Smart Budget Tracker - Project Plan Document

Sabri Mert Piskin - 211101035

June 15, 2025

1. Project Scope

The project will cover the core and some advanced functionalities of the ”Smart Budget Tracker” application. The functional requirements within the scope are:

Core Features:

- **Expense Tracking:** Users can add, edit, and delete expenses with categories, amounts, dates, and descriptions.
- **Budget Management:** Set monthly/weekly budget limits for different expense categories.
- **Smart Notifications:** Automatic alerts when expenses approach or exceed budget limits. **Category Management:** Predefined and custom expense categories (Food, Transportation, Entertainment, etc.).
- **Dashboard:** Visual overview of spending patterns, remaining budgets, and financial summaries.

Advanced Features (to be included in MVP):

- **Expense Analysis:** Graphs and charts showing expense trends over time.
- **Export Functionality:** Export expense data in various formats (CSV, PDF).

This project is focused on MVP (Minimum Viable Product), and some advanced features such as budget suggestions, recurring expenses, and multi-account support may be deferred to later stages.

2. Project Organization - People (Roles and Responsibilities)

Team Members: As stated in Project Assignment 1, the roles and responsibilities of team members will be clarified as the project progresses. Individual contributions of each team member should be documented.

3. Objectives

The primary objective of the project is to develop a comprehensive budget tracking system with category-based expense management, implement a smart notification system for budget limit warnings, create an intuitive user interface for expense entry and budget visualization, and effectively apply multiple design patterns in the application architecture. Secondary objectives include providing detailed expense analysis and reports, enabling budget planning and goal setting, supporting multiple currency formats, and implementing data persistence and backup functionality.

4. Key Phases and Timeline

The project development process will be completed in the following four main phases:

Phase 1 (June 7, 2025 - June 15, 2025): Foundation Building

- Project setup and basic architecture design

Phase 2 (June 16, 2025 - June 29, 2025): Core Features Development

- Implementation of core data models
- Development of basic expense CRUD (Create, Read, Update, Delete) operations
- Integration of simple budget setting functionality
- Development of the category management system
- Creation of the dashboard with basic visualization
- Implementation of the smart notification system
- Integration of budget limit control logic /ul¿

Phase 3 (June 30, 2025 - July 9, 2025): Advanced Features UI Improvements

- Addition of expense analysis and reporting features
- Development of data export functionality
- User interface (UI) improvements and initial testing processes

Phase 4 (July 10, 2025 - July 19, 2025): Finalization Deployment

- Optimization of implemented design patterns
- Performance improvements and debugging
- Completion of project documentation
- Final tests and application deployment

5. Resource Planning

The necessary resources for the project are:

Technology Stack:

- **Frontend:** React.js
- **Backend:** Node.js
- **Database:** SQLite (PostgreSQL option can be evaluated)
- **Graphics:** Chart.js or D3.js **Notifications:** Web Notifications API

Tools:

GitHub (for repository management), code editors, design tools.

6. Risk Management

Potential risks that may be encountered during the project and strategies to mitigate these risks are listed below:

Risk: Integration of Complex Design Patterns

- **Mitigation Strategy:** Regular code reviews, validation of design pattern implementations, and seeking expert opinion when necessary.

Risk: Performance Issues

- **Mitigation Strategy:** Regular performance testing of the application, identification of bottlenecks, and code optimization.

Risk: Adherence to Timeline

- **Mitigation Strategy:** Clear definition of phases, regular progress tracking, and early detection of potential delays.

Risk: Scope Creep

- **Mitigation Strategy:** Clear definition of project scope and careful management of scope changes.

7. Communication Plan

This section outlines how communication will be managed throughout the project lifecycle. An active communication approach will be followed, including **weekly meetings** to discuss progress and challenges, and a **feedback mechanism** where team members provide input to each other upon task completion. This ensures all team members and relevant parties are informed and aligned.

8. Change Management Plan

This section describes the process for managing changes to the project scope, timeline, and resources within the student group.

- **Change Proposal and Discussion:** Any team member can propose a change, which will be discussed collectively during weekly meetings or through asynchronous communication channels. The proposal should detail the change, its rationale, and potential impact.
- **Consensus-Based Decision Making:** All changes will be approved through a consensus-based decision-making process among the team members. Discussions will focus on feasibility, necessity, and potential risks, ensuring collective buy-in.
- **Internal Feedback Loop:** As there is no external product user, development will be driven by the team's internal feedback. Completed tasks will be reviewed by peers, and feedback will be integrated iteratively to refine features and address issues.
- **Documentation of Changes:** Approved changes will be formally documented, including the rationale and the agreed-upon modifications to the project plan, to maintain a clear record for all team members.
- This collaborative approach ensures that any modifications are collectively managed and their impact is assessed before execution, promoting shared ownership and project adaptability.

9. Budget Plan

This section details the financial resources allocated for the project. Since this is a student group project, there are no direct costs associated with its development.

10. Acceptance Tests and Acceptance Criteria

The following criteria will be used to evaluate the success of the project:

* **Functional Success:** All core features and advanced features within the MVP scope, as defined in "Project Definition Document.pdf," work as expected.
* **Technical Success:** Successful implementation of at least three design patterns specified in the application architecture and effective use of the technology stack.

* **User Experience Success:** The user interface is intuitive, responsive, and usable on different devices (measurable through user testing).