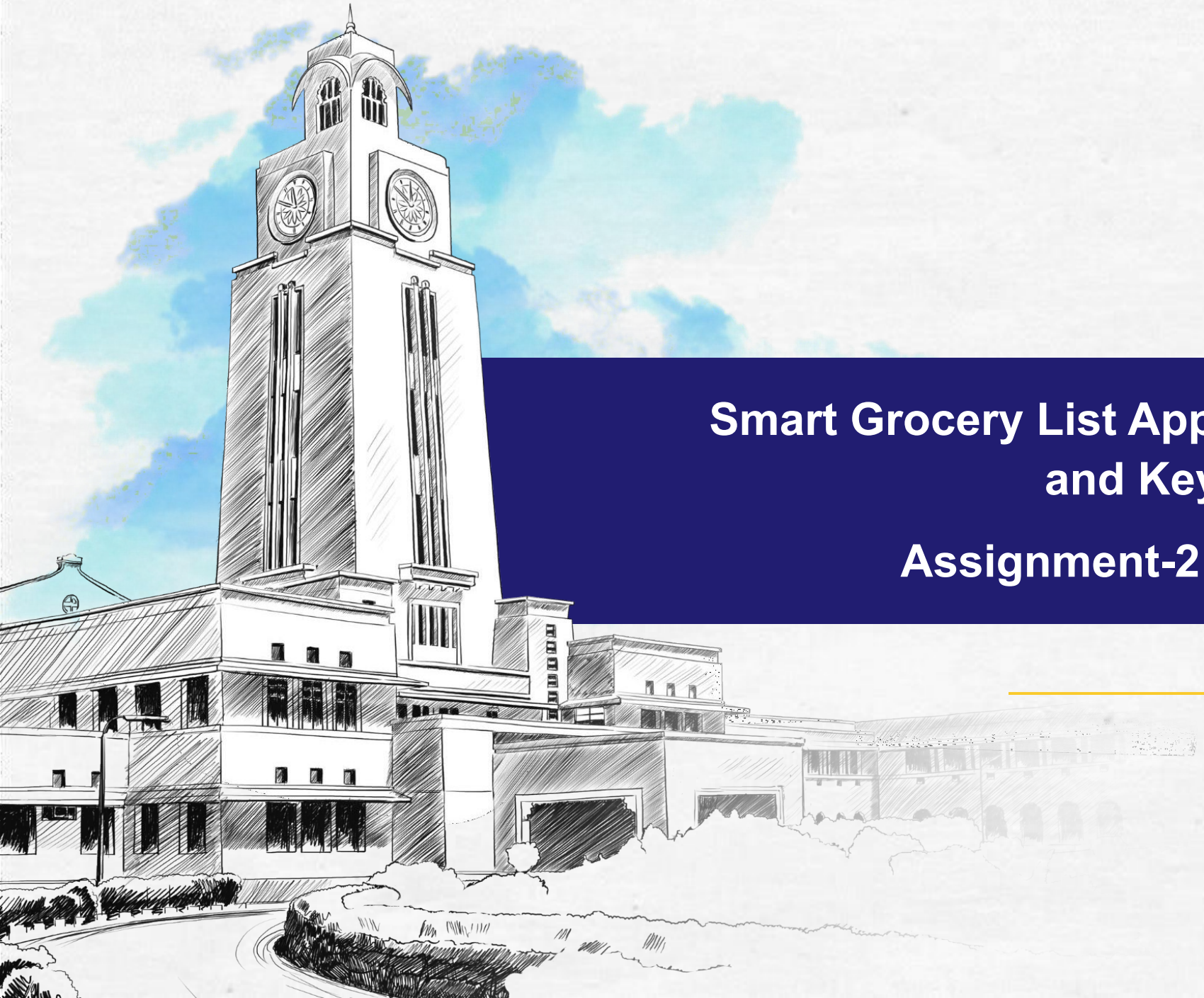


BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Smart Grocery List App: Design, Architecture, and Key Insights

Assignment-2 (SEZG651/SSZG653)

Submitted By:

Kanumuri Sunitha (2024TM93188)

Table Of Contents

Purpose of the Smart Grocery List App

The **Smart Grocery List App** is designed to enhance the grocery shopping experience by providing a personalized, efficient, and user-friendly solution. It aims to:

- **Simplify Grocery Management**: Uses AI to create smart, personalized lists based on user preferences, past purchases, and dietary needs.
- **Optimize Shopping Lists**: Ensures that users only buy what they need, helping to minimize waste and save money.
- **Track Inventory**: Manages home pantry and sends reminders for low-stock items to ensure users never run out of essentials.
- **Enable Real-time Collaboration**: Allows users to share and update lists with family members or roommates, making collaborative shopping seamless.
- **Ensure Data Security**: Protects personal information and shopping preferences with strong security measures.

The app focuses on making grocery management more convenient, tailored, and efficient for users.

Key Functional Requirements of the Smart Grocery List App

Key Functional Requirements of the Smart Grocery List App

The **Smart Grocery List App** offers the following essential features:

1. **User Registration and Profiles**
   - Users can sign up, set preferences, and manage personal profiles.
2. **Personalized Item Suggestions**
   - AI provides recommendations based on past purchases and dietary needs.
3. **Smart List Generation**
   - Creates efficient shopping lists based on user preferences and home inventory.
4. **Inventory Tracking**
   - Helps users track pantry items and sends reminders for low-stock items.
5. **List Sharing**
   - Allows users to share and update lists with others in real-time.
6. **Data Security**
   - Protects user information with encryption and secure access controls.

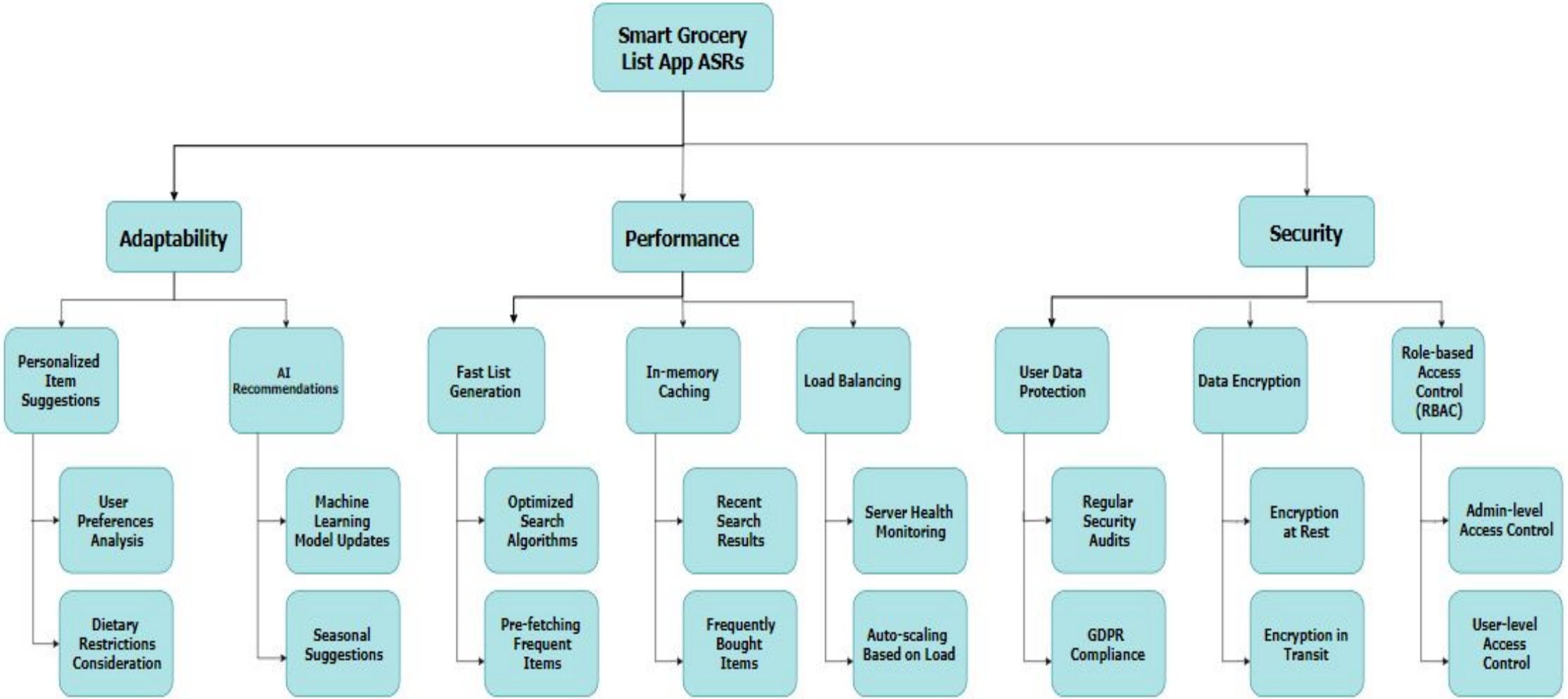
These requirements ensure the app is easy to use, personalized, and secure for managing grocery shopping.

Top 3 Architecturally Significant Requirements (ASRs)

1. **Adaptability: Personalized Item Suggestions**
    - **Significance**: Essential for creating a user-centric experience by tailoring item recommendations based on individual preferences, dietary needs, and past purchases.
2. **Performance: Fast List Generation**
    - **Significance**: Ensures quick response times for generating and updating shopping lists, providing a seamless experience for users during list creation and real-time synchronization.
3. **Security: User Data Protection**
    - **Significance**: Crucial for safeguarding personal information and maintaining user trust by ensuring secure handling of sensitive data, including preferences, shopping history, and profile details.

These ASRs are vital for delivering a personalized, efficient, and secure grocery shopping experience.

Utility Tree of Architecturally Significant Requirements (ASR)



Tactics for ASRs in Smart Grocery List App

1. Adaptability

- **Machine Learning**: Uses AI to analyze preferences for personalized suggestions.
  - Tools: **TensorFlow**, **Scikit-learn**.
- **Modular Design**: Isolates features for easy updates.
  - Tools: **Microservices**.

2. Performance

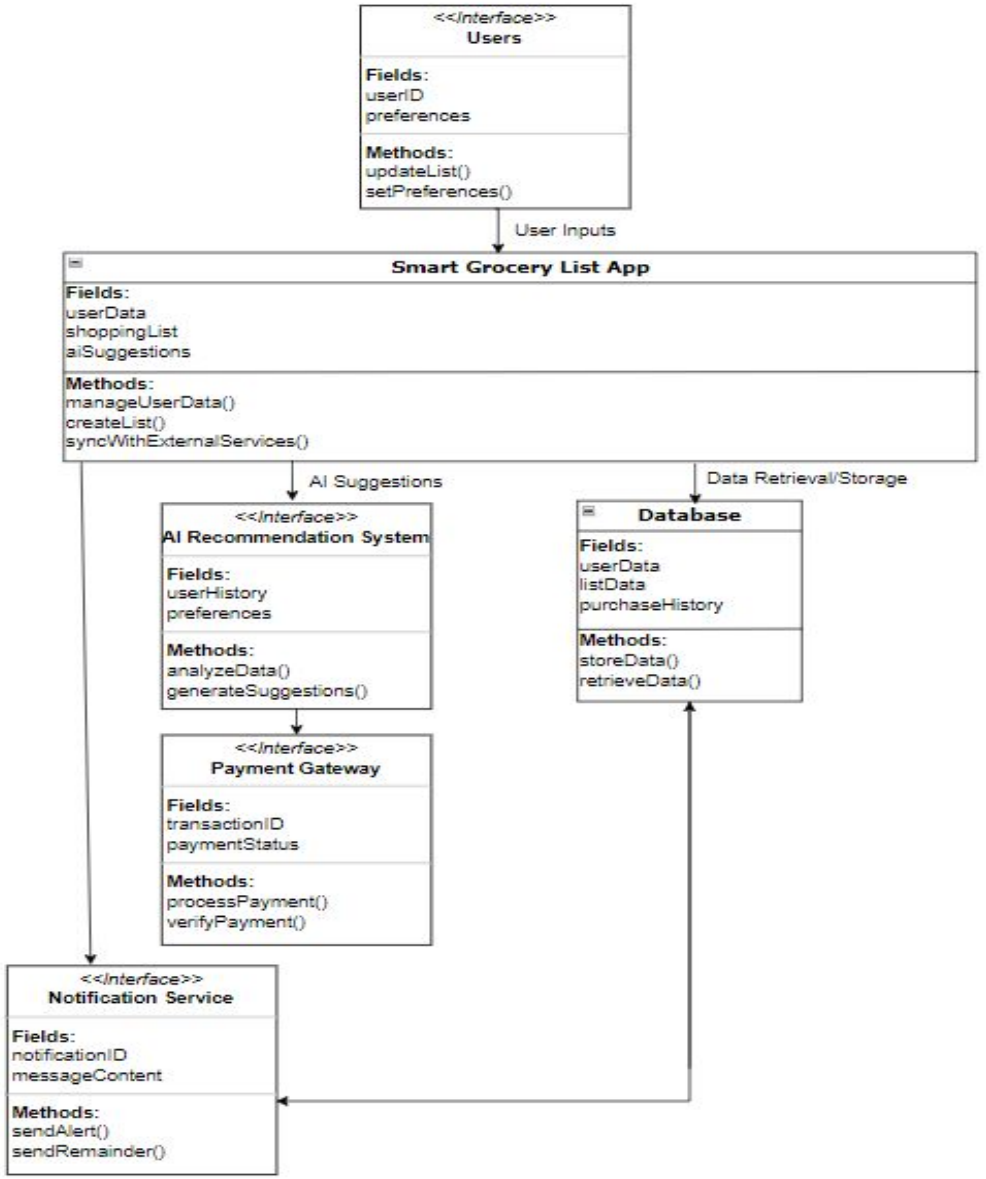
- **In-memory Caching**: Caches frequent items for faster list creation.
  - Tools: **Redis**, **Memcached**.
- **Load Balancing**: Distributes requests across servers.
  - Tools: **NGINX**, **AWS ELB**.

3. Security

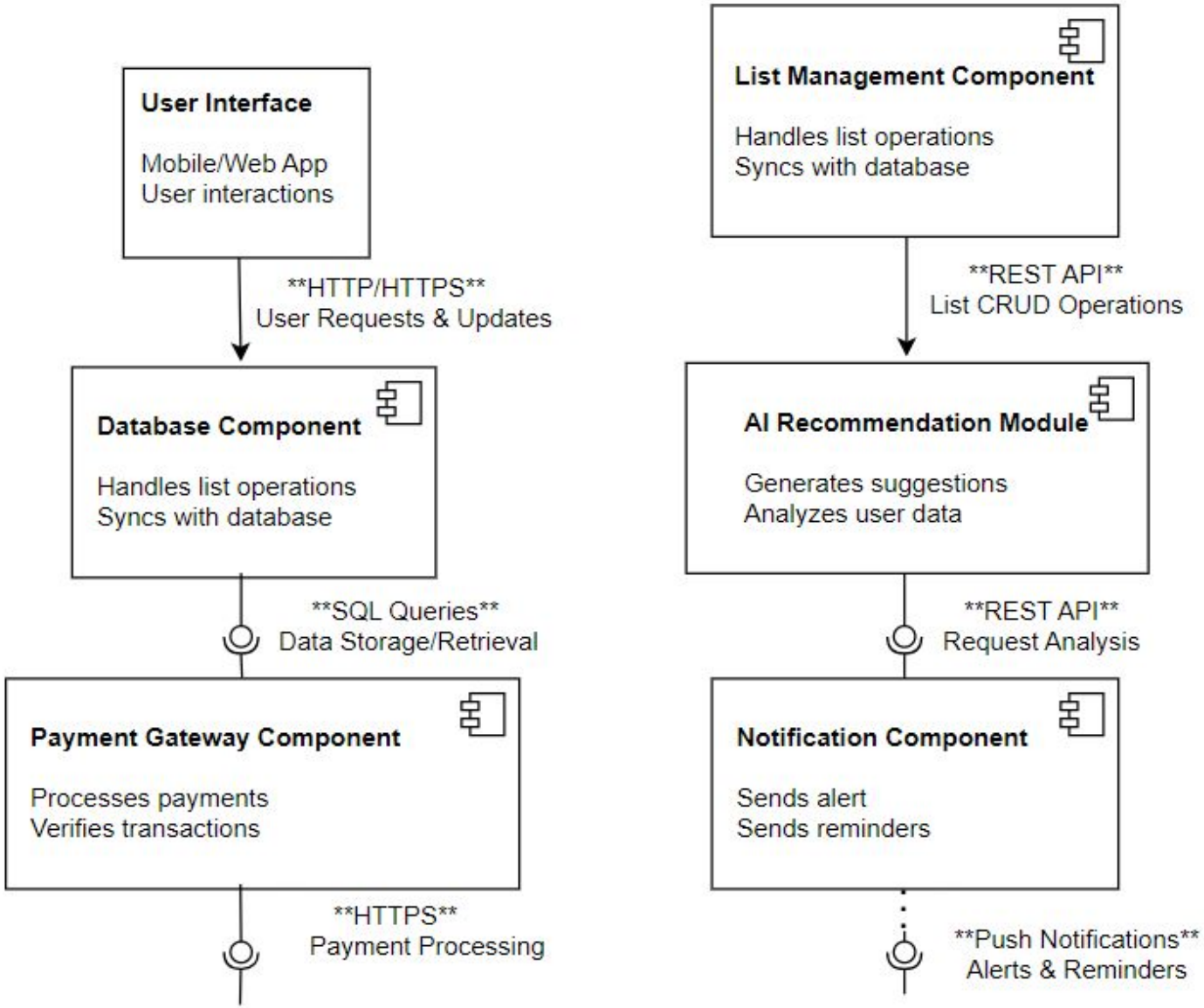
- **Data Encryption**: Encrypts data at rest and in transit.
  - Tools: **AES-256**, **TLS**.
- **RBAC**: Controls access based on user roles.
  - Tools: **Spring Security**, **JWT**.

These tactics ensure adaptability, performance, and security in the app.

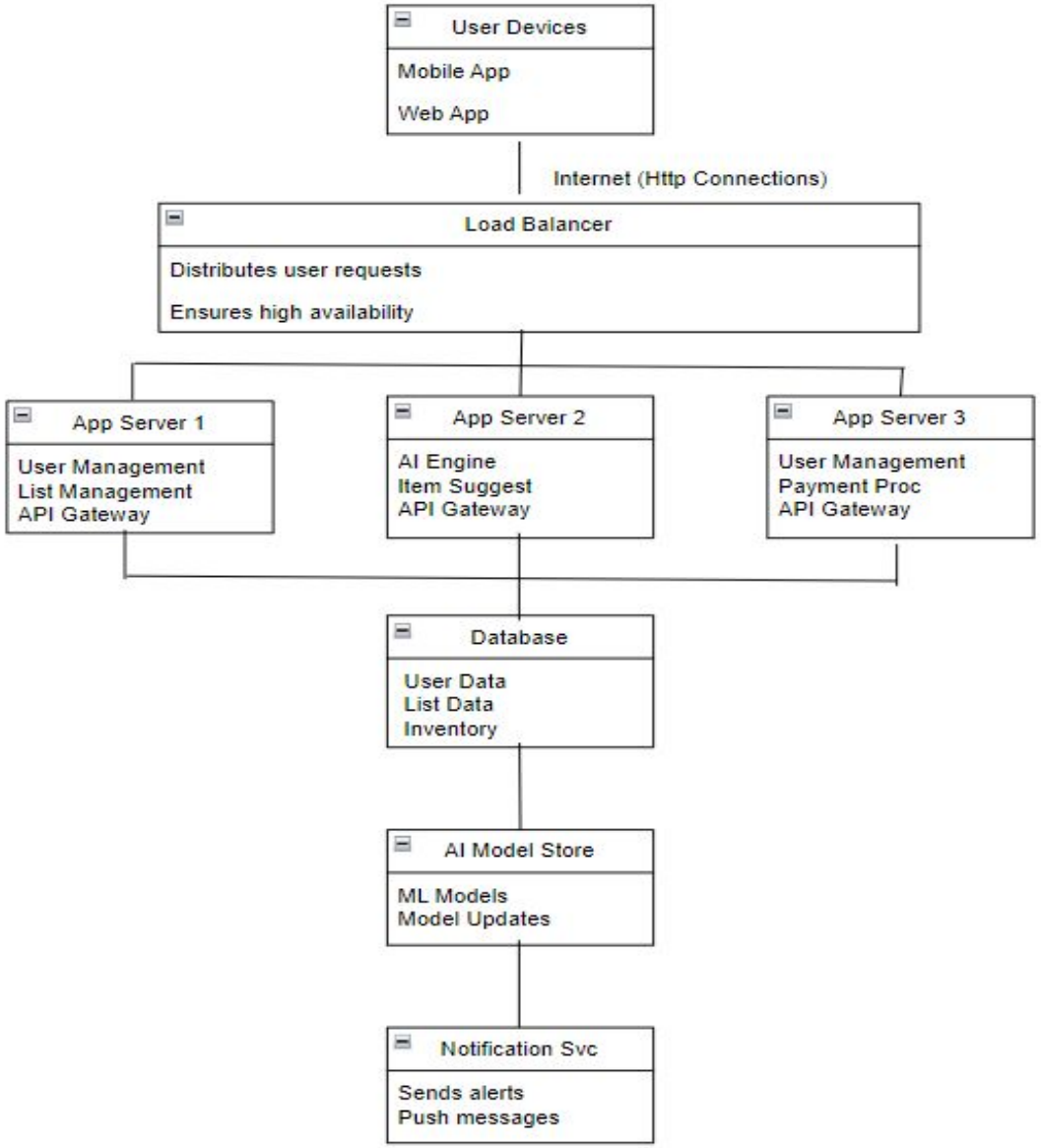
Context view diagram for Smart Grocery List App



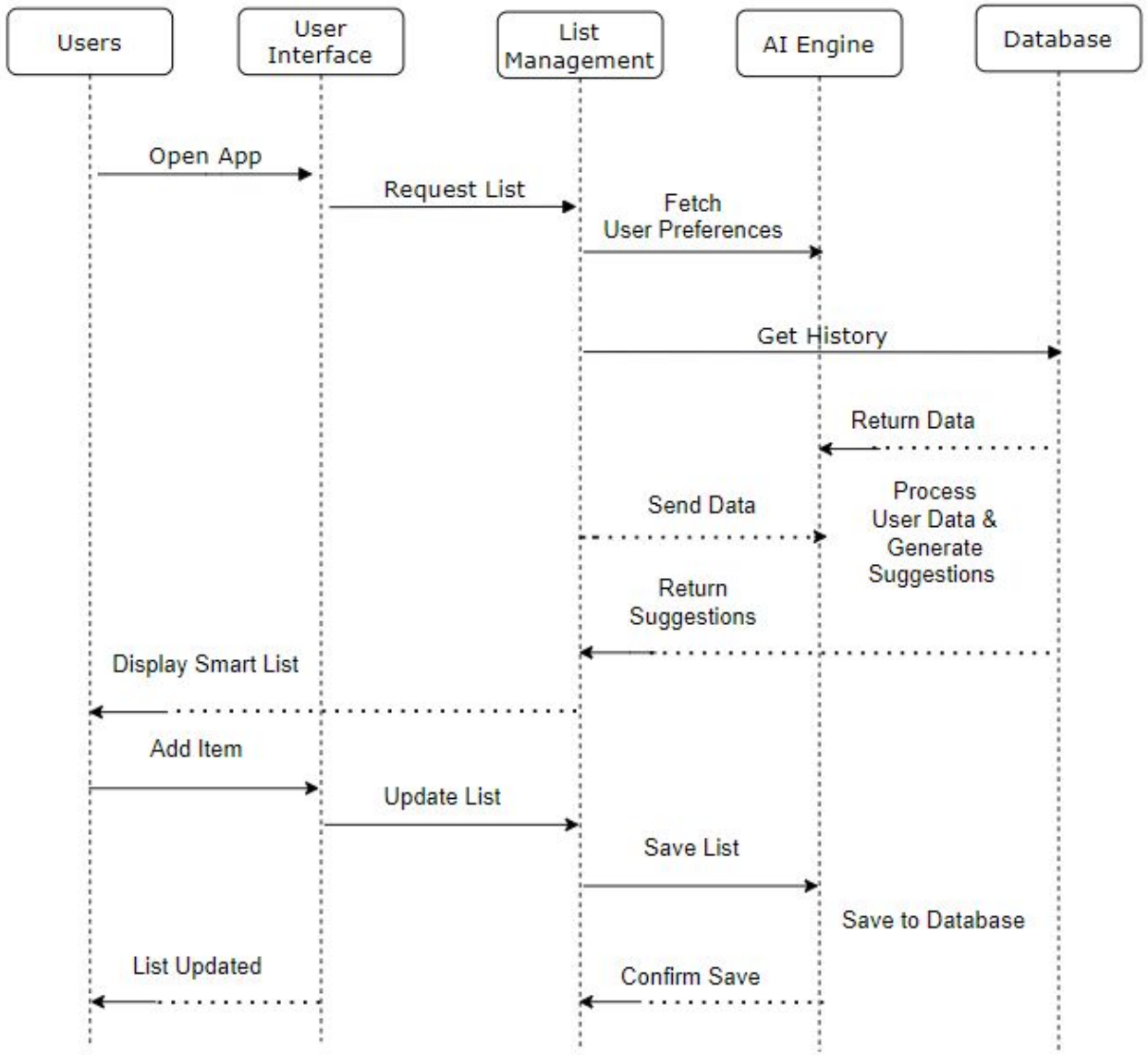
Component & Connector View diagram for the Smart Grocery List App



Deployment view diagram for Smart Grocery List App



Sequence Diagram for the Smart List Creation



Architectural Patterns in Smart Grocery List App

1. **Microservices Architecture**
   - **Used for**: User management, list management, AI recommendations, and notifications.
   - **Benefit**: Allows independent scaling, modular updates, and separate deployment of components, ensuring flexibility and fault isolation.
2. **Event-Driven Architecture**
   - **Used for**: Handling list updates, AI suggestions, and notifications.
   - **Benefit**: Supports real-time processing and asynchronous communication, improving responsiveness and user experience.
3. **Client-Server Pattern**
   - **Used for**: Interaction between the user interface and backend services.
   - **Benefit**: Separates client logic from server processing, enabling easier updates and independent development of front-end and back-end.
4. **Layered Architecture**
   - **Used for**: Organizing presentation, business logic, and data access layers.
   - **Benefit**: Enhances separation of concerns, making the app easier to test, maintain, and scale.

These patterns enable the app to be scalable, responsive, and easy to maintain.

Key Learnings from Smart Grocery List App

- **Designing for Adaptability**
  - Learned how to implement AI-driven personalization for a more user-centric experience, improving adaptability based on user behavior and preferences.
- **Ensuring High Performance**
  - Gained insights into using caching, load balancing, and optimized algorithms to enhance system performance and ensure fast response times.
- **Implementing Strong Security**
  - Understood the importance of data encryption and access control measures to protect user data, maintaining privacy and trust.

These learnings highlight the importance of adaptability, performance, and security in developing scalable and user-friendly applications.

Thank you!