

### **Create Account**

Pre-condition: The user is on the application login screen.

Event Flow:

1. The user fills the login form with the email address and password they wish to sign up with and then clicks the "Create Account" button.
2. The system validates the account. Meaning that it checks to make sure the email address provided does not belong to an existing account.
3. The system creates a new account in the database using the provided email address and password and the user is logged in to the application.
  - a. If an account was found with the matching email address, then an error message displays informing the user that they must use a different email address.
  - b. If the password entered is less than 7 characters, then an error message displays informing the user that they must choose a stronger password.

Post-condition: The user account is created and logged in.

### **Login**

Pre-condition: The user is on the application login screen and has already created an account.

Event Flow:

1. The user fills the login with the email address and password corresponding to their account.
2. The system performs account verification by searching the database for an account with the matching email address and password combination.
  - a. If no such account could be found, then an error message displays informing the user that the provided email address and password do not match an account.
3. The system logs the user in using the account found in account verification.

Post-condition: The user account is logged in.

### **Logout**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks the "Logout" button.
2. The system checks the current screen for any unsaved data and attempts to save it, then the system logs the user out and displays the login screen.

Post-condition: The user is logged out of the application.

### **Manage Profile**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks the "Manage Profile" button.
2. The system displays the "Account Settings" screen with a form containing account information populated from the database.
3. From this screen, the user can choose to edit their email, password, identity information, link and unlink financial accounts, create alerts and delete their account. When the user finishes making changes, they click the "Save" button.
4. The system saves any changes made to the account in the database and displays the main screen.
  - a. If the account settings are missing information, then an error message is displayed informing the user of which fields need to be corrected.

Post-condition: The user's account information is updated.

### **Delete Account**

Pre-condition: The user is logged in and has loaded the "Account Settings" screen.

Event Flow:

1. The user clicks on the "Delete Account" button.
2. The system displays a prompt for the user to enter their password as a confirmation that they wish to delete all their account data from the application database.
3. The user enters their account password and clicks the "Delete" button.
4. The system erases all the data associated with the user's account and displays the application login screen.
  - a. If the supplied password does not match the one for the account, an error message displays informing the user that they entered an incorrect password.

Post-condition: The user's account data no longer exists.

### **Link Financial Account**

Pre-condition: The user is logged in and has loaded the "Account Settings" screen.

Event Flow:

1. The user clicks on the "Link Financial Account" button.
2. The system displays a prompt for the user to choose a supported financial institution from a dropdown list and then provide the username and password associated with the account they wish to link.
3. The user chooses a financial institution, enters their username and password and clicks the "Continue" button.
4. Based on the financial institution selected by the user, the system sends a request to the financial account service to access an account that matches the provided username and password.
  - a. If the user did not select an institution or provide a username and password, then an error message displays asking the user to correct those issues.
5. The financial account service receives the request and looks for an account with a matching username and password. If an account is found, then the service responds with authorization to access the account data. If the account could not be found, the service responds with an error message.
6. The system establishes a connection to the financial account through the financial account service and begins downloading a history of transaction data.
  - a. If the financial account service returned an error, this error message is displayed back to the user instead of establishing a connection.

Post-condition: The financial account is now linked to the user's Cash Stash account.

### **Create Transaction**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks on the "Create Transaction" button.
2. The system displays the "New Transaction" screen. This screen contains a form with fields for the user to enter transaction information. Required fields include a title, date of transaction, amount and type of exchange (cash, check, credit, etc.). Optional fields include a category, memo and transaction number (check #, etc.).
3. When the user finishes filling out the form, they click the "Save" button.
4. The system verifies that all the required information for creating a transaction has been filled out properly.
5. The system creates a new transaction with the details provided by the user and displays the "Transaction Log" screen.
  - a. If transaction detail verification fails, then an error message is displayed informing the user of which fields need to be corrected before attempting to save the transaction.

Post-condition: The transaction is saved in the user's account.

### **Schedule Transaction**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks on the "Schedule Transaction" button.
2. The system displays the "New Transaction" screen. This screen contains a form with fields for the user to enter details about the recurring transaction. Required fields include a title, date of transaction, amount, type of exchange (cash, check, credit, etc.), the next date the transaction should occur and the frequency at which to keep performing the transaction. Optional fields include a category, memo and transaction number.
3. When the user is finished entering the transaction details, they click the "Save" button.
4. The system verifies that all the required information for creating a recurring transaction has been filled out properly.
5. The system creates a new recurring transaction with the details provided by the user and displays the "Transaction Log" screen.
  - a. If transaction detail verification fails, then an error message displays informing the user of which fields need to be corrected before attempting to save the transaction.

Post-condition: The recurring transaction is saved within the user's account.

### **Create Budget**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks the "Create Budget" button.
2. The system displays the "New Budget" screen. On this screen the user is required to provide their gross monthly income and define categories to be used in the budget. For each category the user also needs to provide a percentage or fixed amount of income they wish to set aside for it.
3. The user fills out the required fields in the budget report form. Once the user has finished setting up their budget, they click the "Save" button.
4. The system verifies that the user has filled out all the required fields in the "New Budget" screen.
5. Based on the settings chosen by the user, the system calculates how much of the user's gross monthly income should be set aside for each category in the budget. The system then displays the budget report.
  - a. If the verification step finds missing information, an error message displays that informs the user of which fields need to be corrected.
  - b. If the verification step find that the sum of percentages defined for categories is less than 0 or greater than 100, then an error message displays informing the user to correct the percentage values.

Post-condition: A budget report is displayed based on settings chosen by the user.

### **Calculate Cash Flow**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks the "Calculate Cash Flow" button.
2. The system displays a prompt for the user to choose a month and year as a timeframe for generating the report.
3. The user chooses a month and a year from the provided lists and clicks the "Continue" button.
4. The system calculates the total income and expenses for transactions found within the timeframe chosen by the user. These totals are also broken down by category based on the categories defined within each transaction item. If a transaction item does not contain a category, it is assigned to the category "Other".
  - a. If no transactions were found within the chosen timeframe, an error message displays informing the user that the system couldn't find any transactions.
5. The system displays a cash flow report that summarizes the details previously calculated.

Post-condition: A cash flow report is displayed based on settings chosen by the user.

### **Calculate Slippage**

Pre-condition: The user is logged in to the application and has setup a budget.

Event Flow:

1. The user clicks the "Calculate Slippage" button.
2. The system displays a prompt for the user to choose a month and year as a timeframe for calculating slippage.
3. The user chooses a month and year from the provided lists and clicks the "Continue" button.
4. The system loads the budget associated with the user's account and calculates the cash flow for the chosen timeframe.
5. The system compares the total values from the budget to the cash flow broken down by category. This shows the user how close they stayed to the goals setup in their budget. These results are displayed as a slippage report.

Post-condition: A slippage report is displayed based on settings chosen by the user.

### **Check Credit Score**

Pre-condition: The user is logged in to the application and has completed their account identity information.

Event Flow:

1. The user clicks the "Check Credit Score" button.
2. The system loads the user's identity information from their account and sends this to the credit check service, asking for a credit score estimate in return.
3. The credit check service verifies the identity and performs a credit score lookup.
4. The system displays the user's credit score as received by the credit check service.
  - a. If the credit check service couldn't verify the identity information, it returns an error message. This message is displayed to the user instead of their credit score.

Post-condition: The user's credit score is displayed.

### **Export Data**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks the "Export Data" button.
2. The system displays prompt for the user to choose a location for saving the data file.
3. The user chooses a save location and clicks the "Continue" button.
4. The system compiles an excel file; each row in the file contains information related to a transaction contained within the user's account.
  - a. If the system doesn't find any transactions in the user's account, an error message displays informing the user that it couldn't find any transactions.
5. The system sends a request to the operating system to create the excel file at the user's chosen save location.

Post-condition: The operating system receives the create file request.

### **Print Data**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks the "Print Data" button.
2. The system sends a request to the printer service to print the current screen.

Post-condition: The printer service receives the print request.

## **Refresh Data**

Pre-condition: The user is logged in to the application.

Event Flow:

1. The user clicks the “Refresh Data” button.
2. The system grabs each of the linked financial accounts for the current user. For each financial account the system sends a data sync request to the appropriate financial account service.
3. The financial account service validates the account information provided and respond with transaction data related to that account.
4. After all accounts have synced, the system reloads the current screen.
  - a. If the financial account service returned error messages for any of the requests, then these error messages are displayed to the user before the screen gets reloaded.

Post-condition: The user’s account now contains all transaction data from any linked financial accounts.