Software Development Life Cycle (SDLC) Plan: Smart Farm Management System

Project Goal: To develop a secure, intelligent, and user-friendly farm management system that optimizes budgeting and scheduling for rice farmers.

Methodology: Agile Model

The Agile methodology will be employed to manage this project. This approach prioritizes iterative development, continuous collaboration, and flexibility to adapt to changing requirements. The project will be broken down into a series of five two-week sprints, each culminating in the delivery of a working, testable product increment.

Work Breakdown Structure (WBS) for Smart Farm Management System (Agile Model)

Phase 1: Sprint 1 - Core Data Management & Foundational UI

- **Epic: User & Data Foundations**
    - **User Story:** As a farmer, I want to securely log-in, so my data is protected.
        - Task: Set up local authentication (or user profile storage in SQLite).
        - Task: Design and code the login/registration screen.
    - **User Story:** As a farmer, I want to input and save my farm's basic information so I can get started.
        - Task: Design the database schema for the Farm table (e.g., location, size).
        - Task: Develop the user interface for inputting farm details.
    - **User Story:** As a farmer, I want to input my rice inventory so I can start tracking it.
        - Task: Design the database schema for the Inventory table (e.g., rice type, quantity, storage location).
        - Task: Create the UI for adding new rice entries.

Phase 2: Sprint 2 - Budgeting & Expense Tracking

- **Epic: Financial Management**
    - **User Story:** As a farmer, I want to track my expenses so I can manage my budget.
        - Task: Design the database schema for an Expenses table (e.g., date, item, cost, category like 'seeds' or 'labor').
        - Task: Develop the UI for adding and viewing expense records.
    - **User Story:** As a farmer, I want to see a summary of my spending so I can understand my costs.

        - Task: Write a function to query and sum expenses from the database.
        - Task: Create a simple report or dashboard view to display total costs.

Phase 3: Sprint 3 - Automated Scheduling & Weather Integration

- **Epic: Intelligent Scheduling**
    - **User Story:** As a farmer, I want the system to get the weather forecast so it can help with scheduling.
        - Task: Research and select a free weather API (e.g., OpenWeatherMap).
        - Task: Code the API call to download weather data.
        - Task: Design and create a Weather table in SQLite to store the fetched forecast.
    - **User Story:** As a farmer, I want the system to suggest a planting schedule based on the weather so I can optimize my crop.
        - Task: Design the decision tree algorithm logic for planting schedules.
        - Task: Code the algorithm to process weather data and farm data to make a recommendation.
        - Task: Create a UI to display the recommended schedule and an option to accept it.

Phase 4: Sprint 4 - Historical Data & Reporting

- **Epic: Insights & Analytics**
    - **User Story:** As a farmer, I want to see how my current budget compares to previous years so I can plan better.
        - Task: Write database queries to compare current and historical expense data.
        - Task: Design and develop a report screen with charts and graphs.
    - **User Story:** As a farmer, I want to see my past schedules and yields so I can learn from them.
        - Task: Implement functionality to store historical scheduling data.
        - Task: Create a report showing past planting, harvesting, and yield data.

Phase 5: Sprint 5 - Final Polish, Testing & Deployment

- **Epic: Final System Launch**
    - **User Story:** As a user, I want the system to look professional and be easy to use.
        - Task: Conduct a full UI/UX review of all screens.

        - Task: Refine the design for mobile and desktop access.
    - **User Story:** As a farmer, I want the system to be reliable and free of bugs.
        - Task: Perform comprehensive end-to-end testing of all features.
        - Task: Run user acceptance tests with a target user (e.g., Ignacio).
    - **User Story:** As a project team, we want to deliver the finished system to the user.
        - Task: Package the application for deployment.
        - Task: Create a user manual or training material for the farmers.

GANTT CHART