University of Houston-Victoria

Department of Computer and Information Science

Group Project

Expense Tracker

Software Requirements Specification

Prepared for Partial Fulfillment of the Course

COSC 6342 Software Engineering Project Management

Spring 2024

By

Aradhana Sharma (2250955)

Kapil Suryawanshi (2039409)

Shayan Khan (2244554)

Suraj Odedra (2329291)

Advisor: Dr. Amjad Nusayr

Director of UG Computer Science & Computer Information Systems  
Associate Professor of Computer Science

University of Houston Victoria

Table of Contents

[**I.** **Introduction** 3](#_Toc160376346)

[**II.** **Specification** 5](#_Toc160376347)

[**A.** **Glossary** 5](#_Toc160376348)

[**B.** **User Requirement Definitions** 6](#_Toc160376349)

[**C.** **System Requirement Specifications** 7](#_Toc160376350)

[**D.** **Scenario for 3 user requirements** 18](#_Toc160376351)

[**1.** **Scenario 1: User Self-Registration** 18](#_Toc160376352)

[**2.** **Scenario 2: Creating and managing shared group expenses.** 18](#_Toc160376353)

[**3.** **Scenario 3: User creates a monthly budget.** 19](#_Toc160376354)

[**E.** **Functional requirements & prioritization** 20](#_Toc160376355)

[**F.** **Structured approach for top 5 requirements** 21](#_Toc160376356)

[**1.** **User Registration** 21](#_Toc160376357)

[**2.** **Category Management** 21](#_Toc160376358)

[**3.** **Budget Management** 22](#_Toc160376359)

[**4.** **Transaction Management** 23](#_Toc160376360)

[**5.** **Group Management** 24](#_Toc160376361)

[**G.** **Use case diagrams** 26](#_Toc160376362)

[**1.** **Use Case: User Management Diagram** 26](#_Toc160376363)

[**2.** **Use Case: Dashboard Diagram** 27](#_Toc160376364)

[**3.** **Use Case: Category Management Diagram** 29](#_Toc160376365)

[4. **Use Case: Budget Management Diagram** 31](#_Toc160376366)

[**5.** **Use Case: Transaction Management Diagram** 34](#_Toc160376367)

[**6.** **Use Case: Group Management Diagram** 37](#_Toc160376368)

[**7.** **Use Case: Group Transaction Management Diagram** 40](#_Toc160376369)

[**III.** **Traceability Matrix** 43](#_Toc160376370)

# **Introduction**

Welcome to the "Expense Tracker" project, where we are dedicated to revolutionizing the landscape of personal finance management. In today's fast-paced world, where financial transactions occur at the speed of a click and expenses can quickly accumulate, maintaining a clear understanding of one's finances and staying disciplined in spending habits can often feel like an overwhelming challenge. It is in response to these modern-day complexities that we have embarked on a journey to develop an innovative software solution that simplifies the process of financial management, empowers users with valuable insights, and ultimately fosters greater financial well-being.

Our primary objective with the "Expense Tracker" is to provide individuals and groups alike with a comprehensive and user-friendly platform that not only facilitates the tracking of expenses but also equips users with the tools and resources they need to make informed financial decisions and achieve their long-term financial goals. By leveraging advanced technologies and intuitive design principles, we aim to streamline the often tedious and time-consuming tasks associated with budgeting, expense tracking, and financial planning, thereby empowering our users to take control of their financial futures.

At the core of the "Expense Tracker" lies a robust set of features and functionalities designed to address the diverse needs and preferences of our user base. From the ability to easily categorize expenses and set personalized budgets to tracking spending in real-time and visualizing financial data through interactive reports and charts, our platform offers a comprehensive suite of tools to help users gain a clearer understanding of their financial habits and patterns. Moreover, our innovative group expense management feature allows users to seamlessly split expenses and track shared costs, making it ideal for families, roommates, and other collaborative financial arrangements.

What sets the "Expense Tracker" apart is not only its breadth of features but also its commitment to personalized experiences and user-centric design. Whether you're a student managing a tight budget, a family juggling multiple expenses, or part of a shared household seeking to streamline financial management, our platform is tailored to meet your specific needs and preferences. By providing intuitive interfaces, customizable settings, and personalized recommendations, we strive to ensure that every user has a seamless and rewarding experience with our platform.

We are keenly aware of the challenges that lie ahead, from ensuring robust data security and privacy to fostering user adoption and scalability. However, with a dedicated team of professionals and a shared commitment to excellence, we are confident in our ability to overcome these obstacles and deliver a solution that truly redefines financial management for the modern age.

# **Specification**

## **Glossary**

|  |  |
| --- | --- |
| **Term** | **Description** |
| Budget | A predefined amount of money allocated for a specific purpose or period. |
| Category | A classification or group used to organize transactions based on their nature or purpose. |
| Cash In | Money received or earned as income or proceeds from a transaction. |
| Cash Out | Money spent or paid as an expense or cost for goods or services. |
| Group | A collection of individuals who share common interests or responsibilities. |
| Group Transactions | Financial transactions related to expenses shared among members of a group. |
| Monthly Overview Visualization | A graphical representation of financial data summarizing transactions over a month. |
| Status of Settlement | The current state or progress of settling financial obligations within a group. |
| Transaction | A financial exchange involving the transfer of money or assets between parties. |
| Transaction Entry | A record or entry in the system documenting a transaction. |
| User | An individual who interacts with the system to perform various tasks or functions. |
| External User | An individual who wants to interacts with the system to perform various tasks or functions, but does not have a registered account in the system yet. |
| System User | An individual who wants to interacts with the system to perform various tasks or functions, and has a registered account in the system. |
| Expense | Money spent or paid |
| Income | Money received or earned |

## **User Requirement Definitions**

| **ID** | **Area** | **User Requirement** |
| --- | --- | --- |
| UR-01 | User | As an external user, I should be able to self-register, so that I can access and work with it. |
| UR-02 | User | As a system user, if I am already registered in system, I should be able login to the system, so that I can access and work with it. |
| UR-03 | Dashboard Transaction | As a system user, I should be able to get monthly overview visualization of my transactions (cash in/out) grouped in categories against budget, so that I am enabled to strategies my behavior towards maximum saving. |
| UR-04 | Dashboard Group Transaction | As a system user, I should be able to get overview visualization of my group transactions (cash in/out) against the person responsible, so that I understand how much money I owe and reimburse to individual in the group. |
| UR-05 | Category | As a system user, I should be able to create transaction categories, so that I will be able to group of my transactions to get summary. |
| UR-06 | Category | As a system user, I should be able to view all the transaction categories created by me, so that I will be able to know which categories I have already created. |
| UR-07 | Category | As a system user, I should be able to edit any transaction category created by me, so that I will be able to update the necessary details of the transaction category. |
| UR-08 | Category | As a system user, I should be able to delete a transaction category created by me, if there are not transactions recorded against it, so that I will be able to remove unnecessary transaction categories. |
| UR-09 | Budget | As a system user, I should be able to create a monthly budget against category, so that I will be able to understand how much budget I am consuming and is available during the month. |
| UR-10 | Budget | As a system user, I should be able to view all my monthly budgets against category, so that I will be able to understand how much budget I have planned for each month. |
| UR-11 | Budget | As a system user, I should be able to edit my monthly budgets against category, so that I will be able to adjust the monthly planned budget when needed. |
| UR-12 | Budget | As a system user, I should be able to delete my monthly budgets against category, so that I will be able to remove any budget planning if I need to. |
| UR-13 | Transaction | As a system user, I should be able to create a transaction entry against category, so that I will be able to track my transaction in the system. |
| UR-14 | Transaction | As a system user, I should be able to view all my monthly transaction entries against categories, so that I will be able to understand how much money I have spent/received and when. |
| UR-15 | Transaction | As a system user, I should be able to edit any of my monthly transaction entries, so that I will be able to correct any mistake in recording of the transactions. |
| UR-16 | Transaction | As a system user, I should be able to delete any of my monthly transaction entries, so that I will be able to remove any unnecessary transactions. |
| UR-17 | Group | As a system user, I should be able to create a group of members, so that I can track the shared expenses for the group. |
| UR-18 | Group | As a system user, I should be able to view all the groups I have created or where I have been added as member, so that I know which all shared events I am part of. |
| UR-19 | Group | As a system user, I should be able to edit all the groups I have created, so that I can update details on the group like title, description and membership in the group. |
| UR-20 | Group | As a system user, I should be able to delete the groups I have created, if there are no group transactions recorded against it, so that I can remove the unnecessary groups from the system. |
| UR-21 | Group Transaction | As a group member, I should be able to create group transactions, stating how much money I have spent on behalf of which group members, so that I can track my expenses in a shared event. |
| UR-22 | Group Transaction | As a group member, I should be able to view all the group transactions, either created by me or by any other group members, so that I am aware of the expenses in a shared event. |
| UR-23 | Group Transaction | The system should provide status of settlement within the group (which transfers are not yet settled), so that all the group members are aware of their part of the responsibility to settle the transfer. |
| UR-24 | Group Transaction | As a group member, I should be able to edit all the group transactions that I have created, so that I can correct any mistake in recording of the expenses in a shared event. |
| UR-25 | Group Transaction | As a group member, I should be able to delete all the group transactions that I have created, so that I can remove any unnecessary group transaction in a shared event. |

## **System Requirement Specifications**

Following are the system requirement specifications for these user requirement definitions.

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-01 Create User | FR-01 | The system shall provide a user registration feature accessible from the application interface. |
| FR-02 | The registration process shall require users to provide information such as first name, last name, email address, and password. |
| FR-03 | The system shall validate that the mandatory field first name, last name, email address and password are provided before accepting this information for processing. |
| FR-04 | The system shall validate that the lengths for the information provided in each of the fields does not exceed 50 characters before accepting this information for processing. |
| FR-05 | The system shall validate that the email address provided is unique in the database before accepting this information for processing. |
| FR-06 | Upon successful validation, the system shall store user credentials securely in the database followed by registration successful confirmation back to user. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-02 Login User | FR-07 | The system shall provide a user login feature accessible from the application interface. |
| FR-08 | The login process shall require users to input their registered email address and password. |
| FR-09 | The system shall validate that the mandatory fields email address and password are provided before accepting this information for processing. |
| FR-10 | The system shall ensure that the password field will not show the user typed password on screen. |
| FR-11 | The system shall authenticate user credentials against stored records in the database. |
| FR-12 | Upon successful authentication, the system shall grant access to the user's account dashboard. |
| FR-13 | The system shall implement session management to maintain user authentication during their browsing session. |
| FR-14 | Logged-in users shall have access to the system until user signs out from the system. |

| **Derived From** | **Id** | **System Requirement Specification** |
| --- | --- | --- |
| UR-03 Read Dashboard Transaction | FR-15 | The system shall provide a transaction dashboard feature accessible to users for viewing monthly overviews. |
| FR-16 | The dashboard will default to the current year-month's visualization, while user has options to switch other year-month as per the data he has created in the system. |
| FR-17 | Monthly overviews shall include visualizations depicting cash inflow and outflow transactions categorized according to predefined expense and income categories. |
| FR-18 | The system shall integrate budget data into the visualization to allow users to compare their actual spending and earnings against budgeted amounts. |
| FR-19 | Visualization components shall include charts or graphs displaying transaction amounts by category and budget comparison metrics. |
| FR-20 | The system shall ensure that the visualization accurately reflects the user's transaction data and budget allocations based on the latest information available in the system. |
| FR-21 | Access to the monthly overview visualization feature shall be available to registered users upon logging into their accounts. |
| FR-22 | User will only be able to visualize his transaction data in the system. |
| NFR-01 | The transaction dashboard visualization should be loaded within 5 seconds. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-04 Read Dashboard Group Transaction | FR-23 | The system shall provide a group transaction dashboard feature accessible to users for viewing group transaction overviews. |
| FR-24 | Group transaction overviews shall include visualizations depicting cash inflow and outflow transactions categorized by the person responsible. |
| FR-25 | The system shall calculate and display the net amount owed or owed to each group member based on their transactions. |
| FR-26 | Visualization components shall include bar charts displaying transaction amounts by the responsible person and net balances. |
| FR-27 | The system shall ensure that the visualization accurately reflects the group's transaction data and net balances based on the latest information available in the system. |
| FR-28 | Access to the group transaction overview visualization feature shall be available to registered users upon logging into their accounts |
| NFR-02 | The group transaction dashboard visualization should be loaded within 5 seconds. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-05 Create Category | FR-29 | The system shall provide a feature accessible to users for creating transaction categories. |
| FR-30 | The system shall accept category title and description value for creation of category. |
| FR-31 | The system shall validate the title to be mandatory and not exceed 50 characters. |
| FR-32 | The system shall validate the description to be mandatory and not exceed 100 characters. |
| FR-33 | The system shall store the category title and description in database with unique identifier for the category. |
| FR-34 | The system shall ensure that the category title is unique (case-insensitive) compared to existing categories created by the user. |
| FR-35 | Access to the transaction category creation feature shall be available to registered users upon logging into their accounts. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-06 Read Category | FR-36 | Access to the transaction category’s view shall be available to registered users upon logging into their accounts. |
| FR-37 | The system shall provide a feature accessible to users for viewing their created transaction categories. |
| FR-38 | Upon accessing the transaction category’s view, the system shall display a list or grid layout of all categories created by the user. |
| FR-39 | The grid layout will be sorted by category title in ascending order. |
| FR-40 | In the grid layout, in a category row, the last columns will have actions to edit and delete the category. |
| FR-41 | Each transaction category entry shall include relevant details such as category title and description. |
| FR-42 | The system shall ensure that only transaction categories created by the user are displayed in the view. |
| NFR-03 | The category listing table with up to 100 record must loaded within 5 seconds. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-07 Update Category | FR-43 | Access to the transaction category editing feature shall be available to registered users upon logging into their accounts. |
| FR-44 | The system shall provide an editing feature accessible to users for modifying their created transaction categories. |
| FR-45 | Upon selecting a transaction category for editing, the system shall present a form or interface allowing users to update category details. |
| FR-46 | The system shall accept category title and description value for editing of category. |
| FR-47 | The system shall provide the current category's title and description already set in the form field. |
| FR-48 | User shall click Save button in order to commit his changes against the category. |
| FR-49 | The system shall validate the title to be mandatory and not exceed 50 characters. |
| FR-50 | The system shall validate the description to be mandatory and not exceed 100 characters. |
| FR-51 | The system shall ensure that the category title for this edited category is unique compared to existing categories created by the user. |
| FR-52 | The system shall ensure that changes made to transaction categories are accurately reflected in the database. |
| FR-53 | The system shall confirm the successful category update operation. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-08 Delete Category | FR-54 | Access to the transaction category deletion feature shall be available to registered users upon logging into their accounts. |
| FR-55 | The system shall provide a deletion feature accessible to users for removing transaction categories. |
| FR-56 | Users shall be able to select a transaction category for deletion from the list of categories they have created. |
| FR-57 | Before allowing deletion, the system shall check if any transactions are recorded against the selected category. |
| FR-58 | If no transactions are found associated with the category, the system shall permit the deletion action. |
| FR-59 | If transactions are recorded against the category, the system shall prevent the deletion action by notifying the user. |
| FR-60 | Upon user confirmation for deletion, the system shall remove the selected transaction category from the database. |
| FR-61 | The system shall confirm the successful category delete operation. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-09 Create Budget | FR-62 | Access to the budget creation feature shall be available to registered users upon logging into their accounts. |
| FR-63 | The system shall provide a feature for users to create budgets. |
| FR-64 | The system shall accept an existing category, budget amount, date, and optional notes for the creation of a budget. |
| FR-65 | The system shall validate the budget amount to be a positive number with two decimal places. |
| FR-66 | The system shall store the budget details in the database with a unique identifier for the budget. |
| FR-67 | The system shall ensure that the budget is associated with a valid category that exists in the CATEGORY table. |

| **Derived From** | **Id** | **System Requirement Specification** |
| --- | --- | --- |
| UR-10 Read Budget | FR-68 | Registered users shall have access to view their budgets upon logging into their accounts. |
| FR-69 | The system shall display a list or grid layout of all budgets associated with the user's categories. |
| FR-70 | The system shall sort the budgets by date in descending order. |
| FR-71 | The system shall provide details such as category, budget amount, date, and notes in the budget view. |
| FR-72 | The system shall ensure that only budgets associated with the user's categories are displayed. |
| NFR-04 | The budget listing table with up to 100 record must loaded within 5 seconds. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-11 Update Budget | FR-73 | The system shall provide an editing feature for users to modify their budgets. |
| FR-74 | Upon selecting a budget for editing, the system shall present a form with the current budget details. |
| FR-75 | The system shall accept changes to the budget amount, date, and notes. |
| FR-76 | The system shall validate the updated budget amount to be a positive number with two decimal places. |
| FR-77 | The system shall ensure that the updated budget is still associated with a valid category. |
| FR-78 | The system shall confirm the successful update of the budget. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-12 Delete Budget | FR-79 | The system shall provide a deletion feature for users to remove budgets. |
| FR-80 | Users shall be able to select a budget for deletion from their list of budgets. |
| FR-81 | The system shall check if any transactions are recorded against the selected budget's category before deletion. |
| FR-82 | If no transactions are found, the system shall permit the deletion action. |
| FR-83 | If transactions are found, the system shall prevent deletion by notifying the user. |
| FR-84 | Upon user confirmation for deletion, the system shall remove the budget from the database. |
| FR-85 | The system shall confirm the successful budget deletion operation. |

| **Derived From** | **Id** | **System Requirement Specification** |
| --- | --- | --- |
| UR-13 Create Transaction | FR-86 | Access to the transaction recording feature shall be available to registered users. |
| FR-87 | The system shall provide a feature for users to record transactions. |
| FR-88 | The system shall accept transaction details including type, category, date, amount, and optional notes. |
| FR-89 | The system shall validate the transaction amount to be a positive number with two decimal places. |
| FR-90 | The system shall validate the transaction date is not in future. |
| FR-91 | The system shall store the transaction in the database with a unique identifier. |
| FR-92 | The system shall ensure that the transaction is linked to an existing category in the CATEGORY table. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-14 Read Transaction | FR-93 | Registered users shall have access to view their transactions. |
| FR-94 | The system shall display transactions in a list or grid, including details like type, category, date, amount, and notes. |
| FR-95 | The system shall sort transactions by date in descending order. |
| FR-96 | The system shall ensure that users can only view transactions linked to their categories. |
| NFR-05 | The transaction listing table with up to 100 record must loaded within 5 seconds. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-15 Update Transaction | FR-100 | The system shall validate the updated transaction amount to be a positive number with two decimal places. |
| FR-101 | The system shall validate the updated transaction date is not in future. |
| FR-102 | The system shall ensure that the updated transaction remains linked to a valid category. |
| FR-103 | The system shall confirm the successful update of the transaction. |
| FR-97 | The system shall provide an editing feature for users to modify transaction details. |
| FR-98 | Upon selecting a transaction, the system shall present a form with the current details pre-filled. |
| FR-99 | The system shall accept changes to the transaction type, date, amount, and notes. |

| **Derived From** | **Id** | **System Requirement Specification** |
| --- | --- | --- |
| UR-16 Delete Transaction | FR-104 | The system shall provide a deletion feature for users to remove transactions. |
| FR-105 | Users shall be able to select a transaction for deletion from their list of transactions. |
| FR-106 | The system shall confirm the deletion action with the user before proceeding. |
| FR-107 | Upon user confirmation, the system shall remove the transaction from the database. |
| FR-108 | The system shall confirm the successful transaction deletion operation. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-17 Create Group | FR-109 | Access to the group creation feature shall be available to registered users. |
| FR-110 | The system shall provide a feature for users to create groups. |
| FR-111 | The system shall accept group date, title and description for the creation of a group. |
| FR-112 | The system shall validate the group date to be mandatory and not in future. |
| FR-113 | The system shall validate the group title to be mandatory and not exceed 50 characters. |
| FR-114 | The system shall validate the group description to be mandatory and not exceed 100 characters. |
| FR-115 | The system shall store the group details in the database with a unique identifier. |
| FR-116 | The system shall ensure that the group is associated with the user creating it. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-18 Read Group | FR-117 | Registered users shall have access to view their groups. |
| FR-118 | The system shall display a list of groups including date, title and description. |
| FR-119 | The system shall sort groups by date in descending order. |
| FR-120 | The system shall ensure that only groups associated with the user (owned or member of) are displayed. |
| NFR-06 | The group listing table with up to 100 record must loaded within 5 seconds. |

| **Derived From** | **Id** | **System Requirement Specification** |
| --- | --- | --- |
| UR-19 Update Group | FR-121 | The system shall provide an editing feature for users to modify group details. |
| FR-122 | Upon selecting a group, the system shall present a form with the current details pre-filled. |
| FR-123 | The system shall allow edit of the user's owned groups only. |
| FR-124 | The system shall accept changes to the group date, title and description. |
| FR-125 | The system shall validate the updated group date to be mandatory and not in future. |
| FR-126 | The system shall validate the updated group title to be mandatory and not exceed 50 characters. |
| FR-127 | The system shall validate the updated group description to be between 3 and 100 characters. |
| FR-128 | The system shall confirm the successful update of the group. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-20 Delete Group | FR-129 | The system shall provide a deletion feature for users to remove groups. |
| FR-130 | Users shall be able to select owned group for deletion from their list of groups. |
| FR-131 | The system shall confirm the deletion action with the user before proceeding. |
| FR-132 | The system shall prevent the group deletion if there exists group transactions already recorded against this group. |
| FR-133 | Upon user confirmation, the system shall remove the group from the database. |
| FR-134 | The system shall confirm the successful group deletion operation. |

| **Derived From** | **Id** | **System Requirement Specification** |
| --- | --- | --- |
| UR-21 Create Group Transaction | FR-135 | Access to the group transaction creation feature shall be available to registered users who are members of the user group. |
| FR-136 | The system shall provide a feature for users to create group transactions within a user group. |
| FR-137 | The system shall accept details including the user group, category, date, amount, paid to users, and optional notes for the creation of a group transaction. |
| FR-138 | The system shall validate the transaction amount to be a positive number with two decimal places. |
| FR-139 | The system shall validate the transaction date is not in future. |
| FR-140 | The system shall create one record in Transaction table using details transaction type "Expense", category, date, amount and notes. |
| FR-141 | The system shall create a split amount by splitting the amount equally between the "paid to users". |
| FR-142 | The system shall create multiple records in User Group Transaction table using details group, date, amount, paid by user, paid to user and notes. |
| FR-143 | The system shall ensure that the 'paid by user' and 'paid to user' reference users that are members of the group specified in the group transaction. |
| FR-144 | The system shall store the group transaction details in the database with a unique identifier for the group transaction. |
| FR-145 | The system shall ensure that the user group, transaction, paid by user, and paid to user are all referencing the existing data in database. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-22 Read Group Transaction | FR-146 | Registered users shall have access to view their group transactions within a user group upon logging into their accounts. |
| FR-147 | The system shall display a list or grid layout of all group transactions associated with the user's groups. |
| FR-148 | The system shall sort the group transactions by date in descending order. |
| FR-149 | The system shall provide details such as the group, date, transaction amount, paid by user, paid to user, and notes in the group transaction view. |
| FR-150 | The system shall ensure that only group transactions associated with the user's groups (owned or member of) are displayed. |
| NFR-07 | The group transaction listing table with up to 100 record must loaded within 5 seconds. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-23 Read Group Transaction Settlement | FR-151 | The system shall show a read-only tabular view of group transaction settlement. |
| FR-152 | The group transaction settlement table will have the details group member name, total amount paid, total amount received and unsettled amount. |
| FR-153 | The unsettled amount will be calculated as difference between total amount paid and total amount received. |
| NFR-08 | The group transaction settlement listing table with up to 100 record must loaded within 5 seconds. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-24 Update Group Transaction | FR-154 | The system shall not provide edit functionality for the group transaction directly. |
| FR-155 | To update the group transaction, the user would have to delete the main transaction and recreate it with necessary correction. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| UR-25 Delete Group Transaction | FR-156 | The system shall not provide delete functionality for the group transaction directly. |
| FR-157 | To delete the group transaction, the user would have to delete the main transaction which would delete the related group transactions. |

|  |  |  |
| --- | --- | --- |
| **Derived From** | **Id** | **System Requirement Specification** |
| Global | NFR-09 | The system should automatically scale resources up or down based on real-time demand to maintain performance levels. |
| NFR-10 | The web application should aim for high availability, with a target uptime of 99.9% or higher. |
| NFR-11 | Maintenance windows should be scheduled during off-peak hours to minimize user impact. |
| NFR-12 | The web application should be compatible with operating system Microsoft Windows 10 with browsers Google Chrome, Microsoft Edge and Mozilla Firefox ensuring a wide user reach. |

## **Scenario for 3 user requirements**

### **Scenario 1: User Self-Registration**

|  |  |
| --- | --- |
| **Assumption** | 1. The user does not already have an account registered. 2. The user has a stable internet connection. 3. The user has a valid email address. |
| **Normal Way** | 1. The user navigates to the registration page. 2. The user fills in their details such as email-address, First Name, Last Name, and password. 3. Once the details of the users are validated, the application provides a confirmation message saying, “Account successfully created”. 4. The new user can then start using the application. |
| **What can go wrong** | 1. The user leaves the field blank and click Save. 2. The user enters an invalid email address which does not follow format aaa@bbb.ccc. 3. The user enters a password that does not meet the application requirement. 4. The user unintentionally leaves a mandatory field blank; the application will present a message stating what the error is. 5. There is a technical issue e.g. connection time out that prevents the user registration to which the application will present the user with the necessary error. |
| **Final state** | The user is able to register and gain access to the application successfully. |

### **Scenario 2: Creating and managing shared group expenses.**

|  |  |
| --- | --- |
| **Assumption** | 1. The user has an account registered with the application. 2. The group expense management functionality is working as required. |
| **Normal Way** | 1. The user navigates to the Group Transactions main menu. 2. The user clicks the Add New command. 3. The system will provide the form with fields Group, Category, TxDate, Amount, paid for member and Notes. 4. The user will provide the information and click Save. 5. The system will validate the provided information and then create the main transaction for the category as well as individual group transactions which will split the amount from-to-user information. 6. The system will show the confirmation of the operation. |
| **What can go wrong** | 1. The user leaves the field blank and click Save. 2. The user enters the future date. 3. The user enters the negative amount and pays for members. 4. The system validation fails if the user has entered invalid data or left a field empty. The system will present the user with the corresponding error. |
| **Final state** | The group will have a clear record of the shared expenses and individual balances |

### **Scenario 3: User creates a monthly budget.**

|  |  |
| --- | --- |
| **Assumption** | 1. The user is logged into their account. 2. The user understands their monthly income and expenses. |
| **Normal Way** | 1. The user navigates to the Budget main menu. 2. The user clicks the Add New command. 3. The system will provide the form with fields Category, Effective Date and Amount. 4. The user will provide the information and click Save. 5. The system will validate the provided information and then create the budget for the category. 6. The system will show the confirmation of the operation and then navigate to the budget listing table. |
| **What can go wrong** | 1. The user leaves the field blank and click Save. 2. The user enters negative amounts for the category. 3. The user enters the future date. 4. The system is unable to validate the information and presents the user with the corresponding error. 5. Connection time-out error preventing the budget from being saved. |
| **Final state** | The user is able to create and save an accurate monthly budget. |

## **Functional requirements & prioritization**

Following is the functional requirement prioritization

| **Functional Requirement** | **Description** | **Priority** | **Priority Notes** |
| --- | --- | --- | --- |
| User Registration | Allow users to register accounts with their required details and implement authentication and authorization. | 1 | Users must first register to access the features of the application. |
| Category Management | Allow users to categorize their expenses into customizable categories to manage their spending patterns. | 2 | Defining categories lays the groundwork for tracking expenses effectively. |
| Budget Management | Allow users to set up budgets for different expense categories to establish spending limits that align with their financial goals. | 3 | Establishing spending limits with budgets is the next priority to control their expenses. |
| Transaction Management | Provide functionality for users to add, edit, and delete individual expenses, including details such as amount, date, category, and optional notes. | 4 | After users have established their complete financial information, users can now monitor and track expenses and make informed decisions about their spending. |
| Group Management | Allow users to create and join groups to manage and track group expenses. It shall also enforce roles to ensure that users have access only to their respective groups. | 5 | Management of group expenses is implemented after individual expense tracking because it involves multiple users sharing financial responsibilities. Here onward it is logical add-on module. |
| Group Transaction Management | Allow users within their respective groups to allocate and split costs, and settle debts among group members. | 6 | Group transactions are implemented after group management because users must first create or join a group before engaging in shared financial responsibilities. |
| Visualization Dashboard for transaction | Offer charts to visualize expense data, such as breakdowns by category and budget. | 7 | At this moment all the data management part of the application is ready and to be useful we need to summarize the transactions for user. |
| Visualization Dashboard for group transaction | Offer reports and charts to visualize group transactions to understand the account payable and account receivable. | 8 | This is prioritized last since they rely on accumulated data, and then visualization tools can enhance their understanding of spending patterns and financial trends. |

## **Structured approach for top 5 requirements**

### **User Registration**

|  |  |
| --- | --- |
| **Function** | User Registration |
| **Description** | Allow users to register accounts with their required details and implement authentication and authorization |
| **Input** | 1. Signup Form: User details such as Email, First Name, Last Name, and Password. 2. Login Form: User details such as Email, and Password |
| **Source** | A signup or login form with the text fields for the details above |
| **Output** | A user account saved into the database, along with a confirmation message upon successfully creating accounts or logging in with an existing account |
| **Destination** | Database |
| **Action** | 1. Validate each text field for user input. 2. Store each valid user input into the MySQL database. 3. Provide authentication using tokens and/or cookies. 4. Provide authorization based on permissions. 5. Display confirmation or error message depending on the outcome of the process. 6. Redirect the user to the dashboard of the application. |
| **Requirements** | 1. The system shall provide a user registration feature accessible from the application interface. 2. The system shall verify and validate the user inputs, including checking for unique usernames and email and checking for password requirements. 3. The system shall display a confirmation or error message depending on the outcome of the registration process. 4. The system shall save the user account into the MySQL database and encrypt user passwords. 5. The system shall create a unique userID for each user account. 6. The system shall provide other security measures such as authentication and authorization |
| **Pre-condition** | The user has access to the registration form and provides valid inputs. |
| **Post-condition** | 1. The user account is created and securely stored in the MySQL database. 2. The user account is retrieved from the MySQL database when successfully logging in. 3. A confirmation message will be displayed for both conditions |
| **Side effects** | None |

### **Category Management**

|  |  |
| --- | --- |
| **Function** | Category Management |
| **Description** | Allow users to categorize their expenses into customizable categories to manage their spending patterns |
| **Input** | 1. Category Title 2. Description |
| **Source** | A category form with the text fields for the details above |
| **Output** | A category is saved into the database, along with a confirmation message upon successfully creating a category |
| **Destination** | Database |
| **Action** | 1. Validate each text field for user input. 2. Store each valid user input into the MySQL database to create categories for each expense. 3. Display confirmation or error message depending on the outcome of the process. 4. Redirect the user to the category page of the application. |
| **Requirements** | 1. The system shall provide a category form from the application interface to create custom categories. 2. The system shall verify and validate the user inputs for expense details. 3. The system shall store categorized expenses securely in the database. 4. The system shall create a unique CategoryID for each category within every user account identified by a unique OwnerID. 5. The system shall support the storage and retrieval of categorized expenses for viewing and analysis. 6. The system shall display a confirmation or error message depending on the outcome of creating a category. |
| **Pre-condition** | The user has access to the category form and provides valid inputs |
| **Post-condition** | 1. The category is successfully created and stored in the MySQL database. 2. A confirmation message will be displayed for the success of creating a category. 3. An error message will be displayed if any errors occur, redirecting the user back to the category page |
| **Side effects** | None |

### **Budget Management**

|  |  |
| --- | --- |
| **Function** | Budget Management |
| **Description** | Allow users to set up budgets for different expense categories to establish spending limits that align with their financial goals |
| **Input** | 1. Amount 2. Effective Date 3. A selected category using CategoryID and OwnerID |
| **Source** | A budget form with the text fields for Amount and Effective Date and a dropdown list for categories |
| **Output** | A budget is saved into the database, along with a confirmation message upon successfully creating a budget |
| **Destination** | Database |
| **Action** | 1. Validate each text field for user input. 2. Store each valid user input into the MySQL database to create a budget for each category. 3. Display confirmation or error message depending on the outcome of the process. 4. Redirect the user to the budget page of the application |
| **Requirements** | 1. The system shall provide a budget form from the application interface to create a budget. 2. The system shall verify and validate the user inputs for budget details. 3. The system shall store the budgets securely in the MySQL database. 4. The system shall create a unique BudgetID for each budget corresponding to the user's specific category, CategoryID, within each user account identified by a unique OwnerID. 5. The system shall support the storage and retrieval of budget information for viewing and analysis. 6. The system shall display a confirmation or error message depending on the outcome of creating a budget |
| **Pre-condition** | The user has access to the budget form and provides valid inputs |
| **Post-condition** | 1. The budget is successfully created and stored in the MySQL database. 2. A confirmation message will be displayed for the success of creating a category. 3. An error message will be displayed if any errors occur, redirecting the user back to the budget page |
| **Side effects** | None |

### **Transaction Management**

|  |  |
| --- | --- |
| **Function** | Transaction Management |
| **Description** | Provide functionality for users to add, edit, and delete individual expenses, including details such as amount, date, category, and optional notes |
| **Input** | 1. Transaction Type 2. Selected Category using CategoryID 3. Selected Transaction, if applicable, using TransactionID 4. Transaction Date 5. Amount 6. Notes |
| **Source** | A transaction form with the text fields above and dropdown list for categories to create new transactions.  An interface to edit and delete transactions |
| **Output** | A transaction is saved, edited, or deleted into the database, along with a confirmation message indicating which action occurred |
| **Destination** | Database |
| **Action** | 1. Validate each text field for user input. 2. Store each valid user input into the MySQL database for the transaction. 3. Display confirmation or error message depending on the outcome of the process. 4. Redirect the user to the expense page of the application |
| **Requirements** | 1. The system shall provide a transaction form from the application interface. 2. The system shall verify and validate the user inputs or type of action for the transaction. 3. The system shall reflect the changes of the transactions in the MySQL database. 4. The system shall create a unique TransactionID. 5. The system shall update the contents of the appropriate budget based on the amount of the transaction, using the corresponding CategoryID and BudgetID. 6. The system shall support the storage and retrieval of transaction information for viewing and analysis. 7. The system will display a confirmation or error message depending on the outcome of the transaction |
| **Pre-condition** | 1. The user has access to the transaction form and provides valid inputs for creating a new transaction. 2. The user has selected a specific transaction for editing and deleting purposes |
| **Post-condition** | 1. The transaction is successfully created, edited, or deleted and reflected in the MySQL database. 2. A confirmation message will be displayed for the success of the type of action for the transaction. 3. An error message will be displayed if any errors occur, redirecting the user back to the expense page |
| **Side effects** | None |

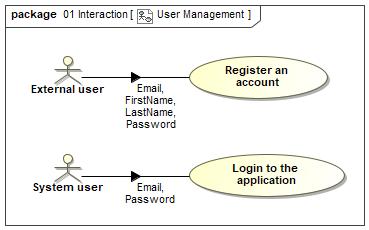
### **Group Management**

|  |  |
| --- | --- |
| **Function** | Group Management |
| **Description** | Allow users to create and join groups to manage and track group expenses. It shall also enforce roles to ensure that users have access only to their respective groups |
| **Input** | 1. Title 2. Member List with UserID’ s 3. OwnerID 4. Description |
| **Source** | 1. A group form to create a new group with the information above. 2. An interface to edit and delete groups. |
| **Output** | A group is saved, edited, or deleted into the database, along with a confirmation message indicating which action occurred |
| **Destination** | Database |
| **Action** | 1. Allow users to manage the groups that they are in. 2. Validate each field for user input and expense details. 3. Store group information into the MySQL database. 4. Display confirmation or error message depending on the outcome of the process. 5. Redirect the user to the group page of the application |
| **Requirements** | 1. The system shall provide interfaces for users to create, join, edit, or delete groups. 2. The system shall verify and validate the user inputs and the type of actions for the group management. 3. The system shall reflect the changes of the group management in the MySQL database. 4. The system shall create a unique GroupID. 5. The system shall enforce rules and permissions to allow users to have access only to their respective groups. 6. The system shall support the storage and retrieval of group information for viewing and management. 7. The system shall display a confirmation or error message depending on the outcome of creating, editing, or deleting a group. |
| **Pre-condition** | 1. The user has access to the group form and provides valid inputs for creating a new group. 2. The user has selected a specific group for editing and deleting purposes. |
| **Post-condition** | 1. The group is successfully created, edited, or deleted and reflected in the MySQL database. 2. A confirmation message will be displayed for the success of the type of action for the group. 3. An error message will be displayed if any errors occur, redirecting the user back to the group page |
| **Side effects** | None. |

## **Use case diagrams**

Following are the use case diagrams.

### **Use Case: User Management Diagram**



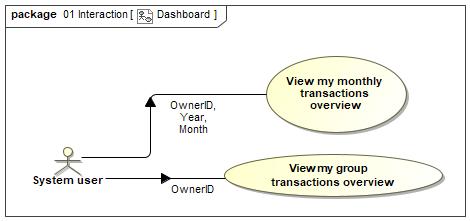
#### Login to the application Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Login to the application |
| **Actors** | * System user |
| **Description** | A system user (the one with account on the application) navigates to the login page.  System provides a login form with Email and Password fields.  User will provide Email and Password and then click Login button.  System will authenticate the credentials and then login to the application showing the home page of the application. |
| **Data** | Email  Password |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### Register an account Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Register an account |
| **Actors** | * External user |
| **Description** | An external user who does not have an account on the application would click the Signup button and system will provide a registration form.  User will provide information like Email, First Name, Last Name and Password and then click Register.  System will register an account for user and provide confirmation.  Then user will click on Login button to login to the system. |
| **Data** | Email  First Name  Last Name  Password |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

### **Use Case: Dashboard Diagram**



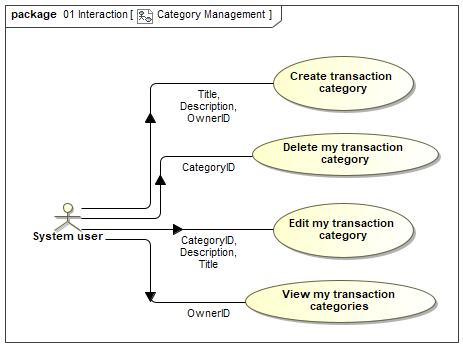
#### View my group transactions overview Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View my group transactions overview |
| **Actors** | * System user |
| **Description** | The user will navigate to Dashboard from main menu.  User will click on the "Group Transactions" sub menu.  System will show charts providing information about the money I should receive and the money I should be paying against the user names. |
| **Data** | OwnerID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### View my monthly transactions overview Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View my monthly transactions overview |
| **Actors** | * System user |
| **Description** | The user will navigate to Dashboard from main menu.  User will click on the "Transactions" sub menu.  System will load the overview page for the current month.  User will select the period, if he wants to view overview for month other than the correct month.  System will consider the selected month and show charts providing information about income and expenses. The information will include the split of different expenses/income types as well as the comparison of expected vs actual expense/income. |
| **Data** | OwnerID  Year  Month |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

### **Use Case: Category Management Diagram**



#### Create transaction category Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Create transaction category |
| **Actors** | * System user |
| **Description** | The user navigates to the Categories main menu.  User clicks the Add New command.  System will provide the form with fields Title and Description.  User will provide the information and click Save.  System will validate the provided information and then create the Category.  System will show the confirmation of the operation and then navigate to the Category listing table. |
| **Data** | Title  Description  OwnerID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### Delete my transaction category Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Delete my transaction category |
| **Actors** | * System user |
| **Description** | The user is seeing the Category listing table.  User clicks on the delete link against the category.  System provides confirmation message.  User responds to confirmation message to proceed to delete operation.  System validation if there are transactions recorded against the categories, if so then provides error message and the operation halts.  Otherwise system will delete the category and refresh the page to show updated categories list table. |
| **Data** | CategoryID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

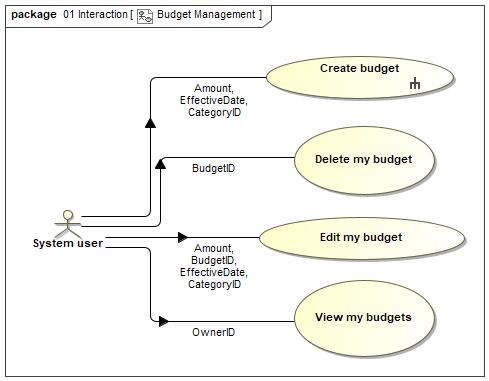
#### Edit my transaction category Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Edit my transaction category |
| **Actors** | * System user |
| **Description** | The user is seeing the Category listing table.  User clicks the name link of a category.  System opens the category edit form with Title and Description field with existing values.  User will provide the information and click Save.  System will validate the provided information and then update the Category.  System will show the confirmation of the operation and then navigate to the Category listing table. |
| **Data** | CategoryID  Description  Title |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### View my transaction categories Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View my transaction categories |
| **Actors** | * System user |
| **Description** | The user navigates to the Categories main menu.  System will by default list the user owned categories in table. |
| **Data** | OwnerID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

### **Use Case: Budget Management Diagram**



#### Create budget Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Create budget |
| **Actors** | * System user |
| **Description** | The user navigates to the Budget main menu.  User clicks the Add New command.  System will provide the form with fields Category, Effective Date and Amount.  User will provide the information and click Save.  System will validate the provided information and then create the budget for the category.  System will show the confirmation of the operation and then navigate to the budget listing table. |
| **Data** | Amount  Effective Date  CategoryID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### Delete my budget Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Delete my budget |
| **Actors** | * System user |
| **Description** | The user is seeing the Budget listing table.  User clicks on the delete link against the budget.  System provides confirmation message.  User responds to confirmation message to proceed to delete operation.  System validation if there are transactions recorded against the categories, if so then provides error message and the operation halts.  Otherwise system will delete the budget and refresh the page to show updated budget list table. |
| **Data** | BudgetID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

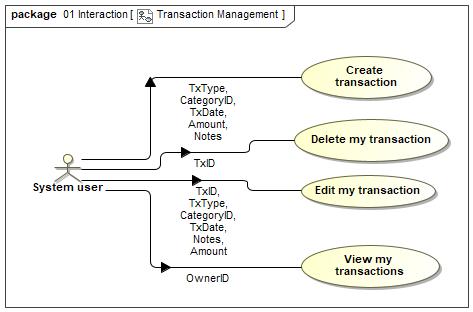
#### Edit my budget Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Edit my budget |
| **Actors** | * System user |
| **Description** | The user is seeing the Budget listing table.  User clicks the name link of the category.  System opens the budget edit form with Category, Effective Date and Amount field with existing values.  User will provide the information and click Save.  System will validate the provided information and then update the budget.  System will show the confirmation of the operation and then navigate to the Budget listing table. |
| **Data** | Amount  BudgetID  Effective Date  CategoryID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### View my budgets Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View my budgets |
| **Actors** | * System user |
| **Description** | The user navigates to the Budget main menu.  System will by default list the user owned budget for categories in table. |
| **Data** | OwnerID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

### **Use Case: Transaction Management Diagram**



#### Create transaction Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Create transaction |
| **Actors** | * System user |
| **Description** | The user navigates to the Transactions main menu.  User clicks the Add New command.  System will provide the form with fields Transaction Type, Category, Transaction Date, Amount and Notes.  User will provide the information and click Save.  System will validate the provided information and then create the transaction for the category.  System will show the confirmation of the operation and then navigate to the transaction listing table. |
| **Data** | TxType  CategoryID  TxDate  Amount  Notes |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### Delete my transaction Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Delete my transaction |
| **Actors** | * System user |
| **Description** | The user is seeing the Transaction listing table.  User clicks on the delete link against the transaction.  System provides confirmation message.  User responds to confirmation message to proceed to delete operation.  System will delete the transaction and any related group transactions and refresh the page to show updated categories list table. |
| **Data** | TxID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

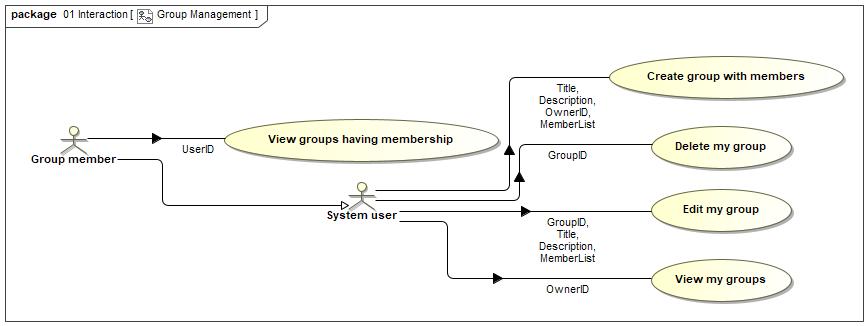
#### Edit my transaction Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Edit my transaction |
| **Actors** | * System user |
| **Description** | The user is seeing the Transaction listing table.  User clicks the name link of a transaction.  System opens the transaction edit form with Transaction Type, Category, Transaction Date, Amount and Notes field with existing values.  User will provide the information and click Save.  System will validate the provided information and then update the transaction and any related group transactions with amount split.  System will show the confirmation of the operation and then navigate to the Transaction listing table. |
| **Data** | TxID  TxType  CategoryID  TxDate  Notes  Amount |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### View my transactions Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View my transactions |
| **Actors** | * System user |
| **Description** | The user navigates to the Transactions main menu.  System will by default list the user owned transactions in table. |
| **Data** | OwnerID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

### **Use Case: Group Management Diagram**



#### Create group with members Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Create group with members |
| **Actors** | * System user |
| **Description** | The user navigates to the Groups main menu.  User clicks the Add New command.  System will provide the form with fields Title, Description and Members.  User will provide the information and click Save.  System will validate the provided information and then create the Group.  System will show the confirmation of the operation and then navigate to the Groups listing table. |
| **Data** | Title  Description  OwnerID  MemberList |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### Delete my group Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Delete my group |
| **Actors** | * System user |
| **Description** | The user is seeing the Group listing table.  User clicks on the delete link against a group.  System provides confirmation message.  User responds to confirmation message to proceed to delete operation.  System validation if there are transactions recorded against the group, if so then provides error message and the operation halts.  Otherwise system will delete the group and refresh the page to show updated groups list table. |
| **Data** | GroupID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### Edit my group Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Edit my group |
| **Actors** | * System user |
| **Description** | The user is seeing the Group listing table.  User clicks the name link of the group.  System opens the group edit form with Title, Description and Members field with existing values.  User will provide the information and click Save.  System will validate the provided information and then update the group. If a group member has created group transaction then this member cannot be removed from the group.  System will show the confirmation of the operation and then navigate to the group listing table. |
| **Data** | GroupID  Title  Description  MemberList |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

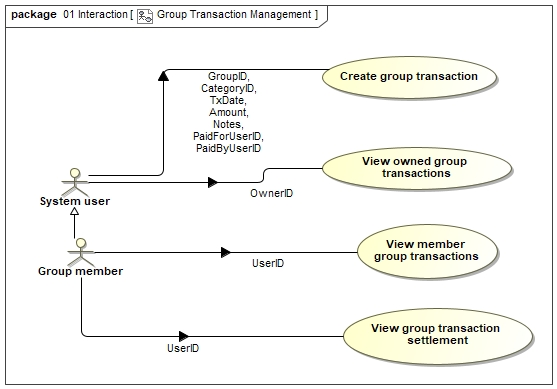
#### View groups having membership Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View groups having membership |
| **Actors** | * Group member |
| **Description** | The user navigates to the Groups main menu.  System will by default list the groups where this user has membership, just below the user owned groups in table. |
| **Data** | UserID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### View my groups Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View my groups |
| **Actors** | * System user |
| **Description** | The user navigates to the Groups main menu.  System will by default list the user owned groups in table. |
| **Data** | OwnerID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

### **Use Case: Group Transaction Management Diagram**



#### Create group transaction Use Case

|  |  |
| --- | --- |
| **Use Case Name** | Create group transaction |
| **Actors** | * System user |
| **Description** | The user navigates to the Group Transactions main menu.  User clicks the Add New command.  System will provide the form with fields Group, Category, TxDate, Amount, and Paid for member and Notes.  User will provide the information and click Save.  System will validate the provided information and then create the main transaction for the category as well as individual group transaction which will have split of the amount from-to user information.  System will show the confirmation of the operation. |
| **Data** | GroupID  CategoryID  TxDate  Amount  Notes  PaidForUserID  PaidByUserID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### View member group transactions Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View member group transactions |
| **Actors** | * Group member |
| **Description** | The user navigates to the Group Transactions main menu and select the group name where this user is member of.  System will list the transactions from the selected group. |
| **Data** | UserID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### View owned group transactions Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View owned group transactions |
| **Actors** | * System user |
| **Description** | The user navigates to the Group Transactions main menu and select the group name where this user is owner of.  System will list the transactions from the selected group. |
| **Data** | OwnerID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

#### View group transaction settlement Use Case

|  |  |
| --- | --- |
| **Use Case Name** | View group transaction settlement |
| **Actors** | * Group member |
| **Description** | The user navigates to the Group Transactions main menu and select the group name where this user is member of. System will list the transactions from the selected group. Below the group transaction listing table, the system shall show a read-only tabular view of group transaction settlement. The group transaction settlement table will have the details group member name, total amount paid, total amount received and unsettled amount. The unsettled amount will be calculated as difference between total amount paid and total amount received. |
| **Data** | UserID |
| **Stimulus** | User command issued from user interface |
| **Response** | Confirmation of the successful execution is provided by system |
| **Comments** | None |

# **Traceability Matrix**

| **User Requirement Definition** | **Derived System Requirement Specs** | **Satisfied By Use Case** |
| --- | --- | --- |
| UR-01 Create User | FR-01 FR-02 FR-03 FR-04 FR-05 FR-06 | Register an account |
| UR-02 Login User | FR-07 FR-08 FR-09 FR-10 FR-11 FR-12 FR-13 FR-14 | Login to the application |
| UR-03 Read Dashboard Transaction | FR-15 FR-16 FR-17 FR-18 FR-19 FR-20 FR-21 FR-22 NFR-01 | View my group transactions overview |
| UR-04 Read Dashboard Group Transaction | FR-23 FR-24 FR-25 FR-26 FR-27 FR-28 NFR-02 | View my monthly transactions overview |
| UR-05 Create Category | FR-29 FR-30 FR-31 FR-32 FR-33 FR-34 FR-35 | Create transaction category |
| UR-06 Read Category | FR-36 FR-37 FR-38 FR-39 FR-40 FR-41 FR-42 NFR-03 | View my transaction categories |
| UR-07 Update Category | FR-43 FR-44 FR-45 FR-46 FR-47 FR-48 FR-49 FR-50 FR-51 FR-52 FR-53 | Edit my transaction category |
| UR-08 Delete Category | FR-54 FR-55 FR-56 FR-57 FR-58 FR-59 FR-60 FR-61 | Delete my transaction category |
| UR-09 Create Budget | FR-62 FR-63 FR-64 FR-65 FR-66 FR-67 | Create budget |
| UR-10 Read Budget | FR-68 FR-69 FR-70 FR-71 FR-72 NFR-04 | View my budgets |
| UR-11 Update Budget | FR-73 FR-74 FR-75 FR-76 FR-77 FR-78 | Edit my budget |
| UR-12 Delete Budget | FR-79 FR-80 FR-81 FR-82 FR-83 FR-84 FR-85 | Delete my budget |
| UR-13 Create Transaction | FR-86 FR-87 FR-88 FR-89 FR-90 FR-91 FR-92 | Create transaction |
| UR-14 Read Transaction | FR-93 FR-94 FR-95 FR-96 NFR-05 | View my transactions |
| UR-15 Update Transaction | FR-97 FR-98 FR-99 FR-100 FR-101 FR-102 FR-103 | Edit my transaction |
| UR-16 Delete Transaction | FR-104 FR-105 FR-106 FR-107 FR-108 | Delete my transaction |
| UR-17 Create Group | FR-109 FR-110 FR-111 FR-112 FR-113 FR-114 FR-115 FR-116 | Create group with members |
| UR-18 Read Group | FR-117 FR-118 FR-119 FR-120 NFR-06 | View groups having membership View my groups |
| UR-19 Update Group | FR-121 FR-122 FR-123 FR-124 FR-125 FR-126 FR-127 FR-128 | Edit my group |
| UR-20 Delete Group | FR-129 FR-130 FR-131 FR-132 FR-133 FR-134 | Delete my group |
| UR-21 Create Group Transaction | FR-135 FR-136 FR-137 FR-138 FR-139 FR-140 FR-141 FR-142 FR-143 FR-144 FR-145 | Create group transaction |
| UR-22 Read Group Transaction | FR-146 FR-147 FR-148 FR-149 FR-150 NFR-07 | View owned group transactions View member group transactions |
| UR-23 Read Group Transaction Settlement | FR-151 FR-152 FR-153 NFR-08 | View group transaction settlement |
| UR-24 Update Group Transaction | FR-154 FR-155 | Create transaction Delete my transaction |
| UR-25 Delete Group Transaction | FR-156 FR-157 | Delete my transaction |