PROJECT REPORT

ON

Mobile Note Sharing Platform

FOR

(Name of the Company or Institute)

BY

(Your Full Name)

YASHWANTRAO CHAVAN MAHARASHTRA OPEN UNIVERSITY

(Your Course Name)

2024–2025

---

1. Introduction

1.1 Company/Institute/Client Profile

(Provide a brief overview of the institute or hypothetical client requesting the app—e.g., an educational institute or startup.)

1.2 Abstract

This project is a mobile application that enables students to upload, browse, and share academic notes. Inspired by Studocu.com, it aims to build a community-based platform to facilitate seamless knowledge sharing among students. It is built using Flutter for the frontend and Spring Boot for the backend.

1.3 Existing System and Need for System

Most institutions lack a centralized platform for students to share notes and materials. Students often rely on informal means like WhatsApp or email, which are inefficient and hard to access. Hence, a structured, scalable, and user-friendly platform is needed.

1.4 Scope of System

- Share, download, and rate notes
- User authentication & role management
- Admin panel for moderation
- Tags and categories for easy search
- Notifications
- Integration with Firebase for storage and Expo for frontend deployment

1.5 Operating Environment

- **Frontend:** Flutter (Dart), Expo
- **Backend:** Spring Boot (Java), REST API
- **Database:** MySQL
- **Cloud:** Firebase Storage
- **Tools:** IntelliJ IDEA, VS Code

1.6 Brief Description of Technology Used

- **OS:** Windows/Linux
- **RDBMS:** MySQL

- **Frameworks:** Spring Boot, Flutter
- **Dev Tools:** Postman, Firebase Console

---

2. Proposed System

2.1 Study of Similar Systems

- Studocu.com
- OneClass
  (Briefly discuss features of these platforms)

2.2 Feasibility Study

- **Technical Feasibility:** Open-source tools used, supported platforms
- **Operational Feasibility:** User-friendly UI, minimal learning curve
- **Economic Feasibility:** Uses mostly free and open-source technologies

2.3 Objectives of Proposed System

- Encourage knowledge sharing
- Maintain content quality through moderation and user feedback
- Enable easy access to study materials anytime, anywhere

2.4 Users of System

- Students

- Faculty
- Admin

---

3. Analysis and Design

3.1 System Requirements

- **Functional:** Upload, Download, Rate notes, Manage user roles, Comment system
- **Non-Functional:** Scalability, Security, Responsive UI, 99% Uptime

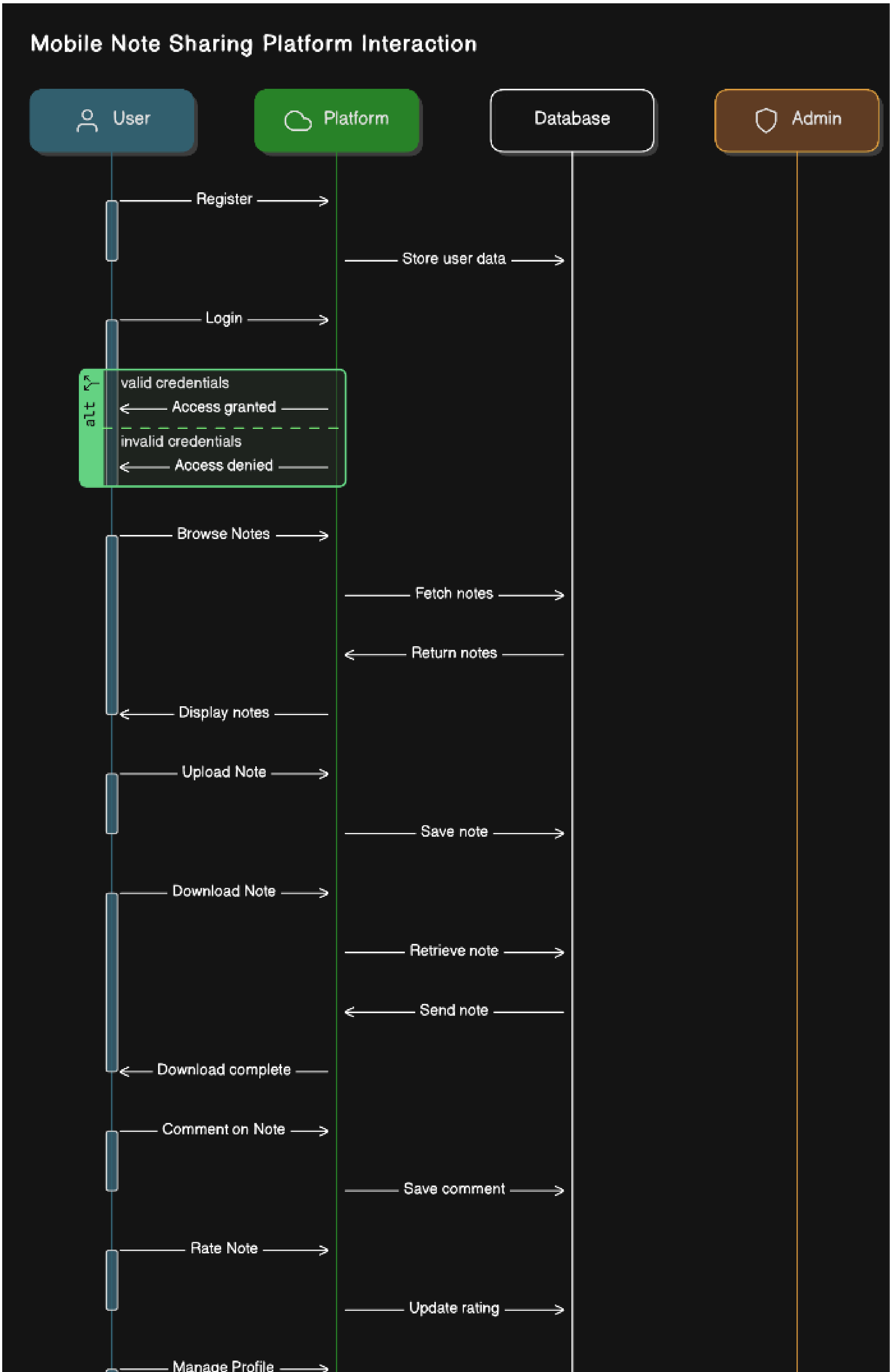
3.2 ER Diagram

(Include ERD with entities like User, Notes, Categories, Comments, etc.)

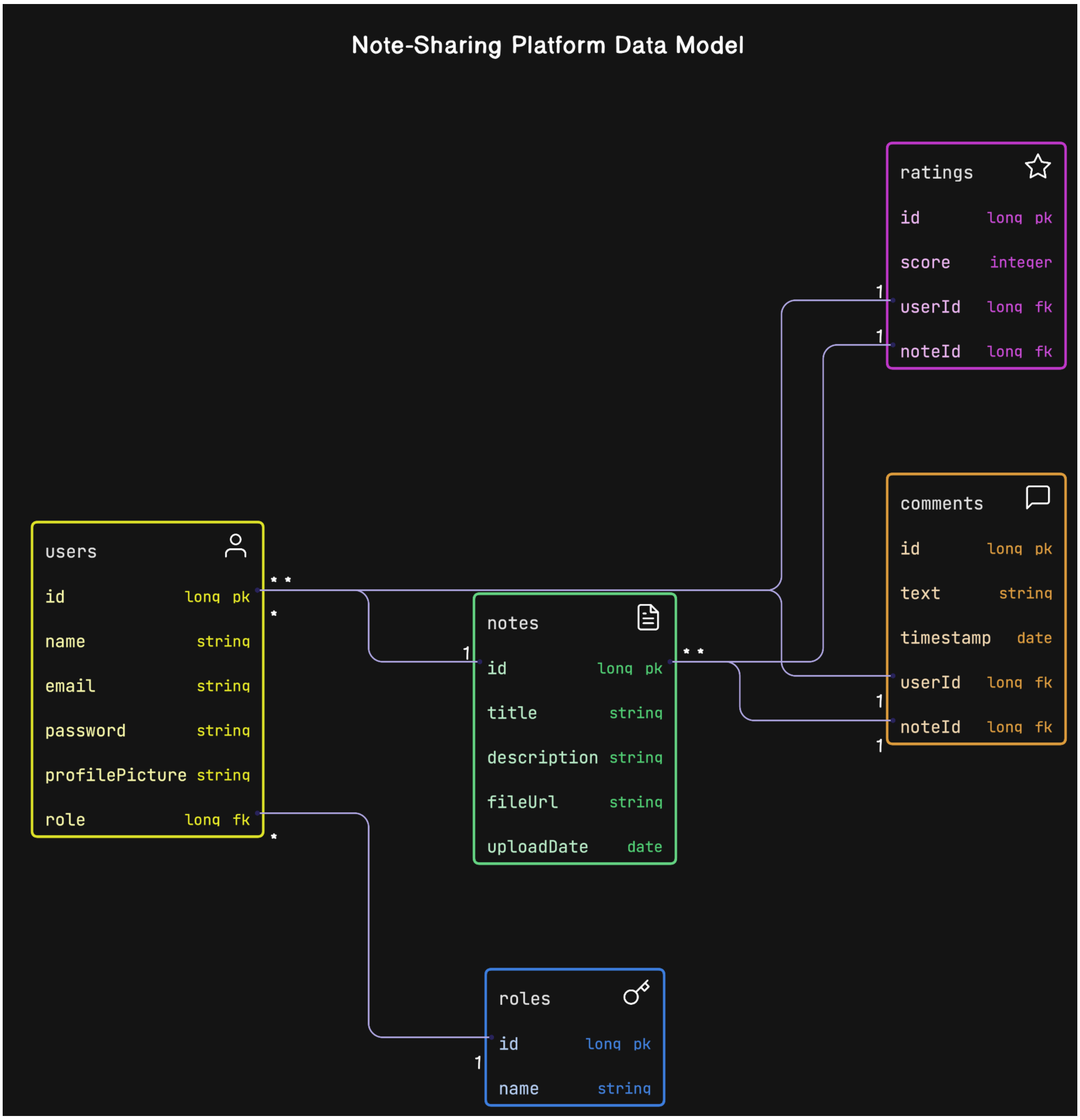
3.3 Table Structure

- users(id, name, email, password, role)
- notes(id, title, description, file_url, uploader_id, upload_date, rating)
- categories(id, name)
- comments(id, user_id, note_id, text, timestamp)

3.4 Use Case Diagrams


Mobile Note Sharing Platform Interaction
User
Platform
Database
Admin
Register
Store user data
Login
alt
valid credentials
Access granted
invalid credentials
Access denied
Browse Notes
Fetch notes
Return notes
Display notes
Upload Note
Save note
Download Note
Retrieve note
Send note
Download complete
Comment on Note
Save comment
Rate Note
Update rating
Manage Profile

3.5 Class Diagram



Note-Sharing Platform Data Model

ratings		
id	long	pk
score	integer	
userId	long	fk
noteId	long	fk

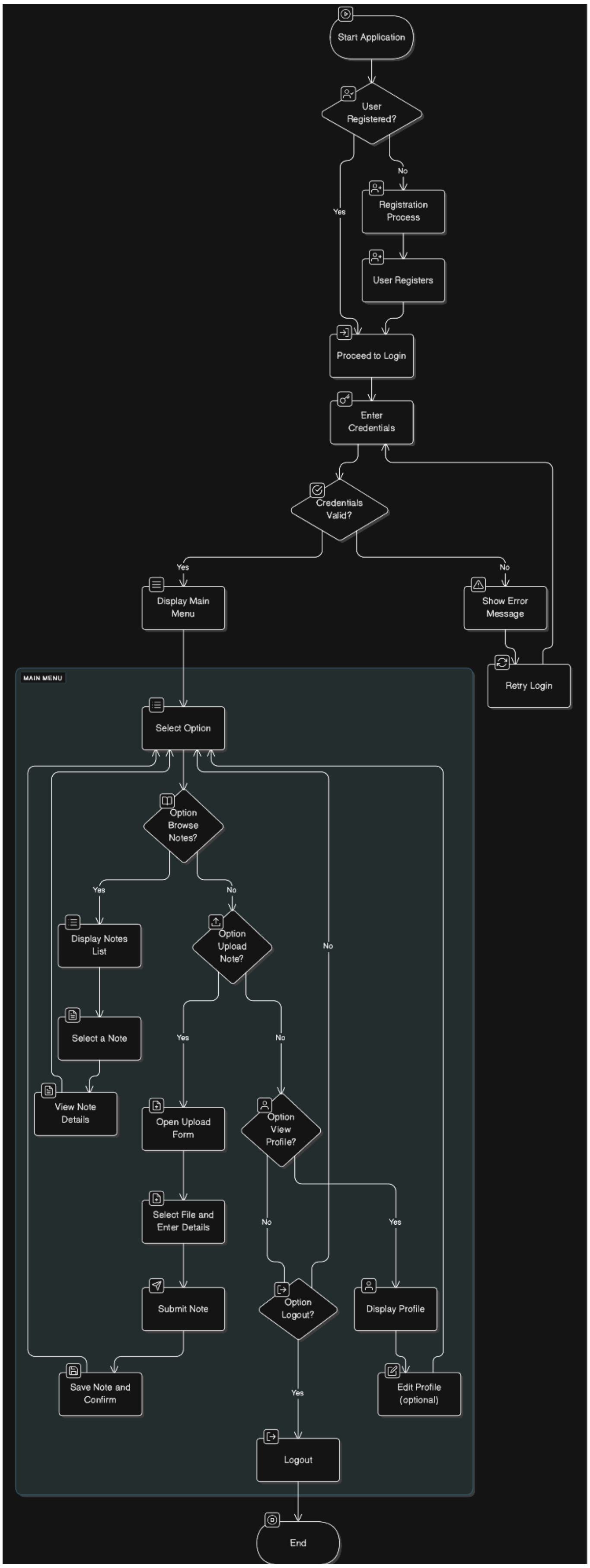
users		
id	long	pk
name	string	
email	string	
password	string	
profilePicture	string	
role	long	fk

notes		
id	long	pk
title	string	
description	string	
fileUrl	string	
uploadDate	date	

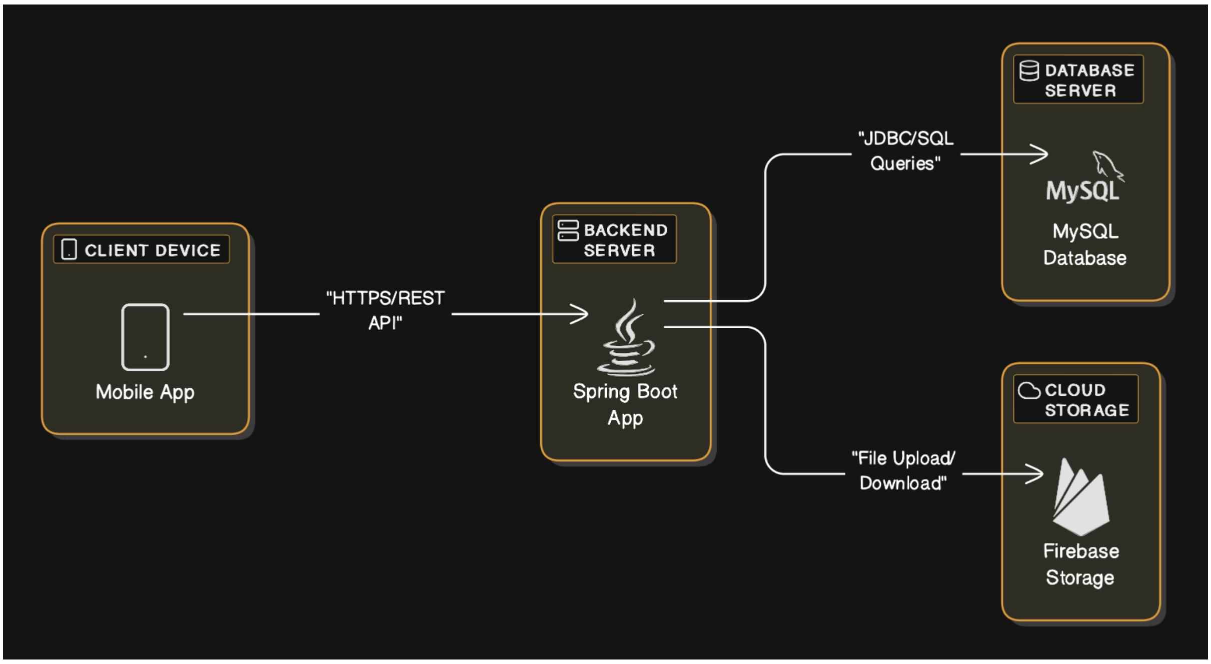
comments		
id	long	pk
text	string	
timestamp	date	
userId	long	fk
noteId	long	fk

roles		
id	long	pk
name	string	

3.6 Activity Diagram


Start Application
User Registered?
No
Yes
Registration Process
User Registers
Proceed to Login
Enter Credentials
Credentials Valid?
Yes
No
Display Main Menu
Show Error Message
Retry Login
MAIN MENU
Select Option
Option Browse Notes?
Yes
No
Display Notes List
Option Upload Note?
No
Select a Note
Yes
No
View Note Details
Open Upload Form
Option View Profile?
Select File and Enter Details
No
Yes
Submit Note
Option Logout?
Display Profile
Save Note and Confirm
Yes
Edit Profile (optional)
Logout
End

3.7 Deployment Diagram



3.8 Module Hierarchy Diagram

```
Mobile Note Sharing Platform
|
├── User Management
├── Notes Management
├── Comment & Rating
├── Admin Moderation
├── Notifications
└── Search & Filter
```

3.9 Sample Screens

- Register/Login
- Home Page (Recent Uploads)
- Note Detail Page
- Upload Page
- Admin Dashboard

4. Project Modules Overview

1. Authentication Module

- User Registration
- User Login
- JWT Token-based Authentication
- Role-based Access (User/Admin)

2. User Management

- View Profile
- Edit Profile
- Change Password
- Delete Account

3. Notes Management

- Upload Notes (PDF, DOC, etc.)

- Download Notes
- View Notes (title, description, preview)
- Note Metadata (date, tags, uploader info)

4. Search & Filter

- Keyword Search
- Filter by Subject/Course/Tags

5. Commenting & Ratings

- Comment on notes
- Like/Dislike system or star rating
- Report inappropriate notes/comments

6. Admin Panel

- Approve/Reject Notes
- Delete Notes or Comments
- Manage Users
- View Reports/Flags

7. Notifications

- In-app notifications (new notes, approvals)
- Optional: Push notifications (Firebase Cloud Messaging)

8. File Storage Integration

- Upload/Download notes to/from Firebase Storage or local server

9. Analytics Module (Optional)

- Most downloaded notes
- Top-rated contributors
- Recently uploaded notes

📂 Suggested File Structure

🔹 Spring Boot (Backend)

```
mobile-note-sharing-backend/
│
├── src/main/java/com/yourapp/
│   ├── MobileNoteSharingPlatformApplication.java
│   │
│   ├── config/
│   │   ├── SecurityConfig.java
│   │   └── JwtAuthenticationFilter.java
│   │
│   ├── controller/
│   │   ├── AuthController.java
│   │   ├── UserController.java
│   │   ├── NoteController.java
│   │   ├── CommentController.java
│   │   └── AdminController.java
│   │
│   ├── model/
│   │   ├── User.java
│   │   ├── Note.java
│   │   ├── Comment.java
│   │   ├── Role.java
│   │   └── Rating.java
│   │
│   ├── repository/
│   │   ├── UserRepository.java
│   │   ├── NoteRepository.java
│   │   ├── CommentRepository.java
│   │   └── RatingRepository.java
│   │
│   ├── service/
│   │   ├── AuthService.java
│   │   ├── UserService.java
│   │   ├── NoteService.java
│   │   └── AdminService.java
│   │
│   └── dto/
│       ├── UserDTO.java
│       ├── NoteDTO.java
│       ├── CommentDTO.java
│       └── AuthRequest/ResponseDTO.java
│
├── src/main/resources/
│   ├── application.properties
│   └── static/
│
└── Dockerfile / README.md / pom.xml
```

🔹 Flutter (Frontend)

arduino

CopyEdit

```
mobile_note_sharing_frontend/
│
├── lib/
│   ├── main.dart
│   │
│   ├── config/
│   │   └── api_constants.dart
│   │
│   ├── models/
│   │   ├── user_model.dart
│   │   ├── note_model.dart
│   │   ├── comment_model.dart
│   │
│   ├── services/
│   │   ├── auth_service.dart
│   │   ├── note_service.dart
│   │   ├── comment_service.dart
│   │
│   ├── screens/
│   │   ├── auth/
│   │   │   ├── login_screen.dart
│   │   │   └── register_screen.dart
│   │   ├── home/
│   │   │   ├── home_screen.dart
│   │   │   └── note_detail_screen.dart
│   │   ├── notes/
│   │   │   ├── upload_note_screen.dart
│   │   │   └── browse_notes_screen.dart
│   │   ├── profile/
│   │   │   ├── profile_screen.dart
│   │   │   └── edit_profile_screen.dart
│   │   ├── admin/
│   │   │   └── admin_dashboard_screen.dart
│   │
│   ├── widgets/
│   │   ├── custom_button.dart
│   │   ├── note_card.dart
│   │   └── comment_tile.dart
│   │
│   └── utils/
│       ├── validators.dart
│       └── theme.dart
│
├── assets/
│   └── images/
│
├── pubspec.yaml
└── README.md
```

---

5. Coding

5.1 Algorithms

- Note ranking based on user ratings
- Search filtering using tags & categories

5.2 Code Snippets

- JWT authentication (Spring Boot)
- File upload to Firebase (Flutter)
- REST endpoints

---

6. Testing

6.1 Test Strategy

- Manual + Automated testing (Unit + Integration)

6.2 Unit Test Plan

- Spring Boot JUnit tests for service logic
- Flutter widget tests for UI components

6.3 Acceptance Test Plan

- User can register, upload, and view notes
- Admin can delete inappropriate content

6.4 Test Cases/Test Scripts

- Test login with invalid credentials
- Upload note with missing metadata

6.5 Defect Report/Test Log

- Maintain a defect log sheet

---

7. Limitations of Proposed System

- No OCR or text extraction from handwritten notes
- Internet-dependent
- No live chat feature (yet)

---

8. Proposed Enhancements

- AI-based recommendation system
- Text extraction and search within documents
- In-app messaging between users

---

9. Conclusion:

The Mobile Note Sharing Platform enables a digital, collaborative learning environment that addresses the Inefficiencies of traditional note-sharing methods. With scalable architecture and intuitive design, it serves as a modern-day academic aid.

---

10. Bibliography

- https://studocu.com
- https://spring.io
- https://flutter.dev
- https://firebase.google.com

---

11. Publication/Competition Certificates

(Add if any)

12. Appendix – Cost Sheet, Datasheet

- Firebase: Free tier
- Hosting: Localhost for dev, Docker setup for production

12. User Manual

- **Login/Register:** Secure sign-in via email/password
- **Upload Notes:** Choose file, add title & category
- **Browse Notes:** Filter by course/category
- **Rate & Comment:** Provide feedback to help others
- **Admin Actions:** Delete/Flag inappropriate content