẾ KIẾN TRÚC HỆ THỐNG.....</b>	<b>8</b>
2.1 SƠ ĐỒ LỚP (CLASS DIAGRAM) .....	8
2.2 THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG (UI DESIGN) .....	10
2.3 THIẾT KẾ CƠ SỞ DỮ LIỆU .....	11
<b>CHƯƠNG 3: PHÁT TRIỂN ỨNG DỤNG.....</b>	<b>13</b>
3.1 KẾT NỐI DỮ LIỆU (DATABASE HELPER).....	13
3.2 CẬP NHẬT DỮ LIỆU.....	15
3.3 XỬ LÝ CHỨC NĂNG .....	16
3.4 CHẠY CHƯƠNG TRÌNH.....	18
<b>KẾT LUẬN .....</b>	<b>26</b>
<b>PHỤ LỤC: TÀI LIỆU THAM KHẢO .....</b>	<b>28</b>

## **Chương 1: Phân tích yêu cầu và đặc tả hệ thống**

### **1.1 Tổng quát đề tài**

Android là một hệ điều hành mã nguồn mở được phát triển bởi Google, được thiết kế chủ yếu cho các thiết bị di động như điện thoại thông minh và máy tính bảng. Với khả năng hỗ trợ đa dạng phần cứng, giao diện thân thiện và hệ sinh thái ứng dụng phong phú, Android hiện chiếm lĩnh phần lớn thị phần trong ngành công nghiệp di động. Android không chỉ mang lại trải nghiệm sử dụng tiện lợi mà còn cung cấp một nền tảng phát triển mạnh mẽ cho các nhà lập trình viên với sự hỗ trợ từ ngôn ngữ lập trình Java, Kotlin và nhiều thư viện tích hợp. Chính sự phổ biến và linh hoạt của Android đã thúc đẩy việc phát triển các ứng dụng đa dạng, đáp ứng nhiều nhu cầu trong đời sống hàng ngày.

Ứng dụng quản lý công việc cơ bản là một phần mềm giúp người dùng tổ chức, theo dõi và quản lý các công việc hàng ngày một cách hiệu quả. Ứng dụng này sẽ bao gồm các chức năng chính như thêm, sửa, xóa và xem chi tiết các công việc, đồng thời hỗ trợ lưu trữ dữ liệu trên cơ sở dữ liệu cục bộ. Giao diện đơn giản, dễ sử dụng của ứng dụng giúp người dùng nhanh chóng tương tác và tối ưu hóa thời gian quản lý công việc. Với sự phát triển mạnh mẽ của các thiết bị di động, việc xây dựng một ứng dụng quản lý công việc trên nền tảng Android mang lại tính tiện dụng cao khi người dùng có thể sử dụng mọi lúc, mọi nơi.

Hơn nữa, việc xây dựng ứng dụng trên nền tảng Android mang lại nhiều lợi ích. Đầu tiên, Android là một hệ điều hành phổ biến, giúp ứng dụng có tiềm năng tiếp cận lượng lớn người dùng. Thứ hai, dự án này cũng giúp các nhà phát triển nâng cao kỹ năng lập trình, đặc biệt trong việc xử lý cơ sở dữ liệu, thiết kế giao diện người dùng và quản lý trạng thái ứng dụng.

Cuối cùng, thực hiện đề tài này không chỉ nhằm giải quyết nhu cầu thực tiễn mà còn góp phần đào sâu kiến thức chuyên môn trong lĩnh vực phát triển ứng dụng di động. Đây là cơ hội để áp dụng các nguyên lý thiết kế phần mềm, học hỏi quy trình phát triển ứng dụng từ khâu ý tưởng đến triển khai và tối ưu hóa sản phẩm. Dự án này còn là bước đệm để các lập trình viên tiến xa hơn trong lĩnh vực công nghệ, nơi sáng tạo và đổi mới không ngừng được khuyến khích.

## 1.2 Mô tả hệ thống

Hệ thống ứng dụng quản lý công việc cơ bản trên nền tảng Android được thiết kế nhằm hỗ trợ người dùng quản lý, tổ chức và theo dõi các công việc hàng ngày một cách thuận tiện và hiệu quả. Ứng dụng tập trung vào các chức năng chính bao gồm:

- Thêm công việc mới: Người dùng có thể tạo mới các công việc, nhập thông tin cần thiết như tên công việc, mô tả chi tiết.
- Sửa công việc: Ứng dụng cho phép người dùng cập nhật thông tin của công việc đã lưu, giúp duy trì tính chính xác và phù hợp với nhu cầu hiện tại.
- Xóa công việc: Người dùng có thể loại bỏ những công việc không còn cần thiết để giữ danh sách gọn gàng.
- Xem chi tiết công việc: Hỗ trợ hiển thị thông tin chi tiết của từng công việc, giúp người dùng dễ dàng nắm bắt nội dung và yêu cầu của công việc đó.

Hệ thống được phát triển dựa trên cơ sở dữ liệu SQLite, một công cụ quản lý dữ liệu nhẹ và phù hợp với ứng dụng di động. Toàn bộ dữ liệu về công việc của người dùng sẽ được lưu trữ cục bộ trên thiết bị, đảm bảo tính riêng tư và truy cập nhanh chóng.

Về giao diện, ứng dụng sử dụng phong cách thiết kế đơn giản, trực quan, phù hợp với các thiết bị di động Android. Các thành phần giao diện như danh sách công việc, nút thao tác (thêm, sửa, xóa) và các trường nhập liệu được sắp xếp hợp lý, giúp người dùng dễ dàng thao tác ngay cả khi lần đầu sử dụng.

Hệ thống này hướng tới nhóm đối tượng là những người có nhu cầu quản lý công việc cá nhân, chẳng hạn như sinh viên, nhân viên văn phòng hoặc bất kỳ ai muốn tổ chức lịch trình một cách khoa học. Với sự tiện lợi và các tính năng cơ bản, hệ thống này là một công cụ hữu ích cho việc quản lý thời gian hiệu quả trong cuộc sống hàng ngày.

## 1.3 Tổng quan lý thuyết

Ứng dụng đảm bảo lưu trữ dữ liệu lâu dài thông qua cơ sở dữ liệu SQLite, đồng thời sử dụng các phương pháp như Bundle để lưu trạng thái tạm thời trong các trường hợp ứng dụng bị tạm dừng hoặc chuyển màn hình.

### 1.3.1 Hệ điều hành Android

Android là một hệ điều hành mã nguồn mở dành cho các thiết bị di động, do Google phát triển. Với nền tảng dựa trên nhân Linux, Android hỗ trợ phong phú các công cụ và thư viện để xây dựng ứng dụng một cách linh hoạt. Hệ điều hành này chiếm phần lớn

thị phần di động trên toàn cầu, làm cho nó trở thành một nền tảng lý tưởng để phát triển các ứng dụng cá nhân và doanh nghiệp.

Một số đặc điểm nổi bật của Android:

- **Hệ sinh thái mở:** Hỗ trợ nhiều loại thiết bị khác nhau từ nhiều nhà sản xuất.
- **Giao diện người dùng linh hoạt:** Dựa trên XML và hỗ trợ đa dạng các thành phần giao diện như TextView, Button, ListView, RecyclerView.
- **Công cụ phát triển mạnh mẽ:** Sử dụng Android Studio và bộ SDK để lập trình.

### ***1.3.2 Kiến trúc ứng dụng Android***

Ứng dụng Android được tổ chức dựa trên kiến trúc gồm các thành phần chính: Activity, Intent, và SQLite Database:

Activity: Là nơi xử lý logic giao diện của ứng dụng. Mỗi màn hình trong ứng dụng (chẳng hạn như màn hình hiển thị danh sách công việc, màn hình thêm/sửa công việc) được biểu diễn bằng một Activity riêng.

Intent: Là cơ chế giao tiếp giữa các Activity, giúp truyền dữ liệu và chuyển đổi giữa các màn hình một cách linh hoạt.

SQLite Database: Được sử dụng để lưu trữ và quản lý dữ liệu công việc, như thông tin tên công việc, mô tả, và liên kết với tài khoản người dùng.

### ***1.3.3 Cơ sở dữ liệu SQLite***

SQLite là một cơ sở dữ liệu cục bộ nhẹ, phù hợp để lưu trữ dữ liệu của ứng dụng trực tiếp trên thiết bị. Trong ứng dụng:

- Cơ sở dữ liệu bao gồm hai bảng chính: bảng người dùng (chứa thông tin tài khoản) và bảng công việc (lưu danh sách công việc).
- Các thao tác chính như thêm, sửa, xóa, và truy vấn dữ liệu được thực hiện thông qua các hàm trong lớp quản lý cơ sở dữ liệu.
- Việc sử dụng khóa ngoại giúp liên kết các công việc với tài khoản tương ứng, đảm bảo rằng mỗi người dùng chỉ thấy được dữ liệu của mình.

### ***1.3.4 Giao diện người dùng***

Giao diện ứng dụng được thiết kế bằng XML và tuân thủ nguyên tắc Material Design, tập trung vào sự đơn giản và trực quan.

- Danh sách công việc: Hiển thị các công việc dưới dạng danh sách có thể tương tác, sử dụng các thành phần như ListView.
- Form nhập liệu: Giao diện thêm/sửa công việc bao gồm các trường nhập liệu để người dùng cung cấp thông tin công việc.
- Nút thao tác: Bao gồm các nút để thêm, sửa, xóa công việc hoặc quay lại màn hình chính

### ***1.3.5 Luồng hoạt động chính của ứng dụng***

Đăng nhập/đăng ký: Người dùng cần đăng nhập hoặc tạo tài khoản mới để bắt đầu sử dụng ứng dụng. Thông tin đăng nhập được kiểm tra trong cơ sở dữ liệu.

- Quản lý công việc: Sau khi đăng nhập, người dùng có thể:
- Xem danh sách các công việc hiện tại.
- Thêm công việc mới với tên và mô tả chi tiết.
- Sửa thông tin của công việc hiện có.
- Xóa công việc nếu không còn cần thiết.

Chuyển đổi giữa các màn hình: Việc di chuyển giữa các màn hình được thực hiện thông qua Intent, đồng thời truyền dữ liệu cần thiết để đảm bảo trạng thái của ứng dụng được duy trì.

### ***1.3.6 Tương tác người dùng và xử lý sự kiện***

Các sự kiện như nhấn nút, chọn công việc từ danh sách, hoặc nhập liệu được xử lý bằng cách sử dụng các trình xử lý sự kiện (OnClickListener, OnItemClickListener). Việc cập nhật giao diện sau khi thêm/sửa/xóa dữ liệu được thực hiện thông qua các adapter và hàm cập nhật danh sách dữ liệu.

### ***1.3.7 Tính bảo mật và hiệu năng***

Bảo mật: Dữ liệu được lưu trữ cục bộ trên thiết bị và chỉ người dùng đăng nhập hợp lệ mới có thể truy cập vào danh sách công việc của họ.

Hiệu năng: Sử dụng SQLite đảm bảo truy xuất dữ liệu nhanh chóng và tiết kiệm tài nguyên hệ thống.



## **1.4 Yêu cầu chức năng**

### **1.4.1 Đăng nhập và đăng ký tài khoản**

Người dùng có thể tạo tài khoản mới với thông tin đăng nhập duy nhất và sử dụng tài khoản này để truy cập ứng dụng.

Hệ thống kiểm tra thông tin đăng nhập hợp lệ trước khi cho phép truy cập.

### **1.4.2 Xem danh sách công việc**

Người dùng có thể xem danh sách các công việc được liên kết với tài khoản của mình.

Danh sách hiển thị rõ ràng tên công việc và mô tả chi tiết của từng công việc.

### **1.4.3 Thêm công việc mới**

Người dùng có thể tạo một công việc mới bằng cách nhập thông tin tên và mô tả.

Công việc mới được lưu vào cơ sở dữ liệu và hiển thị ngay trong danh sách.

### **1.4.4 Sửa công việc hiện tại**

Người dùng có thể chỉnh sửa thông tin của công việc đã có, bao gồm thay đổi mô tả.

Thông tin sửa đổi được cập nhật trực tiếp trong cơ sở dữ liệu.

### **1.4.5 Xóa công việc**

Người dùng có thể xóa các công việc không còn cần thiết.

Sau khi xóa, danh sách công việc được cập nhật để phản ánh thay đổi.

## **1.5 Yêu cầu phi chức năng**

### **1.5.1 Hiệu năng**

Hệ thống phải xử lý các thao tác thêm, sửa, xóa và truy vấn dữ liệu công việc trong thời gian ngắn, không gây chậm trễ đáng kể.

### **1.5.2 Bảo mật**

Dữ liệu người dùng, đặc biệt là thông tin tài khoản, phải được bảo mật trong cơ sở dữ liệu.

Chỉ tài khoản đăng nhập hợp lệ mới có thể truy cập vào danh sách công việc liên quan.

### **1.5.3. Khả năng sử dụng**

Giao diện ứng dụng phải thân thiện, trực quan, dễ dàng thao tác với người dùng không cần kiến thức kỹ thuật cao.

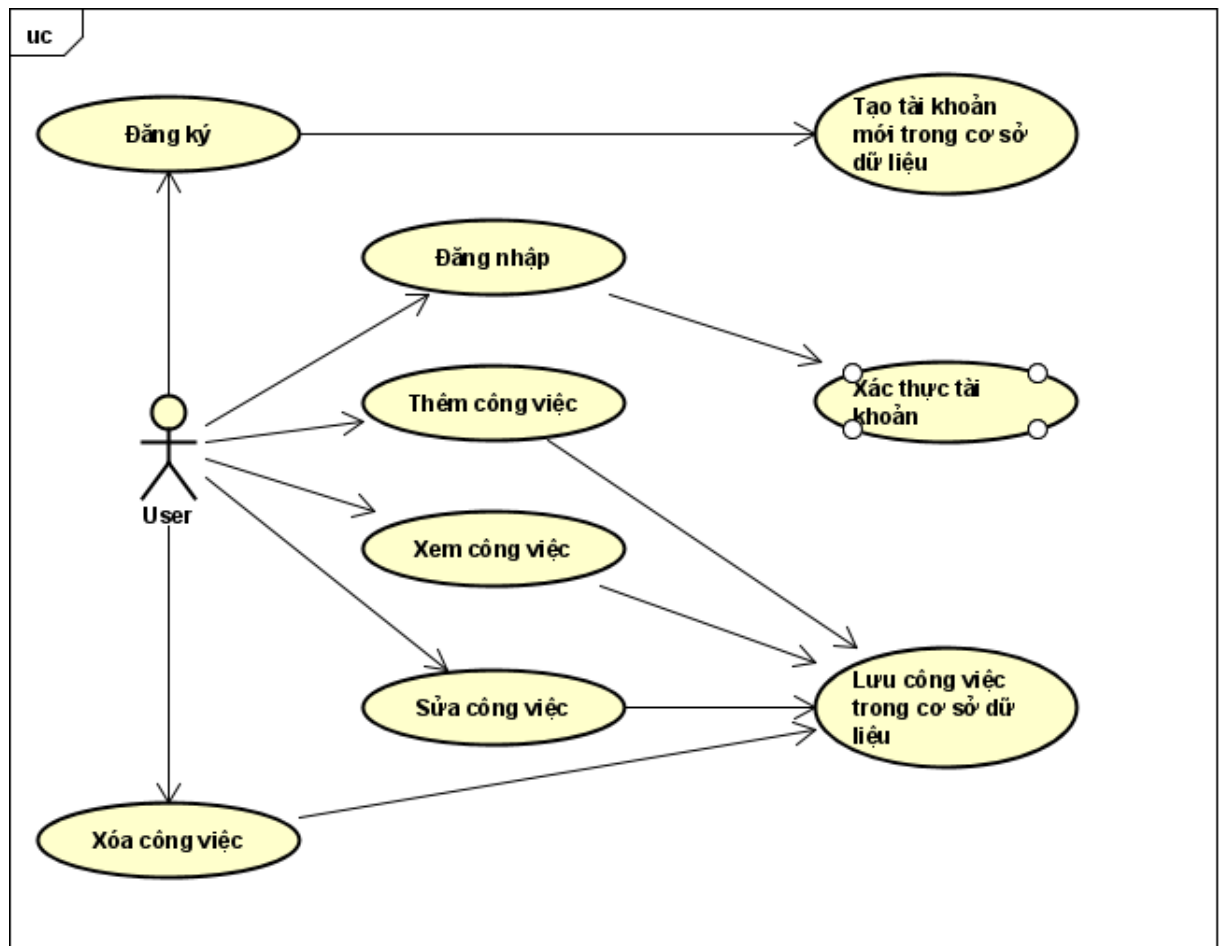
Các nút bấm và tính năng phải rõ ràng và dễ nhận biết.

## 1.6 Sơ đồ Use Case

Sơ đồ Use Case cho hệ thống gồm:

- **(Actor):** Người dùng là thực thể chính tương tác với ứng dụng.
- **Các Use Case chính:**
  - + Đăng ký tài khoản: là bước bắt buộc trước khi sử dụng các chức năng như Đăng nhập, Thêm, Xem, Sửa, hoặc Xóa công việc.
  - Người dùng cung cấp thông tin (tài khoản và mật khẩu).
  - Hệ thống kiểm tra tài khoản đã tồn tại hay chưa.
  - Nếu tài khoản chưa tồn tại, hệ thống lưu thông tin tài khoản mới vào cơ sở dữ liệu.
  - + Đăng nhập: Người dùng gửi thông tin đăng nhập, hệ thống xác thực tài khoản.
  - + Thêm công việc: Người dùng thêm công việc, hệ thống lưu vào cơ sở dữ liệu.
  - + Xem công việc: Người dùng yêu cầu hiển thị danh sách công việc, hệ thống truy xuất từ cơ sở dữ liệu.
  - + Sửa công việc: Người dùng chỉnh sửa công việc, hệ thống cập nhật trong cơ sở dữ liệu.
  - + Xóa công việc: Người dùng xóa công việc, hệ thống xóa khỏi cơ sở dữ liệu.

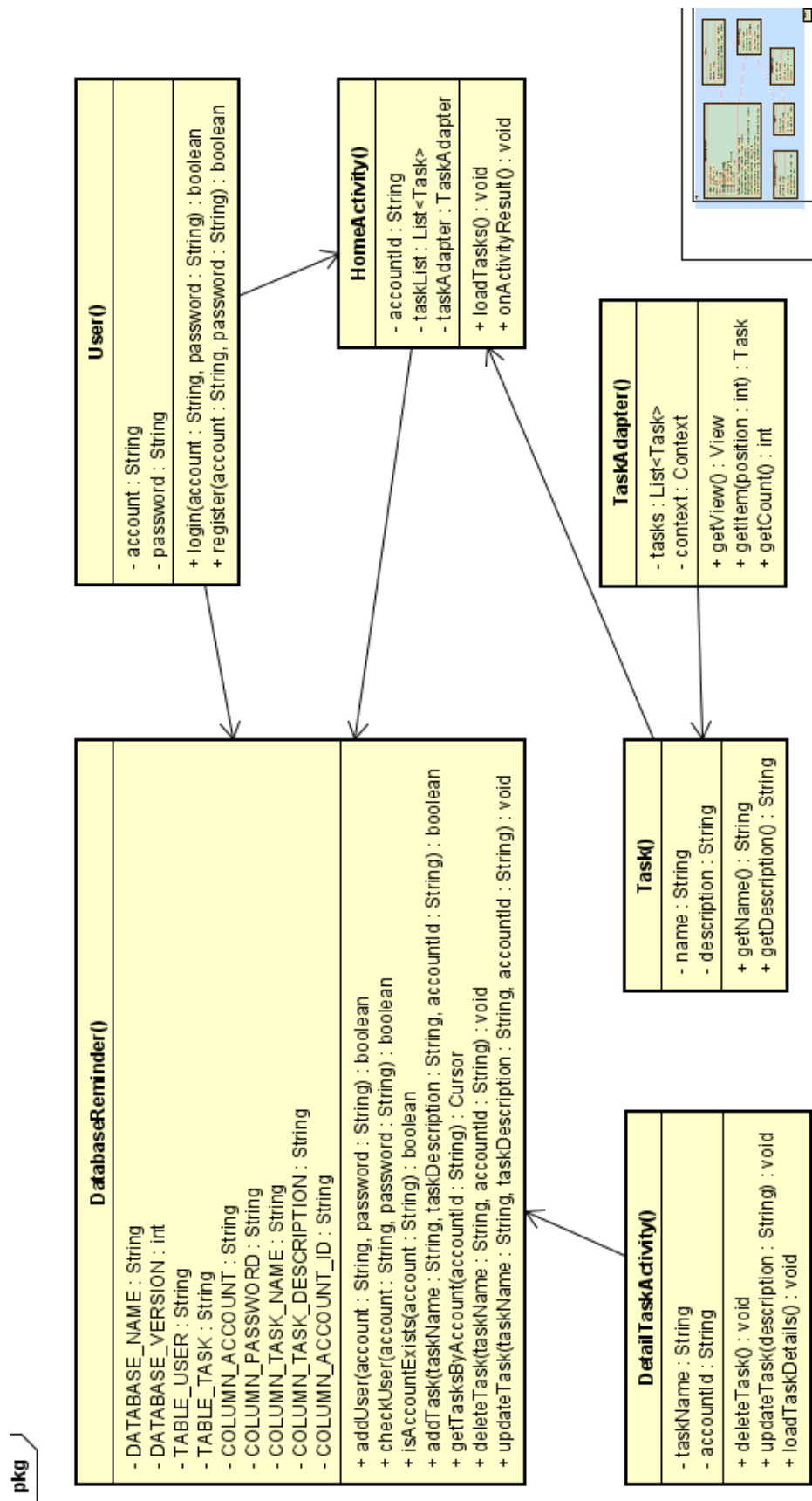
Sơ đồ này mô tả tổng quan cách hệ thống tương tác giữa người dùng và các chức năng chính trong ứng dụng.



(Sơ đồ Use Case)

## Chương 2: Thiết kế kiến trúc hệ thống

### 2.1 Sơ đồ lớp (Class Diagram)



(Sơ đồ Class)

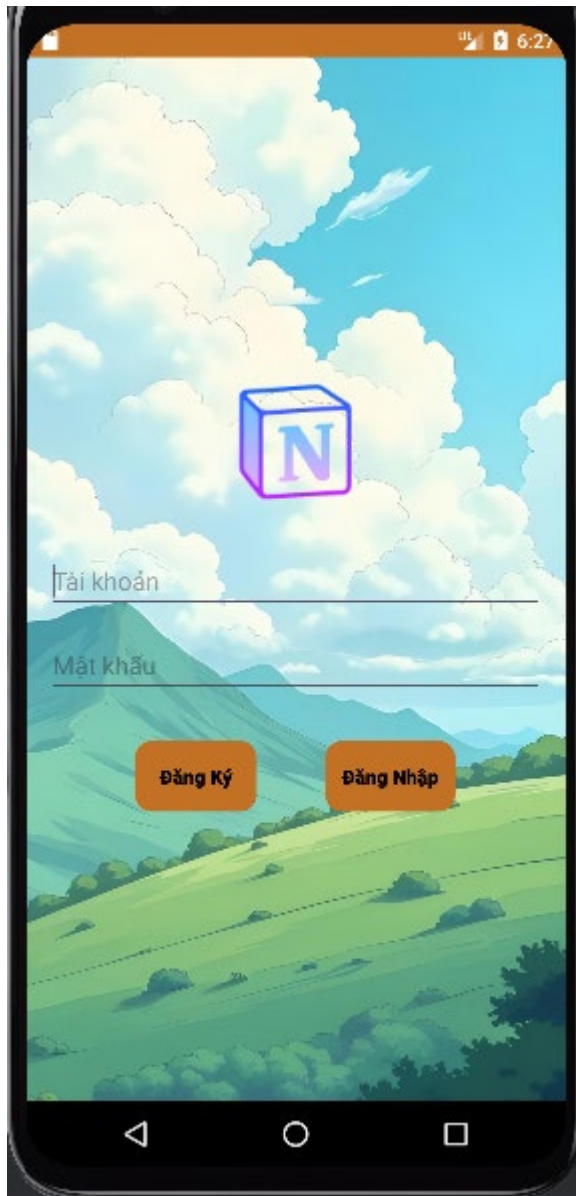
Giải thích sơ đồ:

- DatabaseReminder:
  - + Lớp chịu trách nhiệm quản lý cơ sở dữ liệu.
  - + Cung cấp các phương thức để thêm, sửa, xóa và truy vấn dữ liệu người dùng và công việc.
- Task:
  - + Đại diện cho một công việc với hai thuộc tính chính: tên và mô tả.
- + User:
  - + Đại diện cho thông tin tài khoản của người dùng.
  - + Cung cấp các phương thức liên quan đến đăng nhập và đăng ký tài khoản.
- HomeActivity:
  - + Hoạt động chính của ứng dụng, nơi hiển thị danh sách công việc của người dùng.
  - + Chứa các chức năng tải, thêm, sửa, và xóa công việc.
- DetailTaskActivity:
  - + Hiển thị thông tin chi tiết về một công việc cụ thể.
  - + Cho phép chỉnh sửa hoặc xóa công việc.
- TaskAdapter:
  - + Được sử dụng để kết nối danh sách công việc với giao diện (ListView).
  - + Cung cấp các phương thức hiển thị và quản lý danh sách công việc.

Mối quan hệ giữa các lớp:

- DatabaseReminder được sử dụng bởi cả HomeActivity và DetailTaskActivity để thao tác với dữ liệu.
- TaskAdapter tương tác với Task để hiển thị danh sách công việc trong HomeActivity.
- User được sử dụng trong quá trình đăng nhập và đăng ký.

## 2.2 Thiết kế giao diện người dùng (UI Design)



(Hình ảnh Trang đăng nhập)

Chức năng: Cho phép người dùng nhập tài khoản và mật khẩu để đăng nhập vào ứng dụng.

Thành phần giao diện:

- TextView: Tiêu đề "Đăng nhập".
- EditText:
  - + Trường nhập tài khoản (Hint: "Nhập tài khoản").
  - + Trường nhập mật khẩu (Hint: "Nhập mật khẩu", kiểu nhập mật khẩu).
- Button:
  - + Nút "Đăng nhập".
  - + Nút "Đăng ký" (chuyển đến màn hình Đăng ký).

- Thông báo lỗi: Hiện thị nếu tài khoản hoặc mật khẩu không đúng.

### 2.3 Thiết kế cơ sở dữ liệu

Dưới đây là thiết kế cơ sở dữ liệu cho hệ thống quản lý công việc với ít nhất 3 bảng: Người dùng (Users), Công việc (Tasks), và Danh mục (Categories).

#### 2.3.1 Bảng Users (Người dùng)

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	ID duy nhất của người dùng.
username	TEXT	NOT NULL, UNIQUE	Tên đăng nhập.
password	TEXT	NOT NULL	Mật khẩu của người dùng.
email	TEXT	NOT NULL, UNIQUE	Email của người dùng.

#### 2.3.2 Bảng Tasks (Công việc)

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
task_id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	ID duy nhất của công việc.
user_id	INTEGER	FOREIGN KEY REFERENCES Users(user_id)	ID người dùng liên kết.
category_id	INTEGER	FOREIGN KEY REFERENCES Categories(category_id)	ID danh mục liên kết.
task_name	TEXT	NOT NULL	Tên công việc.

task_description	TEXT	NULL	Mô tả chi tiết của công việc.
status	TEXT	DEFAULT 'Pending'	Trạng thái công việc (Pending, Done).
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Ngày tạo công việc.
updated_at	DATETIME	NULL	Ngày cập nhật công việc.

### 2.3.3 Bảng *Categories* (Danh mục)

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
category_id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	ID duy nhất của danh mục.
category_name	TEXT	NOT NULL, UNIQUE	Tên danh mục (Ví dụ: Công việc cá nhân, Công việc công ty).



## Chương 3: Phát triển ứng dụng

### 3.1 Kết nối dữ liệu (Database Helper)

```
public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "TaskManager.db";

    private static final int DATABASE_VERSION = 1;

    // Table Users

    private static final String TABLE_USERS = "Users";

    private static final String COLUMN_USER_ID = "user_id";

    private static final String COLUMN_USERNAME = "username";

    private static final String COLUMN_PASSWORD = "password";

    private static final String COLUMN_EMAIL = "email";

    // Table Tasks

    private static final String TABLE_TASKS = "Tasks";

    private static final String COLUMN_TASK_ID = "task_id";

    private static final String COLUMN_TASK_NAME = "task_name";

    private static final String COLUMN_TASK_DESCRIPTION = "task_description";

    private static final String COLUMN_STATUS = "status";

    private static final String COLUMN_CREATED_AT = "created_at";

    private static final String COLUMN_UPDATED_AT = "updated_at";

    private static final String COLUMN_USER_ID_FK = "user_id";

    // Table Categories

    private static final String TABLE_CATEGORIES = "Categories";

    private static final String COLUMN_CATEGORY_ID = "category_id";

    private static final String COLUMN_CATEGORY_NAME = "category_name";
```

```

public DatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

@Override

public void onCreate(SQLiteDatabase db) {

    // Create Users table

    String createUsersTable = "CREATE TABLE " + TABLE_USERS + " (" +
        COLUMN_USER_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        COLUMN_USERNAME + " TEXT NOT NULL UNIQUE, " +
        COLUMN_PASSWORD + " TEXT NOT NULL, " +
        COLUMN_EMAIL + " TEXT NOT NULL UNIQUE)";

    db.execSQL(createUsersTable);

    // Create Categories table

    String createCategoriesTable = "CREATE TABLE " + TABLE_CATEGORIES + " (" +
        COLUMN_CATEGORY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,
    " +
        COLUMN_CATEGORY_NAME + " TEXT NOT NULL UNIQUE)";

    db.execSQL(createCategoriesTable);

    // Create Tasks table

    String createTasksTable = "CREATE TABLE " + TABLE_TASKS + " (" +
        COLUMN_TASK_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        COLUMN_TASK_NAME + " TEXT NOT NULL, " +
        COLUMN_TASK_DESCRIPTION + " TEXT, " +
        COLUMN_STATUS + " TEXT DEFAULT 'Pending', " +
        COLUMN_CREATED_AT + " DATETIME DEFAULT CURRENT_TIMESTAMP,
    " +
        COLUMN_UPDATED_AT + " DATETIME, " +

```

```

        COLUMN_USER_ID_FK + " INTEGER, " +
        COLUMN_CATEGORY_ID + " INTEGER, " +
        "FOREIGN KEY(" + COLUMN_USER_ID_FK + ") REFERENCES " +
        TABLE_USERS + "(" + COLUMN_USER_ID + ")", " +
        "FOREIGN KEY(" + COLUMN_CATEGORY_ID + ") REFERENCES " +
        TABLE_CATEGORIES + "(" + COLUMN_CATEGORY_ID + ")");

        db.execSQL(createTasksTable);
    }

```

```

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_TASKS);

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_CATEGORIES);

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_USERS);

    onCreate(db);

}

```

## 3.2 Cập nhật dữ liệu

Ví dụ thêm, sửa, và xóa công việc.

### 3.2.1 Thêm công việc

```

public boolean addTask(String taskName, String description, String status, int
userId, int categoryId) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("task_name", taskName);
    values.put("task_description", description);
    values.put("status", status);
    values.put("user_id", userId);
    values.put("category_id", categoryId);

    long result = db.insert("Tasks", null, values);
}

```

```

        db.close();
        return result != -1;
    }

```

### 3.2.2 Cập nhật công việc

```

public boolean updateTask(int taskId, String taskName, String description, String
status) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("task_name", taskName);
    values.put("task_description", description);
    values.put("status", status);
    values.put("updated_at", System.currentTimeMillis());

    int result = db.update("Tasks", values, "task_id=?", new
String[] {String.valueOf(taskId)});
    db.close();
    return result > 0;
}

```

### 3.2.3 Xóa công việc

```

public boolean deleteTask(int taskId) {
    SQLiteDatabase db = this.getWritableDatabase();
    int result = db.delete("Tasks", "task_id=?", new
String[] {String.valueOf(taskId)});
    db.close();
    return result > 0;
}

```

## 3.3 Xử lý chức năng

Ví dụ lấy danh sách công việc theo người dùng

```

public List<Task> getTasksByUser(int userId) {
    List<Task> tasks = new ArrayList<>();
    SQLiteDatabase db = this.getReadableDatabase();

```

```

String query = "SELECT * FROM Tasks WHERE user_id=?";
Cursor cursor = db.rawQuery(query, new String[]{String.valueOf(userId)});

if (cursor.moveToFirst()) {
    do {
        Task task = new Task(
            cursor.getInt(cursor.getColumnIndex("task_id")),
            cursor.getString(cursor.getColumnIndex("task_name")),
            cursor.getString(cursor.getColumnIndex("task_description")),
            cursor.getString(cursor.getColumnIndex("status")),
            cursor.getInt(cursor.getColumnIndex("user_id")),
            cursor.getInt(cursor.getColumnIndex("category_id"))
        );
        tasks.add(task);
    } while (cursor.moveToNext());
}
cursor.close();
db.close();
return tasks;
}

```

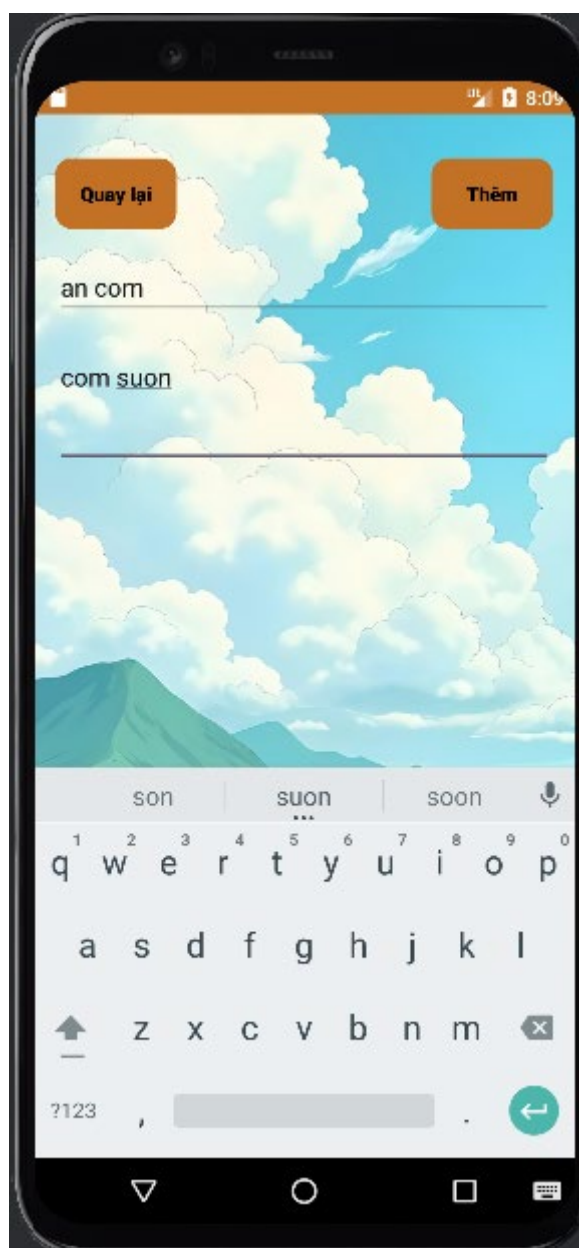
### 3.4 Chạy chương trình

#### 3.4.1 Thêm công việc

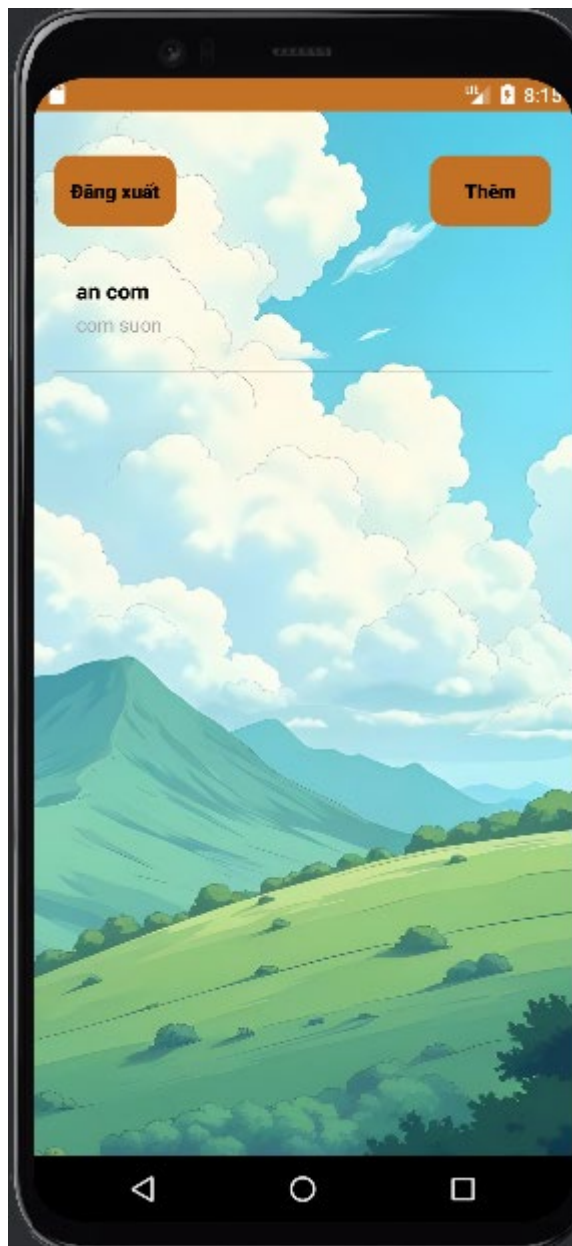
Sau khi đăng nhập thành công, sẽ hiển thị trang chủ



Sau đó, ta nhấn nút Thêm, lúc này sẽ chuyển sang trang “Thêm”



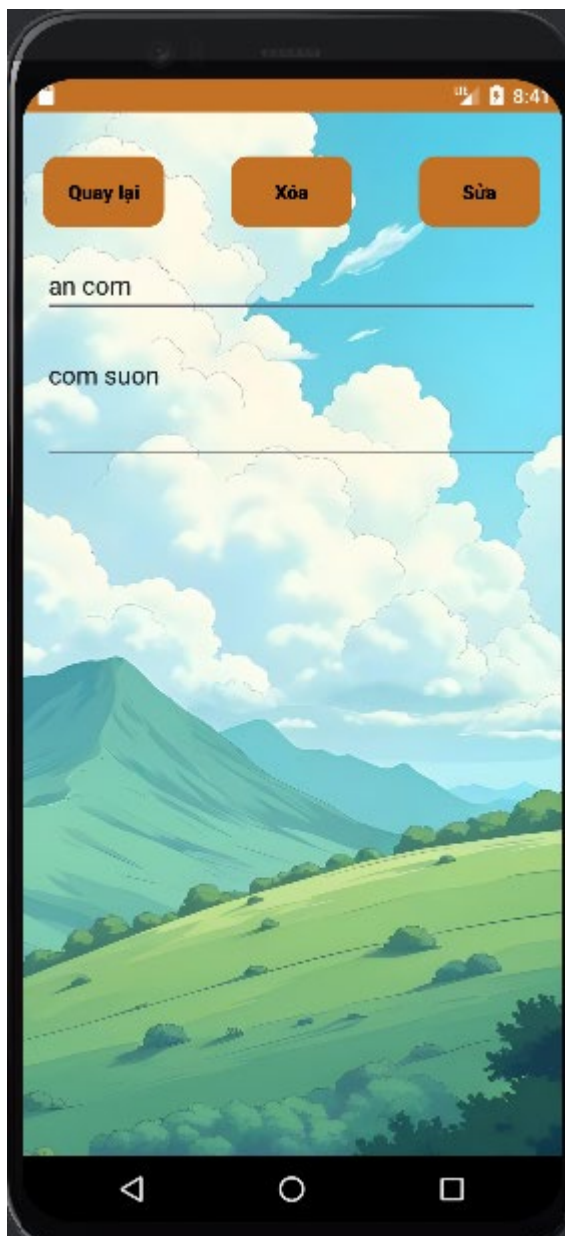
Khi điền xong 2 mục ô Tên công việc và Mô tả công việc, khi nhấn thêm sẽ hiện thông báo đã thêm công việc thành công, và quay lại trang chủ cùng với công việc đã được tạo thêm trước đó



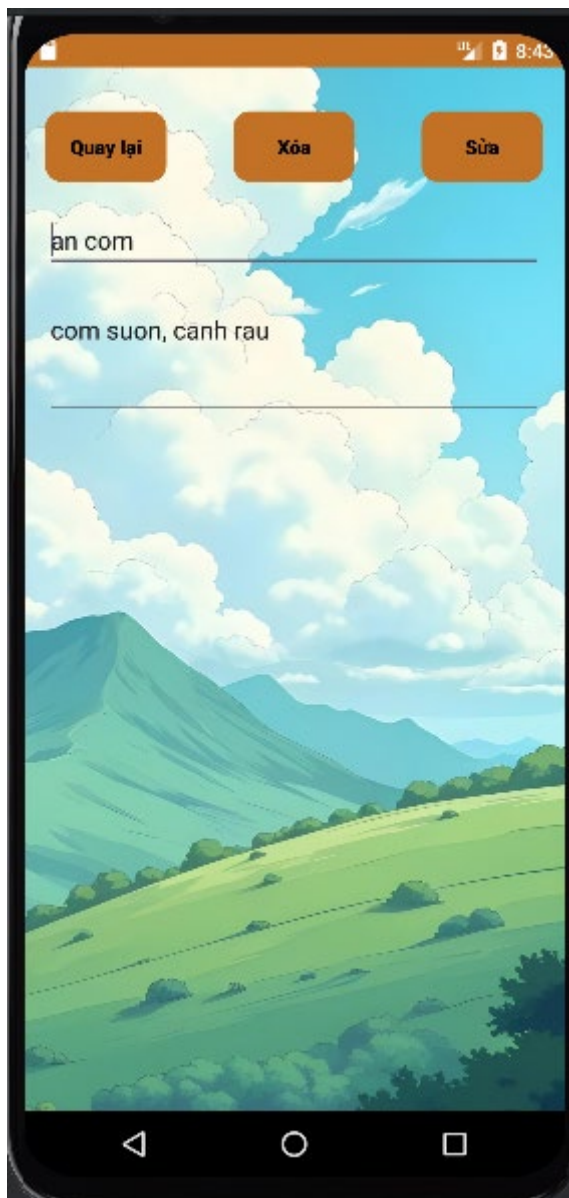


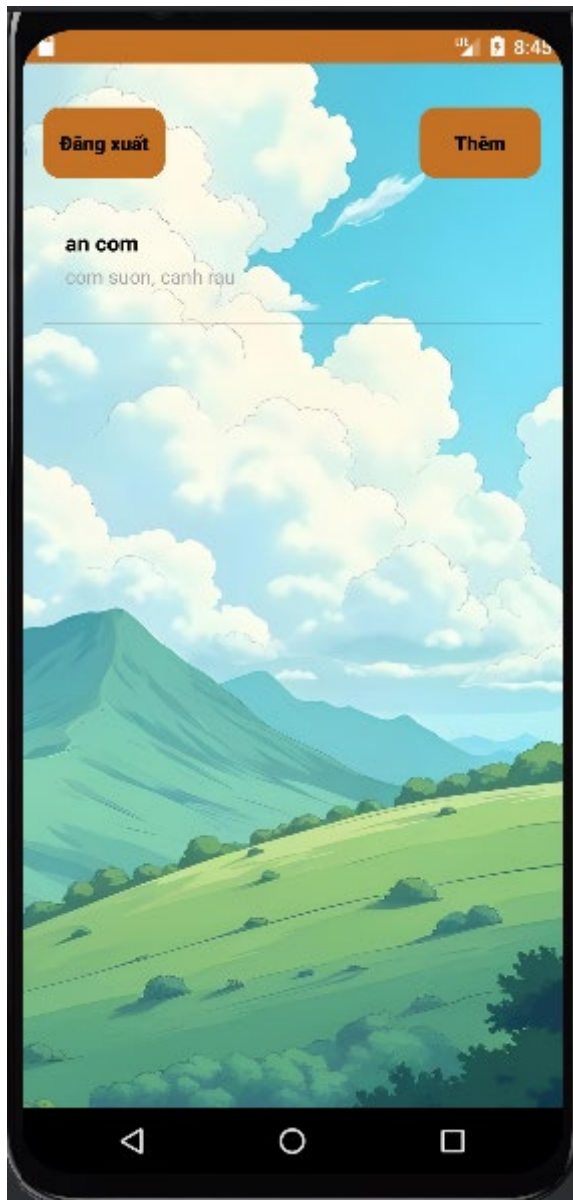
### 3.4.2 Sửa công việc

Khi muốn sửa thông tin “công việc đã thêm” thì hãy nhấn “tên công việc” trong trang chủ



Đổi mô tả công việc và sau đó nhấn nút sửa, khi ấy sẽ hiện thông báo đã sửa thành công và quay về trang chủ

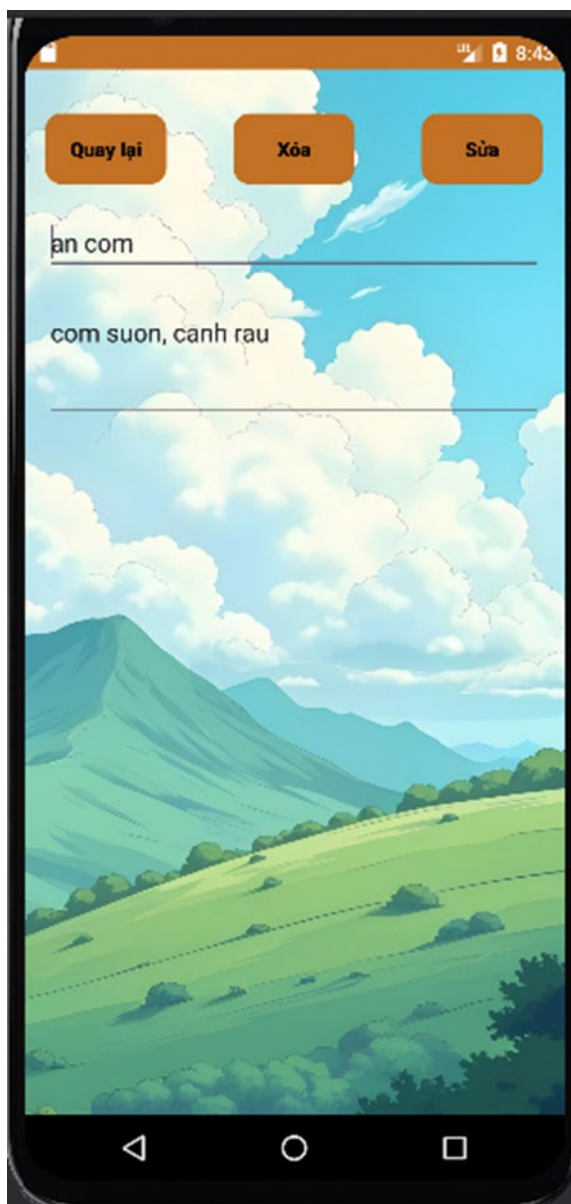




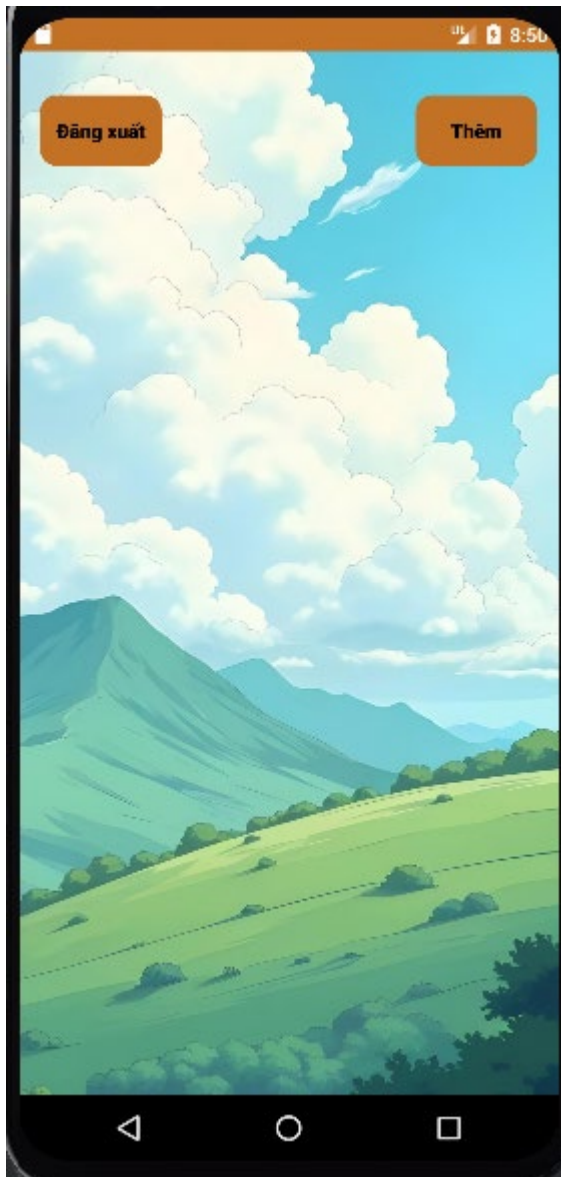
Thông tin đã được thay đổi

### 3.4.3 Xóa công việc

Khi muốn xóa công việc đã thêm, nhấn tên công việc muốn xóa ở trang chủ



Sau đó nhấn nút xóa, sẽ hiện thông báo đã xóa công việc thành công và quay về trang chủ. Lúc này, Trang chủ đã không còn hiện công việc được xóa trước đó.



## Kết luận

Sau khi hoàn thành ứng dụng quản lý công việc cơ bản trên nền tảng Android, hệ thống đã đáp ứng được các yêu cầu đề ra, bao gồm quản lý công việc, người dùng, và các danh mục. Giao diện đơn giản, dễ sử dụng, phù hợp với người dùng cá nhân hoặc nhóm nhỏ muốn theo dõi và quản lý công việc hàng ngày. Dữ liệu được lưu trữ an toàn trong cơ sở dữ liệu SQLite, đảm bảo tính toàn vẹn và khả năng truy xuất nhanh chóng.

### Ưu điểm:

1. Tính tiện lợi: Ứng dụng hoạt động ổn định trên hầu hết các thiết bị Android và không yêu cầu kết nối mạng, giúp người dùng dễ dàng sử dụng mọi lúc, mọi nơi.
2. Giao diện thân thiện: Thiết kế giao diện người dùng đơn giản, trực quan, phù hợp với người mới sử dụng.
3. Chức năng quản lý công việc rõ ràng: Cho phép người dùng thêm, sửa, xóa và theo dõi trạng thái công việc một cách dễ dàng.
4. Khả năng mở rộng: Hệ thống được xây dựng với kiến trúc linh hoạt, dễ dàng mở rộng thêm các chức năng mới trong tương lai.

### Nhược điểm:

1. Giới hạn lưu trữ: Sử dụng SQLite làm cơ sở dữ liệu nên khả năng xử lý dữ liệu lớn bị hạn chế, khó đáp ứng nhu cầu của các tổ chức lớn.
2. Thiếu đồng bộ hóa: Hiện tại ứng dụng chưa hỗ trợ tính năng đồng bộ dữ liệu trên nhiều thiết bị hoặc lưu trữ đám mây.
3. Bảo mật cơ bản: Mặc dù có quản lý người dùng, ứng dụng chưa tích hợp các biện pháp bảo mật nâng cao như mã hóa dữ liệu hoặc xác thực hai yếu tố.

### Hướng phát triển:

1. Tích hợp lưu trữ đám mây: Phát triển tính năng đồng bộ hóa dữ liệu với các dịch vụ đám mây như Google Drive hoặc Firebase để người dùng có thể truy cập công việc từ nhiều thiết bị.
2. Cải thiện giao diện người dùng: Nâng cấp giao diện để hỗ trợ chế độ tối (dark mode) và tùy biến cao hơn cho người dùng.
3. Bổ sung nhắc nhở và thông báo: Tích hợp chức năng nhắc nhở công việc qua thông báo đẩy (push notifications) hoặc email.

4. Mở rộng khả năng bảo mật: Cải thiện bảo mật bằng cách triển khai mã hóa dữ liệu, xác thực đa yếu tố và tích hợp chức năng quản lý mật khẩu.
5. Phát triển ứng dụng đa nền tảng: Xây dựng phiên bản dành cho iOS hoặc phát triển ứng dụng trên nền tảng web để mở rộng phạm vi sử dụng.

Ứng dụng quản lý công việc cơ bản trên Android đã hoàn thành và đáp ứng tốt các yêu cầu đề ra. Tuy nhiên, để cạnh tranh và phát triển hơn nữa, cần khắc phục các hạn chế và bổ sung các tính năng tiên tiến. Đây là bước khởi đầu vững chắc để tiếp tục cải tiến và mở rộng ứng dụng trong tương lai..

## PHỤ LỤC: TÀI LIỆU THAM KHẢO

1. Develop for Android. (n.d.). Android Developers.  
<https://developer.android.com/develop>
2. SQLite Documentation. (n.d.). <https://www.sqlite.org/docs.html>
3. Những lý thuyết và bài tập thực hành đến từ ThS. **Tạ Chí Qui Nhơn**