Mobile Application Design Lab Project – Guideline & Instruction

(Complex Engineering Problem)

Department of CSE

Course: Mobile Application Design Lab
Project Type: Project (Group-based)
Total Marks: 40

Targeted Attributes for Mobile Application Design lab - Complex Engineering Problem:

Attribute	Description
EP1	**Depth of Knowledge Required** - Requires in-depth knowledge of mobile app design principles and tools to implement solutions effectively.
EP2	**Range of Conflicting Requirements** - Requires balancing multiple constraints such as performance, user experience, and cross-platform compatibility.
EP3	**Depth of analysis Required** - Requires to analyze and resolve complex issues involving design patterns and state management.
EP5	**Extend of Applicable Code -** Requires to integrate APIs, DB, modern frameworks.
EP6	**Extend of Stakeholders involvements and Conflicting Requirements -** Requires to target real users & diverse requirements.

Targeted Attributes for Mobile Application Design lab - Complex Engineering Activities:

Attribute	Description
EA1	**Range of Resources-** Involves a wide range of resources including IDEs, emulators, APIs, Firebase, plugins etc.

Instructions to All Student Groups:

Project Selection

Your mobile app must:

- Address a real-world problem
- Include modern UI/UX design principles
- Use **one** of the taught frameworks: Flutter / React Native / Java / Kotlin

- Include **state management** (Provider/Redux/GetX/Bloc etc.)
- Integrate **local database** (SQLite / Hive) or **API**
- Include **minimum 5–7** interactive screens
- Deployable on Emulator / Physical Device

Note: Simple CRUD apps **without responsive design & architecture** are not allowed.

Suggested domains:

- Smart Campus App (DIU: attendance, clubs, events)
- Doctor Appointment, Telemedicine
- Online Food Delivery / E-commerce
- Travel & Tourism Guide with Maps/GPS
- Fitness & Habit Tracker (Notification, Charts)
- Smart Inventory/Billing for Small Business
- Digital Payments (Mock + local DB)

Note: You may propose any unique real-world app.

2. Project Requirements Based on EP Mapping

Your project must address the following **Engineering Principles (EPs):**

Attribute	Requirement in Project
EP1 – Depth of Knowledge Required	The project should require advanced knowledge of mobile application design and development concepts, going beyond simple tutorial-based apps. For example: complex UI/UX design, adaptive layouts, design patterns, state management, background tasks, performance optimization, and integration of device features (GPS, notifications, etc.).
EP2 – Range of Conflicting Requirements	Identify and resolve multiple constraints and trade-offs that arise in real-world applications. Possible conflicts may include: performance vs. battery usage, UI attractiveness vs. development complexity, data security vs. usability, offline capability vs. storage cost, or cross-platform compatibility vs. hardware optimization. Students must describe how these trade-offs were analyzed and addressed.

EP3 – Depth of Analysis Required	The project must involve analytical decision-making for system architecture and development strategies. Students should analyze and select appropriate design patterns, navigation strategies, and state management techniques (e.g., Provider, Bloc, Redux). They must justify their choices based on scalability, maintainability, and performance improvements.
EP5 – Extent of Applicable Codes	The project must incorporate advanced and industry-standard implementation elements. Examples include external API integration, local database (SQLite/Hive), authentication, notifications, background processing, and device sensors (GPS, Camera). The system should demonstrate applicability in real-world deployment scenarios.
EP6 – Extent of Stakeholder Involvement & Conflicting Needs	The project should consider the needs of multiple user groups or stakeholders and handle conflicting requirements among them. Students must conduct requirement analysis (survey/interview) and document how user feedback influenced the design. Accessibility, usability, and personalization should be addressed to ensure the solution fits real users and diverse environments.

Deliverables must clearly indicate how each EP is addressed.

3. Required Documents and Files

Students must submit the following:

1. **Project Report (PDF)**
    - Cover Page — Title, Members & IDs, Section, Semester, Instructor
    - Abstract (120–200 words)
    - Problem Statement & Project Scope
    - Targeted Users (EP6)
    - UI/UX Design
    - Wireframes / Mockups
    - Accessibility consideration
    - System Architecture
    - Navigation Chart
    - State Management Approach
    - Feature List & Screenshots (min 5–7 screens)
    - Data Management
    - API endpoints / SQL structure / Hive Boxes
    - Performance Optimizations

- Conflicting Requirements Handling (EP2)
- Stakeholder Interaction Summary (EP6)
- Challenges & Solutions (EP1, EP3)
- References

2. **Source Code Folder**
   - Full project + assets
   - README with installation steps

3. **Presentation Slides (Optional, if required by instructor)**
   - Summarize project objective, design, implementation, and outcomes.

4. Evaluation Rubric (Total: 40 Marks)

Attribute	Requirement in Project
EP1 – Depth of Knowledge Required	Demonstrate advanced mobile app development knowledge; responsive UI/UX, performance optimization, animations, state management, and structured architecture.
EP2 – Range of Conflicting Requirements	Identify and resolve multiple constraints: performance vs. battery usage, usability vs. security, storage vs. offline capability. Explain trade-offs made.
EP3 – Depth of Analysis Required	Analyze navigation flow, design patterns, and select appropriate state management techniques (Provider/BLoC/Redux/GetX) with justification.
EP4 – Familiarity with Project Context	Consider real-world context: target user needs, environment of use, accessibility, and scalability. Provide justification for design choices.
EP5 – Extent of Applicable Codes	Include diverse real-world features: API/local DB integration, authentication, device sensors (GPS/Camera), notifications, background services.

EP6 – Extent of Stakeholder Involvement & Conflicting Needs	Incorporate stakeholder/user feedback during requirement analysis; ensure usability for multiple user groups and document how feedback influenced design updates.

Total: 40 Marks

5. Submission Instructions

- **Deadline:** 12-12-2025
- **Mode of Submission:** Upload to BLC or submit via email/in-person as instructed.
- **Files Required:**
    - Project Report (PDF)
    - Source Code
    - Optional: APK file

Late submissions will incur penalties as per course policy.

Appendix

How can we address EP1, EP2, EP3, EP5 and EP46 through the Mobile Application Design lab's Mini Project?

EP1 – Apply Knowledge of Mobile Development Fundamentals

Goal: Apply fundamental and advanced concepts of mobile application design and development in a practical solution.

How to address:

- Use appropriate UI components following Material Design / HIG guidelines.
- Build **responsive** and **adaptive** layouts for multiple screen sizes.
- Apply **stateful and stateless widget** concepts (for Flutter) or **Activity/Fragment** (Android native).
- Implement functional features using mobile framework capabilities such as forms, local storage, and navigation.

Example:
Create a health tracking app that uses responsive dashboards, validates user input, and stores daily activities using local storage.

EP2 – Identify, Formulate & Analyze Complex Problems

Goal: Recognize real-world issues users face and break them into solvable app features.

How to address:

- Clearly define the **problem statement** and **user personas** in the report.
- Identify core workflows such as registration, appointment booking, checkout, etc.
- Analyze potential constraints: performance, device compatibility, offline usage.
- Architect the app features logically using **navigation maps**, **use case diagrams**, or **flow charts**.

Example:
For an e-commerce app, analyze the flow from browsing items → cart → checkout → order tracking and design screens accordingly.

EP3 – Conduct Analysis for UI/UX & State Management Decisions

Goal: Demonstrate design thinking and analytical choices behind UI architecture and state handling.

How to address:

- Compare state management options such as Provider, BLoC, Redux, or GetX.
- Justify why the chosen approach improves scalability or performance.
- Use navigation strategies like **Named Routes**, **Bottom Navigation**, or **Deep Linking**.
- Analyze widget rebuilds and optimize where necessary.

Example:
Choose Provider instead of setState due to cleaner state separation and reduced widget rebuild cost in a multi-screen news application.

EP5 – Extend the Application with Real-World Functionality

Goal: Build a system that uses modern mobile technologies beyond local UI features.

How to address:

- Integrate APIs for real-time data (REST/GraphQL/Firebase)
- Use device hardware: Camera, Location/GPS, Push Notifications, Sensors.
- Handle **online & offline modes** using local caching.
- Implement asynchronous programming for network calls.

Example:
A tourism app that uses Google Maps API for location-based recommendations, Firebase for authentication, and caching for offline access.

EP6 – Stakeholder Involvement & Requirement Satisfaction

Goal: Build apps that consider multiple types of users and usability expectations.

How to address:

- Conduct short user interviews or surveys (teachers, students, shop owners, etc.)
- Document their needs & refine the UI/UX accordingly.
- Add accessibility features: Larger text, Screen reader support, High-contrast mode.
- Include a **Client/User Interaction Summary** in the report.

Example:
For a DIU campus app, take input from students about the features they actually need (bus schedule, class routine, exam seat plan, etc.) and redesign screens based on feedback.