Team member:

Menghao Hu: developer, Product Owner.

Liying Qin: developer, Scrum Master

Part 1: extended requirements specification documents and description of our choices regarding our tech stack.

1. Analysis Object Model

Objective: Model the key objects and their relationships in the system.

Approach: Use UML (Unified Modeling Language) to create a class diagram that includes the main entities such as User, Transaction, Budget, Report, etc., and their relationships.

2. Dynamic Models

Objective: Represent the behavior of the system through dynamic models.

Dynamic Model 1: User Interaction Flow

Activities: User Registration, Login, Viewing Dashboard, Setting Budget.

Dynamic Model 2: Financial Data Processing

Activities: Importing Transaction Data, Categorizing Transactions, Updating Budget, Generating Reports.

3. Define Design Goals

Usability: Focus on a clean and intuitive UI/UX design.

Security: Implement robust encryption methods and comply with GDPR and other data protection laws.

Scalability: Use scalable cloud services and design databases for efficient data handling.

Maintainability: Adopt a modular architecture and follow coding best practices.

Reliability: Implement redundancy and regular backups to ensure data integrity and availability.

4. Subsystem Decomposition

User Management Subsystem: Implement OAuth for secure authentication and role-based access control.

Financial Data Management Subsystem: Integrate with Plaid or similar services for bank data synchronization.

Reporting and Analysis Subsystem: Use data visualization libraries like D3.js for charts and graphs.

Financial Education and Resources Subsystem: Regularly update content and use user interaction data for personalized content.

Personalized Recommendation Subsystem: Utilize machine learning frameworks like TensorFlow for financial advice algorithms.

Notifications and Alerting Subsystem: Implement real-time alerting using services like Firebase or AWS SNS.

Tech Stack Description

Frontend: javascript

Backend: java

Deployment Diagram

Objective: Illustrate the deployment architecture of the system.

Components: Include web servers, application servers, database servers, and external integrations.

Tools: Use tools like Lucidchart or Draw.io to create the diagram.

A diagram of a computer security

Description automatically generated

Product backlog:

1. **User Authentication and Management:**
   * **Description:** Implement a secure user registration, login, and management system using OAuth.
   * **Priority:** High
   * **Estimation:** 1 week
2. **Bank Data Integration:**
   * **Description:** Integrate with banking APIs like Plaid for real-time transaction data synchronization.
   * **Priority:** High
   * **Estimation:** 1-2 weeks
3. **Budget Management Tool:**
   * **Description:** Develop a tool for users to set, view, and manage personal budgets.
   * **Priority:** High
   * **Estimation:** 2 weeks
4. **Financial Reporting and Analysis:**
   * **Description:** Create a subsystem for generating financial reports and visualizations using D3.js.
   * **Priority:** Medium
   * **Estimation:** 2 weeks
5. **Financial Education Content Module:**
   * **Description:** Develop a module to provide users with updated financial education and resources.
   * **Priority:** Low
   * **Estimation:** 1 week
6. **Personalized Financial Recommendation Engine:**
   * **Description:** Implement a recommendation system using TensorFlow to offer personalized financial advice.
   * **Priority:** Medium
   * **Estimation:** 3 weeks
7. **Notification and Alert System:**
   * **Description:** Develop a system for sending real-time alerts and notifications using Firebase or AWS SNS.
   * **Priority:** Medium
   * **Estimation:** 1 week
8. **User Interface Design:**
   * **Description:** Design a clean and intuitive UI/UX for the application.
   * **Priority:** High
   * **Estimation:** 2-3 weeks
9. **Cloud Infrastructure Setup:**
   * **Description:** Establish cloud services infrastructure for hosting the application.
   * **Priority:** Medium
   * **Estimation:** 1-2 weeks
10. **Data Security and Compliance:**
    * **Description:** Implement robust encryption methods and ensure GDPR compliance.
    * **Priority:** High
    * **Estimation:** 2 weeks
11. **System Scalability and Performance Optimization:**
    * **Description:** Optimize the system for scalability and efficient data handling.
    * **Priority:** Medium
    * **Estimation:** Ongoing
12. **Codebase Maintainability and Documentation:**
    * **Description:** Ensure modular architecture and maintain well-documented, clean code.
    * **Priority:** Medium
    * **Estimation:** Ongoing
13. **Data Backup and Reliability Measures:**
    * **Description:** Implement data redundancy and regular backup systems.
    * **Priority:** Medium
    * **Estimation:** 1 week