Software Engineering Project

Topic: FinTrack [Expense Tracker]

Subject: Software Engineering

Register No: 242BCA41

Name: Yashas R

Mentor: Mr. Selwyn Paul

Sl.no	Date	Particulars	Remarks

Assignment 1

Abstract

FinTrack is a simple and user-friendly personal finance tracking system designed to help individual people manage their income and expenses in an organized way. The system allows users to securely log in, record income and expenses, and view their financial status through an interactive dashboard. Instead of depending on manual calculations or memory, this provides clear visual idea of their money, transaction history, and help users understand where their money comes from and goes. The main aim of FinTrack is to improve financial awareness, reduce money mistakes, and encourage better money management habits through an easy-to-use digital platform.

Introduction

Managing personal finances is often neglected due to lack of time, awareness, or proper tools. Many people still depend on notebooks or mental calculations to track their daily expenses and income, which often leads to confusion and poor financial decisions. FinTrack is developed to solve this problem by offering a centralized system where users can track, analyse, and manage their money efficiently. The application focuses on simplicity, and clarity. It allows users to log in securely, add income and expense entries, and view their financial overview in one place. FinTrack ensures that each feature works independently while contributing to the overall functionality of the application.

Modules & Design Pattern:

1. **Log In and Authentication Module:**

   This module handles secure user authentication. Users can log in using a username and password, ensuring that personal data remains protected. This is to provide secure access and prevent unauthorized use of financial data.

2. **Home Module:**

   The home page acts as the central navigation point of the application. From here, users can easily access all other modules such as income, expense, dashboard, and profile. This gives users a clear and simple entry point to all system features.

3. **Profile Module:**

   This module allows users to view and manage their personal profile details. It helps maintain user-specific information within the system. This helps in managing user identity and personalize the application experience.

4. **Income Module:**

   The income module allows users to record all sources of income. Each entry includes date and time, amount, category, account type, description. Income history and graphical representations are also provided. This helps in managing accurate records of all income sources and visualize earning patterns.

5. **Expense Module:**

   This module tracks all user expenses with details such as date, time, amount, category (food, travel, rent, etc.), payment method, and notes. Users can view past expense history along with graphs for better understanding. This helps in monitoring spending habits and help users control unnecessary expenses.

6. **Dashboard Module:**

This module tracks the total expenses and income of the individual and creates a graph structure helping the user understand the money better. Users can access everything easily in this module helping them manage the habits and control unnecessary items.

Software Model Selected: Incremental Model

FinTrack consists of multiple independent modules such as authentication, income, expense, and dashboard. Each module can be designed, developed, and tested separately. The Incremental Model allows the system to be built step by step, where a basic working version is available early, and additional features are added gradually.

This approach is especially useful for FinTrack because changes in UI design, categories, or data visualization can be easily made without affecting the entire system.

Benefits of Incremental Model

- **Early Availability of a Functional System:**
  The Incremental Model allows a basic version of the application to be developed and used at an early stage. This helps understand the system's core functionality without waiting for the entire project to be completed.

- **Independent and Modular Development:**
  Each feature or module is developed separately, which ensures clear separation of tasks. This structure improves system organization and makes development more easier.

- **Improved Accuracy (Focused Development):**
  Since each increment focuses on a single functionality, developers can give full attention to one goal at a time. This reduces logical errors and improves the accuracy and quality of each module.

- **Easy Error Detection and Maintenance:**
  Bugs, issues and errors can be identified and fixed within the specific module without affecting the rest of the system. This simplifies testing, debugging, and future maintenance.

- **Flexible Modification and Enhance Features:**
  Changes in requirements and improvements in design can be easily implemented. The system does not need to be rebuilt entirely, which saves time and effort.

- **Reduced Development Risk and Better Control:**
  By developing the system in small increments, risks can be identified early and handled efficiently. This approach ensures better control, lesser cost, and resources throughout the development process

Comparison Study:

Objective	**Waterfall**	**Spiral**	**Prototyping**	**RAD**	**Agile**	**Incremental (Selected model)**
Development Style	Phase-by-phase flow	Cycles with risk analysis	Prototype-first approach	Fast-paced development	Continuous iterations	Feature-wise incremental growth
Ease of Making Changes	Very difficult	Moderate	Easy	Easy	Very easy	Easy
Requirement Adaptability	Fixed after planning	Possible but complex	Easily adaptable	Adaptable	Highly adaptable	Adaptable
Risk Management Level	Very low	High focus on risk	Limited	Limited	Medium	Medium
Suitability for Small Applications	Poor	Poor	Average	Average	Less suitable	Highly suitable
Development Cost & Effort	Low	Very high	Medium	High	High	Low
User Participation	Minimal	Moderate	High	High	Very high	Moderate
Simplicity of Implementation	Simple	Difficult	Simple	Moderate	Complex	Simple
Independence of Modules	Low	Medium	Medium	Medium	Low	Very high