

TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

**CMPE 491 – Project Analysis Report**

by

Arda Ertürk

Bora Kutun

Ece Atakol

Sarp Arslan

22.11.2024

**1. Introduction**

* 1. **Description**

Envolet is a simple, intuitive budget planning and tracking tool to facilitate sustainable living and help the user save money. The program will utilize AI to analyze user spending behavior and provide actionable information in the form of personalized text-based recommendations and visual diagrams. Envolet is saving consumers money but also facilitating more environmentally responsible purchases by pointing out wasteful expenditure while substituting alternatives.

The Node.js application will use the Express.js library to ensure that the back-end data processing and API management are effectively handled. User authentication will securely be provided through Firebase, while MongoDB is to be used for data storage, providing an adaptable structure in tracking user transactions and spending patterns. The Flutter mobile app will be developed for iOS and Android devices, which will provide customers the facility to access their financial data and insights anywhere, anytime.

**2. Current System**

Currently, there is no system like Envolet that provides smart recommendations, encourages financial growth, encourages sustainable living tips, and does this all on one mobile application. Many of the current budgeting applications allow a user to do nothing more than track their spending. While helpful, they do not help users to understand how to save money or make better choices financially. Many of these systems lack any tools for rendering personalized advice based on the user's spending habits.

Other shortcomings of the existing apps are that they do not encourage users toward sustainable or eco-friendly purchases.

These applications are purely designed for handling money-related tasks and do not provide users with ways to reduce waste or make decisions for the environment. Thus, it becomes quite tedious for a person who wants to manage money while living sustainably. From the technical perspective, most of the existing systems use outdated technology. That means they are slower, less reliable, and harder to upgrade as more users join on. For example, a back-end system might not be capable of handling large volumes of data without heating up, which may lead to lags or application crashes. Besides, some apps don't pay enough attention to users' security, which could compromise sensitive financial information. On the side of the app visuals, most of the financial apps offer graphs and charts but lack interactive visuals. Such visuals make it hard for users to clearly understand their spending. Envolet will differentiate from them by addressing major weaknesses of existing apps and offering a unique, user-centric solution.

**3. Proposed System**

**3.1 Overview**

Envolet will eliminate these issues by developing a state-of-the-art budgeting tool that offers its users opportunities to save money by making environmentally conscious choices. The application will be able to offer personalized recommendations with the ability to apply AI in reviewing one's spending. Besides, it will be an easy-to-use application and very functional both on iOS and Android systems.

The back-end will use Node.js to efficiently manage data speedily. Firebase will be used because user authentication needs to be secure, and private data of users’ needs to be tracked. MongoDB, flexible and good, will be used as the database in the tracking of user transactions and spending patterns.

The mobile application will be developed in Flutter so that it works seamlessly on iOS and Android. It will present users with their financial data through clear charts and graphs, as well as real-time useful advice.

**Features and Goals**

• **AI Insights:** Advise users on how to save money and reduce waste.

**• Clear Visuals:** Show spending data with simple and interactive charts.

**• Eco-Friendly Tips:** Help users make better, sustainable choices.

**• Strong Security:** Protect user data with Firebase and encryption.

**• Multi-Platform:** Works well on both iOS and Android devices.

**• Scalable Design:** Handle more users in the future without slowing down.

**3.2 Functional Requirements**

* **Registration & Login:** Users can register or log in using Firebase authentication.
* **Budget Planning:** A user can track their spending across multiple categories and set budget objectives.
* **Spending Analysis:** AI will analyze consumer spending habits to identify areas where consumers could save money.
* **Recommendations:** The application, via text analysis, will make personalized recommendations on how to spend in more eco-friendly ways.
* **Visualization:** The spending data will be visualized in some diagrammatic forms, like charts and graphs, for easier interpretation.
* **Mobile Access:** All functionalities will be cross-platform accessible via a mobile application built using the Flutter application.

**3.3 Non-Functional Requirements**

* **Security:** Firebase will handle user authentication in a secure manner; data should be encrypted.
* **Scalability:** The system should be able to support more users and increased data without any performance degradation.
* **Usability:** The app will provide an easy-to-use interface in moving around.
* **Performance:** The app should respond quickly to whatever the user is doing and make suggestions as they go along.

**3.4 Pseudo Requirements**

* **Platform Restrictions:** Envolet will only work with iOS and Android. The current stage of development does not permit web access.
* **API Constraints:** Firebase, MongoDB, and external APIs will be used by the system as needed. The functionality of the app may be impacted by any modifications or interruptions to these services.
* **User Data Limitations:** The program does not integrate external financial data feeds, and data analysis and suggestions are restricted to user-entered data.

**3.5 System Models**

* + 1. **Scenarios**
1. **Scenario 1: Budget Setup**
* Description: A new user logs into Envolet, creates budget limits for several categories, and starts tracking the amount spent for each category.
* Actions:
	+ The user registers or logs in to the Envolet app via Firebase.
	+ They go to the budget planning page and create budgets for each category.
	+ Envolet stores these selections and starts tracking expenses.
1. **⁠Scenario 2: Spending Analysis and Recommendations**
* Description: A user reviews their monthly spending analysis and receives suggestions.
* Actions:
	+ The user chooses the "Spending Analysis" option.
	+ Envolet analyzes recent transactions to generate insights.
	+ The user analyzes recommended changes for long-term savings, such as ideas for decreasing transportation expenses.
		1. **Use Case Model**

**1.⁠ ⁠Use Case: Register/Login**

* Actors: User, Firebase (authentication service)
* Description: Allows users to safely register and log in to view their personal financial information.

**2.⁠ ⁠Use Case: Set Budget**

* Actors: User
* Description: Allows users to create budgets for specific spending categories.

**3.⁠ ⁠Use Case: View Spending Analysis**

* Actors: User
* Description: Provides individualized spending insights and recommendations.

**4.⁠ ⁠Use Case: View Recommendations**

* Actors: User, AI Analysis System
* Description: The AI technology provides individualized, long-term recommendations based on spending trends.
	+ 1. **Object and Class Model**

**•⁠ Classes:**

* User: Attributes include user ID, name, email, and budget preferences.
* Budget: Attributes include category, spending limit, and actual expenditure.
* Transaction: Attributes include amount, category, date, and description.
* Recommendation: Attributes include recommendation text, relevance, and category.

**•⁠ Relationships:**

* The User class interacts with the Budget and Transaction classes to track and analyze spending.
* Transaction data feeds into Recommendation generation via AI processing.

**3.5.4 Dynamic Models**

**•⁠ Sequence Diagram: User Login and Budget Setup**

* User → App Interface: Enters login credentials.
* App Interface → Firebase: Verifies credentials.
* Firebase → App Interface: Confirms authentication.
* User → App Interface: Allows users to set budgets for different categories.
* App Interface → Database: Stores budget data.

 **•⁠ Activity Diagram: Spending Analysis and Recommendation Generation**

* User logs in and selects “Analyze Spending.”
* System retrieves transaction data.
* AI analyzes data to determine spending patterns.
* Recommendations are generated and provided to the user.

**3.5.5 User Interface - Navigational Paths and Screen Mock-Ups**

**1. Navigational Paths**

* Home Screen: The home screen shows the recent transactions, and account balances and has links to spending analysis for more insights and recommendations.
* Budget Planning Screen: The budget planning screen allows the users to set and modify budget limits for different categories.
* Spending Analysis Screen: The spending analysis screen has charts and graphs that breakdown the spending habits of the users.
* Recommendation Screen: The recommendation screen gives personalized financial tips and suggestions generated by AI.

**2. Screen Mock-Ups**

* Home Screen: The home screen displays account balances of each linked account and shows recent transactions and has links to analyze spending or managing budgets.
* Budget Planning Screen: The transaction management screen has fields to specify the source of funds so that users can categorize their income, expenses and transfers.
* Analysis Screen: The spending analysis screen shows the spending trend of the users with line and bar charts over different time frames (e.g. week, month, year).
* Recommendation Screen: The recommendation screen lists financial insights and tips generated by AI, which users can save or dismiss for future reference.



**4. Glossary**

* **AI:** Artificial Intelligence, used in Envolet to analyze user data and offer personalized recommendations.
* **Firebase:** A platform by Google for app development that includes authentication and database services.
* **Flutter:** An open-source UI framework by Google for creating natively compiled applications for mobile.
* **Node.js:** A JavaScript runtime for executing JavaScript on the server.
* **MongoDB:** A NoSQL database program for storing data in a flexible, scalable way.
* **Express.js:** A framework for Node.js web applications that is used to create the backend, including APIs.
* **Authentication:** A procedure that verifies a user's identification in order to grant them safe access to a program or system.
* **Encryption:** A technique for protecting data by encoding it to prevent unauthorized access.
* **Database:** A data management and storage system; MongoDB serves as the database in Envolet.
* **API (Application Programming Interface):** A collection of guidelines and conventions for creating and using software programs.
* **Scalability:** A system's capacity to accommodate more users or work without affecting performance.
* **User Interface (UI):** The application's graphical user interface allows users to interact with it.
* **Budget Planning:** A feature that lets users track their spending and set financial goals by category.
* **Spending Analysis:** Analyzing user transactions to find trends in spending and potential areas for development.
* **Recommendation System:** An AI-powered function that makes recommendations to customers based on their purchasing patterns.
* **Cross-Platform:** Refers to software that uses Flutter to run on several operating systems, including iOS and Android.
* **Sequence Diagram:** A particular kind of UML diagram that displays object interactions in a sequential fashion.
* **Activity Diagram:** A kind of UML diagram that shows activity and action workflows.
* **NoSQL Database:** A kind of database architecture that offers an adaptable schema, as used with MongoDB.
* **Backend:** The infrastructure and logic on the server side, managed using Node.js and Express.js in Envolet.
* **Frontend:** The client-side interface that users interact with, built using Flutter in this case.

**5. References**

* Firebase documentation: <https://firebase.google.com/docs>
* Node.js with Express.js documentation: <https://expressjs.com>
* MongoDB documentation: <https://www.mongodb.com/docs>
* Flutter documentation: <https://flutter.dev/docs>