TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 491 – High-Level Design Report

by

Arda Ertürk

Bora Kutun

Ece Atakol

Sarp Arslan

27.12.2024

Table of Contents

1. Introduction

Envolet is a modern budgeting and financial management tool that blends AI-driven insights with sustainable recommendations. Unlike traditional budget applications, which are more tracked in nature, Envolet provides actionable insights on how one can optimize his or her spending while choosing environmentally friendly options.

The system is powered by Node.js and Express.js, which provide an efficient backend for data processing and strong application logic. Firebase handles user authentication securely, while MongoDB provides scalable and flexible data storage. The mobile application, developed using Flutter, ensures a seamless experience on both iOS and Android platforms.

Envolet lets users set budgets, track their expenses, and receive personalized recommendations. The system analyzes spending patterns, gives advice on how money might be saved, and builds better habits in support of sustainability. Such a unique combination of financial management and eco-awareness places Envolet in a special position where users can support personal financial growth and responsible living.

1.1 Purpose of the System

Envolet is designed to fill the gap that exists among budgeting applications through the integration of advanced financial tracking along with eco-friendly recommendations. The aim of the system is:

- Allow users to save money by providing them with AI-powered spend pattern insights.

- Encourage living sustainably, pointing out areas of wasteful spending and suggesting eco-friendly alternatives.

- Provide users with a seamless, secure platform where they can manage their budgets and analyze financial habits, ensuring it's usable on both iOS and Android platforms.

Envolet stands alone because of its unique combination of financial analysis and sustainability goals. It will let users align their personal financial growth with positive impacts on the environment.

1.2 Design Goals

The design of Envolet focuses on achieving the following goals:

1. **Intuitive User Experience**: The interface shall focus on ease of use, with easy navigation and presentation of data.

2. **Security and Privacy**: Best practices shall be followed for securely storing sensitive user data, such as financial transactions, and sending it over the network securely.

3. **Scalable Architecture**: The backend shall be designed to handle growth in users and data complexity without performance degradation.

4. **AI-Driven Insights**: Employ AI to provide the user with actionable insights in their unique behavior and allow them to make wiser decisions about money.

5. **Eco-Conscious Functionality**: Through specific recommendations via spending analysis, let users make environmentally friendly choices.

6. **Cross-Platform Accessibility**: Create an identical, seamless user experience on iOS and Android with Flutter.

7. **Maintainability**: Code in a modular fashion in order to facilitate much easier updates and enhancements later on.

1.3 Definitions, Acronyms, and Abbreviations

- **AI (Artificial Intelligence):** The use of machine learning algorithms to analyze expenditure and suggest recommendations.
- **Firebase:** Cloud-based backend service for user authentication, having real-time database capabilities.
- **MongoDB:** NoSQL database system to provide flexible and scalable data storage.
- **Node.js:** An environment for running server-side JavaScript.
- **Flutter:** Framework for building natively compiled applications for mobile, web, and desktop from a single code base.
- **Encryption:** A process that secures data in unreadable code, usable only through decryption.

1.4 Overview

The Envolet system seeks to revolutionize personal budgeting by including financial planning with set goals for sustainability. The effective backend is implemented using Node.js and Express.js for transactions of data and application logics, while the mobile frontend provides an attractive user experience made in Flutter, showing dynamic updates of information and visualization in real time. User data and authentication are handled through Firebase, while MongoDB can handle heavy and scalable storage of data. The integration of AI on the platform further provides analysis of spending patterns and generates insights to guide users toward cost-effective and sustainable choices.

2. Current Software Architecture

Envolet does not have a current system. The application is being developed from scratch to fill shortcomings in current budgeting tools. Existing programs only track spending and provide no meaningful financial advice or motivation to live sustainably. These systems also lack modern technical frameworks, which might compromise scalability and user security. Envolet, on the other hand, aims to address these issues through a secure, scalable, and AI-driven architecture.

3. Proposed Software Architecture

The Envolet system will use cutting-edge architecture to provide scalability, security, and simplicity. The backend will use Node.js and Express.js for carrying out server-side operations. Firebase will be integrated to enable secure authentication and protect sensitive user data. MongoDB, a NoSQL database, will store user transactions and spending data, with a flexible and adaptive structure.

The front-end will be a cross-platform mobile application built with Flutter, allowing users to access their financial data easily on both iOS and Android devices. AI algorithms will evaluate user transactions to create tailored recommendations and environmentally beneficial suggestions. These insights will be presented via interactive charts and graphs to improve the user experience. The system is designed with scalability in mind, so it can handle an increasing number of users without sacrificing performance or stability.

The architecture prioritizes strong access control, with Firebase handling secure user authentication and role-based access control. Encryption will protect all data during transport and storage. The system will effectively handle boundary conditions, with procedures in place to deal with problems like cloud service problems or Firebase and MongoDB downtime. By merging modern frameworks and technologies, Envolet strives to provide its users with a seamless, safe, and insightful budgeting solutions.

3.1 Overview

The Envolet system architecture is designed to be modular, scalable, and secure, built using modern technologies and frameworks. The system is divided into several key subsystems that work together to provide a comprehensive budgeting and sustainability solution:

- **Frontend Mobile Application**: Built with Flutter for cross-platform compatibility, providing the user interface for budget management, spending analysis, and eco-friendly recommendations.

- **Backend Server**: Implemented using Node.js and Express.js, handling business logic, data processing, and API endpoints for client-server communication.

- **Authentication System**: Utilizing Firebase for secure user authentication and session management.

- **Database System**: Leveraging MongoDB for flexible and scalable data storage of user information, transactions, and spending patterns.

- **Flutter Gemini Library**: Implemented using Flutter Gemini package, responsible for processing spending patterns and generating personalized recommendations for financial savings and sustainable choices.

3.2 Subsystem Decomposition

The Envolet system is decomposed into the following major subsystems:

Frontend Mobile Application Subsystem

- UI Components Module manages the presentation layer, including screens for budget planning, spending analysis, and recommendations
- Network Module manages API calls to the backend server
- Local Storage Module handles caching and offline data persistence

Backend Server Subsystem

- API Gateway manages incoming requests and routes them to appropriate services
- Transaction Processing Service handles recording and processing of user transactions
- Budget Management Service manages budget creation, updates, and tracking
- Recommendation Service interfaces with the AI engine to generate and deliver personalized recommendations

Authentication Subsystem

- User Management Service handles user registration and profile management
- Authentication Service manages login, session tokens, and security
- Access Control Service implements role-based access control and permissions

Database Subsystem

- Data Access Layer provides interfaces for database operations
- Transaction Storage Service manages transaction records and history
- User Profile Service handles user-related data storage
- Analytics Storage Service stores processed spending patterns and analysis results

3.3 Hardware/Software Mapping

Envolet will be deployed across multiple environments to ensure optimal performance and reliability:

Cloud Infrastructure

- Backend Server hosted on cloud platforms
  - Node.js runtime environment
  - Express.js framework
  - Load balancers for traffic distribution
  - Auto-scaling configuration for handling varying loads

Database Infrastructure

- MongoDB Atlas cloud-hosted database service
  - Distributed database clusters for high availability
  - Automatic backups and disaster recovery
  - Scaling capabilities based on data volume

Authentication Infrastructure

- Firebase Services cloud-hosted authentication
  - Multiple authentication servers for redundancy
  - Security rules implementation
  - Real-time database capabilities

Client Devices

- Mobile Applications requirements:
  - iOS devices running iOS 12.0 or later

    - Android devices running Android 6.0 or later
    - Minimum 2GB RAM recommended
    - 100MB minimum storage space

Development Environment

- Version Control through GitHub repositories
- CI/CD Pipeline for automated testing and deployment
- Development Tools including Flutter SDK, Node.js runtime, MongoDB tools

3.4 Persistent Data Management

Envolet implements a robust data management strategy to ensure data consistency, security, and accessibility:

Data Storage Strategy

1. Primary Storage: MongoDB
    - User profiles and preferences
    - Transaction records and categories
    - Budget configurations and limits
    - Analysis results and recommendations
    - Historical spending patterns

Data Schema Organization

- Users Collection:
    - Basic profile information
    - Authentication details
    - Preferences and settings
- Transactions Collection:
    - Transaction details
    - Category mappings
    - Timestamps and metadata
    - Environmental impact indicators
- Budgets Collection:
    - Budget categories
    - Spending limits
    - Progress tracking
    - Historical budget data
- Recommendations Collection:
    - Generated insights
    - Sustainability suggestions
    - User feedback and interactions

Data Access Patterns

- Implementation of caching strategies for frequently accessed data
- Use of database indexing for optimal query performance
- Implementation of data partitioning for scalability
- Regular data backups and recovery procedures

Data Security Measures

- Encryption at rest for sensitive data
- Secure transmission using HTTPS
- Regular security audits and monitoring
- Compliance with data protection regulations

3.5 Access Control and Security

To ensure robust access control and security, Envolet integrates several mechanisms:

- **User Authentication:** Utilizes Firebase Authentication to manage user logins and registrations. Firebase provides a variety of secure authentication methods. This service also supports multifactor authentication, enhancing the security for user access.
- **Data Encryption:** All sensitive data, including personal information and transaction details, is encrypted using state-of-the-art encryption techniques before storage. This ensures that even in the event of a data breach, the information remains protected.
- **Security Protocols:** Implements standard security protocols and best practices, such as HTTPS for secure communication, JWT (JSON Web Tokens) for securely transmitting information between parties.

3.6 Global Software Control

Global software control in Envolet is managed through several key technologies and practices:

- **Node.js and Express.js Framework:** The backend is built on Node.js, using the Express.js framework to handle HTTP requests and streamline interactions between the database and the client-side application. This setup aids in managing the application logic and routing, ensuring smooth data flow.

- **GitHub Integration:** Utilizes GitHub for source code management, which includes version control and collaborative development features. This integration helps in maintaining a single source of truth for the project code, facilitates code reviews, and manages updates through pull requests and merges. Continuous integration and deployment are achieved using GitHub Actions, automating the testing and deployment processes.

3.7 Boundary Conditions

Handling boundary conditions involves defining how the system behaves under various edge cases or unusual circumstances:

- **Input Validation:** Strict input validation is implemented to prevent invalid data from entering the system, which could cause errors or security issues. This includes checks for data types, formats, and ranges for all inputs.

- **Error Handling Strategies:** Comprehensive error handling mechanisms are implemented to catch and respond to exceptions or anomalies. This ensures the application does not crash and can provide meaningful error messages to users, aiding in troubleshooting and enhancing user experience.

4. Subsystem Services

Detailed services offered by each subsystem include:

- **User Authentication Subsystem:** Manages secure user access through registration, login, and session management. Integrates with Firebase to provide a reliable and secure authentication mechanism.

- **Budget Management Subsystem:** Allows users to create, modify, and monitor budgets across various categories. This subsystem tracks budget allocations and expenditures, providing notifications when budget limits are approached or exceeded.

- **Transaction Processing Subsystem:** Captures and processes user transactions in real-time. Ensures that all transactions are accurately recorded and reflected in the user's budget and spending analysis.

- **Spending Analysis Subsystem:** Utilizes AI to analyze spending habits and patterns. Offers insights and actionable recommendations to help users optimize their financial decisions.

- **Recommendation Engine Subsystem:** Delivers personalized advice on saving money and making eco-friendly choices based on the user's spending behavior and preferences.

- **Data Storage and Retrieval Subsystem:** Utilizes MongoDB to manage data storage. This subsystem is optimized for high performance and scalability, ensuring that data retrieval is fast and reliable across all application functions.

5. Glossary

- **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.

- **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.

- **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.