

TRƯỜNG ĐẠI HỌC LÂM NGHIỆP  
CƠ ĐIỆN & CÔNG TRÌNH



**ĐỒ ÁN MÔN HỌC LẬP TRÌNH ANDROID**

ĐỀ TÀI: Ứng dụng NoteApp trên Android kết hợp Node.js Backend

Giảng viên hướng dẫn:  
Người thực hiện:  
Lớp:  
Mã SV:

ThS Mai Hà An  
Nguyễn Diệu Linh  
HTTTA-K67  
2275801020693

## MỤC LỤC

1. MỞ ĐẦU.....	2
2. PHÂN TÍCH YÊU CẦU HỆ THỐNG .....	2
2.1. Yêu cầu chức năng.....	2
2.2. Yêu cầu phi chức năng .....	3
3. THIẾT KẾ HỆ THỐNG .....	3
3.1. Thiết kế cơ sở dữ liệu.....	3
3.1.1. Bảng users .....	3
3.1.2. Bảng notes .....	3
3.1.3. Sơ đồ ERD PlantUML .....	4
3.2. Thiết kế kiến trúc ứng dụng .....	4
3.2.1. Sơ đồ kiến trúc PlantUML .....	4
3.2.2. Mô tả kiến trúc.....	5
3.3. Thiết kế giao diện người dùng.....	6
3.3.1. Màn hình chính (MainActivity) .....	6
3.3.2. Màn hình NoteDetailActivity .....	6
3.3.3. Màn hình LoginActivity / RegisterActivity .....	6
4. TRIỂN KHAI ỨNG DỤNG .....	6
4.1. Backend Node.js + Express .....	7
4.1.1. Cấu trúc thư mục.....	7
4.1.2. server.js.....	7
4.2. Frontend Android (Java).....	7
4.3. Kết nối API bằng Retrofit .....	8
4.4. Xử lý dữ liệu local với SQLite .....	8
5. KẾT QUẢ THỰC NGHIỆM .....	8
6. ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN .....	8
6.1. Ưu điểm.....	9
6.2. Hạn chế .....	9
6.3. Hướng phát triển.....	9

# 1. MỞ ĐẦU

Ứng dụng **NoteApp** là một ứng dụng quản lý ghi chú trên Android, cho phép người dùng đăng ký, đăng nhập, tạo, chỉnh sửa, xóa và xem chi tiết ghi chú. Ứng dụng được xây dựng theo mô hình client-server, backend sử dụng **Node.js + Express** và cơ sở dữ liệu **SQLite** (có thể mở rộng sang PostgreSQL hoặc MongoDB). Frontend sử dụng **Android Java**, kết nối với backend thông qua **Retrofit**.

## Mục tiêu của ứng dụng:

- Quản lý ghi chú cá nhân.
- Hỗ trợ đa nền tảng trong tương lai.
- Thực hành kiến thức lập trình Android, REST API, Retrofit, và SQLite.

# 2. PHÂN TÍCH YÊU CẦU HỆ THỐNG

## 2.1. Yêu cầu chức năng

STT	Chức năng	Mô tả	Input	Output
1	Đăng ký	Tạo tài khoản mới	username, email, password	success, message
2	Đăng nhập	Xác thực người dùng	email, password	token, success, message
3	Xem danh sách note	Lấy danh sách ghi chú	token	[[id, title, content, createdAt]]
4	Xem chi tiết note	Lấy chi tiết ghi chú	token, noteId	{id, title, content, createdAt, updatedAt}
5	Tạo note mới	Thêm ghi chú	token, title, content	{id, title, content}

6	Chỉnh sửa note	Cập nhật ghi chú	token, notelId, title, content	{id, title, content}
7	Xóa note	Xóa ghi chú	token, notelId	success/failure

## 2.2. Yêu cầu phi chức năng

- **Bảo mật:** Token JWT, HTTPS (nếu triển khai thực tế).
- **Hiệu năng:** Tải danh sách ghi chú nhanh (<1s).
- **Giao diện:** Đơn giản, thân thiện, responsive với các màn hình Android 5.0+.
- **Tính ổn định:** Xử lý lỗi kết nối, lỗi API, input trống.

# 3. THIẾT KẾ HỆ THỐNG

## 3.1. Thiết kế cơ sở dữ liệu

### 3.1.1. Bảng users

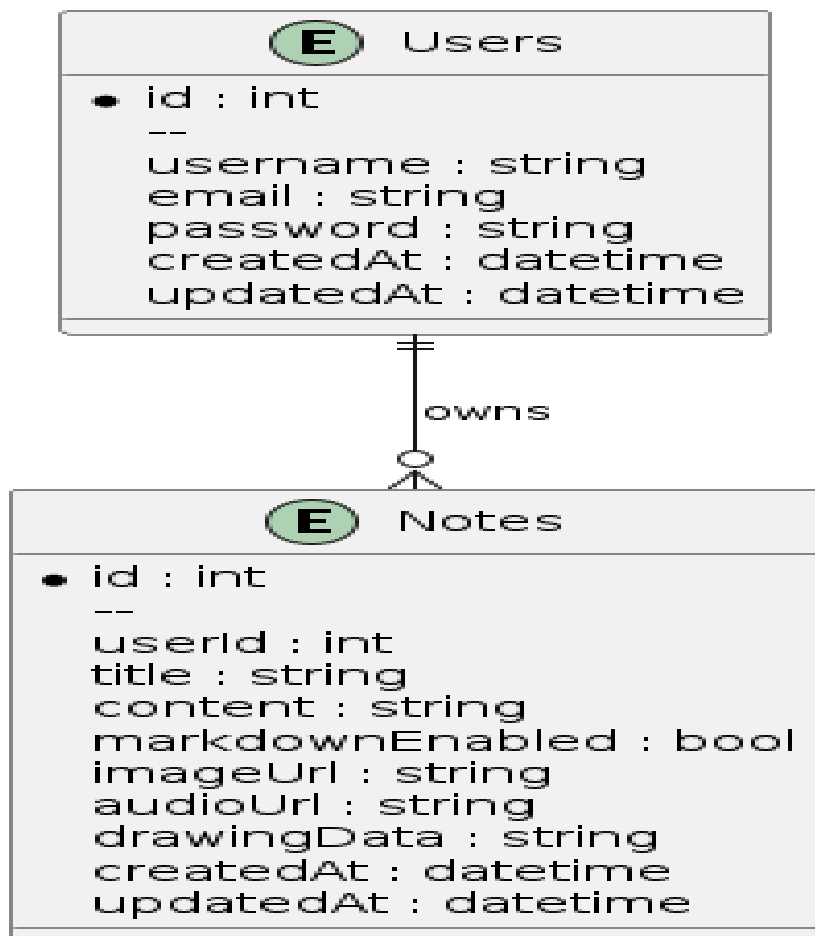
Cột	Kiểu dữ liệu	Khóa	Ghi chú
id	INTEGER	PK, AUTOINCREMENT	ID người dùng
username	TEXT	-	Tên hiển thị
email	TEXT	UNIQUE	Email đăng nhập
password	TEXT	-	Mật khẩu hash
createdAt	DATETIME	-	Thời gian tạo
updatedAt	DATETIME	-	Thời gian cập nhật

### 3.1.2. Bảng notes

Cột	Kiểu dữ liệu	Khóa	Ghi chú
id	INTEGER	PK, AUTOINCREMENT	ID note
userId	INTEGER	FK(users.id)	Người sở hữu note
title	TEXT	-	Tiêu đề note
content	TEXT	-	Nội dung note
markdownEnabled	BOOLEAN	-	Có hỗ trợ markdown
imageUrl	TEXT	-	Đường dẫn ảnh
audioUrl	TEXT	-	Đường dẫn audio

<b>drawingData</b>	TEXT	-	Dữ liệu vẽ tay
<b>createdAt</b>	DATETIME	-	Thời gian tạo
<b>updatedAt</b>	DATETIME	-	Thời gian cập nhật

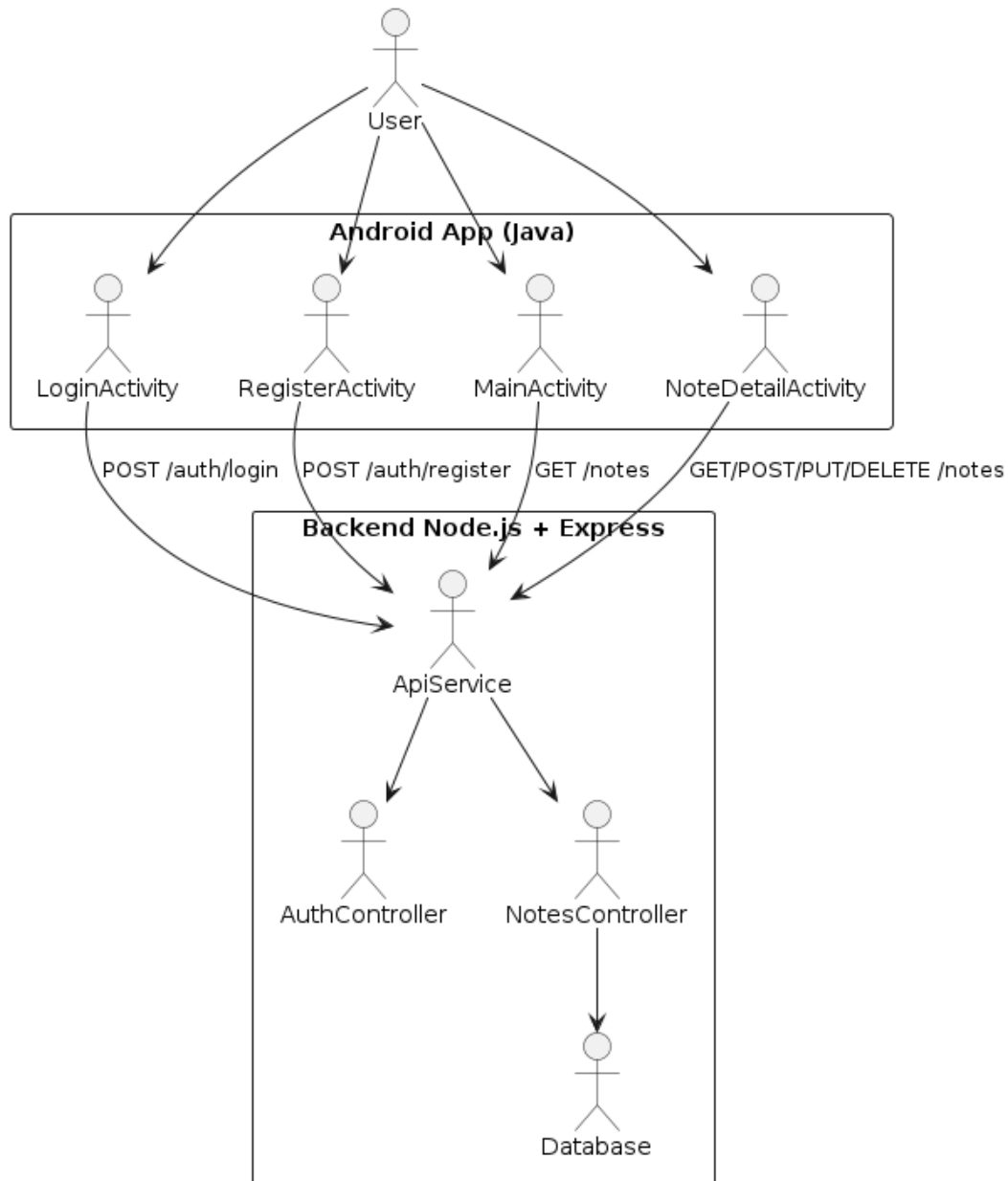
### 3.1.3. Sơ đồ ERD PlantUML



## 3.2. Thiết kế kiến trúc ứng dụng

Ứng dụng sử dụng **kiến trúc client-server**, frontend Android kết nối backend Node.js qua REST API.

### 3.2.1. Sơ đồ kiến trúc PlantUML



### 3.2.2. Mô tả kiến trúc

- **LoginActivity / RegisterActivity:** Xác thực người dùng.
- **MainActivity:** Hiển thị danh sách note, cho phép xóa.
- **NoteDetailActivity:** Tạo / chỉnh sửa note.
- **ApiService (Retrofit):** Kết nối backend.

- **Database (SQLite):** Lưu dữ liệu local, đồng bộ API.

### 3.3. Thiết kế giao diện người dùng

#### 3.3.1. Màn hình chính (MainActivity)

Thành phần	Mô tả
<b>ListView</b>	Hiển thị danh sách note
<b>Button “+”</b>	Tạo note mới
<b>Button Logout</b>	Đăng xuất

#### 3.3.2. Màn hình NoteDetailActivity

Thành phần	Mô tả
<b>EditText title</b>	Tiêu đề note
<b>EditText content</b>	Nội dung note
<b>Button Save</b>	Lưu note (tạo mới hoặc cập nhật)

#### 3.3.3. Màn hình LoginActivity / RegisterActivity

Thành phần	Mô tả
EditText email/username/password	Nhập thông tin
Button Login / Register	Xác nhận
Button chuyển màn hình	Chuyển login ↔ register

## 4. TRIỂN KHAI ỨNG DỤNG

### 4.1. Backend Node.js + Express

#### 4.1.1. Cấu trúc thư mục

```
backend/  
├── server.js  
├── routes/  
│   ├── auth.js  
│   └── notes.js  
├── controllers/  
│   ├── authController.js  
│   └── notesController.js  
├── models/  
│   ├── user.js  
│   └── note.js  
└── database.js
```

#### 4.1.2. server.js

```
import express from 'express';  
import cors from 'cors';  
import authRoutes from './routes/auth.js';  
import notesRoutes from './routes/notes.js';  
  
const app = express();  
app.use(cors());  
app.use(express.json());  
  
app.use('/api/auth', authRoutes);  
app.use('/api/notes', notesRoutes);  
  
app.listen(3000, () => console.log("Server running on port 3000"));
```

## 4.2. Frontend Android (Java)

- **Activities:** LoginActivity, RegisterActivity, MainActivity, NoteDetailActivity.
- **Adapter:** NoteAdapter cho ListView.
- **DatabaseHelper:** SQLite lưu dữ liệu offline.

## 4.3. Kết nối API bằng Retrofit

```
public interface ApiService {  
    @POST("auth/login")  
    Call<LoginResponse> login(@Body LoginRequest request);  
  
    @GET("notes")  
    Call<List<Note>> getNotes(@Header("Authorization") String token);  
  
    @POST("notes")  
    Call<Note> createNote(@Header("Authorization") String token, @Body Note note);  
}
```

## 4.4. Xử lý dữ liệu local với SQLite

- Lưu trữ note offline khi không có mạng.
- Các phương thức: insertNote, updateNote, deleteNoteById, getAllNotes, getNoteById.

# 5. KẾT QUẢ THỰC NGHIỆM

- **Đăng ký / đăng nhập:** Thành công, nhận JWT token.
- **Tạo / chỉnh sửa / xóa note:** Hoạt động trơn tru, cập nhật ListView ngay lập tức.
- **Offline:** Lưu SQLite khi mất mạng, đồng bộ khi kết nối trở lại.
- **Retrofit:** Logging interceptor giúp debug API.

## 6. ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

### 6.1. Ưu điểm

- Kiến trúc client–server rõ ràng.
- Giao diện đơn giản, dễ sử dụng.
- Token JWT bảo mật, dễ mở rộng.
- Hỗ trợ offline với SQLite.

### 6.2. Hạn chế

- Chưa hỗ trợ đa người dùng cùng lúc offline.
- Chưa có sync realtime (WebSocket).
- Giao diện còn cơ bản, chưa responsive cho tablet.

### 6.3. Hướng phát triển

- Thêm **push notification** khi note được cập nhật.
- Hỗ trợ **markdown**, **hình ảnh**, **audio**, **vẽ tay**.
- Đồng bộ realtime với **Firebase** hoặc **Socket.io**.
- Thiết kế giao diện đẹp hơn, hỗ trợ theme sáng/tối.

## 7. TÀI LIỆU THAM KHẢO

1. Android Developers. <https://developer.android.com>
2. Retrofit Documentation. <https://square.github.io/retrofit/>
3. Node.js Documentation. <https://nodejs.org>
4. Express.js Documentation. <https://expressjs.com>
5. SQLite Documentation. <https://www.sqlite.org/docs.html>
6. PlantUML. <https://plantuml.com>

