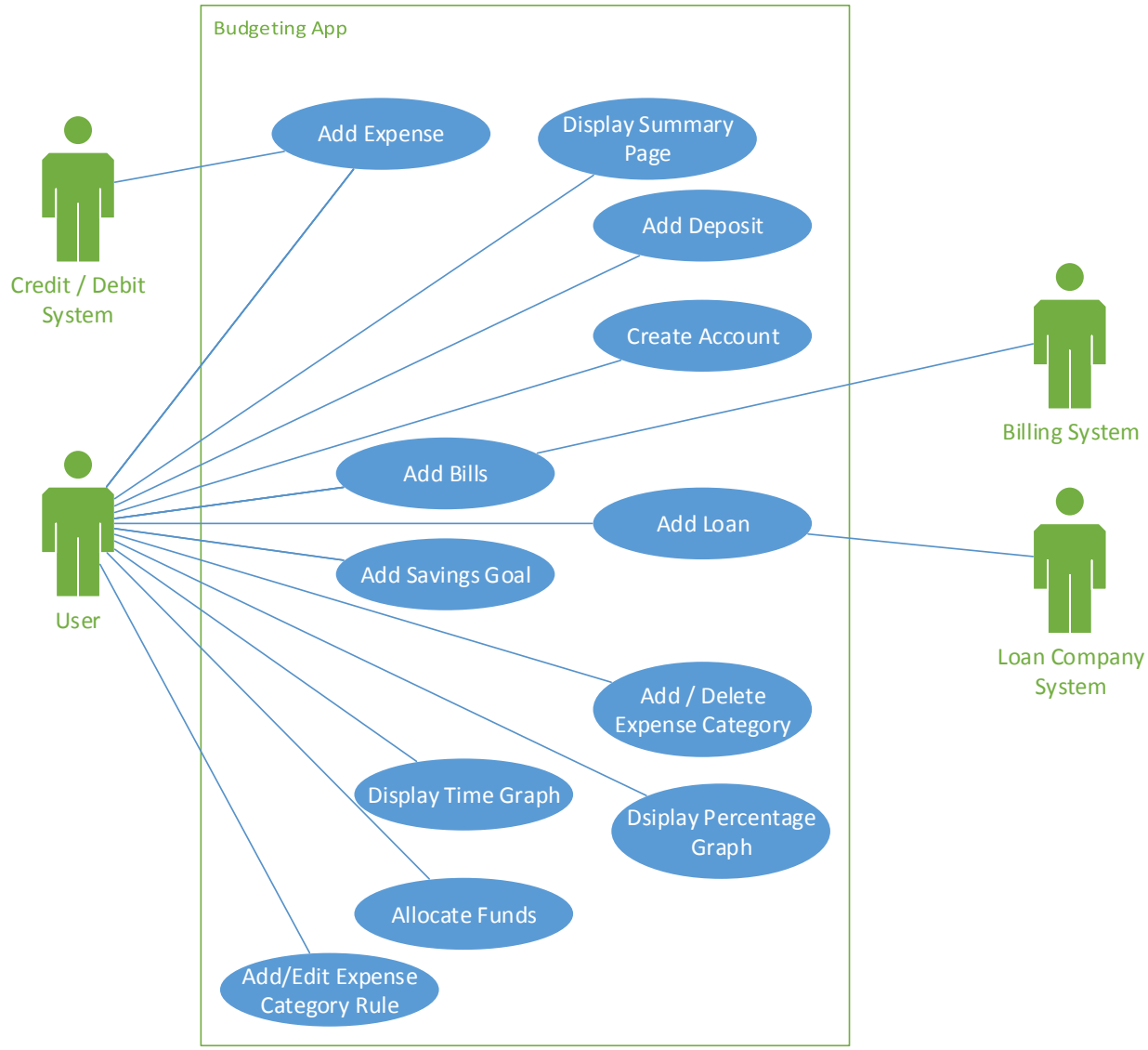
UML Use Case Model

Use Case Diagram



Use Case 1: Create Account

Scope: Budgeting Application

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: Wants to create an account for initial use of the application.

Preconditions: User has opened the application or gone to the web URL.

Success Guarantee: User account is saved and user may begin utilizing other features of the software.

Basic Flow:

1. User presses “Create Account” button.
2. On the following page, user fills out the form with name, email, and desired password.
3. System checks to see if email has not already been used.
4. Account has been created and saved, and user is taken to Summary Page.

Alternative Flow:

1. User presses “Create Account” button.
2. On the following page, user fills out the form with name, email, and desired password.
3. System finds that the given email has already been used.
4. User is notified that the email entered is already in use, and to enter a different one.
5. User enters a new email.
6. System checks and verifies it has not been used.
7. Account has been created and saved, and user is taken to Summary Page.

Frequency of Occurrence: Once per user

Use Case 2: Add Expense

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants to accurately and conveniently record any expense made.

Preconditions: User is logged in to the application.

Success Guarantee: Expense has been recorded and accurately deducted from account total and category total, and these values are updated to reflect the change.

Basic Flow:

1. User presses “Add Expense” button from the Summary Page.
2. User enters price and chooses category from a drop-down list.
3. Once user hits “OK,” they are taken back to the summary page with totals updated.

Frequency of Occurrence: Varies.

Use Case 3: Add Deposit

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants to accurately and conveniently record any deposit made.

Preconditions: User is logged in to the application.

Success Guarantee: Deposit has been recorded and totals have been updated accurately.

Basic Flow:

1. User presses “Add Deposit” button from the Summary Page.
2. User enters total value of the deposit and hits “OK.”
3. System adds deposit information to database.
4. Summary Page is loaded with updated totals.

Alternate Flow (allocate funds):

1. User presses “Add Deposit” button from the Summary Page.
2. User enters total value of the deposit and hits “OK.”
3. System uses allocation rules to apply funds to categories
4. System adds deposit information to database.
5. Summary Page is loaded with updated totals.

Frequency of Occurrence: Varies, typically once a week or less often.

Use Case 4: Display Dashboard

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants an accurate representation of finance summary and recent activity.

Preconditions: User is logged in to the application.

Success Guarantee: Summary information is displayed accurately.

Basic Flow:

1. User clicks on dashboard link from menu bar.
2. System pulls information from database to make sure all information is current.
3. Summary Page is loaded with accurate totals and graphs.

Frequency of Occurrence: Very often, as this is the homepage of the application.

Use case 5: Add Bill

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants to record reoccurring bills and rely on the application to schedule them correctly.

Preconditions: User is logged in to the application.

Success Guarantee: Bill has been recorded and accurately scheduled to be deducted on the correct date.

Basic Flow:

1. User clicks the Bills link from dashboard and is taken to the Bills page.
2. User clicks “Add Bill” button.
3. User inputs name of bill, amount, start date, and frequency of occurrence.
4. System adds this information to the database.
5. Bills page is updated with new information.

Alternate Flow:

1. User clicks the Bills link from dashboard and is taken to the Bills page.
2. User clicks “Add Bill” button.
3. User inputs name of bill, amount, start date, and frequency of occurrence.

4. One or more fields is empty or invalid, and system alerts user to enter valid information.
5. Once all entries are validated, Bills page is updated with new information.

Frequency of Occurrence: Varies, seldom after initial bills are created.

Use Case 6: Add Loan

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants to record loans and rely on the application to schedule payments correctly.

Preconditions: User is logged in to the application.

Success Guarantee: Loan has been recorded and accurately scheduled to be deducted on the correct date.

Basic Flow:

1. User clicks the Loans link from dashboard and is taken to the loans page.
2. User clicks “Add Loan” button.
3. User inputs name of loan, amount, start date, end date, interest rate, and frequency of payment.
4. System calculates monthly payment and adds loan information to the database.
5. Loans page is updated with new information.

Alternate Flow (calculate end date):

1. User clicks the Loans link from dashboard and is taken to the loans page.
2. User clicks “Add Loan” button.
3. User inputs name of loan, amount, start date, interest rate, and desired monthly payment.
4. System calculates end date and adds loan information to database.
5. Once all entries are validated, Loans page is updated with new information.

Invalid input:

1. User clicks the Loans link from dashboard and is taken to the loans page.
2. User clicks “Add Loan” button.
3. User fails to input either name of loan, amount, start date, interest rate, and either end date or monthly payment.
4. System alerts user which entries are invalid.

5. User re-enters data.
6. Once all entries are validated, system calculates either end date or monthly payment and saves loan to database.
7. Loans page is updated with current information.

Frequency of occurrence: Seldom, only once per loan.

Use Case 7: Add Savings Goal

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants to record savings goal and rely on the application to schedule contributions correctly.

Preconditions: User is logged in to the application.

Success Guarantee: Savings goal has been recorded and accurately scheduled to be contributed to as the user specifies.

Basic Flow:

1. User clicks the Savings link from dashboard and is taken to the savings page.
2. User clicks “Add Savings Goal” button.
3. User inputs name of goal, start date, amount of each contribution, and optionally: target amount, end date, interest rate, and frequency of contribution.
4. System calculates any missing information and adds saving information to the database.
5. Savings page is updated with new information.

Alternate Flow:

6. User clicks the Savings link from dashboard and is taken to the savings page.
7. User clicks “Add Savings Goal” button.
8. User fails to input one of the required fields.
9. System alerts user that one or more entries are invalid.
10. User re-enters information.
11. Once entries are validated, system calculates end date and adds loan information to database.
12. Once all entries are validated, savings page is updated with new information.

Frequency of occurrence: Occasionally, not very frequent.

Use Case 8: Add / Delete Expense Category

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants to edit expense categories quickly and conveniently.

Preconditions: User is logged in to the application.

Success Guarantee: Expense categories are updated and accurately displayed.

Basic Flow:

1. User clicks “Add / delete expense categories” button from the Summary Page.
2. System displays category page with current categories listed.
3. User clicks “Add Category” button.
4. System prompts user to input category name
5. Category is added to database
6. Category page is loaded with updated categories.

Alternate Flow (delete):

1. User clicks “Add / delete expense categories” button from the summary page.
6. System displays category page with categories listed.
7. User checks the box next to one or more categories, then hits the delete button.
8. System updates database.
9. Category page is updated with selected categories removed.

Frequency of Occurrence: Occasionally.

Use Case 9: Allocate Funds

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants to automate where deposits are allocated.

Preconditions: User is logged in to the application.

Success Guarantee: Category allocations are updated and accurately displayed.

Basic Flow:

1. User clicks “Allocate Funds” button from the Summary Page.
2. System displays settings page with current categories listed with text boxes beside them.
3. User edits text boxes with either percentages or flat amounts.
4. User hits “commit” button and system checks percentage values.
5. System saves values to database.

Alternate Flow:

1. User clicks “Allocate Funds” button from the Summary Page.
2. System displays settings page with current categories listed with text boxes beside them.
3. User edits text boxes with either percentages or flat amounts.
4. User hits “commit” button and system checks percentage values.
5. If percentage total exceeds 100, user is prompted to check values again.
6. When a valid total is produced, values are saved to database.

Frequency of Occurrence: Occasionally.

Use Case 10: Add/Edit Expense Category Rule

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants to give priorities to certain categories

Preconditions: User is logged in to the application, on the category page.

Success Guarantee: Any changes made on the page are updated accurately.

Basic Flow:

1. User clicks “Rules” button from the Category Page.
2. System displays settings page with current categories listed.
3. User assigns ranks to specified categories.
4. User hits “commit” button and system checks values.
5. System saves values to the database.

Alternate Flow:

1. User clicks “Rules” button from the Category Page.
2. System displays settings page with current categories listed.
3. User assigns ranks to specified categories.
4. User hits “commit” button and system checks values.
5. One or more category has been given the same rank, and user is notified.
5. Once all categories each have unique ranks (or no rank), system saves values to the database.

Frequency of Occurrence: Occasionally.

Use Case 11: Display Time Graph

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants a visual representation of spending / saving over time

Preconditions: User is logged in to the application.

Success Guarantee: A line graph of the given time frame is accurately displayed.

Basic Flow:

1. User clicks on “Graphs” button on summary page.
2. User selects “Running total” graph and enters parameters such as start and end date, interval, and category, if applicable.
3. When user hits “submit,” system grabs information from the database and a line graph is displayed given the specified information.

Alternative Flow:

1. User clicks on “Graphs” button on summary page.
2. User selects “Running total” graph and enters parameters such as start and end date, interval, and category, if applicable.

3a. When user hits “submit,” system checks and notifies user that the specified category did not exist during the given time frame.

3b. When user hits “submit”, system checks and notifies user that one of the information fields is empty or invalid.

4. When all information is valid, system pulls data from database and displays a line graph using the given parameters.

Frequency of Occurrence: As often as user wants to see the graph.

Use Case 12: Display Percentage Graph

Scope: SmartChart

Level: user goal

Primary Actor: User

Stakeholders and Interests:

- User: wants a visual representation of spending in each category.

Preconditions: User is logged in to the application.

Success Guarantee: A pie chart of the given categories is accurately displayed.

Basic Flow:

1. User clicks on “Graphs” button on summary page.
2. User selects “Spending Breakdown” graph and enters parameters such as start date, end date, and which categories to include if applicable.
3. When user hits “submit,” system grabs information from the database and a pie chart is displayed given the specified information.

Alternative Flow:

1. User clicks on “Graphs” button on summary page.
2. User selects “Spending Breakdown” graph and enters parameters such as start date, end date, and which categories to use, if applicable.
3. When user hits “submit”, system checks and notifies user that one or more information fields is empty or invalid.
4. When all information is valid, system pulls data from database and displays a line graph using the given parameters.

Frequency of Occurrence: As often as user wants to see the graph.