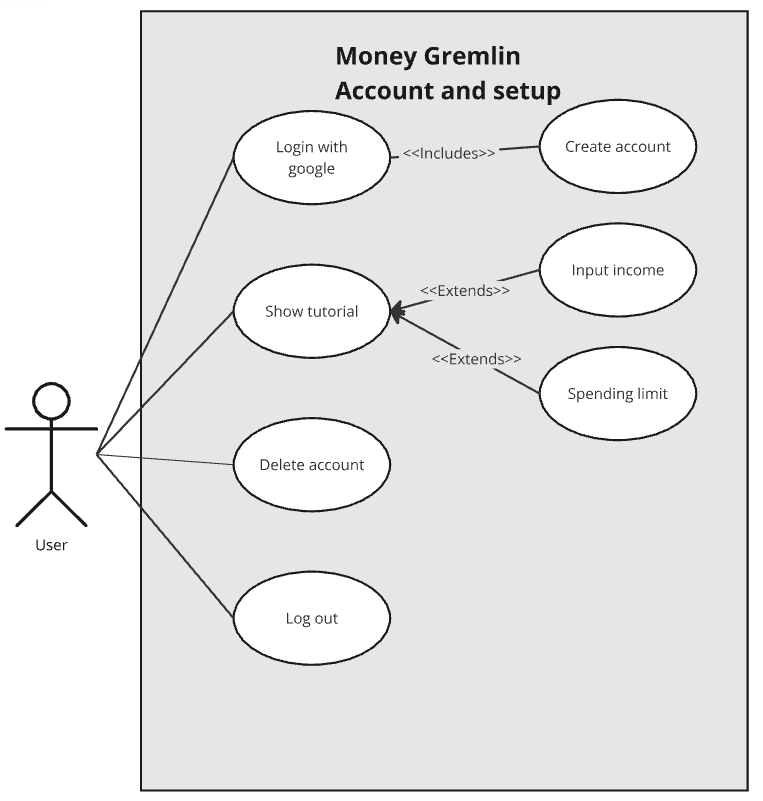
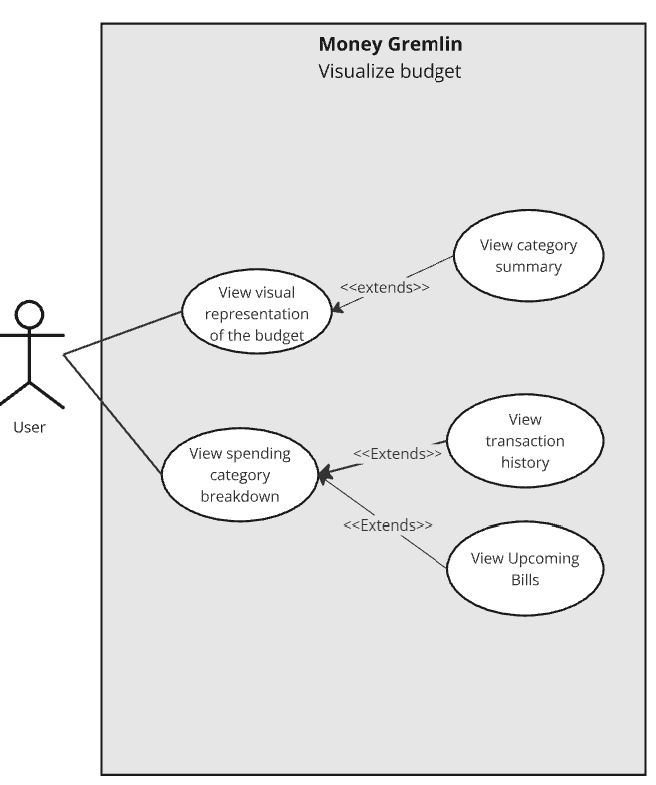
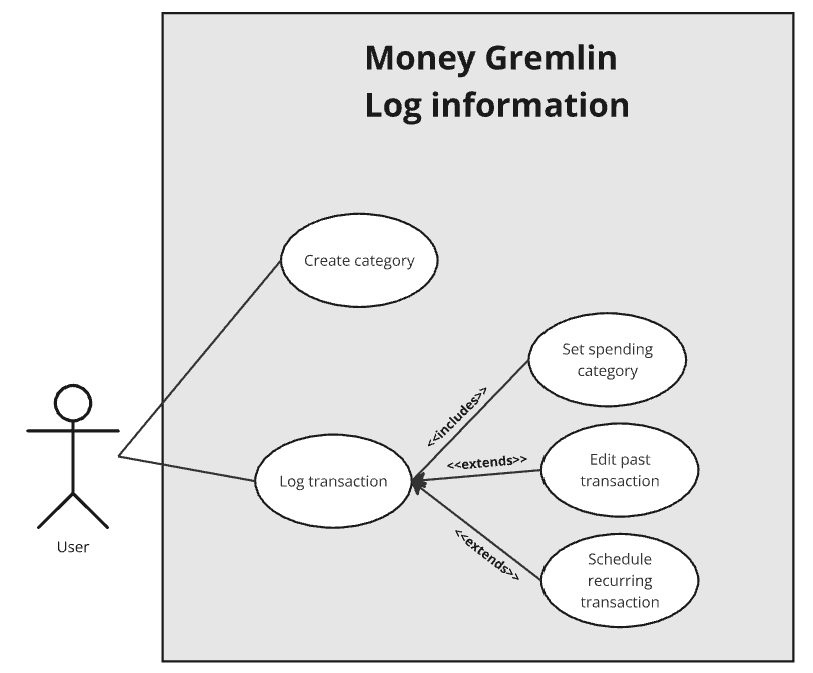
**Use Case Diagrams**

**Team Gremlin**

Diagram Link: <https://miro.com/app/board/uXjVLRWvZhE=/>







**Use Case Descriptions**

| **Name:** | Create category |
| --- | --- |
| **ID:** |  |
| **Description**: | As a user, I want to enter my income into the system so I know how much money I have to work with. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | Enter monthly or yearly income to aid in budgeting. |
| **Related Use Cases**: | Extends: New user experience. |
| **Actor:** | General User |
| **Invariant:** | User has an account. |
| **Precondition:** | User is logged into their account. |
| **Success Postcondition**: | User income data is displayed on the “Budget” interface. |

| **USER STEPS**:  1. | **SYSTEM RESPONSE**:  3. |
| --- | --- |

| **Name:** | Input Income |
| --- | --- |
| **ID:** | 14 |
| **Description**: | As a user, I want to enter my income into the system so I know how much money I have to work with. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | Enter monthly or yearly income to aid in budgeting. |
| **Related Use Cases**: | Extends: New user experience. |
| **Actor:** | General User |
| **Invariant:** | User has an account. |
| **Precondition:** | User is logged into their account. |
| **Success Postcondition**: | User income data is displayed on the “Budget” interface. |

| **USER STEPS**:  1. User navigates to the “Budget” page.  2. User clicks the “Enter Income” button.  4. User income is displayed alongside the monthly budget and spending limit.  5. User confirms the budget by clicking the “Submit” button. | **SYSTEM RESPONSE**:  3. System displays a prompt allowing the user to enter their monthly/yearly income in USD. |
| --- | --- |

| **Name:** | Spending Limit |
| --- | --- |
| **ID:** | 15 |
| **Description**: | As a user, I want to set spending limits each month to aid in keeping me within my budget. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | Allow the user to set a monthly spending limit. |
| **Related Use Cases**: | Extends: New user experience |
| **Actor:** | General User |
| **Invariant:** | User has an account. |
| **Precondition:** | User is logged into their account. |
| **Success Postcondition**: | The user defined a spending limit to stay below each month. |

| **USER STEPS**:  1. The user accesses the “Budget” page.  2. The user defines a monthly spending limit in USD.  4. The user spends money and accumulates transactions.  6. At the beginning of the next month, the user has the option to define a new spending limit or keep the current one. | **SYSTEM RESPONSE**:  3. The system stores this data and displays the spending limit to the user.  5. The system sums transactions within the month and compares that to the defined limit. |
| --- | --- |

| **Name/Title:** | Show tutorial |
| --- | --- |
| **ID:** | 32 |
| **Description**: | REQ-32: The system shall explain its functions to new users |
| **System Under Design:** | Money Gremlin |
| **Primary Actor**: | New user. |
| **Goal**: | Teach the user the basics of the platform. |
| **Related Use Cases**: | Extensions: Input income, Set monthly spending limit. |
| **Invariant**: | User enters all necessary information. |
| **Precondition**: | The user has no data associated with their account. |
| **Success Postcondition**: | The user has an income stored. The user has a monthly limit stored. The user better understands the system. |

| **USER STEPS**:  1. Hit “Sign up for account.”  5. Clicks on relevant tutorial popups.  7. Enter income and click “Next.”  10. Enter the spending limit. | **SYSTEM RESPONSE**:  2. Successfully signs the user up.  3. Brings the user to the home screen.  4. Begin tutorial.  6. Directs the user to the “Enter income” section.  8. Income logged.  9. Direct user to “Set spending limit.”  11. Spending limit logged.  12. Display “Tutorial complete.”  13. Go to the main page. |
| --- | --- |

| **Name/Title:** | Schedule recurring transaction |
| --- | --- |
| **ID:** | 26 |
| **Description**: | REQ-26: The system shall allow the user to schedule recurring transactions such as monthly bills, subscriptions, or income and automatically apply them to the budget. |
| **System Under Design:** | Money Gremlin |
| **Primary Actor**: | Standard user. |
| **Goal**: | Set up a transaction to occur each time period that will be automatically considered in budgeting. |
| **Related Use Cases**: | Instance of: Log transaction.  Inclusions: View upcoming bills. |
| **Invariant**: | User enters all necessary information. |
| **Precondition**: | The user has 1 or more categories created. |
| **Success Postcondition**: | A new recurring transaction is created. The transaction is part of the correct category and automatically added for each period. |

| **USER STEPS**:  1. Click “Add recurring transaction.”  3. Select payment period from daily, weekly, monthly, yearly, or custom.  4. Enter payment periods or end date.  5. Enter the amount of money spent per payment period.  6. Click “Create recurring transaction.”  10. User selects No. | **SYSTEM RESPONSE**:  2. Open recurring transaction popup.  7. Store new recurring transactions in the selected category.  8. Display “New transaction created!”  9. Display “Enter new recurring transaction?” and a Yes/No prompt.  11. Return to the original screen. |
| --- | --- |

| **Name/Title:** | Delete Account |
| --- | --- |
| **ID:** | 24 |
| **Description**: | REQ-24: The system shall allow the user to delete their account. |
| **System Under Design:** | Money Gremlin |
| **Primary Actor**: | Standard user. |
| **Goal**: | Delete a user's account and all associated information. |
| **Related Use Cases**: | N/A |
| **Invariant**: | User has an account. |
| **Precondition**: | None. |
| **Success Postcondition**: | The account is deleted. The email address is unassociated from any accounts. All financial information is removed. |

| **USER STEPS**:  1. Click “Account management.”  3. Select “Delete account.”  5. Select “Yes.”  7. Correctly enter password. | **SYSTEM RESPONSE**:  2. Open account management page.  4. Display “Are you sure?” with yes/no buttons.  6. Display “Enter password” with a text fillable.  9. Display “Deleting account.”  10. Delete email, login information, and all financial information.  11. Return to the login screen. |
| --- | --- |

| **Name:** | Export Transactions |
| --- | --- |
| **ID:** | 13 |
| **Description**: | As a user, I want the ability to export my transaction data so I can keep records of my finances. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | Export transaction data for a user to download. |
| **Related Use Cases**: | Extends: Log Transaction |
| **Actor:** | General User |
| **Invariant:** | User has an account. |
| **Precondition:** | User has entered transactions. |
| **Success Postcondition**: | User downloads file containing transaction data. |

| **USER STEPS**:  1. The user accesses the Money Gremlin homepage.  2. User navigates to the “Transactions” page.  4. User clicks the “Export Transactions” button.  6. User downloads the transaction data file. | **SYSTEM RESPONSE**:  3. System displays transaction data for the user to parse.  5. System outputs transaction data in the form of a downloadable file. |
| --- | --- |

| **Name/Title:** | View a visual representation of the budget |
| --- | --- |
| **ID:** | 1 |
| **Description**: | The system must provide a visual representation of the user’s budget, which will help them plan out their budget properly. |
| **System Under Design:** | Money Gremlin |
| **Primary Actor**: | The User |
| **Goal**: | To display a visual representation of the user’s budget. |
| **Related Use Cases**: | Instance of: N/A  Included by:  1. ID - 14, Enter income  2. ID - 15, Set spending Limits  Inclusions:  1. ID - 12, View Budgets in different currencies  2. ID - 9, Compares budgeting habits with others. Extensions: N/A |
| **Invariant**: | The user must have entered their income and spending categories. |
| **Precondition**: | * The system has access to the user’s financial data, such as income and expenses. * The user must have set budget categories and spending limits. |
| **Success Postcondition**: | The system generates and displays an accurate visual representation of the user's budget, and the user can view insights regarding their spending habits. |

| **USER STEPS**:  1. User enters income and incoming bills  2. The User sets spending limits for specific spending categories.  4. The user clicks on a “view budget”.  7. User reviews the visual representation of the budget | **SYSTEM RESPONSE**:  3. The system processes the entered data.  5. The system then allocates a budget for the user.  6. The system then makes a visual representation of the budget. |
| --- | --- |

| **Name:** | Login with Google |
| --- | --- |
| **ID:** | 3 |
| **Description**: | As a user, I want to log into Money Gremlin using my Google account. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | Allow the user to log in via their Google account. |
| **Related Use Cases**: | Extensions: User login |
| **Actor:** | General User |
| **Invariant:** | N/A |
| **Precondition:** | The user must have a valid Google account. |
| **Success Postcondition**: | The user successfully logs in via their Google account. |

| **USER STEPS**:  1. The user clicks on the “Sign in with Google” button on the login page.  3. The user enters their Google credentials. | **SYSTEM RESPONSE**:  2. The system opens up a different window where the user can enter their Google credentials.  4. If the credentials match, successfully log into Money Gremlin. |
| --- | --- |

| **Name:** | Undo Transaction |
| --- | --- |
| **ID:** | 4 |
| **Description**: | As a user, I want to undo a transaction entry for accidental entries. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | Allow the user to undo accidental transaction entries. |
| **Related Use Cases**: | Extends: Enter transactions. |
| **Actor:** | General User. |
| **Invariant:** | N/A |
| **Precondition:** | The user must have entered a transaction. |
| **Success Postcondition**: | The system successfully removes the transaction, and the budget is updated accordingly. |

| **USER STEPS**:  1. The user navigates to the transaction history.  2. The user selects the transaction they wish to undo.  3. The user clicks on the “Undo” button. | **SYSTEM RESPONSE**:  4. The system removes the selected transaction from the records.  4. The system updates the budget to reflect the change and confirms the undo action to the user. |
| --- | --- |

| **Name:** | Log Transaction |
| --- | --- |
| **ID:** | 11 |
| **Description**: | As a user, I want to be able to import or log my transactions effortlessly so that I can overcome the common barrier to starting and maintaining my budget. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | To log a new transaction in the app, which decreases money from the budget. |
| **Related Use Cases**: | Enter Income |
| **Actor:** | General User |
| **Invariant:** | User logs the correct amount of money and sets it under a category. |
| **Precondition:** | The User has created spending categories. |
| **Success Postcondition**: | User transaction is logged and can be used in the transaction history, user budget is decreased by logged amount. |

| **USER STEPS**:  1. User clicks ‘Log Transaction’  3. Adds values to the Category, Amount and Memo fields.  4. User clicks “Log!” | **SYSTEM RESPONSE**:  2. The system opens a log transaction window.  5. The system updates the transaction history, budget amount, and account balance. |
| --- | --- |

| **Name:** | Set and adjust spending alerts |
| --- | --- |
| **ID:** | 12 |
| **Description**: | As a student, I want to receive alerts if I’m approaching my spending limits so I can adjust my expenses |
| **System Under Design:** | Money Gremlin |
| **Goal:** | Notify users when they approach their set spending limits. |
| **Related Use Cases**: | Enter Income |
| **Actor:** | General User |
| **Invariant:** | User has an account. |
| **Precondition:** | The User has defined spending limits for categories. |
| **Success Postcondition**: | The User receives an alert if they approach their limits. |

| **USER STEPS**:  1. User navigates to the "Notification Settings."  3. User sets alerts for specific percentages of spending limits.  4. User clicks "Save Alerts." | **SYSTEM RESPONSE**:  2. System opens the notification settings page.  5. System stores the alert settings.  6. System monitors spending and triggers alerts when limits are approached. |
| --- | --- |

| **Name:** | View Transaction History |
| --- | --- |
| **ID:** | 13 |
| **Description**: | As a student, I want to be able to get an overview of my spending during the semester so that I can analyze my spending and plan better for future semesters. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | Display Transaction history to users. |
| **Related Use Cases**: | Export transactions |
| **Actor:** | General User |
| **Invariant:** | User has an account. |
| **Precondition:** | User has logged past transactions |
| **Success Postcondition**: | User receives a detailed record of their transaction history |

| **USER STEPS**:  1. User navigates to the "Budget dashboard".  3. User opens “View Transaction History” | **SYSTEM RESPONSE**:  2. System displays the Budget Dashboard page.  4. System displays Transaction History. |
| --- | --- |

| **Name:** | Display budget progress |
| --- | --- |
| **ID:** | 10 |
| **Description**: | As a user, I want to set spending limits each month to aid in keeping me within my budget. |
| **System Under Design:** | Money Gremlin |
| **Goal:** | To display what percentage of the budget a user has used. |
| **Related Use Cases**: | View a visual representation of the budget |
| **Actor:** | General User |
| **Invariant:** | User has an account. |
| **Precondition:** | User has created budgets and categories |
| **Success Postcondition**: | The User is shown progress on their current budgets. |

| **USER STEPS**:  1. User clicks ‘Budget Dashboard’  3. Clicks ‘View Current Balance’ | **SYSTEM RESPONSE**:  2. The system Budgets Dashboard window..  4. The system displays Budget Balance |
| --- | --- |