Mobile Application Development Final Exam Project Plan Fall 2025

Period of submission: 3 Weeks

Project Overview

This project serves as a comprehensive assessment of Flutter development skills, replacing traditional written examinations. Students will develop a fully functional mobile application that demonstrates ability in Flutter development practices, state management, backend integration, and UI/UX design principles.

Week 1: UI Foundation & Data Modeling

Learning Objectives

- Implement Material Design 3 principles
- Create a cohesive design system
- Design robust data models
- Apply software engineering best practices

Tasks & Deliverables

1.1 Design System Creation

- Create a custom theme with light and dark modes
- Implement consistent color palette using Material 3 color system
- Design custom typography using Google Fonts
- Create a component library with at least 5 reusable widgets

1.2 Character Model Implementation

- Create a comprehensive Character class with properties:
    - Basic info: name, description, image
    - Metadata: creation date, last modified
    - Attributes: skills, stats
- Implement JSON serialization/deserialization

1.3 AI-Assisted Implementation

- Use AI tools to generate color palette suggestions
- Generate initial Character class with AI
- Document AI prompts used and rationale for selected elements

Evaluation Criteria (30% of final grade)

- Implementation of Material Design 3 principles (10%)

- Quality and reusability of custom widgets (10%)
- Proper implementation of data models (10%)

Week 2: State Management & Backend Integration

- Implement appropriate state management solution
- Connect Flutter application to Firebase
- Implement real-time data synchronization
- Handle authentication and authorization

Tasks & Deliverables

2.1 State Management Implementation

- Choose between Provider or Riverpod (simplified from original options)
- Implement state management for:
    - Character list
    - User preferences (theme)
    - Authentication state

2.2 Firebase Integration

- Set up Firebase project with Firestore
- Implement authentication (email/password)
- Configure Firestore database with proper data structure
- Set up Firebase Storage for character images

2.3 CRUD Operations

- Implement all CRUD operations for Character data
- Add real-time updates using streams
- Handle network errors and offline scenarios

2.4 AI-Assisted Implementation

- Use AI to generate Firestore security rules
- Create repository classes with AI assistance
- Document AI-assisted backend implementation

Evaluation Criteria (40% of final grade)

- Correct implementation of state management (15%)
- Successful Firebase integration (15%)

- Implementation of all required features (10%)

Week 3: Advanced Features & Final Presentation

Learning Objectives

- Implement advanced Flutter features
- Optimize application performance
- Create comprehensive technical documentation
- Develop effective presentation skills

Tasks & Deliverables

3.1 Feature Implementation (Choose 2 of the following)

- Search and filtering functionality
- Character export/import functionality
- Push notifications for character updates
- Character sharing capabilities

3.2 Performance Optimization

- Implement lazy loading
- Optimize image loading and caching
- Reduce app bundle size

3.3 Documentation & Presentation

- README file with setup instructions
- 5-minute video demonstration of the application
- Slides explaining technical decisions
- Reflection on AI usage throughout the project
- AI Integration Log documenting all AI usage

Evaluation Criteria (30% of final grade)

- Implementation of advanced features (15%)
- Quality of documentation and presentation (15%)

Additional Requirements

AI Usage Documentation

Students must maintain an "AI Integration Log" documenting:

- AI tools used (ChatGPT, GitHub Copilot, etc.)
- Prompts used and their effectiveness
- How AI-generated code was modified
- Challenges encountered with AI assistance
- Ethical considerations of AI usage

Submission Requirements

1. Source code repository with Git history
2. Technical documentation
3. AI Integration Log

Evaluation

CATEGORY	EXCELLENT (90-100%)	GOOD (80-89%)	SATISFACTORY (70-79%)	NEEDS IMPROVEMENT (60-69%)
UI/UX DESIGN	Exceptional design with consistent theming	Good design with consistent theming	Adequate design with some inconsistencies	Poor design with multiple inconsistencies
CODE QUALITY	Clean, well-structured code with documentation	Good code structure with adequate documentation	Functional code with minimal documentation	Disorganized code with poor documentation
STATE MANAGEMENT	Sophisticated implementation with optimal performance	Proper implementation with good performance	Basic implementation that meets requirements	Incomplete or problematic implementation
BACKEND INTEGRATION	Seamless integration with robust error handling	Functional integration with adequate error handling	Basic integration with minimal error handling	Incomplete integration with significant issues
ADVANCED FEATURES	Well-implemented features that add value	Basic implementation of required features	Partial implementation of features	Missing or poorly implemented features
AI INTEGRATION	Thoughtful use of AI with clear documentation	Appropriate use of AI with adequate documentation	Basic use of AI with minimal documentation	Minimal or inappropriate use of AI

DOCUMENTATION & PRESENTATION	Comprehensive documentation and polished presentation	Good documentation and clear presentation	Adequate documentation and basic presentation	Incomplete documentation and poor presentation

---

Timeline & Milestones

WEEK	MILESTONE	DELIVERABLES
1	UI Foundation & Data Modeling	Theme implementation, component library, Character class
2	State Management & Backend Integration	State management setup, Firebase connection, CRUD operations
3	Advanced Features & Final Presentation	2 advanced features, documentation, presentation materials

---

Academic Integrity Policy

While AI tools are encouraged for this project, students must:

1. Clearly document all AI-generated code
2. Ensure they understand and can explain all code in their project
3. Not use AI to generate entire solutions without understanding
4. Cite AI usage appropriately in their documentation
5. Not use AI to plagiarize or create misleading content

Any violation of these guidelines will be treated according to the institution's academic integrity policy.