

## Requirements Gathering and Analysis Phase

Date	15 April 2025
Team ID	SWTID1744094910
Project Title	Personal Expense Tracker App
Maximum Marks	2 Marks

### System Requirements

The solution requirements for an Expense Tracker App encompass both functional and non-functional aspects, ensuring the application is robust, user-friendly, secure, and scalable. Below is an in-depth breakdown based on industry best practices and current trends.

#### 1. Functional Requirements

These define what the system should do—its features and capabilities.

##### 1.1 User Management

Registration & Onboarding:

- Simple, multi-step registration process (email, social logins, and optional biometric authentication).
- Smooth onboarding with guided walkthroughs.

Login/Logout:

- Secure login with password, OTP, or biometric options.
- Logout and session management.

##### 1.2 Expense & Income Tracking

Add/Edit/Delete Expenses and Income:

- Users can record, update, or remove transactions with details like amount, date, category, payment mode, and notes.

Receipt Scanning and Upload:

- OCR-enabled receipt scanning for automatic data extraction and categorization.
- Cloud storage for digital receipts.

### 1.3 Categorization and Organization

Customizable Categories:

- Predefined and user-created categories for expenses and income.

Tagging and Notes:

- Add tags or notes to transactions for better organization.

### 1.4 Budgeting and Planning

Budget Setup:

- Set monthly or category-wise budgets.

Budget Tracking and Alerts:

- Real-time notifications and alerts when nearing or exceeding budget limits.

### 1.5 Analytics and Reporting

Dashboard:

- Personalized dashboard showing summaries, trends, and insights.

Graphical Reports:

- Visual representation of spending patterns (pie charts, bar graphs, trend lines).

Export Reports:

- Download reports in PDF/CSV formats.

### 1.6 Account Integration (Optional/Advanced)

Bank/Card Sync:

- Integration with bank accounts, credit/debit cards for automatic transaction import.

Multi-account Support:

- Manage multiple accounts within the app.

### 1.7 Multi-user/Family/Team Support

Shared Accounts:

- Invite family members or team members to collaboratively manage expenses.

Role-based Access:

- Different permissions for admin and regular users.

## 1.8 Notifications & Reminders

Expense Reminders:

- Remind users to log expenses or pay bills.

Budget/Goal Alerts:

- Notify users when approaching limits or achieving goals.

## 1.9 Money Management Tips

Financial Advice:

- AI-driven or rule-based tips for better money management and savings.

## 2. Non-Functional Requirements

These define how the system should behave—its qualities and constraints.

### 2.1 Usability

Mobile-Friendly & Intuitive UI:

- Responsive design for mobile and web.
- Clean, easy-to-navigate interface with clear calls to action.

Accessibility:

- Support for users with disabilities.

### 2.2 Security

Data Encryption:

- Encrypt sensitive data in transit and at rest.

Authentication & Authorization:

- Secure login, role-based access, and session management.

Compliance:

- Adherence to data privacy regulations (GDPR, etc.).

## 2.3 Reliability & Performance

### Availability:

- High uptime and quick response times.

### Scalability:

- Ability to handle increasing users and transactions without performance degradation.

### Backup & Recovery:

- Regular backups and disaster recovery plans.

## 2.4 Maintainability & Extensibility

### Modular Architecture:

- Easy to add new features or integrate with third-party services.

### Documentation:

- Comprehensive user and technical documentation.

## 2.5 Integration

### Third-party APIs:

- Support for integrating with banks, payment gateways, and analytics tools.

## 3. Technical Requirements

### Platform Support:

- Web and/or mobile (native or cross-platform, e.g., React Native, Flutter).

### Cloud Infrastructure:

- Secure cloud storage for data and receipts.

### Technology Stack:

- Modern frontend and backend frameworks (e.g., React, Node.js, MongoDB).

#### 4. Business & Operational Requirements

Cost-effectiveness:

- Efficient use of resources for development and maintenance.

Revenue Model:

- Freemium, subscription, or ad-supported options.

Support & Maintenance:

- Ongoing technical support and regular updates.

#### Summary Table

Requirement Type	Key Points
Functional	Registration, expense/income tracking, receipt scanning, categorization, budgeting, analytics
Non-Functional	Usability, security, reliability, scalability, maintainability
Technical	Platform support, cloud storage, modern tech stack, API integration
Business/Operational	Cost-effective, scalable, support, clear revenue model