**Finance Management App**

**Project Overview:**

The Finance Management App is designed to empower families, individuals, and small businesses to take control of their finances. This app offers an intuitive platform for managing income, tracking expenses, setting financial goals, and accessing expert financial advice. By providing tools to create realistic budgets and monitor spending, the app helps users achieve financial stability and reduce stress**.**

**Problem Statement:**

Many families and small businesses struggle with financial planning due to inadequate tools and lack of guidance. This leads to overspending, poor savings habits, and difficulty in reaching financial goals. A comprehensive and user-friendly budget planner is essential to bridge this gap, offering a structured approach to financial management and informed decision-making.

**Goals:**

* Financial Empowerment:

Empower families/small businesses to take control of their finances and make informed decisions.

* Improved Financial Health:

Help families/small businesses achieve financial stability and reach their long-term financial goals.

* Reduced Financial Stress:

Alleviate financial stress by providing a clear overview of income and expenses.

**Objectives:**

1. Input Income resources
2. Show categories for expenses (fixed)
3. User can also add their desired categories
4. Specific financial goals with desired amount and deadline
5. Guide users in creating a realistic budget based on their income
6. Notifications when users are nearing their budget limits
7. Articles, tips, and tutorials on budgeting, saving, and investing
8. Allow users to add custom expense categories based on their specific needs.

**Customers:**

The app primarily targets the following customer groups:

* Families: For household expense management and goal-oriented savings.
* Individuals: To facilitate personal budgeting, saving, and financial health tracking.
* Small Businesses: To monitor business expenses, income streams, and profitability.

**Stakeholders:**

Key stakeholders include:

* Families: Users managing shared household finances.
* Individuals: Young professionals, students, and retirees seeking personal financial control.
* Small Companies: Entrepreneurs and small business owners tracking business budgets.
* Financial Advisors/Experts: Professionals offering articles, tutorials, and tips integrated into the app.
* Developers: Responsible for building and maintaining the app, ensuring functionality and user satisfaction.
* Investors: Supporting the app's development and expansion to new features and markets.

**Tools:**

Flutter  
 UiZard

**Scope:**

The scope of a Finance Management App for families, individuals, and small companies focuses on helping users achieve financial stability and make informed budgeting decisions.

**Requirements Analysis**

**F.R. 1: Create Account:** Users should be able to register by entering a valid email address or phone number and setting a password. An email/phone verification mechanism should be in place.

**F.R. 2: Login:** Registered users can log in using their email or phone number and password.

**F.R.3: Third-Party Login:** Integrate third-party authentication options (e.g., Google, Facebook) for a faster sign-in process.

**F.R. 4: Forgot Password:** Users can reset their password via an email or phone verification process.

**F.R. 5: Update Profile:** Users can update their name, contact information, and profile picture.

**F.R. 6: Account Security:** Include two-factor authentication (2FA) for secure login.

**F.R. 7: Add Income Sources:** Input various income sources (e.g., salary, freelance, rental income) with details such as amount, frequency, and source type.

**F.R. 8: Edit Records:** Users can modify income entries if needed.

**F.R. 9: Delete Records:** Users can delete income entries if needed.

**F.R. 10: Categorization:** Predefined categories such as rent, utilities, groceries, and entertainment. Users can add custom categories.

**F.R. 11: Expense Entry:** Log expenses with details like amount, date, and category.

**F.R. 12: Delete Expense:** Users can delete expense.

**F.R. 13: Edit Expense:** Users can edit expense.

**F.R. 14: Goal Creation:** Users can set specific financial goals (e.g., saving for a vacation, debt repayment) by entering the target amount and deadline.

**F.R. 15: Delete goal:** Users can delete goals.

**F.R. 16: Edit goal:** Users can edit goals.

**F.R. 17: Create budget:** System create budget for the user.

**F.R. 18: View articles:** User view articles and watch videos for effective tips.

**F.R. 19: Bookmark sources:** User can bookmark articles/videos.

**F.R. 20: Access bookmarks:** Users can access bookmarks.

**F.R. 21: Goal Reminders:** Send periodic updates on goal progress.

**F.R. 22: Detailed Reports:** Generate detailed reports for selected timeframes, showing income, expense trends, and budget performance.

**F.R. 23: Export Reports:** User exports reports for external use (e.g., PDF).

**F.R. 24: Upload Articles/videos:** User can upload articles and watch videos for effective tips on finance management.

**N.F.R 1:** The system must handle up to 10,000 concurrent users with a response time of <2 seconds for core operations like login, adding income, or viewing reports.

**N.F.R 2:** The system should scale seamlessly to accommodate an increase in users, transactions, and data volume.

**N.F.R 3:** User data must be encrypted using AES-256 for data at rest and HTTPS/TLS for data in transit.

**N.F.R 4:** All data changes (e.g., adding income, expenses) should reflect across all devices in real time with consistent backups to ensure no data loss during failures.

**N.F.R 5:** The UI should be intuitive and responsive on desktop and mobile devices. Users should be able to perform core tasks (e.g., adding income/expenses or updating profiles) within 3 clicks or fewer.

**Actors**

* **Families:** To manage shared household budgets.
* **Individuals:** To track personal financial health and savings.
* **Small Businesses:** To monitor operational income and expenses.
* **System Administrator:** Manages app operations, data security, and user issues.
* **Financial Advisors:** Provide expert content such as articles and tutorials.
* **Investors:** Back the development and feature expansion of the app.

**Use Case List**

**1. User Login and Registration**

* **Register a New Account:** User creates a new account using email or phone number.
* **Log In to Account:** User logs in using valid credentials.
* **Third-Party Authentication:** User logs in with Google or Facebook.
* **Password Recovery:** User resets their password via email or phone verification.

**2. User Account Management**

* **Update Profile Information:** User updates name, email, or contact information.
* **Set Preferences:** User customizes notification and display settings.
* **Enable Two-Factor Authentication:** User activates enhanced security for their account.

**3. Income Management**

* **Add` Income Source:** User logs income details such as source, amount, and frequency.
* **Edit Income Source:** User modifies an existing income record.
* **Delete Income Source:** User removes an income entry.
* **View Income Trends:** User visualizes income through charts or graphs.

**4. Expense Tracking**

* **Log an Expense:** User adds expense details with category and amount.
* **Edit Expense Entry:** User modifies an expense record.
* **Delete Expense Entry:** User removes an expense.
* **View Expense Trends:** User analyzes expenses by category and timeframe.
* **Create Custom Expense Category:** User adds personalized categories for expenses.

**5. Financial Goals**

* **Create a Financial Goal:** User sets a savings or financial target.
* **Track Goal Progress:** User monitors goal achievement.
* **Edit Financial Goal:** User updates target amount or deadline.
* **Delete Financial Goal:** User removes a goal.

**6. Educational Content**

* **Access Articles:** User views articles on financial management.
* **Watch Tutorials:** User watches interactive financial tutorials.
* **Search Educational Materials:** User searches and bookmarks resources.

**7. Reminders and Notifications**

* **Receive Expense Alerts:** System sends notifications for overdue expenses.
* **Receive Goal Reminders:** System reminds users about their financial goals.

**8. Visual Reports**

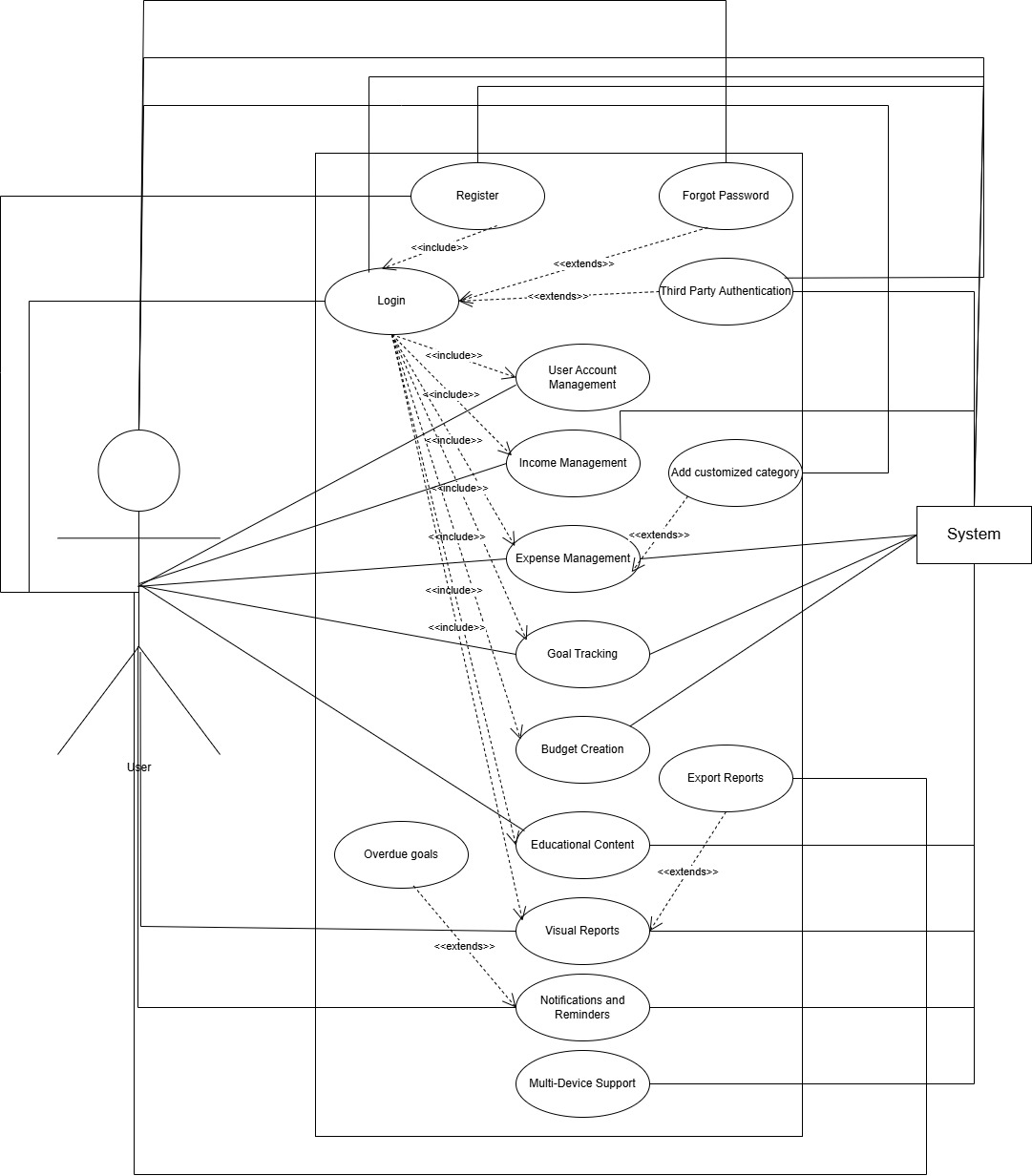
* **View Dashboard:** User sees a summary of key metrics.
* **Generate Reports:** User creates detailed financial reports.
* **Export Reports:** User exports reports for external use (e.g., PDF).

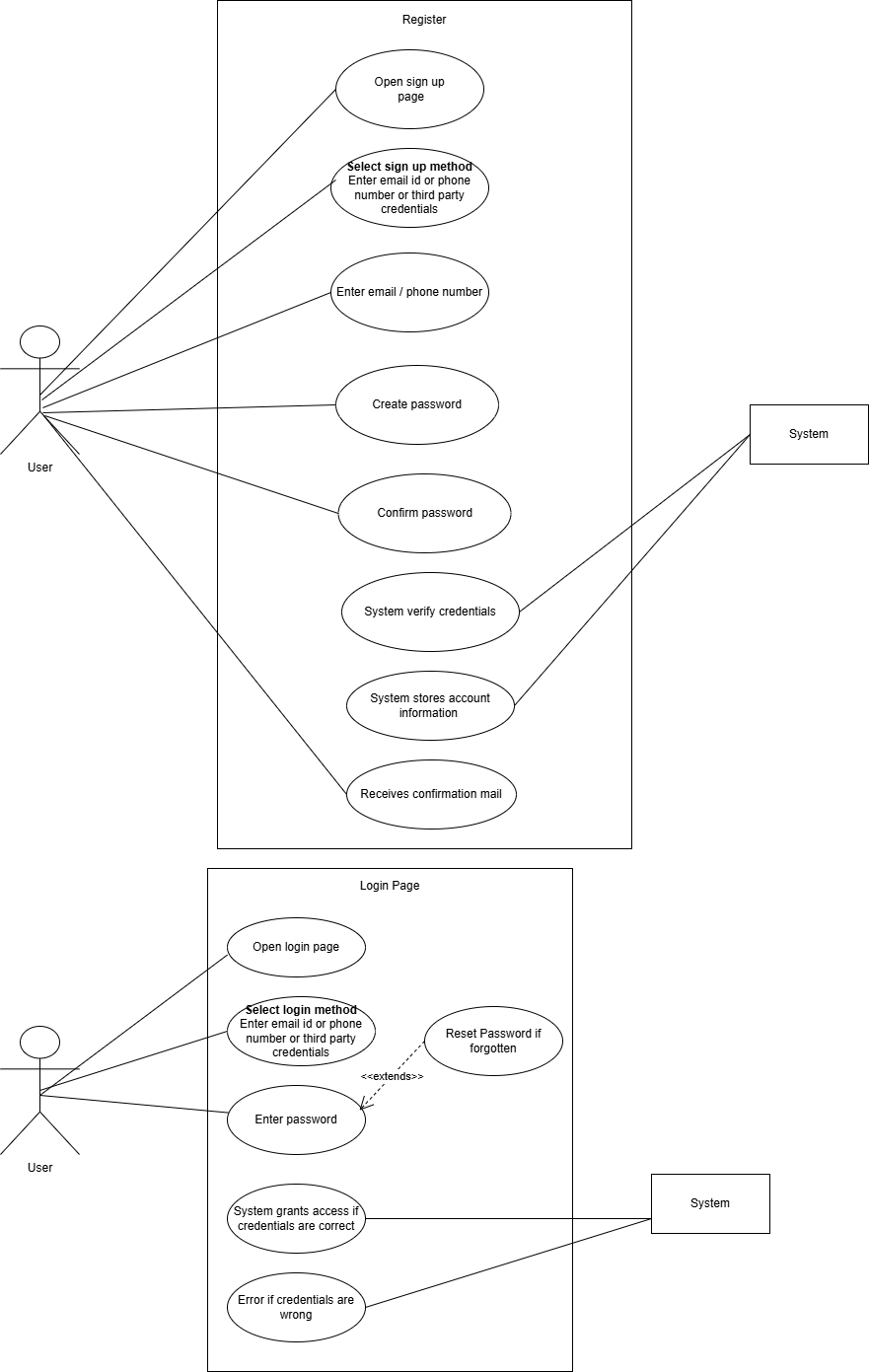
**9. Multi-Device Support**

* **Sync Data Across Devices:** User accesses synchronized financial data across devices.
* **Backup and Restore Data:** User restores app data from the cloud.

**10. Budget Creation**

* System analyze the income sources, goals and expense categories and create budget and notifies the user.

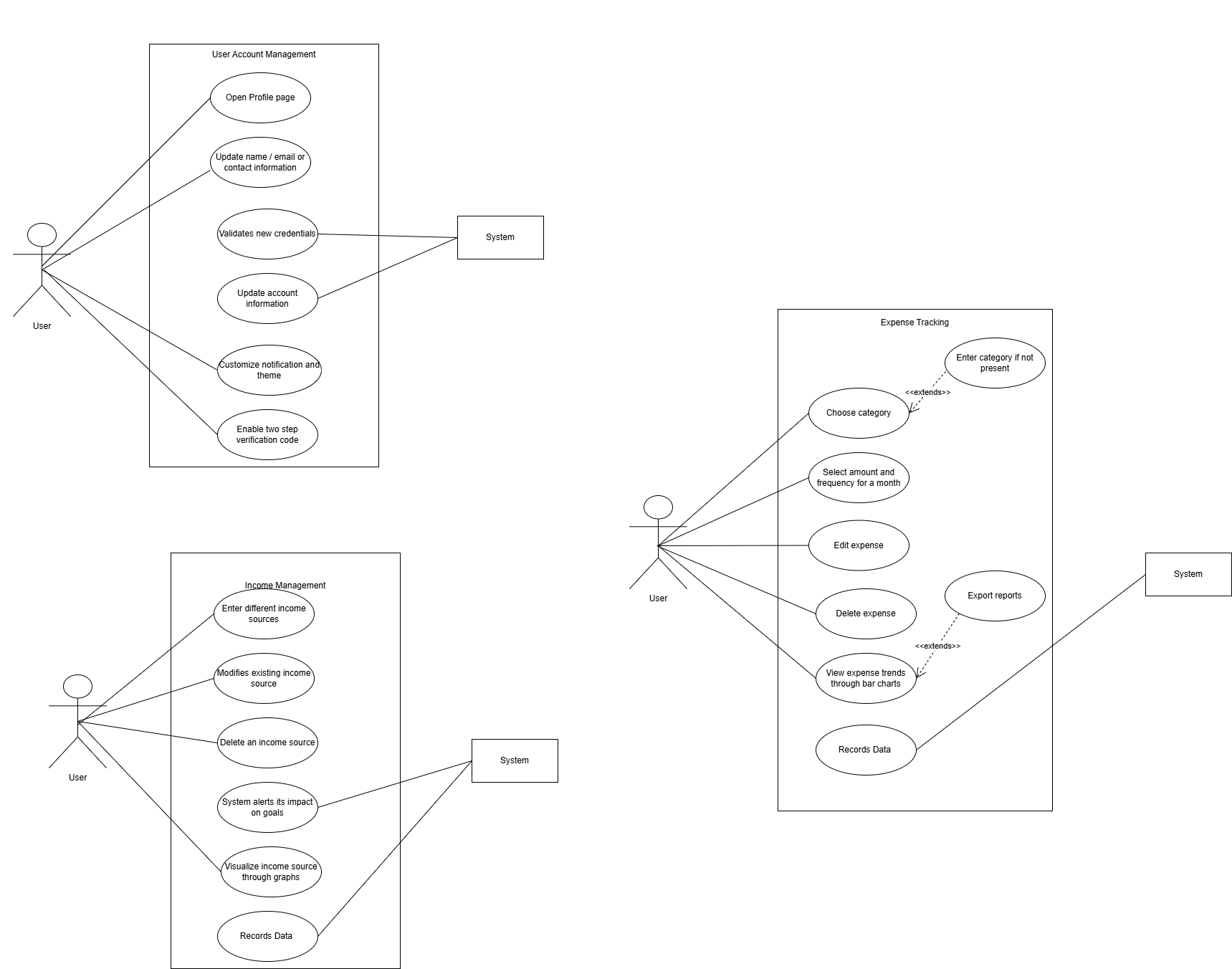




**Use Case 1: Login**

|  |  |
| --- | --- |
| **Section** | **Details** |
| **Use Case Name** | Login |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User, System |
| **Description** | Allows a user to authenticate and gain access to the system. |
| **Trigger** | User enters login credentials and submits the login form. |
| **Preconditions** | User must be registered and have valid credentials. |
| **Postconditions** | User is authenticated and redirected to their dashboard. |
| **Normal Flow** | 1. User enters their username and password.  2. System validates the credentials.  3. If valid, user is logged in and redirected. |
| **Alternative Flows** | 1A. User clicks on "Forgot Password" to recover credentials.  2A. System sends a password recovery email. |
| **Exceptions** | 1. Invalid credentials.  2. Network issues prevent login. |
| **Includes** | "Forgot Password" use case. |
| **Frequency of Use** | Multiple times per day. |
| **Special Requirements** | Secure connection (HTTPS), CAPTCHA to prevent bots. |
| **Assumptions** | System is operational, and the database is accessible. |
| **Notes and Issues** | May require multi-factor authentication for sensitive systems. |
| **Use Case ID** | UC-1 |

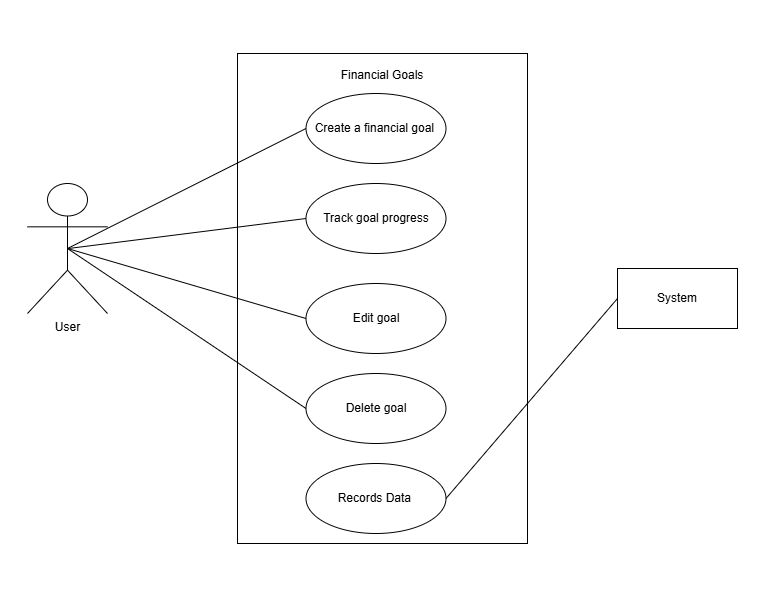
|  |  |
| --- | --- |
| **Section** | **Details** |
| **Use Case Name** | Register |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User, System |
| **Description** | Allows a new user to create an account in the system. |
| **Trigger** | User accesses the registration page and submits the form. |
| **Preconditions** | User must provide all mandatory information (e.g., email, password). |
| **Postconditions** | A new user account is created, and the user is logged in or redirected to the login page. |
| **Normal Flow** | 1. User fills in the registration form with valid details.  2. System validates the input.  3. If valid, system creates the user account.  4. User is notified of successful registration. |
| **Alternative Flows** | 1A. User enters an already registered email address.  2A. System displays an error message and suggests logging in instead. |
| **Exceptions** | 1. Network issues prevent registration.  2. System encounters a server error during account creation. |
| **Includes** | Email verification process. |
| **Frequency of Use** | Once per user, initially. |
| **Special Requirements** | Validation of email address format, strong password enforcement. |
| **Assumptions** | User has internet access and a valid email address. |
| **Notes and Issues** | Ensure data protection regulations are met (e.g., GDPR). |
| **Use Case ID** | UC-2 |



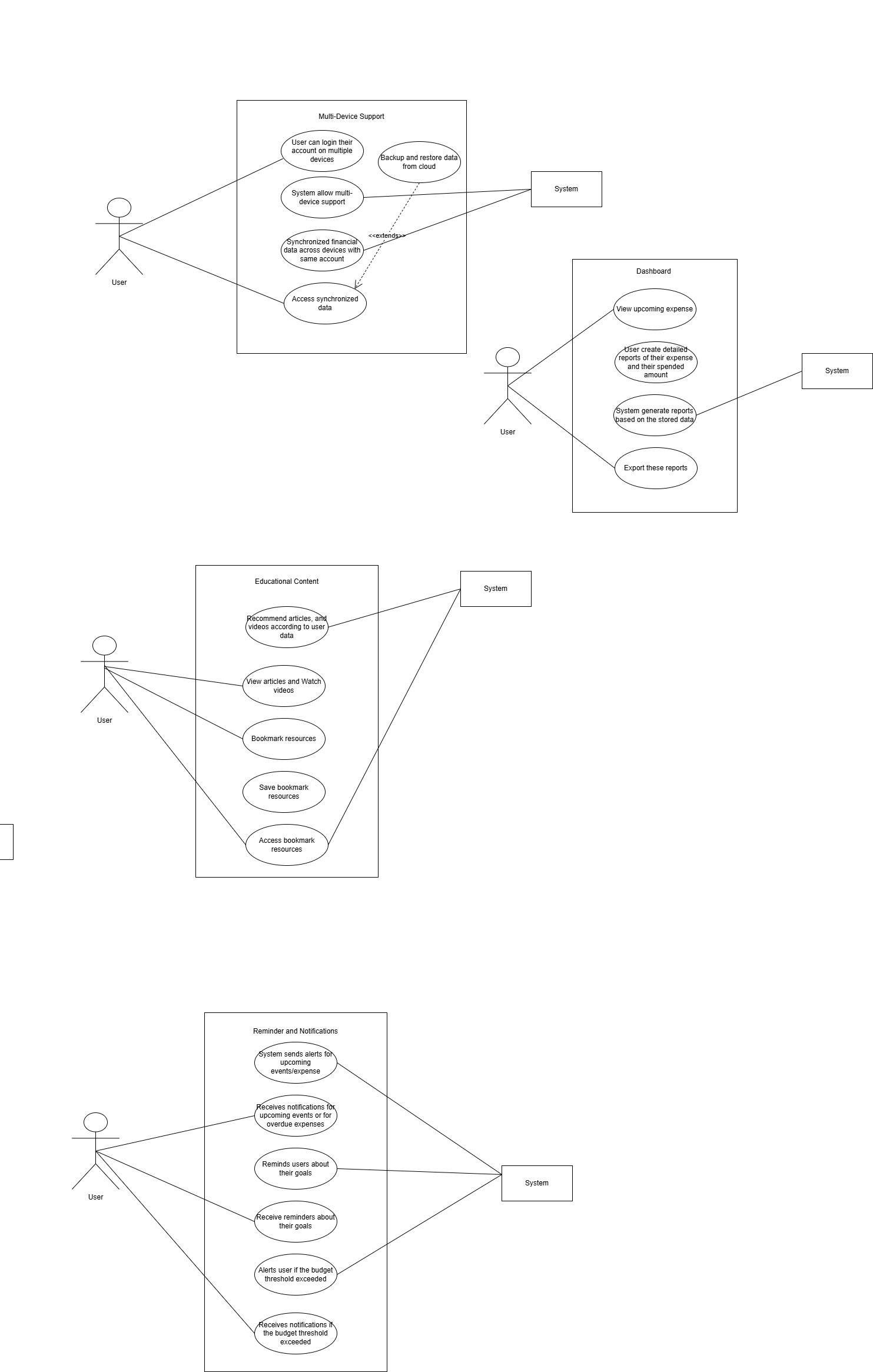
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Section** | **Details** | | **Use Case Name** | User Account Management | | **Created By** | Rabia | | **Date Created** | 1st January 2025 | | **Last Updated By** | Rabia | | **Last Revision Date** | 1st January 2025 | | **Actors** | User | | **Description** | Allows users to manage their account by updating profile details, setting preferences, and enabling two-factor authentication. | | **Trigger** | User selects the "Account Settings" option from the menu. | | **Preconditions** | User is logged in and has access to the account settings feature. | | **Postconditions** | Profile information and preferences are updated, credentials are validated, and security is enhanced if two-factor authentication is enabled. | | **Normal Flow** | 1. User accesses account settings. | |  | 2. User updates profile information (name, email, contact). | |  | 3. System validates the new credentials. | |  | 4. User customizes notification and display preferences. | |  | 5. User enables two-factor authentication. | |  | 6. System saves changes and updates the account. | | **Alternative Flows** | - User skips enabling two-factor authentication and updates only profile or preferences. | |  | - User reverts changes before saving. | | **Exceptions** | - Invalid email or contact number entered. | |  | - Failure to enable two-factor authentication due to network issues. | |  | - System rejects invalid credentials during validation. | | **Includes** | Email validation, Preference saving, Two-factor authentication setup. | | **Frequency of Use** | As needed | | **Special Requirements** | Intuitive UI for updating profile and preferences. | | **Assumptions** | User provides valid details and has access to email or phone for two-factor authentication. | | **Notes and Issues** | Ensure email and contact validation; notify the user if any issue arises during two-factor authentication setup. | | **Use Case ID** | UC-3 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Section** | **Details** | | **Use Case Name** | Income Management | | **Created By** | Rabia | | **Date Created** | 1st January 2025 | | **Last Updated By** | Rabia | | **Last Revision Date** | 1st January 2025 | | **Actors** | User, System | | **Description** | Enables users to manage their income details by adding, editing, deleting income sources, and viewing income trends. The system updates the impact of income changes on financial goals. | | **Trigger** | User selects the "Income Management" option from the menu. | | **Preconditions** | User is logged in and has access to the income management feature. | | **Postconditions** | Income records are updated, and the system reflects changes in financial goals based on modified income data. | | **Normal Flow** | 1. User navigates to the income management section. | |  | 2. User adds a new income source, entering source name, amount, and frequency. | |  | 3. System saves the income source and updates financial goals accordingly. | |  | 4. User modifies an existing income source. | |  | 5. System validates the changes and updates financial goals. | |  | 6. User deletes an income source. | |  | 7. System removes the income source and recalculates the impact on financial goals. | |  | 8. User views income trends in charts or graphs. | | **Alternative Flows** | - User adds income details but cancels before saving. | |  | - User edits an income source but reverts changes before saving. | |  | - User attempts to delete an income source but cancels the operation. | | **Exceptions** | - Invalid income amount or frequency entered. | |  | - System fails to save income details due to technical issues. | |  | - System fails to generate income trends due to insufficient data. | | **Includes** | Data validation, Visualization of income trends, Goal recalculations. | | **Frequency of Use** | Regularly as income details change or for tracking purposes. | | **Special Requirements** | - Intuitive UI for income management. | |  | - Charts/graphs should be visually appealing and easy to understand. | | **Assumptions** | User provides valid and accurate income details. | | **Notes and Issues** | Ensure system efficiently recalculates financial goals after any income update. | | **Use Case ID** | UC-4 | |

|  |  |
| --- | --- |
|  |  |
| **Use Case Name** | Expense Tracking |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User, System |
| **Description** | Allows users to log, edit, and delete expenses, create custom categories, and view expense trends by category or timeframe. The system records the data for analysis. |
| **Trigger** | User selects the "Expense Tracking" option from the menu. |
| **Preconditions** | User is logged in and has access to the expense tracking feature. |
| **Postconditions** | Expense details are updated, and the system records the data for further analysis or goal adjustments. |
| **Normal Flow** | 1. User navigates to the expense tracking section. |
|  | 2. User logs an expense, entering details like category, amount, and date. |
|  | 3. System saves the expense data. |
|  | 4. User edits an existing expense entry. |
|  | 5. System validates the changes and updates the record. |
|  | 6. User deletes an expense entry. |
|  | 7. System removes the expense record from the database. |
|  | 8. User views expense trends in charts or tables by category or timeframe. |
|  | 9. User creates custom expense categories for personalization. |
|  | 10. System saves the new category for future use. |
| **Alternative Flows** | - User logs an expense but cancels before saving. |
|  | - User edits an expense entry but reverts changes before saving. |
|  | - User attempts to delete an expense but cancels the operation. |
| **Exceptions** | - Invalid category or amount entered. |
|  | - System fails to save, update, or delete expense data due to technical issues. |
|  | - Insufficient data for generating expense trends. |
| **Includes** | Data validation, Visualization of expense trends, Goal recalculations. |
| **Frequency of Use** | Regularly for tracking expenses and managing finances. |
| **Special Requirements** | - Intuitive UI for expense tracking. |
|  | - Charts/graphs should be visually appealing and easy to interpret. |
| **Assumptions** | User provides accurate and valid expense details. |
| **Notes and Issues** | Ensure system efficiently handles data storage and retrieval for trends analysis. |
| **Use Case ID** | UC-5 |



|  |  |
| --- | --- |
|  |  |
| **Use Case Name** | Set Financial Goals |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User |
| **Description** | Allows users to set, track, and adjust financial goals. |
| **Trigger** | User selects the "Set Financial Goal" option. |
| **Preconditions** | User is logged in and has defined income and expenses. |
| **Postconditions** | Financial goals are set and tracked against the user's progress. |
| **Normal Flow** | 1. User enters goal details (target amount, timeline). |
|  | 2. System saves the goal and monitors progress. |
|  | 3. User receives updates on goal status. |
| **Alternative Flows** | - Notify user if goal progress is lagging behind. |
| **Exceptions** | - Unrealistic or unachievable goals entered. |
| **Includes** | Goal adjustment recommendation. |
| **Frequency of Use** | Monthly or as needed |
| **Special Requirements** | Provide recommendations for goal adjustments based on progress. |
| **Assumptions** | Users have a clear understanding of their financial capabilities. |
| **Notes and Issues** | Notify users when a goal is achieved. |
| **Use Case ID** | UC-6 |

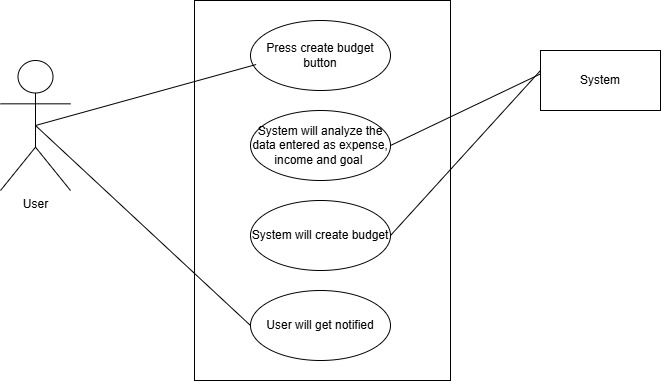


|  |  |
| --- | --- |
| **Section** | **Details** |
| **Use Case Name** | Multi-Device Support |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User, System |
| **Description** | Enables users to access synchronized financial data across multiple devices and restore app data from the cloud. |
| **Trigger** | User logs into their account on a new device or initiates a backup/restore operation. |
| **Preconditions** | - User has a valid account. |
|  | - User is logged in on at least one device. |
|  | - Cloud storage is configured and operational. |
| **Postconditions** | - User’s data is synchronized across all logged-in devices. |
|  | - User’s data is backed up or restored successfully. |
| **Normal Flow** | 1. User logs into their account on a new device. |
|  | 2. System authenticates the user and retrieves financial data from the cloud. |
|  | 3. System synchronizes the latest data across all devices. |
|  | 4. User initiates a backup operation. |
|  | 5. System stores the latest data to the cloud. |
|  | 6. User initiates a restore operation. |
|  | 7. System retrieves the backup data from the cloud and restores it on the device. |
| **Alternative Flows** | - User cancels synchronization midway. |
|  | - User attempts to log in on a new device but fails authentication. |
| **Exceptions** | - Cloud storage service is unavailable or fails. |
|  | - Synchronization conflicts occur due to changes from multiple devices. |
|  | - Backup or restore operation fails due to insufficient storage or corrupted data. |
| **Includes** | - Login and authentication process. |
|  | - Data encryption for secure synchronization and backups. |
| **Frequency of Use** | - Regularly for keeping data consistent across devices. |
| **Special Requirements** | - Secure cloud storage. |
|  | - High-speed data transfer for real-time synchronization. |
| **Assumptions** | - User has an active internet connection. |
|  | - Cloud storage is set up and configured. |
| **Notes and Issues** | - Ensure secure data handling during synchronization and backup processes. |
| **Use Case ID** | UC-7 |

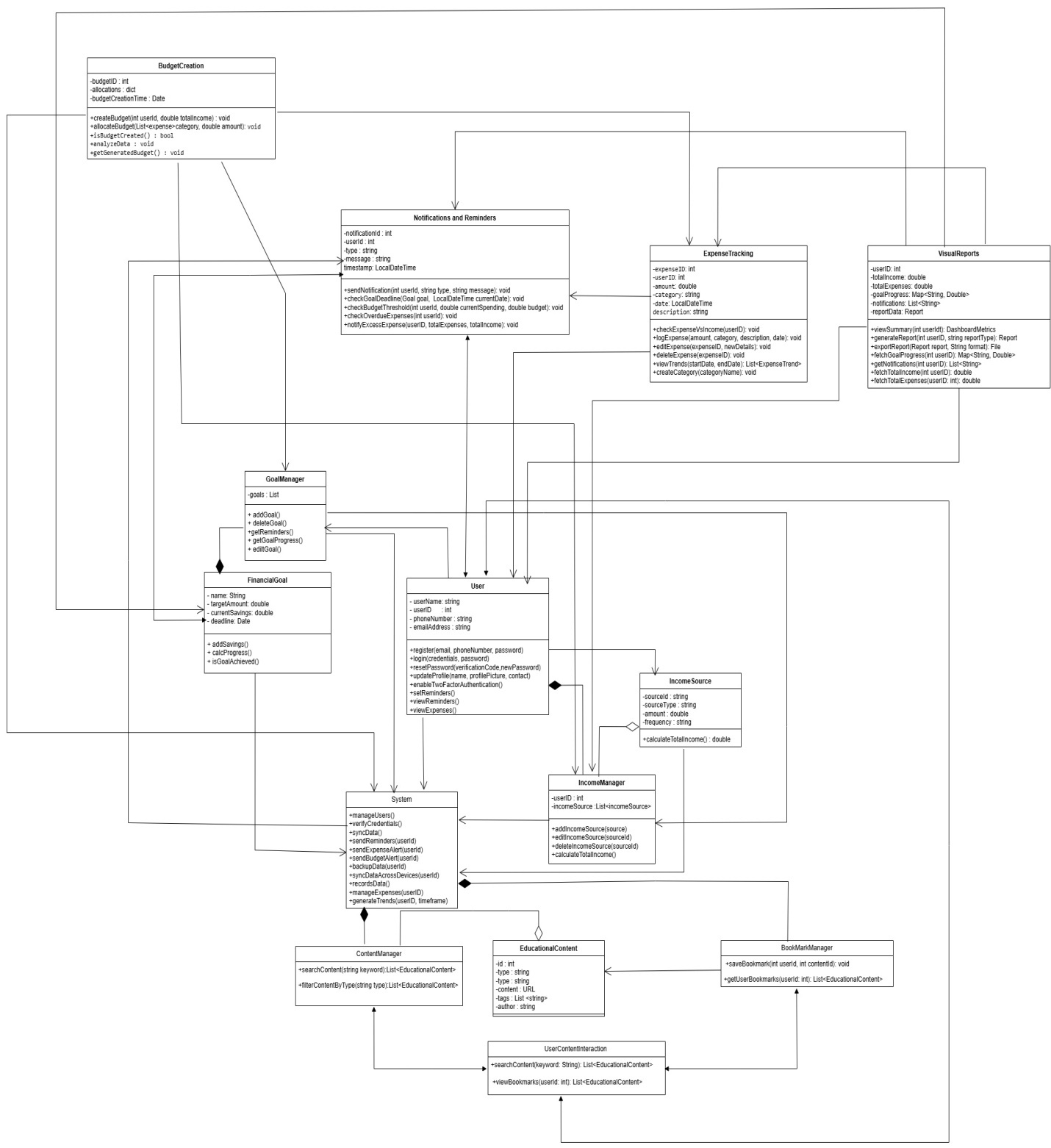
|  |  |
| --- | --- |
|  |  |
| **Use Case Name** | Reminders and Notifications |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User, System |
| **Description** | The system notifies users of untracked expenses, budget thresholds, and goal progress through periodic alerts. |
| **Trigger** | - User sets reminders or budget thresholds. |
|  | - System detects spending patterns or evaluates goal progress. |
| **Preconditions** | - User has enabled notifications in settings. |
|  | - System has access to user’s expense and goal data. |
| **Postconditions** | - User receives notifications for untracked expenses, nearing budget limits, or updates on goal progress. |
| **Normal Flow** | 1. System analyzes user’s spending patterns and identifies untracked expenses. |
|  | 2. System sends an “Expense Alert” to notify the user of missing entries. |
|  | 3. User sets a budget threshold. |
|  | 4. System tracks spending and sends “Budget Alert” when the threshold is nearing or exceeded. |
|  | 5. User sets financial goals. |
|  | 6. System evaluates progress and sends periodic “Goal Reminders.” |
| **Alternative Flows** | - User disables specific notifications. |
|  | - User updates budget thresholds or goal settings after receiving alerts. |
| **Exceptions** | - Notifications fail to send due to server downtime or connectivity issues. |
|  | - System misidentifies untracked expenses due to incomplete data. |
| **Includes** | - Integration with notification system (push notifications, emails, etc.). |
| **Frequency of Use** | - Regularly, depending on user preferences for reminders and alerts. |
| **Special Requirements** | - Real-time analytics for spending patterns. |
|  | - Customizable notification preferences. |
| **Assumptions** | - User actively updates their expense and goal data. |
|  | - Notification delivery services are reliable. |
| **Notes and Issues** | - Ensure that notifications are concise and actionable to avoid user fatigue. |
| **Use Case ID** | UC-8 |

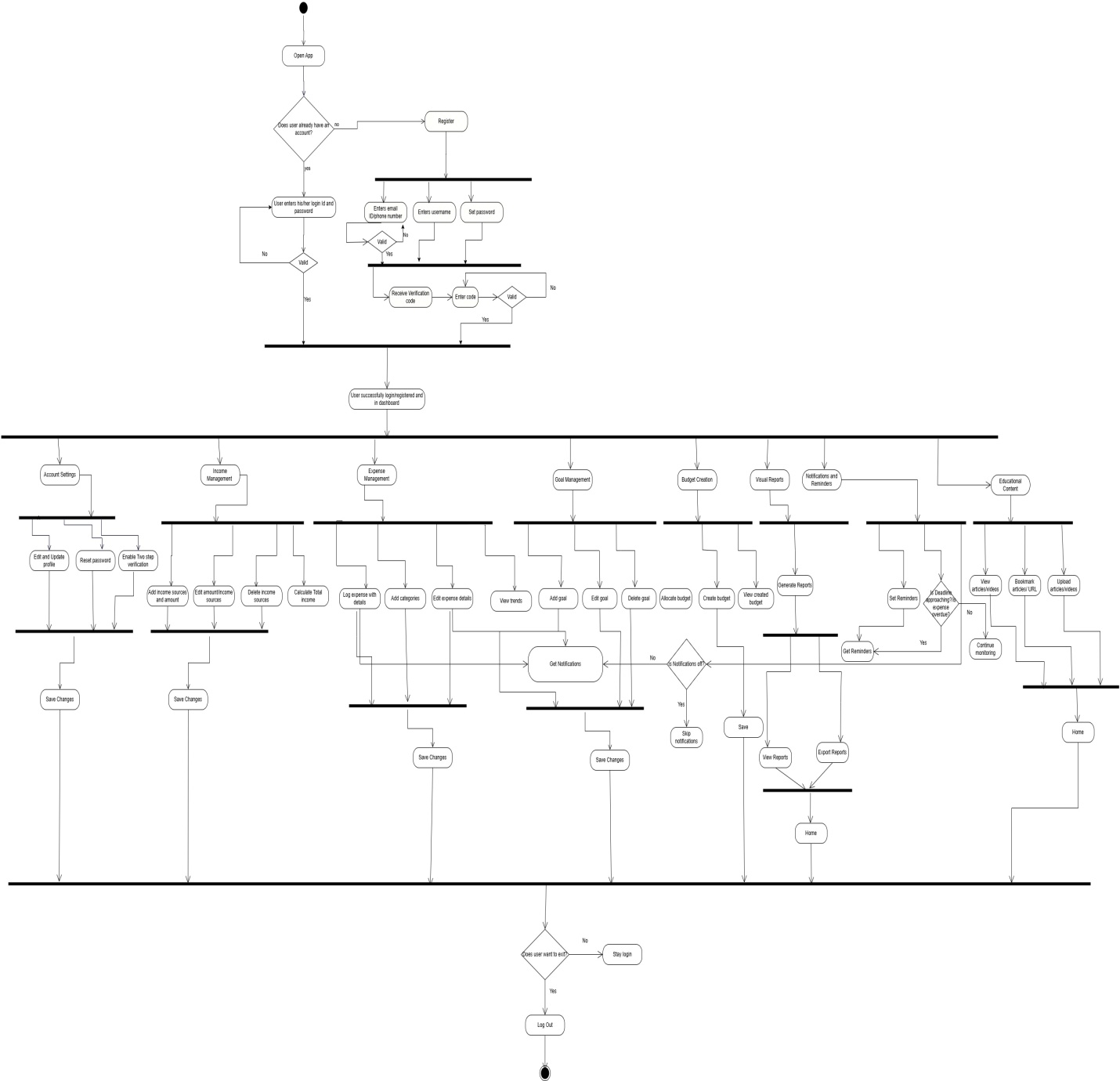
|  |  |
| --- | --- |
|  |  |
| **Use Case Name** | Dashboard and Visual Reports |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User, System |
| **Description** | The system provides users with a summary of key financial metrics, generates detailed reports, and allows exporting reports for external use. |
| **Trigger** | - User accesses the dashboard or selects the option to generate/export a report. |
| **Preconditions** | - System has updated financial data for the user. |
|  | - User has logged into their account. |
| **Postconditions** | - User views a dashboard with summarized metrics. |
|  | - Reports are generated and optionally exported in the selected format. |
| **Normal Flow** | 1. User logs into the application and navigates to the dashboard. |
|  | 2. System retrieves and displays a summary of key metrics, such as income, expenses, and goal progress. |
|  | 3. User selects the "Generate Reports" option and specifies report parameters. |
|  | 4. System generates a detailed financial report based on the parameters. |
|  | 5. User selects the "Export Reports" option and chooses a format (e.g., PDF, Excel). |
|  | 6. System exports the report and notifies the user of successful completion. |
| **Alternative Flows** | - User refreshes the dashboard to update metrics. |
|  | - User cancels the report generation process. |
| **Exceptions** | - Dashboard fails to load due to incomplete or outdated financial data. |
|  | - Report generation or export fails due to insufficient data or system errors. |
| **Includes** | - Integration with reporting and exporting modules. |
| **Frequency of Use** | - Frequently, depending on user needs for insights and documentation. |
| **Special Requirements** | - Dashboard must display real-time or near-real-time metrics. |
|  | - Reports must support multiple export formats. |
| **Assumptions** | - User’s financial data is accurate and up-to-date. |
|  | - Exported reports are compatible with standard tools (e.g., Adobe Reader, Excel). |
| **Notes and Issues** | - Ensure dashboard is visually intuitive and easy to navigate. |
|  | - Optimize report generation for speed and accuracy. |
| **Use Case ID** | UC-9 |

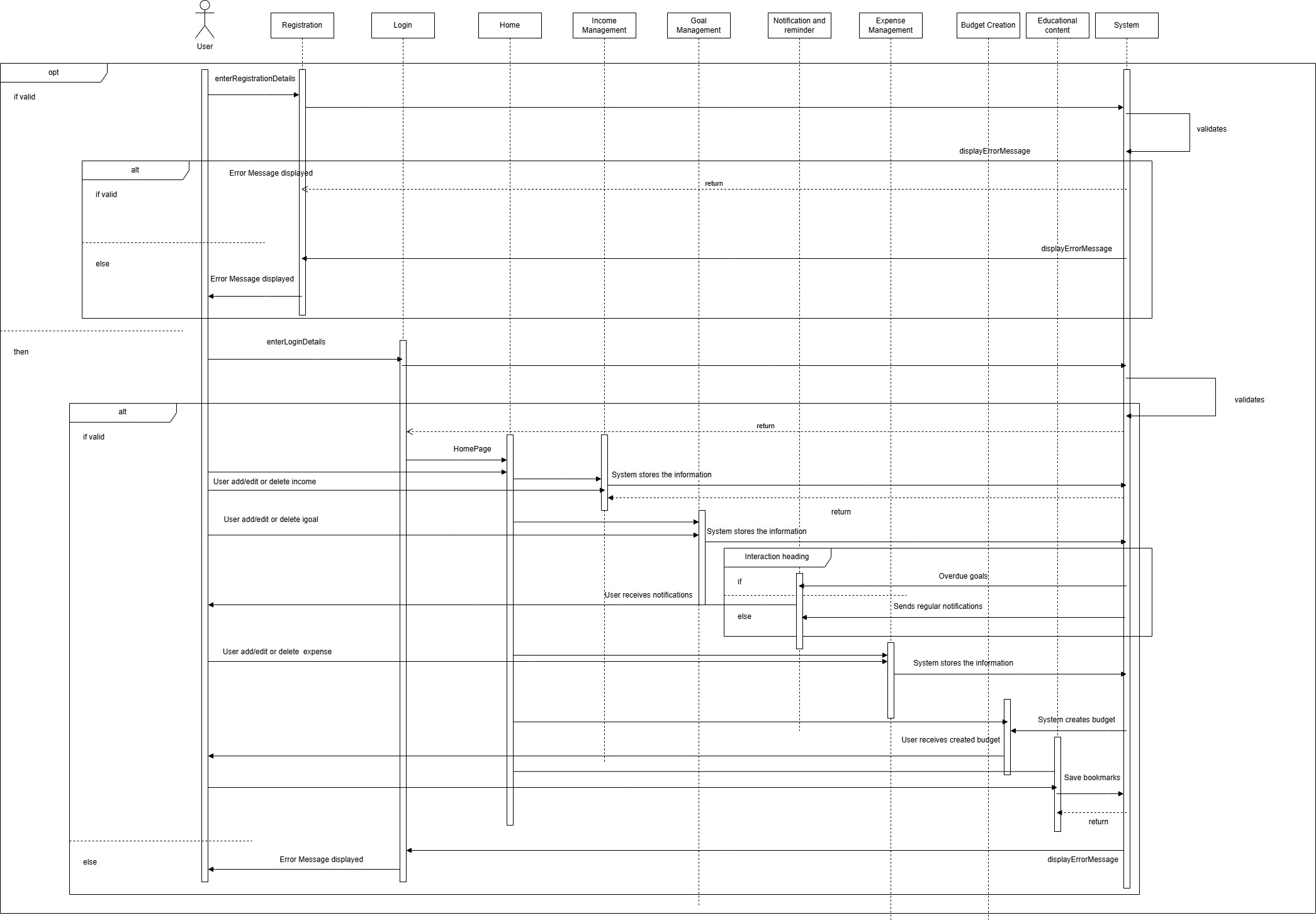
|  |  |
| --- | --- |
|  |  |
| **Use Case Name** | Educational Content |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User, System |
| **Description** | The system provides users with access to educational resources, including articles, tutorials, and a search functionality to explore and bookmark content. |
| **Trigger** | - User selects the "Educational Content" option or searches for resources. |
| **Preconditions** | - User is logged into their account. |
|  | - Educational content is available in the system. |
| **Postconditions** | - User views articles, tutorials, or bookmarked resources. |
|  | - User can bookmark and revisit desired materials. |
| **Normal Flow** | 1. User logs into the application and navigates to the educational section. |
|  | 2. System displays available articles and tutorials categorized by topic. |
|  | 3. User selects an article or tutorial to view. |
|  | 4. System loads the selected content for the user. |
|  | 5. User searches for specific resources using keywords. |
|  | 6. System displays search results matching the keywords. |
|  | 7. User bookmarks preferred resources for later access. |
| **Alternative Flows** | - User filters content by category or level (beginner, advanced). |
|  | - User navigates to bookmarked resources directly. |
| **Exceptions** | - No matching results are found for the search. |
|  | - Educational content fails to load due to system error or network issues. |
| **Includes** | - Integration with search and bookmarking modules. |
| **Frequency of Use** | - Frequently, depending on user interest in financial education. |
| **Special Requirements** | - Content should be engaging and easy to navigate. |
|  | - Tutorials must support video playback. |
|  | - Search functionality should be fast and accurate. |
| **Assumptions** | - System hosts a variety of high-quality financial educational materials. |
|  | - Users are interested in improving their financial literacy. |
| **Notes and Issues** | - Ensure content is updated regularly to remain relevant. |
|  | - Optimize video tutorials for different devices and network conditions. |
| **Use Case ID** | UC-10 |

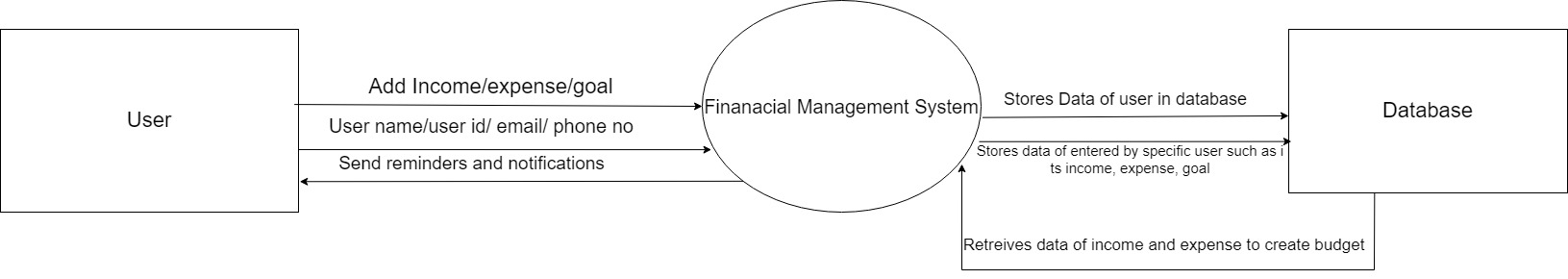


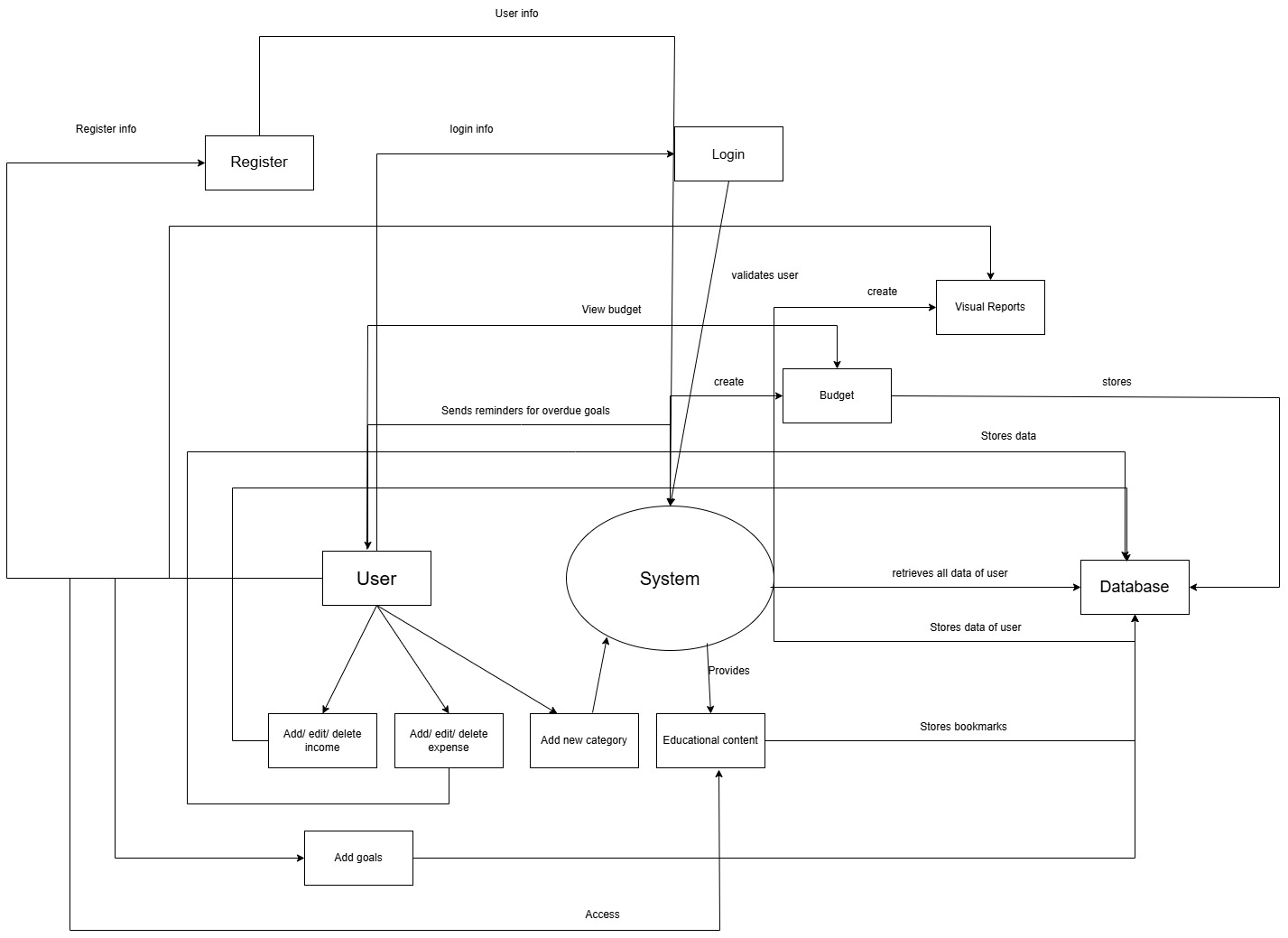
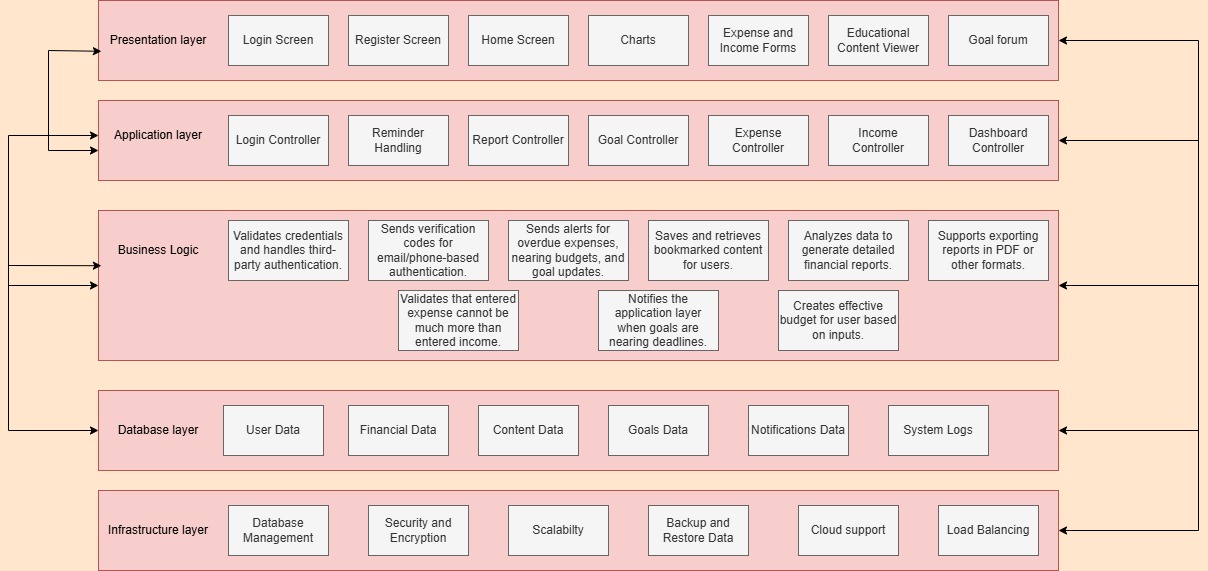
|  |  |
| --- | --- |
| **Section** | **Details** |
| **Use Case Name** | Create Budget |
| **Created By** | Rabia |
| **Date Created** | 1st January 2025 |
| **Last Updated By** | Rabia |
| **Last Revision Date** | 1st January 2025 |
| **Actors** | User |
| **Description** | Allows users to create a budget by analyzing their income, expense, and goals. If any input is missing, the system notifies the user. |
| **Trigger** | User selects the "Create Budget" button. |
| **Preconditions** | User is logged in and has defined income and expenses. |
| **Postconditions** | Budget is created, and the user is notified of success or any missing data. |
| **Normal Flow** | 1. User enters income, expenses, and goals. 2. User clicks "Create Budget" button. 3. System processes input data. 4. System creates budget and displays success message. |
| **Alternative Flows** | - Notify user if income or expense fields are missing. - Send email or message notification if data is incomplete. |
| **Exceptions** | - Missing income or expense input. - Invalid or incomplete data entries. |
| **Includes** | Sends notifications to the user if there is missing data. |
| **Frequency of Use** | As needed when users want to create or adjust their budget. |
| **Special Requirements** | - User should be notified by message or email if fields are missing. - System should notify user if budget is successfully created. |
| **Assumptions** | Users have accurate data for income and expenses. |
| **Notes and Issues** | Ensure that notifications are clear and actionable for missing data. |
| **Use Case ID** | UC-11 |









**Module Name: Register**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **register\_01** | | **Test Case Description** | To test the account registration functionality for users to create an account with a valid email/phone and password. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User not already registered. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking that users can register with a valid email/phone and password, handle errors for invalid formats, and verify the registration via email or phone. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| register\_01.1 | Enter valid email and password | Email: [valid\_email@example.com](mailto:valid_email@example.com) Password: valid\_password | Account is created and a verification email/phone is sent | Account is created and verification email/phone is sent | Pass | No issues found. |
| register\_01.2 | Enter invalid email format | Email: invalid\_email Password: valid\_password | Error message: "Invalid email format." | Error message: "Invalid email format." | Pass | Handled invalid email format. |
| register\_01.3 | Enter valid phone number but invalid format | Phone: 12345 Password: valid\_password | Error message: "Invalid phone number format." | Error message: "Invalid phone number format." | Pass | Handled invalid phone number format. |
| register\_01.4 | Leave email/phone and password blank | Email: blank Password: blank | Error message: "Fields cannot be empty." | Error message: "Fields cannot be empty." | Pass | Empty input handled properly. |
| register\_01.5 | Enter valid email and weak password | Email: [valid\_email@example.com](mailto:valid_email@example.com) Password: weak\_password | Error message: "Password is too weak." | Error message: "Password is too weak." | Pass | Handled weak password case. |
| register\_01.6 | Attempt to register with already registered email/phone | Email: [already\_registered@example.com](mailto:already_registered@example.com) Password: valid\_password | Error message: "Email/Phone already registered." | Error message: "Email/Phone already registered." | Pass | Handled already registered email/phone case. |
| register\_01.7 | Attempt registration without internet connection | Email: [valid\_email@example.com](mailto:valid_email@example.com) Password: valid\_password | Error message: "No internet connection." | Error message: "No internet connection." | Pass | Internet connection error displayed. |

**Module Name: Login**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **login\_01** | | **Test Case Description** | To test the login functionality of the application. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. Valid user account for testing. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking that users can log in with correct credentials and are handled properly for incorrect credentials. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| login\_01.1 | Enter valid username and password | Username: valid\_user Password: valid\_pass | User is redirected to the Home Dashboard. | User is redirected to the Home Dashboard. | Pass | No issues found. |
| login\_01.2 | Enter invalid username and valid password | Username: invalid\_user Password: valid\_pass | Error message: "Invalid credentials." | Error message: "Invalid credentials." | Pass | Invalid credentials rejected successfully. |
| login\_01.3 | Enter valid username and invalid password | Username: valid\_user Password: invalid\_pass | Error message: "Invalid credentials." | Error message: "Invalid credentials." | Pass | Invalid credentials rejected successfully. |
| login\_01.4 | Leave both fields blank | Username: blank Password: blank | Error message: "Fields cannot be empty." | Error message: "Fields cannot be empty." | Pass | Empty input handled properly. |
| login\_01.5 | Enter valid email but invalid format | Username: user@example Password: any\_password | Error message: "Invalid email format." | Error message: "Invalid email format." | Pass | Invalid email format handled properly. |
| login\_01.6 | Attempt login without internet connection | Username: valid\_user Password: valid\_pass | Error message: "No internet connection." | Error message: "No internet connection." | Pass | Internet connection error displayed. |

**Module Name: Third-Party Login**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **third\_party\_login\_01** | | **Test Case Description** | To test the third-party authentication functionality (e.g., Google, Facebook) for users to log in via their third-party accounts. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. Valid third-party account (Google, Facebook, etc.). | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking that users can successfully log in using third-party authentication (Google/Facebook) and that the login process integrates seamlessly with the app. | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** | | third\_party\_login\_01.1 | Tap on the Google login button | Google account: valid\_account | User is redirected to Google login page and logs in successfully. | User redirected to Google login page and logged in. | Pass | No issues found. | | third\_party\_login\_01.2 | Tap on the Facebook login button | Facebook account: valid\_account | User is redirected to Facebook login page and logs in successfully. | User redirected to Facebook login page and logged in. | Pass | No issues found. | | third\_party\_login\_01.3 | Tap on the Google login button with invalid credentials | Google account: invalid\_account | Error message: "Invalid Google credentials." | Error message: "Invalid Google credentials." | Pass | Handled invalid credentials properly. | | third\_party\_login\_01.4 | Tap on the Facebook login button with invalid credentials | Facebook account: invalid\_account | Error message: "Invalid Facebook credentials." | Error message: "Invalid Facebook credentials." | Pass | Handled invalid credentials properly. | | third\_party\_login\_01.5 | Attempt login via Google without internet connection | Google account: valid\_account | Error message: "No internet connection." | Error message: "No internet connection." | Pass | Internet connection error displayed. | | third\_party\_login\_01.6 | Tap on the third-party login button and cancel login process | Google account: valid\_account Facebook account: valid\_account | User is returned to the login screen without logging in. | User returned to login screen without login. | Pass | Handled cancellation correctly. | | third\_party\_login\_01.7 | Attempt login with third-party account linked to multiple apps | Google account: valid\_account | User is prompted to choose an account and log in successfully. | User prompted to choose account and logged in. | Pass | Handled account selection correctly. | |

**Module Name: Forgot Password**

|  |  |
| --- | --- |
| **Test Case ID** | **forgot\_password\_01** |
| **Test Case Description** | To test the functionality of the password reset process via email or phone verification. |
| **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. Valid registered email or phone number for the user. |
| **Tester's Name** | Rabia |
| **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. |
| **Test Scenario** | Checking the password reset functionality via email or phone number for a registered user and ensuring proper handling of incorrect or unregistered user details. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** | | forgot\_password\_01.1 | Enter valid registered email address for password reset | Email: [valid\_user@example.com](mailto:valid_user@example.com) | Reset email sent to the user’s email address with a link to reset password. | Reset email sent successfully. | Pass | No issues found. | | forgot\_password\_01.2 | Enter valid registered phone number for password reset | Phone number: +1234567890 | Reset SMS sent to the user's phone number with a link to reset password. | Reset SMS sent successfully. | Pass | No issues found. | | forgot\_password\_01.3 | Enter unregistered email address | Email: [unregistered\_user@example.com](mailto:unregistered_user@example.com) | Error message: "Email address not registered." | Error message: "Email address not registered." | Pass | Unregistered email handled properly. | | forgot\_password\_01.4 | Enter unregistered phone number | Phone number: +0987654321 | Error message: "Phone number not registered." | Error message: "Phone number not registered." | Pass | Unregistered phone number handled properly. | | forgot\_password\_01.5 | Enter invalid email format | Email: invalid\_email\_format | Error message: "Invalid email format." | Error message: "Invalid email format." | Pass | Invalid email format handled properly. | | forgot\_password\_01.6 | Enter email with no internet connection | Email: [valid\_user@example.com](mailto:valid_user@example.com) | Error message: "No internet connection." | Error message: "No internet connection." | Pass | Internet connection error displayed. | | forgot\_password\_01.7 | Click on the reset link in the email/SMS | Reset link from email/SMS | User is redirected to the password reset screen to set a new password. | User redirected to password reset screen. | Pass | No issues found. | | forgot\_password\_01.8 | Enter new password and confirm password | New password: new\_pass Confirm password: new\_pass | Password updated successfully and user is redirected to the login screen. | Password updated successfully. | Pass | No issues found. | | forgot\_password\_01.9 | Enter new password and mismatched confirm password | New password: new\_pass Confirm password: different\_pass | Error message: "Passwords do not match." | Error message: "Passwords do not match." | Pass | Mismatch in password confirmation handled properly. | |

**Module Name: Update Profile**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **update\_profile\_01** | | **Test Case Description** | To test the functionality of updating a user's profile, including name, contact information, and profile picture. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with valid credentials. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking that users can update their name, contact information, and profile picture, and ensuring the system validates the inputs correctly. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| update\_profile\_01.1 | Tap on the "Edit Profile" button | N/A | User is redirected to the profile update screen. | User redirected to profile update screen. | Pass | No issues found. |
| update\_profile\_01.2 | Update name field with a valid name | Name: John Doe | Name is updated successfully. | Name updated successfully. | Pass | No issues found. |
| update\_profile\_01.3 | Update phone number with a valid format | Phone number: +1234567890 | Phone number updated successfully. | Phone number updated successfully. | Pass | No issues found. |
| update\_profile\_01.4 | Update email with a valid email format | Email: [john.doe@example.com](mailto:john.doe@example.com) | Email updated successfully. | Email updated successfully. | Pass | No issues found. |
| update\_profile\_01.5 | Update profile picture with a valid image file | Profile picture: image\_file.png | Profile picture updated successfully. | Profile picture updated successfully. | Pass | No issues found. |
| update\_profile\_01.6 | Update phone number with invalid format | Phone number: 12345 | Error message: "Invalid phone number format." | Error message: "Invalid phone number format." | Pass | Invalid phone number format handled properly. |
| update\_profile\_01.7 | Update email with invalid format | Email: invalid\_email\_format | Error message: "Invalid email format." | Error message: "Invalid email format." | Pass | Invalid email format handled properly. |
| update\_profile\_01.8 | Leave the name field empty | Name: (blank) | Error message: "Name cannot be empty." | Error message: "Name cannot be empty." | Pass | Empty name field handled properly. |
| update\_profile\_01.9 | Leave the phone number field empty | Phone number: (blank) | Error message: "Phone number cannot be empty." | Error message: "Phone number cannot be empty." | Pass | Empty phone number field handled properly. |
| update\_profile\_01.10 | Tap on "Save" after updating details | Name: John Doe Phone: +1234567890 Email: [john.doe@example.com](mailto:john.doe@example.com) Profile picture: image\_file.png | User is redirected back to the profile screen with updated details. | User redirected to profile screen with updated details. | Pass | No issues found. |
| update\_profile\_01.11 | Attempt to update profile without internet connection | N/A | Error message: "No internet connection." | Error message: "No internet connection." | Pass | Internet connection error displayed. |

**Module Name: Account Security**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **account\_security\_01** | | **Test Case Description** | To verify the functionality of two-factor authentication (2FA) for secure login, including the correct handling of 2FA codes. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with valid credentials and 2FA enabled. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking that two-factor authentication is triggered correctly after login and ensuring users can authenticate successfully using their second factor (e.g., SMS code, authentication app). | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| account\_security\_01.1 | Enter valid username and password. | Username: valid\_user Password: valid\_pass | User is prompted for 2FA code. | User is prompted for 2FA code. | Pass | No issues found. |
| account\_security\_01.2 | Enter valid 2FA code received via SMS | 2FA Code: 123456 | User is successfully logged in and redirected to the home screen. | User successfully logged in. | Pass | No issues found. |
| account\_security\_01.3 | Enter invalid 2FA code | 2FA Code: 654321 | Error message: "Invalid 2FA code. Please try again." | Error message: "Invalid 2FA code." | Pass | Invalid code handled properly. |
| account\_security\_01.4 | Enter expired 2FA code | 2FA Code: (expired code) | Error message: "The code has expired. Please request a new one." | Error message: "The code has expired." | Pass | Expired code handled properly. |
| account\_security\_01.5 | Attempt login without internet connection | Username: valid\_user Password: valid\_pass | Error message: "No internet connection." | Error message: "No internet connection." | Pass | Internet connection error displayed. |
| account\_security\_01.6 | Enter valid username and password, but skip 2FA code | N/A | Error message: "2FA code is required to complete login." | Error message: "2FA code is required." | Pass | 2FA code requirement handled properly. |
| account\_security\_01.7 | Request a new 2FA code after failure | N/A | User receives a new 2FA code. | User receives new code. | Pass | New code generated successfully. |
| account\_security\_01.8 | Attempt to login with correct 2FA code after multiple failed attempts | 2FA Code: 123456 | User is prompted to wait before retrying or asked for a security question. | User prompted to wait. | Pass | Retry handling works. |
| account\_security\_01.9 | Log out and attempt login again with 2FA | Username: valid\_user Password: valid\_pass 2FA Code: 123456 | User successfully logs in and is redirected to home screen. | User successfully logged in. | Pass | Logout and login process works. |

**Module Name: Add Income Sources**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **add\_income\_sources\_01** | | **Test Case Description** | To verify the functionality of adding income sources such as salary, freelance, and rental income, including details like amount, frequency, and source type. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking the ability to add different types of income sources with valid inputs and ensuring that the income source is saved correctly in the system. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| add\_income\_sources\_01.1 | Enter valid salary income details. | Income Type: Salary Amount: 5000 Frequency: Monthly Source: Company A | Income source is successfully added and displayed in the income list. | Income source added successfully. | Pass | Salary income added correctly. |
| add\_income\_sources\_01.2 | Enter valid freelance income details. | Income Type: Freelance Amount: 1500 Frequency: Weekly Source: Freelance Client | Income source is successfully added and displayed in the income list. | Income source added successfully. | Pass | Freelance income added correctly. |
| add\_income\_sources\_01.3 | Enter valid rental income details. | Income Type: Rental Amount: 2000 Frequency: Monthly Source: Property A | Income source is successfully added and displayed in the income list. | Income source added successfully. | Pass | Rental income added correctly. |
| add\_income\_sources\_01.4 | Enter invalid amount (negative value). | Income Type: Salary Amount: -5000 Frequency: Monthly Source: Company A | Error message: "Amount cannot be negative." | Error message displayed correctly. | Pass | Negative amount handled properly. |
| add\_income\_sources\_01.5 | Leave frequency field empty. | Income Type: Freelance Amount: 1500 Frequency: (blank) Source: Freelance Client | Error message: "Frequency is required." | Error message displayed correctly. | Pass | Empty frequency field handled properly. |
| add\_income\_sources\_01.6 | Enter valid details but choose an invalid income type. | Income Type: (invalid type) Amount: 1000 Frequency: Monthly Source: Client B | Error message: "Invalid income type selected." | Error message displayed correctly. | Pass | Invalid income type handled properly. |
| add\_income\_sources\_01.7 | Enter valid details and submit with a source that already exists. | Income Type: Salary Amount: 3000 Frequency: Monthly Source: Company A | Error message: "Income source already exists." | Error message displayed correctly. | Pass | Duplicate source handled properly. |
| add\_income\_sources\_01.8 | Enter valid details and save the income source. | Income Type: Other Amount: 1000 Frequency: One-time Source: Miscellaneous | Income source is successfully added and displayed in the income list. | Income source added successfully. | Pass | Miscellaneous income added correctly. |
| add\_income\_sources\_01.9 | Try adding income source with no input in any field. | Income Type: (blank) Amount: (blank) Frequency: (blank) Source: (blank) | Error message: "All fields are required." | Error message displayed correctly. | Pass | All fields required handled properly. |
| add\_income\_sources\_01.10 | Verify income source persistence after app restart. | Income Type: Freelance Amount: 2000 Frequency: Monthly Source: Freelance Client | Income source should still appear in the income list after restarting the app. | Income source persisted correctly. | Pass | Income source persisted correctly after restart. |

**Module Name: Edit Income**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **edit\_income\_01** | | **Test Case Description** | To verify the functionality of editing existing income sources, ensuring the correct modification of income details such as amount, frequency, and source type. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. Existing income sources added for editing. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking the ability to edit income details for an existing income source and ensuring that the changes are saved correctly. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| edit\_income\_01.1 | Select an existing income source and click on 'Edit'. | Income Source: Freelance | The income details for the selected source are displayed for editing. | Income details displayed for editing. | Pass | Editing screen loaded correctly. |
| edit\_income\_01.2 | Modify the income amount. | New Amount: 1800 | The income amount is updated and saved correctly. | Income amount updated successfully. | Pass | Amount updated correctly. |
| edit\_income\_01.3 | Change the frequency of income. | New Frequency: Bi-weekly | The frequency of the income is updated and saved correctly. | Frequency updated successfully. | Pass | Frequency updated correctly. |
| edit\_income\_01.4 | Modify the source type (e.g., from "Freelance" to "Salary"). | New Source Type: Salary | The source type is updated correctly. | Source type updated successfully. | Pass | Source type updated correctly. |
| edit\_income\_01.5 | Leave income amount blank. | Income Type: Salary Amount: (blank) | Error message: "Amount is required." | Error message displayed correctly. | Pass | Blank amount handled properly. |
| edit\_income\_01.6 | Leave frequency field blank. | Income Type: Salary Amount: 3000 Frequency: (blank) | Error message: "Frequency is required." | Error message displayed correctly. | Pass | Blank frequency handled properly. |
| edit\_income\_01.7 | Edit income source to an already existing source (duplicate). | New Source: Freelance | Error message: "This income source already exists." | Error message displayed correctly. | Pass | Duplicate source handled correctly. |
| edit\_income\_01.8 | Cancel editing and return to the previous screen. | Income Type: Freelance Amount: 2000 Frequency: Weekly | No changes are saved, and the user is returned to the income list. | No changes saved. User returned to list. | Pass | Edit cancel functionality working correctly. |
| edit\_income\_01.9 | Save changes and verify the updated details are reflected in the income list. | Income Type: Freelance Amount: 1800 Frequency: Bi-weekly | Updated income source should appear in the income list with the new details. | Income source updated correctly in list. | Pass | Changes reflected correctly in the list. |
| edit\_income\_01.10 | Verify income source persistence after app restart. | Income Type: Freelance Amount: 1800 Frequency: Bi-weekly | Edited income source should persist after app restart. | Edited income source persisted correctly. | Pass | Changes persisted correctly after restart. |

**Module Name: Delete Income**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **delete\_income\_01** | | **Test Case Description** | To verify the functionality of deleting an existing income source and ensuring that the source is permanently removed from the system. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. Existing income sources added for deletion. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking the ability to delete an income source and ensuring it is removed from the list and the data is updated correctly. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| delete\_income\_01.1 | Select an existing income source and click on 'Delete'. | Income Source: Freelance | The income source is marked for deletion. A confirmation prompt appears. | Deletion prompt appeared. | Pass | Prompt displayed correctly. |
| delete\_income\_01.2 | Confirm the deletion by selecting 'Yes'. | Confirmation: Yes | The selected income source is permanently deleted, and the list is updated. | Income source deleted and list updated. | Pass | Source deleted and list updated. |
| delete\_income\_01.3 | Select an income source and click 'Delete', then select 'No' to cancel. | Confirmation: No | No deletion occurs, and the user is returned to the income list. | No deletion, user returned to list. | Pass | Deletion cancelled successfully. |
| delete\_income\_01.4 | Attempt to delete an income source that is linked to a financial goal. | Income Source: Freelance (linked to goal) | Error message: "This income source is linked to a goal and cannot be deleted." | Error message displayed correctly. | Pass | Linked source handled properly. |
| delete\_income\_01.5 | Try to delete an income source while offline. | Income Source: Salary | Error message: "No internet connection." Deletion request cannot be processed. | Error message displayed correctly. | Pass | Offline error handled properly. |
| delete\_income\_01.6 | Check if the income source is removed from the list after deletion. | Income Source: Freelance | Income source should no longer appear in the income list. | Income source removed from list. | Pass | Source removed as expected. |
| delete\_income\_01.7 | Verify persistence of deletion after app restart. | Income Source: Freelance (deleted) | The deleted income source should not appear after restarting the app. | Deleted income source not present. | Pass | Deletion persisted after restart. |
| delete\_income\_01.8 | Attempt to delete a non-existing income source. | Income Source: Non-existent | Error message: "Income source not found." | Error message displayed correctly. | Pass | Non-existing source handled properly. |
| delete\_income\_01.9 | Check if the user is logged out or redirected after the deletion. | Income Source: Freelance | The user remains logged in and is not redirected. The app state remains unchanged. | User remains logged in. No redirection. | Pass | User session remains active. |

**Module Name: Categorization**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **categorization\_01** | | **Test Case Description** | To verify the categorization functionality, including predefined categories such as rent, utilities, groceries, entertainment, and the ability for users to add custom categories. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking predefined categories, the ability to add custom categories, and ensuring they are saved and displayed properly. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| categorization\_01.1 | Open the category section and view predefined categories. | Predefined Categories | Categories: Rent, Utilities, Groceries, Entertainment | Categories listed correctly. | Pass | Predefined categories displayed as expected. |
| categorization\_01.2 | Add a new custom category. | Custom Category Name: "Travel" | Category "Travel" should be added and displayed in the list. | Custom category "Travel" added. | Pass | Category added as expected. |
| categorization\_01.3 | Add another custom category with a duplicate name. | Custom Category Name: "Rent" | Error message: "Category name already exists." | Error message displayed. | Pass | Duplicate category handling works. |
| categorization\_01.4 | Edit a custom category name. | Current Category: "Travel", New Name: "Holiday" | Category name should be updated to "Holiday" in the list. | Category name updated to "Holiday". | Pass | Category edited successfully. |
| categorization\_01.5 | Delete a custom category. | Category Name: "Holiday" | Category should be removed from the list. | Category removed successfully. | Pass | Category deleted correctly. |
| categorization\_01.6 | Try to delete a predefined category. | Predefined Category: "Groceries" | Error message: "Cannot delete predefined category." | Error message displayed. | Pass | Predefined categories protected from deletion. |
| categorization\_01.7 | Check if custom categories are displayed after app restart. | Custom Category: "Travel" | Custom categories should be saved and displayed correctly after restarting the app. | Custom category "Travel" displayed after restart. | Pass | Custom categories persisted after restart. |
| categorization\_01.8 | Verify that new custom categories can be used when adding an expense. | Custom Category: "Travel" | "Travel" should be available in the category dropdown when logging an expense. | "Travel" appears in the expense category list. | Pass | Custom category available for expense entry. |
| categorization\_01.9 | Verify behavior when no category is selected while adding an expense. | Expense Category: None | Error message: "Category cannot be empty." | Error message displayed. | Pass | Empty category input handled properly. |
| categorization\_01.10 | Check if the app allows for editing an existing predefined category. | Predefined Category: "Rent", New Name: "Housing" | Error message: "Cannot edit predefined category." | Error message displayed. | Pass | Predefined category editing restricted. |
| categorization\_01.11 | Check if expense categories are synced across devices. | Category: "Travel" | "Travel" should be visible across all devices where the user is logged in. | "Travel" category visible across devices. | Pass | Category synchronization works as expected. |

**Module Name: Expense Entry**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **expense\_entry\_01** | | **Test Case Description** | To verify that expenses can be logged with the required details such as amount, date, and category. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking if the user can log expenses correctly by entering amount, date, and category, and verifying if they are saved and displayed properly. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| expense\_entry\_01.1 | Open the expense entry form. | None | Expense entry form should open. | Expense entry form opened. | Pass | Form opened successfully. |
| expense\_entry\_01.2 | Enter valid expense amount, date, and category. | Amount: 100, Date: 2025-01-18, Category: Groceries | Expense should be saved and shown in the expense list. | Expense saved and displayed. | Pass | Expense added correctly. |
| expense\_entry\_01.3 | Leave the amount field empty. | Amount: blank, Date: 2025-01-18, Category: Groceries | Error message: "Amount cannot be empty." | Error message displayed. | Pass | Empty amount handled properly. |
| expense\_entry\_01.4 | Leave the date field empty. | Amount: 100, Date: blank, Category: Groceries | Error message: "Date cannot be empty." | Error message displayed. | Pass | Empty date handled properly. |
| expense\_entry\_01.5 | Leave the category field empty. | Amount: 100, Date: 2025-01-18, Category: blank | Error message: "Category cannot be empty." | Error message displayed. | Pass | Empty category handled properly. |
| expense\_entry\_01.6 | Enter an invalid amount (negative number). | Amount: -50, Date: 2025-01-18, Category: Groceries | Error message: "Amount must be positive." | Error message displayed. | Pass | Invalid amount handled properly. |
| expense\_entry\_01.7 | Enter an invalid date format. | Amount: 100, Date: 2025/01/18, Category: Groceries | Error message: "Invalid date format." | Error message displayed. | Pass | Invalid date format handled. |
| expense\_entry\_01.8 | Enter valid expense details and submit the form. | Amount: 100, Date: 2025-01-18, Category: Groceries | Expense should be saved and shown in the expense list. | Expense saved and displayed. | Pass | Expense added and displayed correctly. |
| expense\_entry\_01.9 | Verify if the expense is reflected in the report. | Expense Entry: 100 on 2025-01-18 under Groceries | Expense should appear in the financial report for the date and category. | Expense reflected in the report. | Pass | Expense is included in the report. |
| expense\_entry\_01.10 | Verify if the expense is reflected across devices. | Expense Entry: 100 on 2025-01-18 under Groceries | Expense should appear on other devices upon syncing. | Expense displayed on other devices. | Pass | Expense sync works correctly. |
| expense\_entry\_01.11 | Try logging an expense without internet. | Amount: 100, Date: 2025-01-18, Category: Groceries | Error message: "No internet connection." | Error message displayed. | Pass | No internet error displayed. |
| expense\_entry\_01.12 | Verify if the expense is saved after restarting the app. | Expense Entry: 100 on 2025-01-18 under Groceries | Expense should be visible in the expense list after restarting the app. | Expense visible after restart. | Pass | Expense persisted after app restart. |
| expense\_entry\_01.13 | Enter an expense with a custom category. | Amount: 200, Date: 2025-01-18, Category: Travel | Expense should be saved and shown under the custom category. | Expense saved under "Travel". | Pass | Custom category used correctly. |
| expense\_entry\_01.14 | Verify that the amount entered is reflected correctly in the expense list. | Amount: 50, Date: 2025-01-18, Category: Groceries | Amount should be displayed correctly in the expense list. | Amount displayed as 50. | Pass | Amount displayed correctly. |

**Module Name: Delete Expense**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **delete\_expense\_01** | | **Test Case Description** | To verify that users can delete an expense from the expense list. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. At least one expense entry exists in the system. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking if users can delete an expense and if the system reflects the deletion correctly. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| delete\_expense\_01.1 | Open the expense list. | None | Expense list should open. | Expense list opened. | Pass | List opened successfully. |
| delete\_expense\_01.2 | Select an expense from the list. | Expense: 100 on 2025-01-18 under Groceries | Expense should be selected. | Expense selected. | Pass | Expense selected correctly. |
| delete\_expense\_01.3 | Click the delete button for the selected expense. | Expense selected: 100 on 2025-01-18 under Groceries | Expense should be deleted from the list. | Expense deleted from the list. | Pass | Expense deleted successfully. |
| delete\_expense\_01.4 | Verify the expense is deleted from the list. | Expense: 100 on 2025-01-18 under Groceries | Expense should no longer be displayed in the list. | Expense not found in the list. | Pass | Expense deletion confirmed. |
| delete\_expense\_01.5 | Try deleting an expense without internet connection. | Expense selected: 100 on 2025-01-18 under Groceries | Error message: "No internet connection." | Error message displayed. | Pass | No internet error displayed. |
| delete\_expense\_01.6 | Verify that the deleted expense is removed after restarting the app. | Expense: 100 on 2025-01-18 under Groceries | Expense should not appear in the list after restarting the app. | Expense not visible after restart. | Pass | Expense deletion persisted after restart. |
| delete\_expense\_01.7 | Attempt to delete an expense that does not exist. | Expense selected: Non-existent expense | Error message: "Expense not found." | Error message displayed. | Pass | Error for non-existent expense handled properly. |
| delete\_expense\_01.8 | Verify that the total expense amount updates after deleting an expense. | Expense deleted: 100 on 2025-01-18 under Groceries | Total expense amount should decrease by 100. | Total expense amount updated correctly. | Pass | Total updated correctly. |
| delete\_expense\_01.9 | Try deleting an expense with a custom category. | Expense: 200 on 2025-01-18 under Travel | Expense should be deleted from the custom category. | Expense deleted from the custom category. | Pass | Custom category deletion handled correctly. |

**Module Name: Goal Creation**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **goal\_creation\_01** | | **Test Case Description** | To verify that users can set specific financial goals, including entering the target amount and deadline for the goal. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking that users can create a financial goal with valid target amount and deadline. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| goal\_creation\_01.1 | Open the "Create Goal" screen. | None | Goal creation screen should appear. | Goal creation screen appeared. | Pass | Screen appeared as expected. |
| goal\_creation\_01.2 | Enter valid goal details (target amount and deadline). | Target amount: 5000, Deadline: 2025-12-31 | Goal details should be entered in the respective fields. | Goal details entered correctly. | Pass | Details entered successfully. |
| goal\_creation\_01.3 | Click "Save" to save the goal. | Goal details: 5000, Deadline: 2025-12-31 | Goal should be saved and appear in the list of goals. | Goal saved and displayed in list. | Pass | Goal saved and displayed correctly. |
| goal\_creation\_01.4 | Verify the goal is listed with correct details. | Goal: 5000 for 2025-12-31 | Created goal should appear in the goal list with target amount and deadline. | Goal listed with correct details. | Pass | Goal displayed correctly in the list. |
| goal\_creation\_01.5 | Enter invalid goal details (e.g., negative target amount). | Target amount: -1000, Deadline: 2025-12-31 | Error message: "Invalid target amount." | Error message displayed. | Pass | Invalid target amount rejected successfully. |
| goal\_creation\_01.6 | Leave required fields blank and try to save. | No input provided | Error message: "All fields are required." | Error message displayed. | Pass | Missing fields handled correctly. |
| goal\_creation\_01.7 | Enter a deadline in the past. | Target amount: 5000, Deadline: 2020-12-31 | Error message: "Deadline must be in the future." | Error message displayed. | Pass | Past deadline rejected successfully. |
| goal\_creation\_01.8 | Verify if the created goal remains saved after restarting the app. | Goal: 5000 for 2025-12-31 | Goal should remain saved after restarting the app. | Goal persists after restart. | Pass | Data persistence verified. |
| goal\_creation\_01.9 | Try creating a goal with a custom category. | Target amount: 3000, Deadline: 2025-06-30, Category: Vacation | Goal with custom category should be created successfully. | Goal with custom category saved. | Pass | Custom category handled correctly. |
| goal\_creation\_01.10 | Attempt to create a goal without internet connection. | Goal: 5000, Deadline: 2025-12-31 | Error message: "No internet connection." | Error message displayed. | Pass | Error for no internet connection handled correctly. |

**Module Name: Edit Goal**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **goal\_edit\_01** | | **Test Case Description** | To verify that users can edit the details of an existing financial goal, including updating the target amount, deadline, and category. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. At least one financial goal already created. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking that users can successfully edit the target amount, deadline, and category of an existing goal. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| goal\_edit\_01.1 | Navigate to the "Edit Goal" screen. | Select an existing goal. | Goal editing screen should appear with current goal details. | Goal editing screen appeared with correct details. | Pass | Screen appeared as expected. |
| goal\_edit\_01.2 | Update the target amount. | New Target Amount: 6000 | Target amount should be updated to the new value. | Target amount updated correctly. | Pass | Amount updated as expected. |
| goal\_edit\_01.3 | Update the deadline. | New Deadline: 2026-12-31 | Deadline should be updated to the new value. | Deadline updated correctly. | Pass | Deadline updated as expected. |
| goal\_edit\_01.4 | Update the category. | New Category: Vacation Fund | Category should be updated to the new value. | Category updated correctly. | Pass | Category updated as expected. |
| goal\_edit\_01.5 | Save the edited goal. | Updated Goal: 6000 for 2026-12-31, Category: Vacation Fund | Goal should be saved and reflected in the goal list with updated details. | Goal saved with updated details. | Pass | Goal saved correctly with updated details. |
| goal\_edit\_01.6 | Attempt to update the goal with invalid data (e.g., negative target amount). | Target Amount: -2000, Deadline: 2026-12-31 | Error message: "Invalid target amount." | Error message displayed. | Pass | Invalid target amount rejected correctly. |
| goal\_edit\_01.7 | Leave required fields blank and try to save. | No input provided. | Error message: "All fields are required." | Error message displayed. | Pass | Blank fields handled properly. |
| goal\_edit\_01.8 | Attempt to set a deadline in the past. | Target Amount: 6000, Deadline: 2020-12-31 | Error message: "Deadline must be in the future." | Error message displayed. | Pass | Past deadline rejected successfully. |
| goal\_edit\_01.9 | Verify if the edited goal persists after restarting the app. | Edited Goal: 6000 for 2026-12-31 | Goal should persist with the updated details after restarting the app. | Goal persists with updated details after restart. | Pass | Persistence of edited goal verified. |
| goal\_edit\_01.10 | Try editing a goal without internet connection. | Target Amount: 6000, Deadline: 2026-12-31 | Error message: "No internet connection." | Error message displayed. | Pass | Error for no internet connection handled correctly. |

**Module Name: Delete Goal**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **goal\_delete\_01** | | **Test Case Description** | To verify that users can delete an existing financial goal and that it is removed from the system. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. At least one financial goal already created. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Checking that users can successfully delete a financial goal and that the goal is removed from the goal list. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| goal\_delete\_01.1 | Navigate to the "Goals" screen. | Select an existing goal. | Goal details should be displayed. | Goal details displayed. | Pass | Goal list loaded correctly. |
| goal\_delete\_01.2 | Click on the delete option for a goal. | Select goal: "Vacation Fund" | Goal should be marked for deletion. | Goal marked for deletion. | Pass | Goal selected for deletion. |
| goal\_delete\_01.3 | Confirm deletion. | Confirm deletion action. | Goal should be permanently deleted from the list. | Goal deleted successfully. | Pass | Goal deleted as expected. |
| goal\_delete\_01.4 | Verify that the goal is no longer in the goal list after deletion. | - | Goal should not appear in the goal list. | Goal removed from the list. | Pass | Goal removed successfully from the list. |
| goal\_delete\_01.5 | Attempt to delete a goal with missing internet connection. | Select goal to delete, confirm deletion. | Error message: "No internet connection." | Error message displayed. | Pass | Error for no internet connection handled correctly. |
| goal\_delete\_01.6 | Try to delete a goal that doesn’t exist. | Select non-existent goal. | Error message: "Goal not found." | Error message displayed. | Pass | Non-existent goal handled properly. |
| goal\_delete\_01.7 | Verify if the deleted goal is completely removed from the system after app restart. | - | Goal should not be present after restarting the app. | Goal is not present after restart. | Pass | Goal correctly removed after restart. |
| goal\_delete\_01.8 | Attempt to delete a goal with invalid data (e.g., incomplete data). | Select goal with incomplete data. | Error message: "Invalid goal data." | Error message displayed. | Pass | Invalid goal data rejected correctly. |

**Module Name: Create Budget**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **budget\_create\_01** | | **Test Case Description** | To verify that the system correctly creates a budget for the user based on income, expenses, and financial goals. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. User has inputted income and expenses. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Verifying that the system can create and display a budget for the user based on the financial data entered (income, expenses, goals). | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| budget\_create\_01.1 | Navigate to the "Budget" screen. | - | Budget creation option should be visible. | Budget creation option visible. | Pass | Budget creation option accessible. |
| budget\_create\_01.2 | Click on "Create Budget" button. | - | The system should start processing the budget creation. | Budget creation started. | Pass | Budget creation started successfully. |
| budget\_create\_01.3 | Enter income data (e.g., salary, freelance). | Salary: $3000, Freelance: $1000 | Income details entered successfully. | Income entered successfully. | Pass | Income entered correctly. |
| budget\_create\_01.4 | Enter expense data (e.g., rent, utilities). | Rent: $800, Utilities: $200 | Expense details entered successfully. | Expense entered successfully. | Pass | Expenses entered correctly. |
| budget\_create\_01.5 | Enter financial goals (e.g., saving for vacation). | Goal: Vacation, Target Amount: $2000, Deadline: 6 months | Goal data entered successfully. | Goal entered successfully. | Pass | Goal entered correctly. |
| budget\_create\_01.6 | Click "Generate Budget" button after entering details. | - | The system should generate a budget based on income, expenses, and goals. | Budget generated successfully. | Pass | Budget created as expected. |
| budget\_create\_01.7 | Verify that the generated budget shows accurate allocations based on income, expenses, and goals. | Income: $4000, Expenses: $1000, Goals: $2000 | Budget should reflect income, expenses, and goal amounts. | Budget reflects income, expenses, and goals correctly. | Pass | Budget values are correct. |
| budget\_create\_01.8 | Check that the budget summary is displayed with categories like total income, total expenses, and savings. | - | Budget summary should display income, expenses, savings. | Budget summary displayed with all categories. | Pass | Summary displayed correctly. |
| budget\_create\_01.9 | Verify if the budget can be edited after creation. | Edit income/expenses. | The system should allow edits to the created budget. | Budget edited successfully. | Pass | Budget edited as expected. |
| budget\_create\_01.10 | Verify if the budget is saved and retained after app restart. | - | Budget should be retained after restarting the app. | Budget saved and retained. | Pass | Budget retained correctly. |
| budget\_create\_01.11 | Attempt to create a budget with missing income or expense data. | Missing income or expense. | Error message: "Income/Expense data missing." | Error message displayed. | Pass | Missing data error handled. |
| budget\_create\_01.12 | Try to create a budget without a valid internet connection. | - | Error message: "No internet connection." | Error message displayed. | Pass | Internet connection error handled. |

**Module Name: View Articles and Videos**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **articles\_videos\_view\_01** | | **Test Case Description** | To verify that users can view articles and watch videos with effective tips within the app. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Verifying that users can view articles and videos for financial tips and advice, ensuring proper content display. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| articles\_videos\_view\_01.1 | Navigate to the "Articles & Videos" section. | - | The section containing articles and videos should be displayed. | Articles and videos section displayed. | Pass | Section visible and accessible. |
| articles\_videos\_view\_01.2 | Click on an article to read. | Article title: "Budgeting Tips for Beginners" | The article should open and display its content. | Article opened and content displayed. | Pass | Article displayed as expected. |
| articles\_videos\_view\_01.3 | Click on a video to watch. | Video title: "Saving for Retirement" | The video player should open, and the video should play. | Video played successfully. | Pass | Video played without issues. |
| articles\_videos\_view\_01.4 | Scroll through the list of available articles. | - | The list of articles should be scrollable and show more articles as needed. | Article list is scrollable. | Pass | Scrolling works fine. |
| articles\_videos\_view\_01.5 | Scroll through the list of available videos. | - | The list of videos should be scrollable and show more videos as needed. | Video list is scrollable. | Pass | Scrolling works fine. |
| articles\_videos\_view\_01.6 | Verify that the article content includes proper formatting (headings, paragraphs, etc.). | - | Article content should be formatted correctly, with headings, paragraphs, etc. | Article formatting is correct. | Pass | Article is formatted as expected. |
| articles\_videos\_view\_01.7 | Verify that the video player has basic controls (play, pause, volume, full-screen). | - | Video player should have play, pause, volume control, and full-screen option. | Video player controls available. | Pass | Controls working correctly. |
| articles\_videos\_view\_01.8 | Try to view an article or video with no internet connection. | - | Error message: "No internet connection." | Error message displayed. | Pass | Handled no internet scenario. |
| articles\_videos\_view\_01.9 | Verify that the article or video is fully loaded and viewable without interruptions. | - | The content should load completely without interruption. | Article/video loaded completely. | Pass | Content loaded without issues. |
| articles\_videos\_view\_01.10 | Attempt to access an article or video with missing content or errors. | - | Error message: "Content unavailable." | Error message displayed. | Pass | Error handling for missing content. |
| articles\_videos\_view\_01.11 | Try viewing an article or video with a slow internet connection. | - | The article or video should buffer and display properly. | Article/video buffered and displayed. | Pass | Content buffered as expected. |

**Module Name: Bookmark Sources**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **bookmark\_sources\_01** | | **Test Case Description** | To verify that users can bookmark articles and videos for easy access later. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Verifying that users can successfully bookmark articles and videos and access them from the bookmark section. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| bookmark\_sources\_01.1 | Navigate to the "Articles & Videos" section. | - | The section containing articles and videos should be displayed. | Section displayed. | Pass | Section accessible. |
| bookmark\_sources\_01.2 | Select an article or video to bookmark. | Article title: "Budgeting Tips" | Bookmark icon should be visible on the article/video. | Bookmark icon visible. | Pass | Bookmark icon visible. |
| bookmark\_sources\_01.3 | Click the bookmark icon on an article/video. | Article/video selected | The article/video should be added to the "Bookmarked" section. | Article/video added to bookmarks. | Pass | Bookmark functionality works. |
| bookmark\_sources\_01.4 | Navigate to the "Bookmarked" section. | - | The bookmarked articles and videos should be listed here. | Bookmarked content listed. | Pass | Bookmarked content displayed correctly. |
| bookmark\_sources\_01.5 | Click on a bookmarked article or video. | Bookmarked article/video | The article or video should open and display its content. | Article/video opened correctly. | Pass | Opened from bookmarks without issues. |
| bookmark\_sources\_01.6 | Try bookmarking the same article/video multiple times. | Article/video title: "Saving for Retirement" | The article/video should only be bookmarked once. Duplicate bookmarking should not occur. | Only one bookmark created. | Pass | Duplicate bookmarking prevented. |
| bookmark\_sources\_01.7 | Remove an article/video from the "Bookmarked" section. | Article/video in bookmarks | The article/video should be removed from the "Bookmarked" section. | Article/video removed. | Pass | Bookmark removal works correctly. |
| bookmark\_sources\_01.8 | Attempt to bookmark an article/video with no internet connection. | - | Error message: "No internet connection." | Error message displayed. | Pass | Error handling for no internet connection. |
| bookmark\_sources\_01.9 | Verify that bookmarked articles/videos are available after app restart. | - | The bookmarked articles/videos should persist after restarting the app. | Bookmarked content available after restart. | Pass | Persistence tested successfully. |
| bookmark\_sources\_01.10 | Verify that the number of bookmarks is correctly updated after bookmarking/removing content. | - | The bookmark count should reflect accurately when items are added or removed. | Bookmark count updated. | Pass | Bookmark count accurate. |

**Module Name: Access Bookmarks**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **access\_bookmarks\_01** | | **Test Case Description** | To verify that users can access their bookmarked articles and videos. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. At least one article or video has been bookmarked. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Verifying that users can access their bookmarked articles and videos from the "Bookmarked" section. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| access\_bookmarks\_01.1 | Navigate to the "Bookmarked" section. | - | The section containing all bookmarked articles and videos should be displayed. | Bookmarked section displayed. | Pass | Bookmarked content is visible. |
| access\_bookmarks\_01.2 | Click on a bookmarked article/video. | Bookmarked article/video title: "Budgeting Tips" | The article/video should open and display its content. | Article/video opened correctly. | Pass | Opened from the bookmarked list. |
| access\_bookmarks\_01.3 | Verify that a recently bookmarked article/video appears in the "Bookmarked" section. | Article/video just bookmarked | The article/video should immediately appear in the "Bookmarked" section. | New bookmark displayed immediately. | Pass | Bookmark appears instantly. |
| access\_bookmarks\_01.4 | Remove an article/video from the "Bookmarked" section. | Bookmarked article/video title: "Saving for Retirement" | The article/video should be removed from the "Bookmarked" section. | Article/video removed. | Pass | Removed from bookmarks. |
| access\_bookmarks\_01.5 | Attempt to access a bookmarked article/video after it has been removed. | Article/video removed previously | Error message: "This bookmark is no longer available." | Error message displayed. | Pass | Correct message displayed for removed items. |
| access\_bookmarks\_01.6 | Verify that the "Bookmarked" section remains accessible even after restarting the app. | - | The "Bookmarked" section and its contents should be available after the app is restarted. | Bookmarked section available post-restart. | Pass | Persistence works as expected. |
| access\_bookmarks\_01.7 | Verify that the number of bookmarks is correctly updated after adding/removing items. | - | The bookmark count should reflect the correct number after items are added or removed. | Bookmark count updated. | Pass | Bookmark count accurate. |
| access\_bookmarks\_01.8 | Attempt to access bookmarks with no internet connection. | - | Error message: "No internet connection." | Error message displayed. | Pass | Proper error handling for no internet connection. |

**Module Name: Upload Articles/Videos**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **upload\_articles\_videos\_01** | | **Test Case Description** | To verify that users can upload articles or videos to the system. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. User has necessary permissions to upload articles or videos. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Verifying that users can upload articles or videos successfully. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| upload\_articles\_videos\_01.1 | Navigate to the "Upload" section. | - | The upload page/section should open. | Upload section displayed. | Pass | Page opened successfully. |
| upload\_articles\_videos\_01.2 | Select the type of content to upload (Article/Video). | Option selected: "Article" | The system should allow uploading an article. | Article upload option shown. | Pass | Selection displayed correctly. |
| upload\_articles\_videos\_01.3 | Upload a valid article (text or file). | Article title: "Saving Tips" Article content: "Tips for better savings." | The article should be uploaded successfully. | Article uploaded successfully. | Pass | Article uploaded and available in the system. |
| upload\_articles\_videos\_01.4 | Upload a valid video (video file). | Video title: "How to Budget" Video file: "budgeting.mp4" | The video should be uploaded successfully. | Video uploaded successfully. | Pass | Video uploaded and playable. |
| upload\_articles\_videos\_01.5 | Try to upload an unsupported file type (e.g., PDF when expecting video). | File: "document.pdf" | Error message: "Invalid file type. Please upload a supported file." | Error message displayed. | Pass | Error message shown for unsupported file types. |
| upload\_articles\_videos\_01.6 | Upload an article without required fields (e.g., title or content missing). | Article title: "" Article content: "" | Error message: "Title and content are required." | Error message displayed. | Pass | Proper validation for missing fields. |
| upload\_articles\_videos\_01.7 | Upload a video with a large file size that exceeds the system limit. | File: "large\_video.mp4" (above size limit) | Error message: "File size exceeds limit." | Error message displayed. | Pass | System handles file size limit properly. |
| upload\_articles\_videos\_01.8 | Attempt to upload content without an internet connection. | - | Error message: "No internet connection." | Error message displayed. | Pass | Proper error handling for no internet connection. |
| upload\_articles\_videos\_01.9 | Verify that the uploaded article/video is visible in the content library after uploading. | Uploaded content: "Saving Tips" (Article) / "How to Budget" (Video) | The uploaded article/video should appear in the content library. | Uploaded content visible in library. | Pass | Content correctly displayed in the library. |

**Module Name: Detailed Reports**

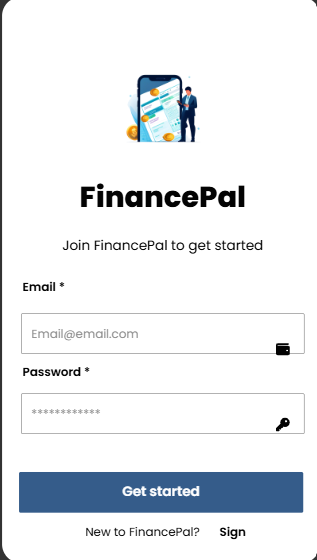
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **detailed\_reports\_01** | | **Test Case Description** | To verify the generation of detailed reports for selected timeframes, showing income, expense trends, and budget performance. | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. Income, expenses, and budgets already added in the system. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Verifying that users can generate accurate, detailed reports for selected timeframes, including income, expenses, and budget performance. | |

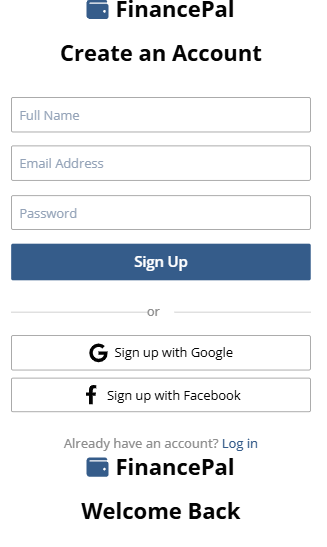
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| detailed\_reports\_01.1 | Navigate to the "Reports" section in the app. | - | The "Reports" page/section should open. | Reports section displayed. | Pass | Page loaded successfully. |
| detailed\_reports\_01.2 | Select a specific timeframe for the report. | Timeframe: "Last Month" | The system should filter data for the selected timeframe. | Timeframe data filtered successfully. | Pass | Timeframe selection working correctly. |
| detailed\_reports\_01.3 | Generate a report with income trends for the selected timeframe. | Selected option: "Income Trends" | A detailed income trend report is generated. | Income trend report generated successfully. | Pass | Income trends displayed accurately. |
| detailed\_reports\_01.4 | Generate a report with expense trends for the selected timeframe. | Selected option: "Expense Trends" | A detailed expense trend report is generated. | Expense trend report generated successfully. | Pass | Expense trends displayed accurately. |
| detailed\_reports\_01.5 | Generate a report showing budget performance for the selected timeframe. | Selected option: "Budget Performance" | A detailed budget performance report is generated. | Budget performance report generated successfully. | Pass | Budget performance data displayed correctly. |
| detailed\_reports\_01.6 | Attempt to generate a report without selecting a timeframe. | Timeframe: "None" | Error message: "Please select a timeframe to generate a report." | Error message displayed. | Pass | Proper validation for missing timeframe. |
| detailed\_reports\_01.7 | Attempt to generate a report when no income/expenses exist for the selected timeframe. | Timeframe: "Last Year" (no data) | Message: "No data available for the selected timeframe." | Message displayed correctly. | Pass | Proper handling for missing data. |
| detailed\_reports\_01.8 | View graphical representation of trends (e.g., pie charts, bar graphs). | Selected option: "View Graphs" | Graphs for income and expense trends are displayed. | Graphs displayed successfully. | Pass | Visualizations are clear and accurate. |
| detailed\_reports\_01.9 | Generate a combined report showing both income and expense trends for a custom timeframe. | Timeframe: "01 Jan 2023 - 31 Dec 2023" | A detailed combined report is generated. | Combined report generated successfully. | Pass | Combined trends displayed correctly. |
| detailed\_reports\_01.10 | Attempt to generate a report with an invalid timeframe (e.g., start date after end date). | Timeframe: "01 Jan 2024 - 31 Dec 2023" | Error message: "Invalid timeframe. Start date must be before end date." | Error message displayed. | Pass | Proper validation for invalid timeframes. |

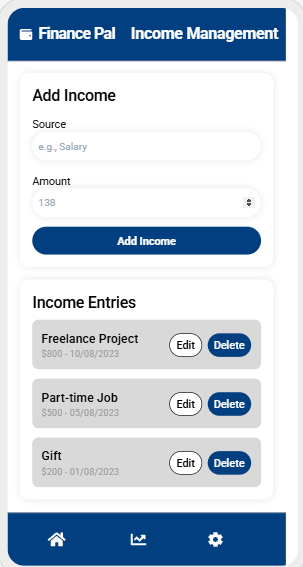
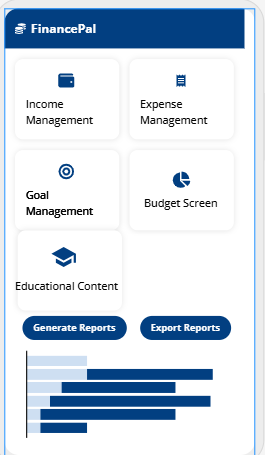
**Module Name: Export Reports**

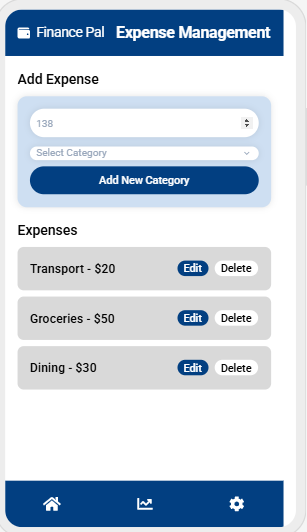
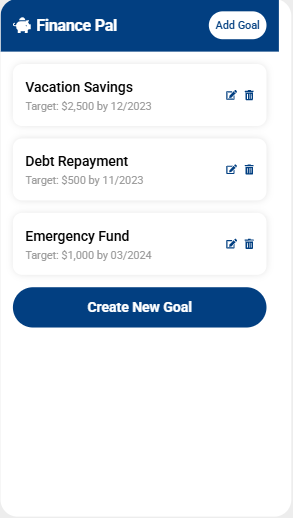
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID** | **export\_reports\_01** | | **Test Case Description** | To verify the functionality of exporting reports in various formats (e.g., PDF, Excel). | | **Prerequisites** | 1. Stable internet connection. 2. Mobile device with the app installed. 3. User logged in with a valid account. 4. Detailed reports successfully generated and available for export. | | **Tester's Name** | Rabia | | **Environmental Information** | 1. OS: Android/iOS 2. Device: Smartphone/Tablet. | | **Test Scenario** | Verifying that users can export detailed reports in different formats and save them to their device or share them. | |

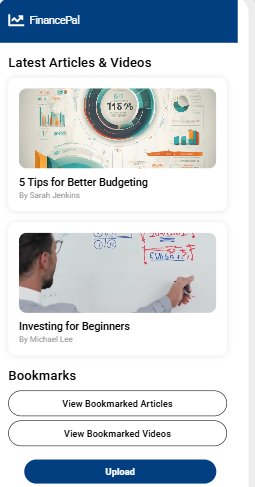
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Test Input** | **Expected Results** | **Actual Results** | **Status** | **Comments** |
| export\_reports\_01.1 | Navigate to the "Reports" section and generate a report. | Timeframe: "Last Month" | The report is generated and displayed. | Report displayed successfully. | Pass | Report generation is a prerequisite. |
| export\_reports\_01.2 | Select the "Export" option for the generated report. | Selected option: "Export" | Export options (PDF, Excel) are displayed. | Export options displayed successfully. | Pass | Export options are accessible. |
| export\_reports\_01.3 | Choose the export format as PDF. | Format: "PDF" | The report is downloaded as a PDF file. | PDF file downloaded successfully. | Pass | PDF format works correctly. |
| export\_reports\_01.4 | Choose the export format as Excel. | Format: "Excel" | The report is downloaded as an Excel file. | Excel file downloaded successfully. | Pass | Excel format works correctly. |
| export\_reports\_01.5 | Attempt to export a report when no data exists for the selected timeframe. | Timeframe: "No Data" | Error message: "No data available to export." | Error message displayed. | Pass | Proper handling for missing data. |
| export\_reports\_01.6 | Select the "Share" option after exporting the report. | Option: "Share" | Share options (e.g., email, messaging apps) are displayed. | Share options displayed successfully. | Pass | Sharing functionality is working. |
| export\_reports\_01.7 | Attempt to export a report without generating one first. | No report generated | Error message: "Please generate a report before exporting." | Error message displayed. | Pass | Proper validation for missing reports. |
| export\_reports\_01.8 | Verify that the exported file is saved to the device's storage. | Format: "PDF" | File is saved in the designated downloads folder. | File saved successfully. | Pass | File storage path works as expected. |
| export\_reports\_01.9 | Attempt to export a report while offline. | Internet: "Offline" | Error message: "Internet connection is required to export reports." | Error message displayed. | Pass | Offline restriction handled properly. |
| export\_reports\_01.10 | Export a combined report showing income and expense trends. | Timeframe: "01 Jan 2023 - 31 Dec 2023", Format: "PDF" | The combined report is exported as a PDF file. | Combined report exported successfully. | Pass | Combined export functionality works. |

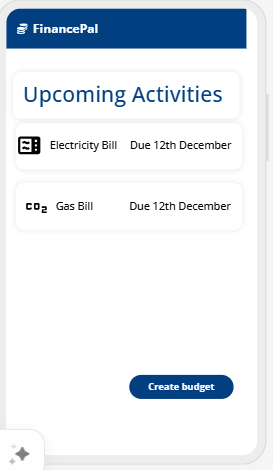




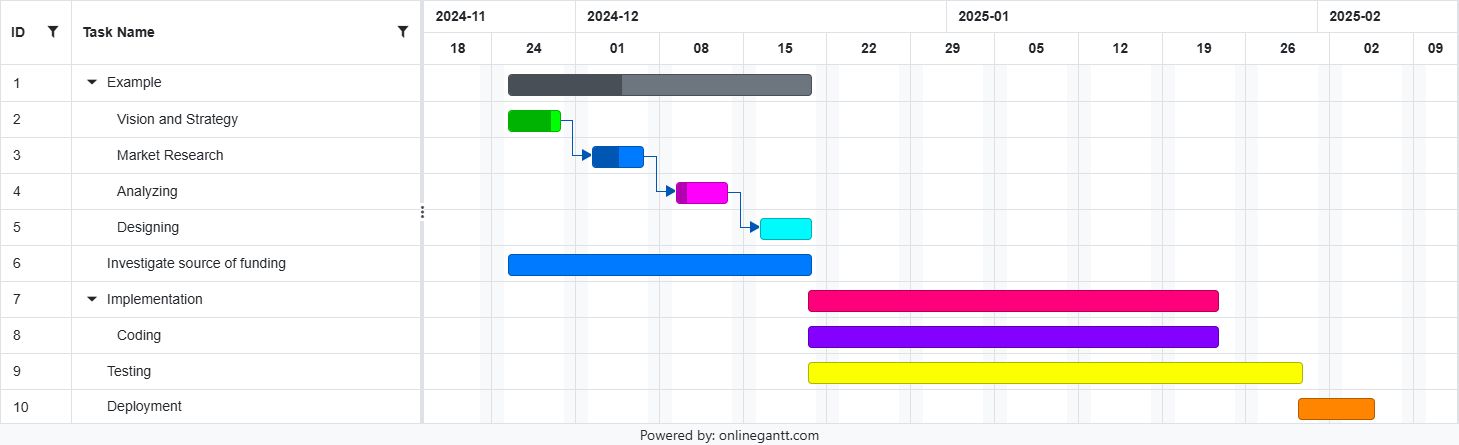








Gantt Chart:



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Level** | **ID** | **Task Name** | **Start Date** | **End Date** | **Notes** | | **1** | 1 | Daily Paws Project | 2024-11-18 | 2025-01-09 | Overall project timeline. | | **2** | 2.1 | Phase 1: Planning | 2024-11-18 | 2024-11-29 | Planning and strategy tasks. | | **3** | 2.1.1 | Define project scope and objectives | 2024-11-18 | 2024-11-20 | Identify key objectives and scope. | | **3** | 2.1.2 | Identify stakeholders and resources | 2024-11-21 | 2024-11-23 | Assess roles and available resources. | | **3** | 2.1.3 | Develop project timelines and milestones | 2024-11-24 | 2024-11-26 | Create roadmap with deadlines. | | **2** | 3.1 | Phase 2: Design | 2024-11-30 | 2024-12-09 | UI/UX design and workflows. | | **3** | 3.1.1 | Design Login/Signup/Forget Password screens | 2024-11-30 | 2024-12-02 | Related to F.R. 1–4. | | **3** | 3.1.2 | Design Home Dashboard and Income Layouts | 2024-12-03 | 2024-12-05 | Related to F.R. 5–9. | | **3** | 3.1.3 | Create wireframes for Goals and Budgeting | 2024-12-06 | 2024-12-09 | Related to F.R. 14–17. | | **2** | 4.1 | Phase 3: Development | 2024-12-10 | 2024-12-30 | Coding core features. | | **3** | 4.1.1 | Implement Login/Signup/Forget Password features | 2024-12-10 | 2024-12-14 | Covers F.R. 1–4. | | **3** | 4.1.2 | Add income, expenses, and categorization | 2024-12-15 | 2024-12-20 | Covers F.R. 7–13. | | **3** | 4.1.3 | Develop Goal Creation and Budgeting | 2024-12-21 | 2024-12-30 | Covers F.R. 14–17. | | **2** | 5.1 | Phase 4: Testing | 2025-01-02 | 2025-01-06 | Ensuring system functionality. | | **3** | 5.1.1 | Perform unit and integration testing | 2025-01-02 | 2025-01-03 | Validate F.R. and N.F.R. compliance. | | **3** | 5.1.2 | Conduct user acceptance testing (UAT) | 2025-01-04 | 2025-01-05 | Test user experience and functionality. | | **3** | 5.1.3 | Ensure data security and encryption standards | 2025-01-06 | 2025-01-06 | Validate N.F.R. 3–4 compliance. | | **2** | 6.1 | Phase 5: Deployment | 2025-01-07 | 2025-01-09 | Deploying the application. | | **3** | 6.1.1 | Launch app to app stores | 2025-01-07 | 2025-01-07 | Submit app to Apple and Google stores. | | **3** | 6.1.2 | Monitor app performance and feedback | 2025-01-08 | 2025-01-09 | Analyze usage and resolve issues. | |