Project: Finance Buddy

Plan Introduction

This Software Development Plan (SDP) outlines the process, resources, and schedule for developing Finance Buddy, an AI powered personal finance mobile app designed to help users track spending, set savings goals, and learn financial literacy through gamified challenges and personalized insights.

The project will be developed using Agile methodology, with iterative sprints focusing on incremental feature delivery. Each sprint will include planning, development, testing, and reflection. The goal is to produce a fully functional Minimum Viable Product (MVP) by the end of the semester.

Key Milestones

Milestone	Description	Target Date
SRS (Requirements)	Finalize user stories, functional/non-functional requirements	Week 5
SDP	Define schedule, resources, and organization	Week 8
Prototype Test	Implement core UI and dashboard functionality	Week 11
Beta Testing	Integration of AI categorization + goal tracking	Week 13

Final Presentation	Complete MVP with all core modules	Week 15

Project Deliverables

Software Requirements Specification (SRS)
A detailed document defining all functional and non-functional requirements, user stories, and acceptance criteria for Finance Buddy.

Software Development Plan (SDP)
The current document, describing the project organization, resources, schedule, and timeline.

Prototype Implementation
A preliminary demo including the login screen, dashboard layout, and dummy financial data visualization.

Beta Version (Integration Phase)
Integration of Firebase authentication, expense tracking, and goal tracking modules.

Final MVP Submission
Fully functional Finance Buddy application with AI-powered categorization, gamified lessons, and notifications.

Presentation & Demo Video
Recorded presentation demonstrating app functionality, development process, and reflection on results.

Project Resources

Hardware Resources

Hardware	Purpose	Specifications
MacBook Air (M4, 2024)	Primary development machine	macOS Sonoma

iPhone 17	Testing device	iOS 18
Cloud Firestore Server	Backend data storage	Firebase cloud infrastructure

Software Resources

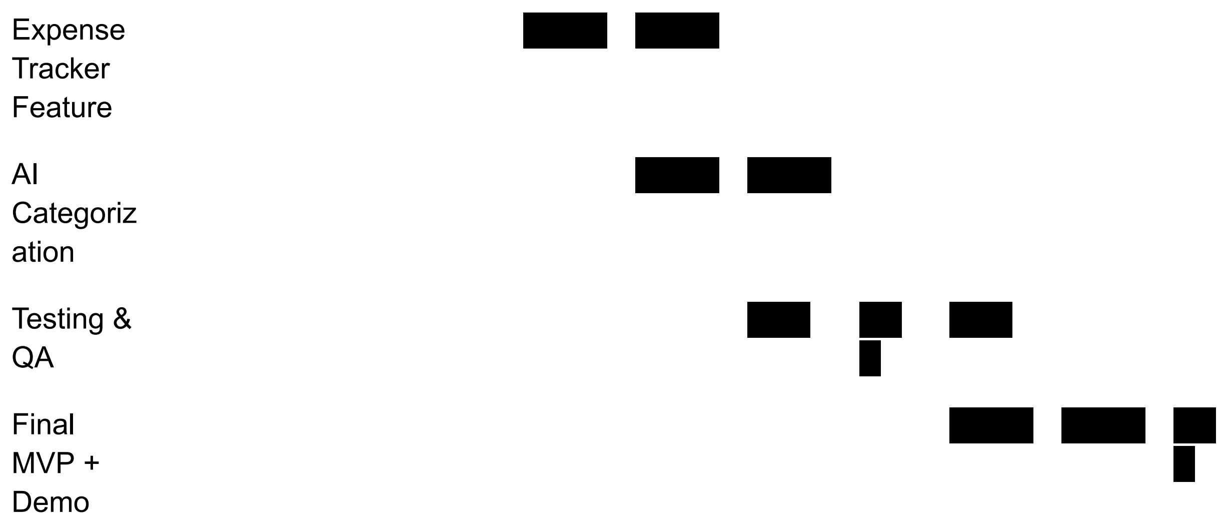
Software	Version	Purpose
Xcode	16.0	Swift UI mobile app development
Firebase		Authentication, Firestore, Hosting, Storage
GitHub		Version control and collaboration
Canva / Figma		UI/UX design and prototyping
Python + OpenAI CLIP	3	AI categorization and analytics module
Google Sheets		Schedule and budget tracking
Notion		Sprint planning and Agile tracking

Project Organization

Team Member	Role	Responsibilities
A’Kaia Phelps	Project Manager	Full-stack development (frontend + Firebase backend), AI integration, sprint management
Natasha Cordova-Diba	UI/UX Designer	App layout design, component styling, prototyping
Hannah Holden	Backend Developer	Database schema, authentication logic, API handling

Schedule

Task	Week 6	7	8	9	10	11	12	13	14	15
Requirements (SRS)	█									
Software Development Plan		█								
UI Design		█	█							
Firebase Setup			█	█						



Critical Path: Firebase Setup → Expense Tracker → AI Categorization → Testing → Final Demo

Task/Resource Table

Task	Assigned Developer	Hardware	Software
UI/UX Design	Natasha	MacBook	Figma, Xcode
Firebase Integration	Hannah	MacBook	Firebase, Xcode
Authentication System	Hannah	MacBook	Firebase Auth
Expense Tracker	Akaia	MacBook	SwiftUI, Firestore
AI Categorization	Natasha	MacBook	Python, CLIP, Firebase
Testing & QA	Akaia	iPhone	Xcode Test, Firebase
Presentation	Hannah, Akaia, Natasha	Laptop	PowerPoint, Canva

Finance Buddy – Requirements Specification (Agile)

1. Introduction

Purpose:

Finance Buddy is an innovative mobile application designed to empower users on their journey toward financial literacy and stability. Unlike traditional, static budgeting tools, Finance Buddy leverages cutting-edge artificial intelligence (AI) and gamification techniques to deliver a uniquely engaging and adaptive financial experience. Its core purpose is to provide personalized budgeting, robust goal tracking, and an interactive learning platform that makes understanding and managing personal finances accessible and enjoyable for everyone.

Scope:

The development of Finance Buddy will proceed incrementally through Agile sprints, each lasting 2–3 weeks, ensuring continuous delivery of value and flexibility to adapt to evolving requirements.

- **Minimum Viable Product (MVP):** The initial release will focus on foundational features critical for establishing a core user base and demonstrating the app's value proposition. This includes a smart, intuitive dashboard, comprehensive goal-setting capabilities, timely reminders to keep users on track, and an AI-powered system for intelligent budget categorization.
- **Future Sprints:** Subsequent development phases will build upon the MVP, introducing more advanced and engaging features. This will include the expansion of gamified learning modules, transforming financial education into an interactive and rewarding experience, and the integration of social competition elements to foster a supportive and motivating community among users.
- **Incremental Delivery:** Adhering to Agile principles, the application will be delivered in small, functional increments, allowing for regular feedback incorporation and rapid iteration.

2. Overall Description

Product Perspective:

Finance Buddy is envisioned as a mobile-first application, meticulously designed for the iOS ecosystem. Its architecture will be robust, utilizing a Firebase backend for scalable and secure data management.

- **Mobile-First Design:** The primary focus is on delivering a seamless and intuitive user experience on iOS devices, ensuring accessibility and convenience for users on the go.
- **Data Flexibility:** The application will support both real financial data, integrated through bank statements or CSV uploads, and simulated transactions, providing a safe

environment for users to practice financial management without risk and enabling thorough testing during development.
- **RAG-Powered AI Module:** A key differentiator, the Retrieval-Augmented Generation (RAG) powered AI module will deliver highly personalized insights, offering tailored advice and recommendations based on individual user spending patterns and financial goals.

User Classes:

Finance Buddy is designed to cater to a diverse range of users, each with unique financial needs and goals:

- **Students:** This demographic often seeks to save money for specific objectives, such as travel or educational expenses, and requires tools to manage small, often fluctuating budgets effectively.
- **Young Professionals:** As they progress in their careers, young professionals typically focus on larger financial goals, such as planning for significant purchases like cars or housing, and require more sophisticated tools for long-term planning.
- **General Users:** This broad category encompasses individuals interested in improving their overall financial literacy, seeking long-term financial planning strategies, and striving for greater financial independence.

Constraints:

Several constraints will guide the development process, ensuring project feasibility and success:

- **MVP Timeframe:** The Minimum Viable Product must be fully functional and ready for deployment within a single academic semester, emphasizing efficient development and focused feature prioritization.
- **Security:** Given the sensitive nature of financial data, security is paramount. Rigorous measures will be implemented to protect user information from unauthorized access, breaches, and other cyber threats.
- **Bank API Integration Limitations:** Acknowledging potential limitations or complexities in integrating with various bank APIs, a robust fallback mechanism will be provided, allowing users to upload CSV files or manually enter transactions, ensuring continued functionality regardless of API availability.

3. Functional Requirements (User Stories)

FR1. User Authentication
As a user, I want to create an account and securely log in so that my data is private and protected.
Acceptance Criteria: Users can sign up, log in, and reset passwords via Firebase Authentication.

FR2. Smart Dashboard
As a user, I want to see a clear summary of my spending so that I can identify patterns.
Acceptance Criteria: Dashboard displays categorized data in chart format; refresh ≤ 3 seconds.

FR3. Goal Setting
As a user, I want to set personalized savings goals so that I can track progress toward objectives.
Acceptance Criteria: Users can create, edit, and delete goals, visualize progress, and receive reminders.

FR4. AI Categorization
As a user, I want my expenses automatically categorized so that I save time.
Acceptance Criteria: Uploaded transactions are categorized by AI; accuracy verified in organized views.

FR5. Gamified Learning
As a user, I want to earn rewards for completing lessons so that I stay motivated.
Acceptance Criteria: Lessons unlock progressively; points tracked and shown on a leaderboard.

FR6. Notifications
As a user, I want to receive reminders for goals and lessons so that I stay on track.
Acceptance Criteria: Push notifications sent for milestones, deadlines, and new lesson availability.

Account & User Management

FR7. Profile Management
As a user, I want to edit my profile information so that I can keep my account details up to date.
Acceptance Criteria: Users can update their name, email, and profile picture; changes reflect immediately.

FR8. Multi-Factor Authentication
As a user, I want optional two-factor authentication so that my account is more secure.
Acceptance Criteria: Users receive a one-time code via email or SMS when logging in if MFA is enabled.

FR9. Account Deletion
As a user, I want to permanently delete my account so that my personal data is removed from the system.
Acceptance Criteria: The app must provide a delete option; all user data is removed within 24 hours.

Expense Tracking

FR10. Manual Expense Entry
As a user, I want to manually add an expense so that I can record cash or missing transactions.
Acceptance Criteria: Users can input amount, category, date, and description.

FR11. Recurring Expenses
As a user, I want to set recurring expenses so that I don't have to enter repeating charges each month.
Acceptance Criteria: Users can mark expenses as recurring daily/weekly/monthly.

FR12. Receipt Upload
As a user, I want to upload a receipt so that the app can parse the details automatically.
Acceptance Criteria: App accepts JPG/PNG/PDF, extracts date, merchant, and amount.

FR13. Edit/Delete Expenses
As a user, I want to edit or delete expenses so that I can correct mistakes.
Acceptance Criteria: Users can update expense details or remove records.

FR14. Bulk Import
As a user, I want to import CSVs of transactions so that I can quickly add historical data.
Acceptance Criteria: CSV upload validates headers and imports all records.

Visualization & Analytics

FR15. Category Breakdown
As a user, I want a pie chart of expenses by category so that I can see proportions of spending.

FR16. Time-Series Graphs
As a user, I want line charts of spending over time so that I can track monthly changes.

FR17. Top 5 Categories
As a user, I want to see my top 5 spending categories so that I can quickly identify problem areas.

FR18. Custom Date Ranges
As a user, I want to filter expenses by date range so that I can analyze specific periods.

FR19. Budget vs. Actual
As a user, I want to compare actual spending against my set budget so that I know if I'm overspending.

FR20. Savings Progress Chart
As a user, I want a progress bar for each savings goal so that I can visually track progress.

Exports & Reporting

FR21. CSV Export

As a user, I want to export my expenses to CSV so that I can analyze them externally.

FR22. PDF Report
As a user, I want monthly PDF reports so that I can review and share my financial summary.

FR23. Email Reports
As a user, I want automated monthly reports via email so that I get insights without logging in.

AI & Smart Features

FR24. Merchant Recognition
As a user, I want merchants automatically recognized so that receipts are labeled correctly.

FR25. Smart Search
As a user, I want to search expenses by merchant or keyword so that I can find records quickly.

FR26. Anomaly Detection
As a user, I want alerts for unusual expenses so that I can spot potential fraud.

Gamification & Learning

FR27. Daily Challenges
As a user, I want financial mini-challenges so that I can improve my money habits daily.

FR28. Badges
As a user, I want to earn badges for achieving goals so that I feel rewarded.

FR29. Leaderboard
As a user, I want to see how I rank against friends so that I feel motivated.

FR30. Lesson Progress Tracking
As a user, I want a tracker for completed lessons so that I know how far I've come.

Notifications & Reminders

FR31. Budget Alerts
As a user, I want alerts when I exceed my budget so that I can control spending.

FR32. Goal Milestone Alerts
As a user, I want notifications when I hit 50% or 100% of a savings goal.

FR33. Lesson Reminder Alerts
As a user, I want reminders if I haven't studied for a week so that I stay consistent.

Admin & System Features

FR34. Admin Dashboard
As an admin, I want to view system metrics so that I can monitor usage.

FR35. Content Management
As an admin, I want to add or edit financial lessons so that content stays relevant.

FR36. User Management
As an admin, I want to suspend accounts so that I can handle abuse.

Error Handling & Usability

FR37. Invalid Input Handling
As a user, I want helpful error messages so that I understand what went wrong.

FR38. Offline Mode
As a user, I want to add expenses offline so that data syncs later when online.

FR39. Auto-Save
As a user, I want unsaved changes auto-saved so that I don’t lose progress.

FR40. Dark Mode
As a user, I want a dark mode so that I can use the app comfortably at night.

Security & Privacy

FR41. Data Encryption
As a user, I want my data encrypted in transit and at rest so that it is secure.

FR42. Privacy Settings
As a user, I want to choose what data is shared so that I feel in control.

FR43. Session Timeout
As a user, I want my session to auto-expire after inactivity so that my account is safe.

Collaboration & Social

FR44. Share Goals
As a user, I want to share a savings goal with friends so that we can achieve it together.

FR45. Group Challenges
As a user, I want to join group savings challenges so that I stay accountable.

FR46. Invite Friends
As a user, I want to invite others to the app so that I can compare progress with peers.

Miscellaneous Enhancements

FR47. Currency Support
As a user, I want to select my local currency so that amounts are displayed correctly.

FR48. Multi-Language Support
As a user, I want the app in different languages so that it's accessible globally.

FR49. Help Center
As a user, I want a help center with FAQs so that I can resolve issues independently.

FR50. Contact Support
As a user, I want in-app support messaging so that I can get help quickly.

4. Non-Functional Requirements

- **Usability:** The user interface (UI) must be designed for simplicity, intuitive navigation, and accessibility, adhering to mobile-first design principles for an optimal user experience on smartphones.
- **Security:** All sensitive financial data, both at rest (stored on servers) and in transit (during communication between the app and backend), must be robustly encrypted using industry-standard protocols to ensure maximum data protection.
- **Performance:** The application must exhibit high performance, with the dashboard loading within 3 seconds and AI categorization processing transactions in under 5 seconds to provide a smooth and responsive user experience.
- **Reliability:** The system must maintain a high level of reliability, targeting 99% uptime for all Firebase services to ensure continuous availability of the application.
- **Scalability:** The architecture must be designed to support future growth, with the MVP capable of handling up to 1,000 active users, and readily scalable to accommodate a larger user base as the app gains popularity.
- **Portability:** While initially designed for iOS, the application will be developed with adaptability in mind, allowing for a future transition to frameworks like React Native for potential cross-platform deployment.

5. Agile Process

- **Methodology:** The project will follow the Scrum framework, employing 2-week sprints to manage development cycles and facilitate iterative progress.
- **Backlog Management:** All requirements, expressed as user stories, will be meticulously tracked and managed using popular Agile tools such as Jira or Trello, providing transparency and clear prioritization.
- **Prioritization:** Features will be prioritized based on their value and necessity for the MVP. Core functionalities like the smart dashboard, goal setting, and AI categorization will be developed first, with gamification and social features introduced in later stages.
- **Deliverables by Sprint:**

Sprint	Features Delivered
1	Focus on establishing the foundational elements, including User Authentication and a basic Dashboard skeleton.
2	Implementation of the Goal Setting feature and the integration of reminder functionalities.
3	Development of the AI Categorization module, tested with sample data to refine accuracy.
4	Introduction of Gamified Learning lessons and the implementation of the Notifications system.
5	Dedicated to comprehensive Polishing and rigorous Testing across all implemented features to ensure stability and user satisfaction.

6. System Models

Visual and structural models will be used to clearly define the system's architecture and functionalities.

- **Architecture:** The system architecture will consist of an iOS frontend, communicating with a Firebase backend for data storage and user management, which in turn interacts with a dedicated AI Categorization model.
- **Use Cases:** Key user interactions will be mapped out through use case diagrams, including Login, Add Transaction, View Dashboard, Set Goal, and Complete Lesson, illustrating user flows and system responses.
- **Mockups:** Visual mockups will provide concrete representations of the user interface, including the Dashboard screen layout, the Goal progress bar visualization, and the Gamified learning streak view, guiding the UI/UX design.

7. Risks & Mitigation

- **Bank API Integration May Be Blocked:**
  - **Mitigation:** A robust fallback mechanism will be implemented, allowing users to upload bank statements via CSV files or manually enter transactions, ensuring uninterrupted functionality.
- **AI Miscategorizes Transactions:**
  - **Mitigation:** The application will incorporate a user feedback loop, enabling users to easily edit miscategorized transactions. This feedback will be used to retrain and improve the AI model's accuracy over time, leading to adaptive learning.
- **Gamification Scope Creep:**
  - **Mitigation:** To manage the scope effectively, the MVP will initially ship with a limited set of 3–4 core gamified lessons, preventing over-engineering and ensuring timely delivery. Future iterations can then expand on this foundation.

6.4 Database Design and Description Section

Finance Buddy uses Firebase Cloud Firestore as its primary database. Firestore is a NoSQL, document-oriented database that supports real-time synchronization, scalable storage, and secure access through Firebase Authentication. This section describes the structure of the database, how the application interacts with it, and how security is enforced across all modules.