Use Cases

for

Unified Smart Wallet with Budgeting and Insights

Version 1.0 approved

Prepared by: Taha Alabyad, Majd Eddin Alraie, Abdaladeem Zahaf

Horizon

5 December 2024

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

# Use Case Identification

## Use Case ID

**UC-1**

## Use Case Name

Manage Monthly Budget

## Use Case History

### Created By

Taha Alabyad

### Date Created

05 December 2024

### Last Updated By

Taha Alabyad

### Date Last Updated

20 December 2024

# Use Case Definition

## Actor

User

## Description

This use case makes it straightforward for the user to come up with a monthly financial plan in terms of a total budget as well as various expense categories. The app assists the user to track his or her spending and inform the user when certain expenses are in the process of reaching or have already reached certain limits that the user has set.

## Preconditions

1. The user has to be signed in to the system.
2. The user must have the permission to use the Budget Management feature.

## Postconditions

1. The monthly budget is successfully saved.
2. Notifications are triggered when expenses approach the set budget limits.

## Priority

High

## Frequency of Use

Once per month or as needed.

## Normal Course of Events

1. The user logs into the app.
2. The user navigates to the “Budget Management” feature.
3. The system prompts the user to enter a budget amount.
4. The user selects budget categories (e.g., Food, Transport).
5. The system confirms the budget setup and activates notifications.

## Alternative Courses

**UC-1.AC.1:** If the user skips category assignment, the system defaults to general categories.

## Exceptions

**UC-1. EX.1:** If the system fails to store the budget as a result of a connection problem, an error message appears.

## Includes

Budget Management

Notifications

Transaction Tracking

## Special Requirements

The system must store budget data using Java OOP classes like Budget, Category, and UserAccount.

## Assumptions

The user has a stable internet connection.

## Notes and Issues

It is recommended to include future enhancement options that can be incorporated for allowing the users to set up their personal notifications and alerts.

Use Case Template

**Use Case 1: Manage Monthly Budget**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-1** | | |
| Use Case Name: | Manage Monthly Budget | | |
| Created By: | Taha Alabyad | Last Updated By: |  |
| Date Created: | 5 December 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: |  |
| Preconditions: | 1. The user has to be signed in to the system. 2. The user must have the permission to use the Budget Management feature. |
| Postconditions: | 1. The monthly budget is successfully saved. 2. Notifications are triggered when expenses approach the set budget limits. |
| Priority: | High |
| Frequency of Use: | Once per month or as needed. |
| Normal Course of Events: | 1. The user logs into the app. 2. The user navigates to the “Budget Management” feature. 3. The system prompts the user to enter a budget amount. 4. The user selects budget categories (e.g., Food, Transport). 5. The system confirms the budget setup and activates notifications. |
| Alternative Courses: | If the user skips category assignment, the system defaults to general categories. |
| Exceptions: | If the system fails to store the budget as a result of a connection problem, an error message appears. |
| Includes: | Budget Management  Notifications  Transaction Tracking |
| Special Requirements: | The system must store budget data using Java OOP classes like Budget, Category, and UserAccount. |
| Assumptions: | The user has a stable internet connection. |
| Notes and Issues: | It is recommended to include future enhancement options that can be incorporated for allowing the users to set up their personal notifications and alerts. |

**Use Case 2: Transaction Tracking**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-2** | | |
| Use Case Name: | Track Transactions | | |
| Created By: | Taha Alabyad | Last Updated By: |  |
| Date Created: | 20 December 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Registered User |
| Description: | Automatically tracks and categorizes expenses from connected wallets. |
| Preconditions: | User must link a wallet. |
| Postconditions: | Transaction data successfully saved and categorized. |
| Priority: | High |
| Frequency of Use: | Daily |
| Normal Course of Events: | 1. System retrieves transactions. 2. Categorizes expenses. 3. Stores transaction details. |
| Alternative Courses: | Transactions manually entered if no internet is available. |
| Exceptions: | API connection failure. |
| Includes: | Wallet Integration, Notifications |
| Special Requirements: | PCI DSS compliance for secure data transfer. |
| Assumptions: | Wallet API services are operational. |
| Notes and Issues: | Add custom categories in future updates. |

**Use Case 3: Spending Insights Report**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-3** | | |
| Use Case Name: | Generate Spending Report | | |
| Created By: | Taha Alabyad | Last Updated By: |  |
| Date Created: | 20 December 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Registered User |
| Description: | Generates detailed reports on monthly spending, grouped by category. |
| Preconditions: | User must have transactions logged. |
| Postconditions: | A spending report is generated and available for download. |
| Priority: | Medium |
| Frequency of Use: | Monthly |
| Normal Course of Events: | 1. User requests a spending report. 2. System generates the report. 3. Report is available for viewing or download. |
| Alternative Courses: | None |
| Exceptions: | Data server downtime may delay report generation. |
| Includes: | Transaction Tracking, Notifications |
| Special Requirements: | PDF export capabilities. |
| Assumptions: | User has an active internet connection. |
| Notes and Issues: | Consider adding CSV export functionality. |

**Use Case 4: Wallet Integration**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-4** | | |
| Use Case Name: | Connect Wallet | | |
| Created By: | Taha Alabyad | Last Updated By: |  |
| Date Created: | 20 December 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Registered User |
| Description: | Connects an external wallet like Google Wallet or Apple Wallet. |
| Preconditions: | User must verify identity. |
| Postconditions: | Wallet successfully linked. |
| Priority: | High |
| Frequency of Use: | One-time or as needed. |
| Normal Course of Events: | 1. User logs in. 2. Selects “Link Wallet.” 3. Enters wallet credentials. 4. System verifies and links the wallet. |
| Alternative Courses: | Re-attempt linking on failure. |
| Exceptions: | Incorrect wallet credentials. |
| Includes: | Transaction Syncing |
| Special Requirements: | Secure wallet API integration. |
| Assumptions: | Wallet service must be operational. |
| Notes and Issues: | Add support for more wallet providers. |

**Use Case 5: Send Money**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-5** | | |
| Use Case Name: | Send Money | | |
| Created By: | Taha Alabyad | Last Updated By: |  |
| Date Created: | 20 December 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Registered User |
| Description: | Allows users to send money to other wallet users securely. |
| Preconditions: | The user must be logged in, with a verified wallet and sufficient funds. |
| Postconditions: | Funds successfully transferred and the transaction recorded. |
| Priority: | High |
| Frequency of Use: | As needed. |
| Normal Course of Events: | 1. User logs in. 2. Selects “Send Money” 3. Enters recipient details and amount. 4. System processes the transfer and displays confirmation. |
| Alternative Courses: | If recipient details are invalid, prompt for correction. |
| Exceptions: | Insufficient balance, incorrect wallet ID, or failed authentication. |
| Includes: | Money management, notifications |
| Special Requirements: | AES-256 encryption, secure APIs for transactions. |
| Assumptions: | Wallet services are operational. |
| Notes and Issues: | Add recurring transfers as a future feature. |

**Use Case 6: Send Money**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-6** | | |
| Use Case Name: | Receive Money | | |
| Created By: | Taha Alabyad | Last Updated By: |  |
| Date Created: | 20 December 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Registered User |
| Description: | Allows users to receive money from other wallet users or financial accounts. |
| Preconditions: | The user must have a linked and verified wallet. |
| Postconditions: | Funds successfully received and reflected in the wallet balance. |
| Priority: | High |
| Frequency of Use: | As needed. |
| Normal Course of Events: | 1. Sender initiates payment. 2. System verifies recipient details. 3. Funds are credited to the recipient’s wallet. 4. Notification is sent to the recipient. |
| Alternative Courses: | System prompts for verification if wallet linking is incomplete. |
| Exceptions: | Incorrect recipient details or technical failure. |
| Includes: | Notifications, Transaction History |
| Special Requirements: | Secure wallet linking protocols. |
| Assumptions: | The sending system is operational. |
| Notes and Issues: | Add payment request functionality in future updates. |