Android App Development from ER Diagram with Free Cloud Database

Pre-Final and Final Exam Instructions

Part 1: Choose Your Tech Stack

- **Option A: React Native**
  - JavaScript/TypeScript-based
  - Write once, deploy on both Android and iOS
  - Uses React components with native rendering
- **Option B: Flutter**
  - Dart programming language
  - Rich widget library
  - Excellent performance and UI capabilities

Part 2: From ER Diagram to Cloud-Based Mobile App

1. **Translate your ER Diagram to a MySQL Database Schema**

   - Convert entities to tables
   - Map relationships to foreign keys
   - Define data types for all attributes
2. **Set Up Free Cloud MySQL Database**

   - Options:
     - **InfinityFree** (free MySQL hosting with phpMyAdmin)
     - **PlanetScale** (free tier with limitations)
     - **Railway** (free tier with limitations)
     - **Clever Cloud** (free tier MySQL)
     - **GearHost** (free tier with limitations)
     - **AWS RDS** (free tier for 12 months)
   - Create database instance using chosen provider
   - Set up security settings (firewall, user access)
   - Execute your schema creation SQL statements
3. **Develop RESTful API**

   - Choose API framework:
     - Node.js + Express
     - PHP with Slim/Laravel (works well with InfinityFree)
     - Python with Flask/FastAPI
   - Deploy API to free hosting services:
     - **InfinityFree** (includes free PHP hosting with MySQL)
     - **Render** (free tier)
     - **Heroku** (free tier with limitations)
     - **Vercel** (free tier)
     - **Glitch** (free tier)
   - Create CRUD endpoints for each entity in your ERD
   - Implement authentication and proper error handling
4. **Set Up Your Development Environment**

   - Install necessary SDKs, emulators, and code editors
   - Create a new mobile app project with your chosen framework
5. **Create Data Models in Mobile App**

   - Create classes/models representing each entity in your ERD

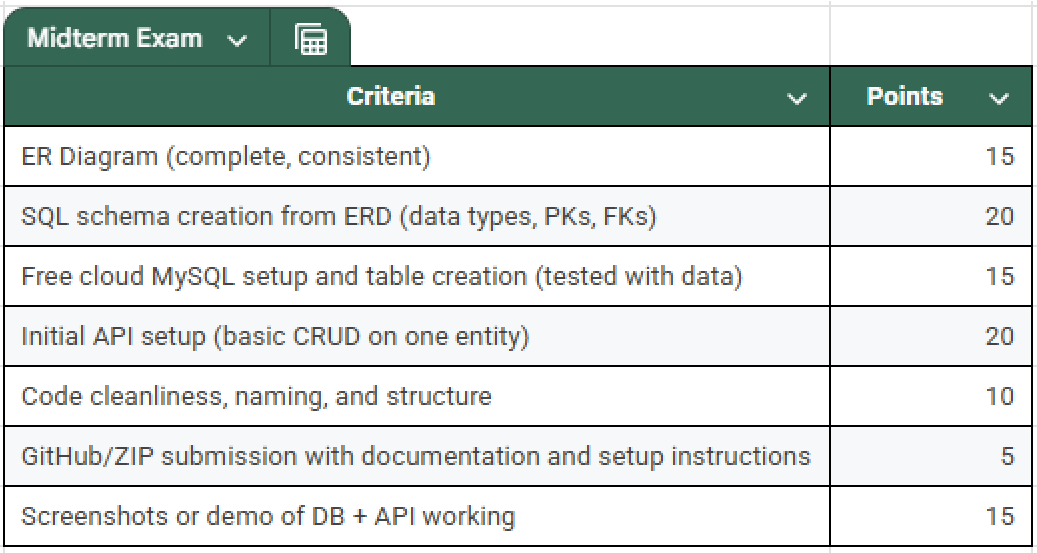
   - Implement API service to connect with your backend
6. **Develop Core Screens**
   - Create at least one CRUD screen for each main entity
   - Implement navigation between screens
   - Design UI components following Material Design guidelines
7. **Connect UI to API Layer**
   - Implement HTTP client to communicate with your API
   - Create service methods to call your API endpoints
   - Handle loading states and error scenarios
8. **Add Basic Authentication**
   - Implement user login/registration screens
   - Store and manage authentication tokens
   - Add authorization controls for protected data
9. **Polish and Test**
   - Test on multiple screen sizes
   - Debug and fix any issues
   - Generate a signed APK or AAB file

Evaluation Criteria Analytic Rubric

Midterm Exam – Database Design & Initial Backend Setup (100 Points)

Tasks:

1. Submit a complete and well-labeled ER Diagram.
2. Convert ERD to SQL schema.
3. Set up a free cloud MySQL database (e.g., InfinityFree, PlanetScale).
4. Create tables using your schema.
5. Start API development with at least one entity implemented.
6. Use GitHub or submit as a ZIP with proper documentation.

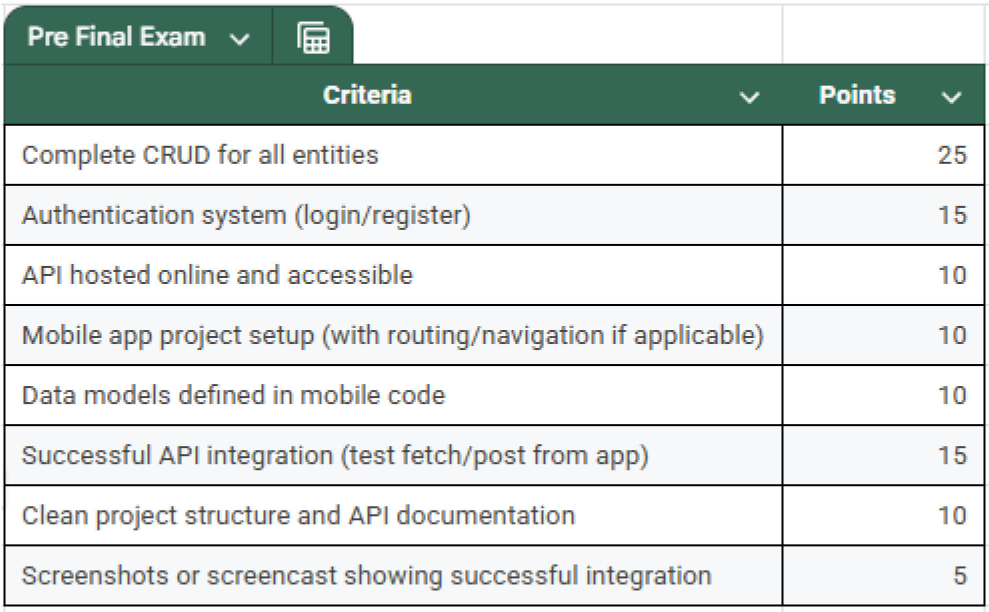

Midterm Exam

Criteria	Points
ER Diagram (complete, consistent)	15
SQL schema creation from ERD (data types, PKs, FKs)	20
Free cloud MySQL setup and table creation (tested with data)	15
Initial API setup (basic CRUD on one entity)	20
Code cleanliness, naming, and structure	10
GitHub/ZIP submission with documentation and setup instructions	5
Screenshots or demo of DB + API working	15

Pre-Final Exam – Full API Development & Mobile App Bootstrapping (100 Points)

Tasks:

1. Develop CRUD endpoints for all entities in your ERD.
2. Implement user authentication (register/login).
3. Deploy API using services like Render, Heroku, or InfinityFree.
4. Set up a mobile app project (React Native or Flutter).
5. Create data models and connect mobile app to the API.

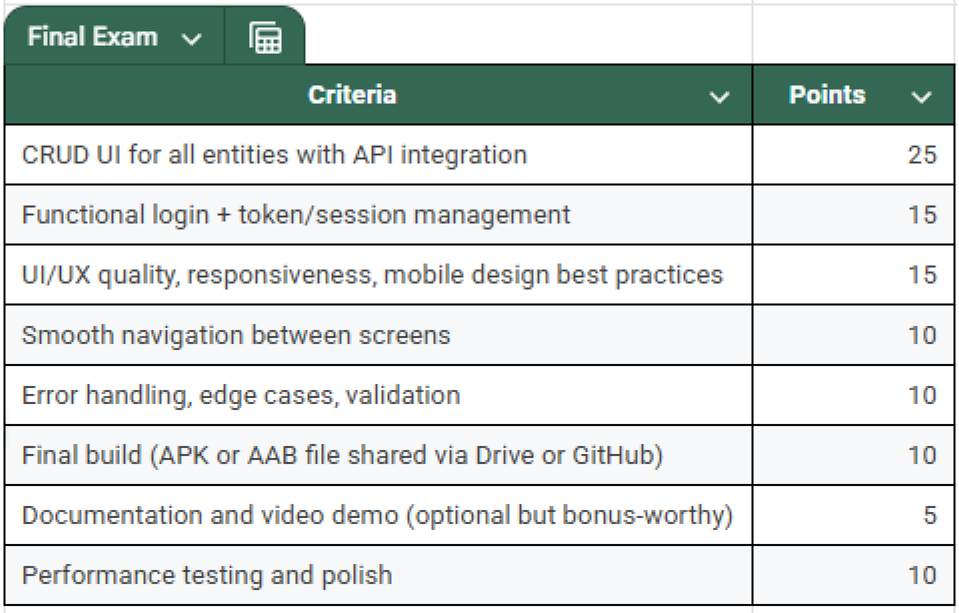

Pre Final Exam

Criteria	Points
Complete CRUD for all entities	25
Authentication system (login/register)	15
API hosted online and accessible	10
Mobile app project setup (with routing/navigation if applicable)	10
Data models defined in mobile code	10
Successful API integration (test fetch/post from app)	15
Clean project structure and API documentation	10
Screenshots or screencast showing successful integration	5

<u>Final Exam – Complete Mobile App Implementation & Polish (100 Points)</u>

Tasks:

1. Implement CRUD screens for all major entities.
2. Integrate API with full data flow.
3. Apply Material Design (React Native Paper or Flutter Widgets).
4. Add responsive layouts and navigation.
5. Finalize login/session system and role-based access if applicable.
6. Generate and submit APK or AAB.



Criteria	Points
CRUD UI for all entities with API integration	25
Functional login + token/session management	15
UI/UX quality, responsiveness, mobile design best practices	15
Smooth navigation between screens	10
Error handling, edge cases, validation	10
Final build (APK or AAB file shared via Drive or GitHub)	10
Documentation and video demo (optional but bonus-worthy)	5
Performance testing and polish	10