Project Plan for Coffee Shop Management Application

**Author:Thanh Nam Trinh**

**1. Executive Summary**

* Purpose: Explain that the Coffee Shop Management app aims to streamline operations for coffee shop staff and enhance the customer experience. The app will manage orders, inventory, employee schedules, and customer loyalty programs.
* Audience: Small to medium-sized coffee shop owners and managers.
* Persuasive Argument: Highlight how the app will improve order accuracy, track inventory in real-time, streamline employee management, and create a better customer experience through a loyalty program.

**2. Functional Requirements for App**

* **User Stories**:
  1. *As a barista, I want to view incoming orders in real-time so that I can prepare them efficiently.*
  2. *As a cashier, I want to quickly add items to a customer’s order so that I can reduce wait times.*
  3. *As a manager, I want to monitor inventory levels so that I can order new stock before supplies run out.*
  4. *As a customer, I want to earn rewards for each purchase so that I feel valued and encouraged to return.*
  5. *As an employee, I want to view my schedule so that I can manage my work-life balance.*
  6. *As a manager, I want to analyze sales reports so that I can make data-driven decisions.*

**3. Technical (Non-Functional) Requirements for App**

* C# and Visual Studio: The application will be developed using C# within Visual Studio for optimal performance and maintainability.
* GitHub Repository: All code will be version-controlled on GitHub for collaborative development.
* Code Comments: Key sections of the codebase will be commented for clarity, especially complex functions like order processing.
* README File: Include setup instructions, an app overview, and troubleshooting tips.
* Variables: Variables will be used to store data such as order details, inventory quantities, and customer information.
* Calculations: The app will use mathematical operators to calculate order totals and customer rewards.
* Boolean Operators: To manage logic, such as checking if an order is complete or if inventory needs replenishment.
* Conditional Statements: To handle operations like checking stock levels or validating customer rewards eligibility.
* Loops: For tasks such as listing menu items or processing batch data for reports.
* Data Structures: Lists and dictionaries will manage orders, inventory, and employee data.
* Object-Oriented Programming: Methods for adding new items, calculating discounts, and processing orders.
* GUI Development: A user-friendly interface will allow employees to take orders, view schedules, and manage inventory efficiently.
* Database: A database will store data related to orders, customers, inventory, and employee schedules.
* Entity Framework (EF) Core or ADO.NET: For database connectivity, handling CRUD operations to keep all records current.

**4. Task List**

* **Detailed Task Breakdown**:
  + Design GUI layout for order taking and inventory tracking
  + Implement order processing functionality
  + Set up database schema for inventory, orders, and customers
  + Develop customer loyalty program logic
  + Create reports for sales and inventory
  + Integrate employee scheduling and management tools
* **Sprint Assignment**: Allocate tasks into sprints, focusing on foundational features (e.g., orders, inventory) in earlier sprints.
* **Deadlines**: Set realistic deadlines to ensure timely completion, leaving room for testing and refinement.