Software Requirements Specification (SRS)

BudgetBuddy - Smart Budgeting Tool

Team Members:

- Hannah Belton
- Cameron Hackett
- Faisal Sayed

Date:

Feb 25, 2025

1. System Requirements

1.1 Functional Requirements

1.1.1 User Account Management

- The system shall allow users to create, manage, and delete accounts securely.
- The system shall support OAuth authentication for third-party logins (Google, Apple, etc.).

1.1.2 Expense Categorization

- The system shall automatically classify transactions into categories such as groceries, dining, rent, and subscriptions.
- Users shall be able to manually adjust the category of any transaction.

1.1.3 AI-Based Expense Prediction

- The system shall predict upcoming expenses based on historical spending patterns.
- Users shall receive notifications about predicted high-spending periods.

1.1.4 Smart Savings Suggestions

- The system shall analyze spending habits and suggest customized savings strategies.
- Users shall be able to set and track personal savings goals.

1.1.5 Bank Integration

- The system shall support integration with major banks via APIs to fetch transaction data.
- Transactions shall be updated in real-time or via scheduled syncs.

1.1.6 Alerts & Notifications

- The system shall send alerts for bill due dates, overdraft risks, and unusual spending patterns.
- Users shall be able to configure notification preferences.

1.1.7 Data Visualization

- The system shall provide interactive charts and graphs showing spending trends and financial insights.

1.2 Non-Functional Requirements

1.2.1 Performance Requirements

- The system shall handle at least 1000 concurrent users.
- The system shall update transaction data within 5 seconds of receiving bank API updates.

1.2.2 Security Requirements

- The system shall protect all sensitive user financial data against unauthorized access, with no security breaches during penetration testing.
- Authentication mechanisms shall prevent at least 99.9% of unauthorized access attempts.
- The system shall detect and alert administrators of suspicious login attempts within 60 seconds.
- All data in transit and at rest shall maintain confidentiality through industry-standard encryption.

1.2.3 Usability Requirements

- The system shall enable 85% of first-time users to complete basic budgeting tasks (adding expenses, categorizing transactions) without assistance within their first 15 minutes of use.
- The system shall maintain a task success rate of at least 90% across all core functions as measured in user testing.
- Error messages shall be understandable to non-technical users, with 80% of users able to successfully recover from errors without support.
- The system shall provide user interface elements with sufficient color contrast (minimum 4.5:1 ratio) to ensure readability for users with visual impairments.

---

2. System Constraints

2.1 Tool Constraints

- Development is limited to the technology stack already established in the organization.
- Third-party libraries and frameworks are restricted to those approved by the organization's security team.

2.2 Language Constraints

- Initial release is limited to English language support due to localization resource constraints.

2.4 Hardware Constraints

- The solution must operate within the organization's existing cloud service infrastructure.
- No additional hardware procurement is permitted within the project timeline.

2.5 Network Constraints

- The application must function effectively under typical consumer internet conditions.
- The solution must comply with organizational network security policies.

2.8 Budget & Schedule Constraints

- Development must be completed within the allocated budget.
- Market release deadline cannot be extended beyond the established timeline.
- Team size and composition is fixed for the duration of the project.

2.9 Regulatory Constraints

- Development must adhere to all applicable financial data regulations.
- The solution must comply with data privacy laws in target markets.

---

3. Requirements Modeling

3.1 Use Case: Automated Expense Categorization

Actors: User, System, Bank API

Flow:

1. User links a bank account.
2. System retrieves transaction data.
3. AI model categorizes transactions.
4. User reviews and modifies category if needed.
5. System updates the user’s financial dashboard.

---

4. Evolutionary Requirements

4.1 Functional Requirements

4.1.1 Future Enhancements

- Investment Tracking: Expanding the platform to include investment analysis and portfolio tracking.
- Cryptocurrency Support: Adding support for tracking crypto assets alongside traditional finances.
- AI-Based Financial Coaching: Implementing a virtual assistant for personalized financial advice.

4.2 Non-Functional Requirements

4.2.1 Future Non-Functional Enhancements

- The system shall improve UI/UX design based on user feedback.
- The system shall implement advanced AI models for more accurate financial predictions.
- Future updates shall optimize system performance to reduce latency.