

Date: 10/18/2025

ACKNOWLEDGMENT

We would like to express our sincere gratitude to each member of our group for their dedication, collaboration, and hard work throughout the planning of the proposal. We are proud of what we accomplished together as a team.

TABLE OF CONTENTS

ACRONYMS

ABSTRACT

INTRODUCTION	1
Background of Family Spending	1
The Existing System of Budgeting	1
Statement of the Problem	2
Objective of the Project	3
General Objective	3
Specific Objective	3
Proposed System	3
Feasibility Study	4
Economic Feasibility	4
Developmental cost	4
Operational Cost	4
Technical Feasibility	4
Schedule Feasibility	4
Scope	5
Methodology	5
Project Management plan	7
Time Management plan	8
Quality Management Plan	9
Requirement Management Plan	10
Communication Management Plan	10
Risk Management Plan	11
APPENDIX	12
REFERENCE	14

ACRONYMS

- **FBP**

Full name: Family Budget planner

Description: The proposed web-based system designed to help families manage, track, and plan their financial budgets effectively.

- **AGILE**

Full Name: Iterative Development Methodology

Description: A flexible approach to software development that delivers work in small incremental cycles.

- **API**

Full Name: Application Programming Interface

Description: Interface A set of rules that allows different software applications to communicate with each other.

- **CURD**

Full Name: Create, Read, Update, Delete

Description: The four basic operations used in database management.

- **DBMS**

Full Name: Database Management System

Description: Software used to store, manage, and organize data efficiently.

- **MVS**

Full Name: Model-View-Controller

Description: A software design pattern that separates data (Model), interface (View), and logic (Controller).

- **UI**

Full Name: User Interface

Description: The visual part of a software system through which users interact.

- **UX**

Full Name: User Experience

Description: The overall experience and satisfaction a user feels when using a product or system.

- **SPMP**

Full Name: Software Project Management Plan

Description: A detailed document describing how a software project will be managed, executed, and controlled.

- **SRS**

Full Name: Software Requirement Specification

Description: A document that describes the system's functional and non functional requirements.

ABSTRACT

The Family Budget Planner (FBP) is a web app that helps families keep track of their income and spending. Many families find it hard to manage their money and save properly. This project aims to make budgeting simple and easy. Users can add their income and expenses, see spending reports, and get reminders for bills.

The system is designed to be user friendly and will help families understand their money better and save more. It is a lightweight, user-centered family budget planner designed to help households record income, classify expenses, set and monitor savings goals, and visualize cash flow trends. The system supports multi-member households with role-based access, recurring transactions, category budgets, and automated alerts for overspending. A clear dashboard presents weekly, monthly, and annual summaries along with interactive charts and forecasted balances based on historical patterns. Built with a modular architecture, FBP integrates local data persistence for offline use and optional cloud synchronization for multi-device access, employing encryption for sensitive data and role-based authentication for privacy.

The project focuses on usability for non-technical users, offering guided on boarding, predefined expense templates, and quick-entry modes for mobile use. Key technical components include a relational data model for transactions and budgets, a RESTful API layer, a front-end single-page application, and lightweight backend services. Success metrics include reduced monthly overspend, user adoption in pilot households, and system reliability under typical family workloads. Deliverables for this fundamental software engineering project comprise Requirements documentation, Data collection, Design diagrams, Implemented prototype, Test results, user manual, and Final project.

INTRODUCTION

Background

Family spending refers to how households allocate their income to cover daily needs such as food, housing, education, health care, and entertainment. Managing family expenses efficiently is vital for maintaining financial stability and avoiding unnecessary debt. In most families, budgeting decisions are done manually or without a structured system, which often leads to poor financial tracking, overspending, and a lack of savings.

As living costs continue to rise, effective budgeting has become more essential than ever. Families need a simple, reliable, and digital method to plan their monthly expenses, track income sources, and control spending habits. Modern technologies such as web and mobile applications now provide better solutions to help families monitor their finances in real time.

The Existing System

Currently, most Ethiopian households depend on manual budgeting systems or do not keep financial records at all. Those who attempt to manage their budgets often use notebooks, diaries, or simple mobile calculators to track daily expenses. Others may use spreadsheet programs like Microsoft Excel but without consistency or proper formulas.

The major limitations of this existing approach include:

- Lack of accuracy: Manual recording is prone to human error and forgotten entries.
- Difficulty in tracking expenses: It is hard to identify where most of the money is being spent.
- No long-term analysis: Families cannot easily compare their spending patterns across months or years.
- Data loss risk: Paper records can be damaged, misplaced, or lost.
- Limited awareness: Many families are not familiar with digital tools that could simplify financial management.

As a result, families often face financial uncertainty, especially when unexpected costs arise. A modern digital budgeting system that automates expense tracking, categorizes spending, and generates monthly summaries could transform the way Ethiopian households manage money. It would also align with the country's move toward digital transformation and improved financial literacy.

Statement of the Problem

Family Financial Fragmentation

- Modern families face the everyday problem: financial fragmentation. With joint income, costs, and saving goals divided among many different instruments and improvised conventions, families fall out of sync. Tension thereby builds up, goals are not met, and avoidable tension arises. The source of the issue is the lack of a single unified system that allows households to see their money clearly, work together in real time, and make sane decisions
- The target audience for this project are any family unit (couples, solo parent households, multi-generational families) who wish to have a shared budget succeed

Pain points addressed

- Splintered tracking: transactions are track inconsistently on multiple platforms so one can never know the true up-to-date budget status
- Delayed budget synchronization: budget updates are not in real-time, meaning that families cannot see the impact of their spend and therefore make impulse decisions
- Lack of responsibility: without an exact record of who spent how much, financial conflicts escalate and budgetary boundaries are easily exceeded

User needs

- The basic needs of the target audience is for one, accessible web site that provides
 - One source for income monitoring and expenditure
 - Clear visualization of progress and open contribution tracking
 - Instant, multi user transaction logging

Real world applications

- A family decides to save for a vacation. They know that the target is some thousands of dollars, but since the saving is in the main account, it feels far from their normal spendings. Without an open tracker, they end up borrowing from the goal fund for non-necessities. The lack of apparent progress propels the target buy repeatedly again and again

Objective of the Project

- The primary purpose is to develop an amicable user, cooperative domestic budget planner system

Global objective

- To assist families with total financial transparency, make decisions together, and reach their savings goals in a single virtual portal that bridges income, spending, and goal management

Specific objective (SMART)

- Upon completion of the project, the system will possess the following specific, measurable, achievable, relevant, and time bound adjectives

S: establish secure authentication with tiered access, implement a particular feature for future saving goals

M: allows users to construct customisable categories, allow users to record a contribution goal several times weekly

A: monitors routine activity, designed for performance optimisation

R: ensures system reliability and trust by users

T: integrated and tested by week 8

Proposed System

The proposed system is a budget planner : a web application that helps families manage their finances more effectively and in a more efficient way.

It is a platform where users can record income, track daily expenses, set monthly budgets, and monitor savings goals.

Unlike traditional paper-based or manual tracking, this system automates calculations, provides annual or monthly summaries, and generates visual reports like (charts and graphs) to help users understand their financial patterns.

Feasibility Study

Economic Feasibility

Development Cost:

The system requires minimal financial investment since it will use open-source technologies like HTML, CSS, JavaScript, Node.js, react and MongoDB.

Operational Cost:

Hosting and maintenance costs are low; the system can be deployed locally or on free hosting platforms like Vercel,Render, Netlify, or GitHub Pages.

Technical Feasibility

The budget planning system is technically feasible by using available technology, skills, and resources. This system will use HTML, CSS, JavaScript (maybe React) for the front end design and Java for backend; My SQL will be used as a database. Our version control will be GitHub; then use canva and figma as a design tool for mockups. These technologies and tools are cost-effective, easy for maintenance & upgrade and can integrate or run with existing software or systems. The Project team has the technical skills required to design,develop,manage,test and maintain the system effectively.

Schedule Feasibility

The estimated timeline for the Budget Planning System Project is approximately 8-12 weeks. This period allows enough and sufficient time for planning, design, development and testing.

Phase/Stage	Description	Timeline(Weeks)
Planning & Requirement Analysis	Define goals & scope, study current system, collect requirements	1-2
System Design and Development	Create database, code and develop the system	5-6

Testing	Check the errors and the performance	1-2
Documentation, Finalization and Presentation	To prepare reports in each phase, finalize and present the document	1-2
Total timeline/duration	-	8-12

Scope

The scope of the Budget Planning System Project is to clarify what the project will (and will not) deliver, in order to avoid future shifts in the level of ambition.

This project will deliver a platform that allows users to register, login & securely access their financial data and also add, edit and delete their daily expenses. It enables users to plan and track their budgets, also generates financial summaries and reports and secures data using MySQL database. The system provides simple and user-friendly interfaces.

However, this project will not deliver mobile application version, cloud or online deployment and no third-party integration. It will also not include advanced analytics like currency conversion, investment tracking, etc.

The Budget Planning System will have a basic and functioning platform for managing income and expenses. It will allow users to plan and maintain their budgets while ensuring the project remains achievable with the team's available time, skill and resources.

Methodology

To develop the Family Budget Planner, we will follow the Agile Software Development Life Cycle (Agile SDLC), which emphasizes iterative development, continuous user feedback, and flexibility in adapting to changes. The project will begin with the Requirements Gathering phase during the first sprint, where we will review previous research works, conduct user interviews with 5–7 households of various types and distribute surveys to at least 20 respondents to understand common budgeting challenges, must-have features, and user

expectations. From there, we'll make a clear list of tasks and features, based on what users need most, to help us stay organized as we build the project.

In the Design phase, we will define system constraints such as usability, simplicity, performance, and maintainability. We will design mockups for the UI/UX using tools like Figma, and get it reviewed by peers and target users for early feedback on accessibility, responsiveness on mobile phones etc. Development will proceed using tools such as HTML, CSS, JavaScript (or React.js), and optionally Node.js for backend logic and authentication. Git and GitHub will be used to collaborate, track changes, and manage our codebase throughout the project. We plan to use jQuery for interactive features and Bootstrap to build a responsive, user-friendly interface, while using localStorage to manage data directly in the browser. In each stage of the project, we will build important features like expense tracking, budget categories, and savings goals one at a time, using small, reusable parts. We'll regularly review our progress and adjust based on our priorities.

Testing will be integrated throughout the process, including manual testing for UI/UX issues, and user testing with original interviewees to validate ease of use and usefulness. In the final stage, we will conduct system testing across devices and browsers, and success will be measured by how effectively users can manage their budgets with minimal assistance. By using Agile, we ensure that the project remains aligned with real user needs, and we are able to improve the product continuously throughout development.

Project Management plan Overview

The Project Management Plan will lay out the details of the management plan that will be followed in order to develop the Budget planner software. It outlines the project development approach, captures the project baselines and all the subsidiary management plans as well as a project status report updated periodically as we go through with the project.

A. Purpose, Scope and Objective

The Software Project Management Plan (SPMP) will relay all the details regarding the development plan and the development cycle. It will assign each member roles and duties regarding the development, and how and what methods they will use in order to finish their tasks. It will also assign deadlines so that project will be developed on time. It will follow the specifications declared in the most current version of the SRS.

B. Assumptions and Constraints

- Team members will attend all meetings
- Team members will meet all the deadlines
- Team members will follow the requirements specified in SRS
- The system must run on web browsers only (no mobile app version).
- Team members will work on the project outside the class to finish it on time
- Advanced features like multi-user accounts, online transactions, or AI-based suggestions are outside the project scope.

Roles and Responsibilities

Role	Member	Responsibilities
Project Manager / Team Leader	Tsadena Abraha	Coordinates team activities, assigns tasks, monitors progress, communicates with the sponsor, ensures deadlines are met.
System Analyst	Hiruye Dereje	Collects and analyzes requirements, defines system features, ensures the system meets user needs.
System Analyst	Oumer Jemal	Designs user interface layouts and user experience flow, prepares mockups, ensures usability and consistency.
Frontend Developer	Maryamawit Getu	Develops the user interface using HTML, CSS, and JavaScript; ensures responsive and interactive design.
Backend Developer	Yeabfikir Kedir	Handles server logic, connects database to frontend, ensures secure and efficient data handling.
Database Administrator (DBA)	Habiba Ziyad	Designs and manages the project's database, maintains data integrity and performance.
Quality Assurance (Tester)	Amanuel Dejene	Tests the software for errors or bugs, ensures all requirements are met before final submission.
Documentation & Presentation Lead	Shahad Ahmed	Prepares all written documents (proposal, reports, SRS), creates slides, and leads presentation efforts.

Time Management plan

This is to outline the schedule creation process and the basis schedule estimates, display the high level milestones and project schedule as well as the project calendar and schedule impacts.

Aside from the regular class meeting time, the team might be required to meet outside the class in order to finish and deliver the product on time. Managers will make sure that each member is completing their tasks and on time. The managing team will keep a constant eye on the progress of the project and make sure that everything is completed by the deadline. The project team will maintain a detailed project calendar outlining major milestones and deliverables. Progress will be reviewed weekly to ensure adherence to schedule.

The project schedule will be managed using Google Calendar, where all key milestones, deadlines, and meetings will be recorded and shared with the team. Each member is responsible for keeping track of their assigned tasks and upcoming deliverables through this calendar.

Task progress and weekly goals will also be tracked using GitHub Projects, which provides visibility into the current status of each feature or component being developed. This ensures transparency and accountability across all team members.

The team will follow a structured schedule to ensure timely completion of all project deliverables.

Week 1 – 2: Project proposal and requirement analysis

Week 3: System and interface design

Week 4 – 7: Backend and frontend development

Week 8: Integration and testing

Week 9: Final documentation and presentation preparation

(A visual version of this plan is provided in **Appendix A** for reference.)

Quality Management Plan

The managing and the developing team will constantly perform a quality check on the software at least once a week to make sure that the project meets all the expectations. Our mentor/ Supervisor will be present throughout the development of the Software and therefore will be informed of the quality.

During development and testing phases (like uni testing, user acceptance testing), every member of the development team will test and report any deficiencies in the software. Design flaws or bugs will be reported and documented for immediate fix or future review.

A weekly review will be held to evaluate code quality and application performance against predefined acceptance criteria.

Requirement Management Plan

Each member of the group is required to attend the meetings in the class. Also each member is required to document their code, follow the guidelines decided in the SRS and meet each deadline as well. Any unexpected issues, technical difficulties or requests by the clients will be assessed by the Managing team and decided upon.

Requirements will be tracked and updated through Google Docs and Notion. All requirements changes will be documented and version-controlled using GitHub to maintain traceability.

Communication Management Plan

The managing team will notify our mentor of the weekly meetings. The project manager will generally send the attendance of the group members and make sure that the software is up to date. The managing team will also make sure that each week every developer completes their tasks so that the project is on track and progressing as expected. All key decisions and updates will be documented and shared via a group Google Drive folder.

Meetings will be held in person or through Google Meet and GitHub will be used for technical collaborations.

The managing team will make sure that the code is efficient, and meet all the standards. Each member of the development team will keep the managing team members up to date on any issues that they might encounter. Afterwards the managing team will decide on how to handle those issues and make sure that the project is completed. They will also make any changes necessary to make sure that the project runs as smoothly and as efficiently as possible.

Risk management Plan

Development

- The team will meet regularly to make sure that the production is not stopped at one particular point.
- There will be scheduled deadlines that everyone will be following to make sure the project is not delayed
- Each team member will be kept up to date, and inform the managing team of any changes or difficulties that might affect their ability to complete the task they are given on time.

Project Failure

- If the technology does not exist or is not viable the managing team will discuss with the client and make sure that there is another route that will result in a satisfactory outcome for the client.

To minimize potential disruptions, the team identified key project risks and defined strategies for prevention and mitigation.

The main risks include communication delays, technical challenges, and scheduling conflicts.

A detailed breakdown is provided in **Appendix B**.

APPENDIX

Appendix A - Time Management Plan (Gantt Chart Overview)

Task	Start Date	End Date	Duration	Responsible Members	Status
Project proposal & approval	Oct 11	Oct 18	1 week	All	Completed
Requirement analysis	Oct 19	Oct 26	1 week	Team Lead + Analyst	In Progress
System design (UI & architecture)	Oct 27	Nov 3	1 week	UI/UX + Developer	Planned
Backend development	Nov 4	Nov 18	2 weeks	Developers	Planned
Frontend development	Nov 19	Dec 2	2 weeks	Developers	Planned
Integration & testing	Dec 3	Dec 9	1 week	QA + Developers	Planned
Documentation & final report	Dec 10	Dec 13	4 days	Documentation Lead	Planned
Presentation & submission	Dec 14	Dec 16	3 days	All	Planned

Appendix B - Risk Management Plan

Risk	Impact	Likelihood	Mitigation
Team member unavailable due to illness or workload	High	Low	Assign backup tasks; track updates on GitHub
Delay in backend development	High	Medium	Divide tasks; use milestones to monitor progress
Miscommunication or missed meetings	Medium	High	Weekly meetings; shared minutes on Google Docs
Unclear requirements or scope changes	Low	Medium	Confirm all changes with instructor

Appendix C - Github Repository Link: direct link to our repo for code and updates.

<https://github.com/tsadena/Budget-Planner>

Appendix D - Meeting schedule: Planned meeting days through a shared google calendar.

<https://calendar.google.com/calendar/u/0?cid=NGI1ZGVhYjMzYTFlkYjVhNjU0YzYxZDA5YWU2MDZhOTA0ZTU4NjkyNWUyMzZhYjIwZDkyM2I4Mjk2NTc1NTE5N0Bncm91cC5jYWxlbmRhci5nb29nbGUuY29t>

References

- Sommerville, I. *Software Engineering*, 10th Edition, Pearson, 2015.
- “How to Create a Gantt Chart in Excel,” *ProjectManager.com*,
<https://www.projectmanager.com/guides/gantt-chart>
- GitHub. *GitHub – Version Control Platform*. <https://github.com>
- Software Project Management Plan, MobileAppDev. Academic
- Google. Google Calendar - Team collaboration Tool. <https://calendar.google.com>