

Bill Splitter for Roommates

Group 6

Zoe Moccia, Alexander Sirotkin, Lucy Gavrich, Naysa Arora, Niharika Elangovan
Zainub Waqas (Project Manager)

Description:

This web application will help roommates/housemates manage shared expenses, such as how much each roommate owes for rent, utilities, and groceries by creating a “household.” The application will support splitting bills equally among roommates, as well as custom splits for different expenses. All roommates will be able to see these bills, as no one roommate will have “admin” privileges over the group.

The application will also help them keep track of who owes who for non-shared expenses (e.g. if one roommate borrows money from another, that “bill” can be posted to be kept track of). Both the roommate who owes money, and the roommate who is owed can access those bills.

User Stories:

Title	Create Account
Description	This use case tells how a new user would create an account in the system. By creating an account a user can access the application, join a household and begin tracking and splitting expenses with their roommates.
Primary Actors	New user
Secondary Actors	No secondary Actor
Goal	Allows a user to register an account and secure access to the system
Preconditions	<ul style="list-style-type: none"> ● User does not have an account already ● Access to a working device and internet connection
Postconditions	<ul style="list-style-type: none"> ● New user account created and stored in system ● Once logged in they are redirected to dashboard page ● On the dashboard page, they can access features like joining a household, adding bills etc.
Trigger	The application's homepage would have a create account button which they would have to click
Main success scenario	<ol style="list-style-type: none"> 1. The user clicks on the create account button 2. The system shows the account registration form 3. The user enters information like name, email/userID, password 4. The user submits the information 5. The system checks this information 6. After validating the information, the system creates a new account for the user 7. The new account created is confirmed if successful 8. The user is directed to the dashboard once or login page once account creation is successful
Exceptions	<ul style="list-style-type: none"> ● The create account page or registration form fails to load (1) ● The email/userID entered is not in a valid format.(3) ● The password does not meet security requirements.(5) ● The email/userID is already associated with an existing account(5) ● The user loses internet connection while submitting the form.(4) ● A server or database error occurs during account creation.(6) ● The system fails to redirect the user after successful account

	creation.(8)
Error scenarios	<ul style="list-style-type: none"> Duplicate account : If the user already has an account associated with a given email address. The system would display an image saying an account already exists and ask the user to try logging in from that account or to click on ‘forget password’ to reset the password. (3/4/5) Invalid input: If the user enters invalid information like an incorrect email address. The system would show a message that says invalid input and ask the user to correct that information. (3/5) If there is a network error or system failure. The system would display an error message and ask the user to try again. (Any)

Title	Creating a household
Description	A registered user creates a group for themself and their other room/housemates, inviting them to join by providing their contact information.
Primary Actors	Registered user creating the household
Secondary Actors	Invited roommates
Goal	Allow a registered user to create a household so they can track bills with the roommates in said household.
Preconditions	User must have made an account User must be signed in
Postconditions	In the event of success: <ul style="list-style-type: none"> - A new household is created with some unique identifier - The name of the household is stored in the system - The household creator is added as a member of the household - Invitations are sent to provide contact information In the event of failure: <ul style="list-style-type: none"> - No household is created - The system state is unchanged - No invitations are sent
Trigger	User selects "Create Household" in the interface
Main success scenario	<ol style="list-style-type: none"> 1: User navigates to household creation 2: System displays household creation form 3: User enters a household name 4: System validates household name 5: User inputs at least one roommate's contact 6: System validates each contact's format 7: User submits form when all roommates are added 8: System creates household with a unique ID 9: System adds user as a member of the household 10: System sends invitations to provided contacts 11: System displays confirmation of the household's creation 12: System redirects to the new household's "dashboard"
Exceptions	4: User enters an invalid household name. System displays message indicating requirements and prevents form from submitting. User corrects

	<p>the name and resubmits.</p> <p>5a: User enters a contact that has already been added. System displays warning and prevents the duplicate. User can continue</p> <p>5b: User enters their own contact information. System displays warning and prevents the contact from being added. User provides alternate information</p> <p>6: User enters contacts in invalid format. System displays message indicating requirements and prevents form submission. User corrects format</p> <p>7: User attempts to submit form without at least one contact. System displays error message and prevents form submission. User adds at least one valid contact</p>
Error scenarios	<p>Any: Session times out. System displays timeout message and redirects to login. User may log in again and restart process</p> <p>8/9: Database connection fails, server error, timeout, etc. System displays generic message and no changes occur. User can retry household creation</p> <p>10: Invitations fail to deliver. Household is still created and message is displayed indicating which invitations failed. Option to resend is provided. User can retry or provide alternate information</p>

Title	Posting a bill
Description	A household member adds a bill to the splitter, designating who is responsible for various portions of the amount.
Primary Actors	Registered user, and household member, who creates the bill
Secondary Actors	Other household members who are responsible for portions of the bill
Goal	Allow a household member to create a bill to be added to the household split. They should be able to designate specific billees and either amount or percentage splits for responsible parties.
Preconditions	<ul style="list-style-type: none"> - User must have an account and be signed in - User must be a member of a household
Postconditions	<p>In the event of success:</p> <ul style="list-style-type: none"> - A new bill is created, with its relevant data stored - Notifications are sent to billees asking them to approve owed amount <p>In the event of failure:</p> <ul style="list-style-type: none"> - No bill is created - No debts are changed - No notifications are sent
Trigger	User selects “New Bill” in the interface
Main success scenario	<ol style="list-style-type: none"> 1. User selects “New Bill” 2. System displays bill creation form 3. User enters a bill title 4. System validates bill title 5. User enters a bill amount 6. System validates bill amount 7. User selects a bill category 8. User selects a splitting option (amount or percent) 9. User selects household members to add to bill 10. User designates amounts/percentages to household members 11. System validates split 12. User submits form when all info is entered and valid 13. System creates a new bill, marked as pending, and stores its data 14. System notifies billees that a new bill has been created and asks them to approve (or reject) amount owed 15. System redirects biller to household dashboard

Exceptions	<p>4. User enters an invalid bill title (blank, etc.). System displays message indicating requirements and prevents form from submitting. User corrects the bill title and resubmits.</p> <p>6. User enters an invalid bill amount. System displays message indicating requirements and prevents form from submitting. User corrects the bill amount and resubmits.</p> <p>9. User enters amounts that do not add up to bill total or percentages that do not add up to 100. System displays message with the difference and a prompt to adjust amounts/percentages. User corrects the amounts/percentages and resubmits.</p> <p>10. User submits bill without adding a billee. System displays error message prompting user to add at least one billee and prevents form submission. User adds at least one billee and resubmits.</p>
Error scenarios	<p>Any. Session times out. System displays timeout message and redirects to login. User may log in again and restart process.</p> <p>13. Database connection fails, server error, timeout, etc. System displays generic message and no changes occur. User can retry bill creation.</p> <p>14. Notification(s) fail to deliver. Bill is still created and message is displayed indicating which notifications failed. Option to resend is provided. User can retry or provide alternate information.</p>

Title	Pay amount
Description	A household member indicates that they have paid their share of a bill using an external method (such as cash or Zelle). The payment is recorded but must be approved before balances are officially updated.
Primary Actors	The household member making the payment.
Secondary Actors	The Household members receiving the payment.
Goal	To allow a user to mark a bill as paid while keeping balances unchanged until the payment is approved.
Preconditions	<ul style="list-style-type: none"> - User is logged in - User is part of a household - There is an approved bill with money owed by the user
Postconditions	<p>In the event of success:</p> <ul style="list-style-type: none"> - The payment is saved as pending - A balance change is suggested but not applied - Other household members are notified <p>In the event of Failure:</p> <ul style="list-style-type: none"> - No payment is saved - Balances remain the same
Trigger	User selects “Pay Amount” on a bill they owe money on
Main success scenario	<ol style="list-style-type: none"> 1. User opens the list of unpaid bills 2. User selects a bill 3. System shows bill details and amount owed 4. User selects “Pay Amount” 5. The user chooses how they paid (cash, Zelle, etc.) 6. User confirms the payment 7. System saves the payment as pending 8. System notifies other household members 9. User is taken back to the dashboard
Exceptions	<p>5. The user tries to continue without choosing a payment method. The system shows an error and asks the user to select one.</p> <p>6. The user cancels before confirming the payment. No payment is saved and no changes are made.</p>
Error	Any: System displays timeout message and redirects to login. User may log

scenarios	<p>in again and restart process.</p> <p>7. Database connection fails, server error, or timeout occurs while saving the pending payment. The system displays a generic message and no payment is recorded.</p> <p>8. Notification delivery fails. Pending payment is still recorded, but the system displays a message indicating which notifications failed. The option to resend is provided. The user may retry or continue without resending.</p>
------------------	--

Title	Accepting a proposed bill/payment
Description	A household member accepts a proposed bill/payment another member has made. After all affected parties accept the proposal, the bill will be added or the payment will be confirmed and balance will be updated.
Primary Actors	User accepting the proposal
Secondary Actors	User who initiated the proposal
Goal	To allow the user to ensure that the proposed bill is warranted or the payment was actually made.
Preconditions	<ul style="list-style-type: none"> - User is logged in - User is a member of a household - There is a pending bill/payment that must be accepted by the user
Postconditions	<p>Success:</p> <ul style="list-style-type: none"> - If accepting a <i>bill</i>: <ul style="list-style-type: none"> - Unresolved bill will be added to the household - Members included in the bill will be notified - If accepting a <i>payment</i>: <ul style="list-style-type: none"> - Balance of the payer will be updated - If the paying user was the last one needed to pay the bill, the bill will be marked as resolved - Paying member will be notified that the payment was accepted <p>Failure:</p> <ul style="list-style-type: none"> - Upon failure of the “accept” action (not necessarily “denying” the proposal), the proposal will be left as pending - The option to accept/deny is still available - No changes are made to the proposer’s balance (if it was a payment proposal)
Trigger	User selects “Accept” action on a pending bill or payment proposal.
Main success scenario	<p><i>Bill:</i></p> <ol style="list-style-type: none"> 1. User selects the proposed bill on the dashboard/bill page 2. System displays the details of the bill (amount, proposer, included members, etc.) 3. User selects the “Accept” option 4. User’s selection is registered on the bill

	<ol style="list-style-type: none"> 5. If the user is the last one who needed to accept the bill, the bill is added to the household as an unresolved bill <ol style="list-style-type: none"> a. Else, the bill is still pending until all other members included on the bill accept 6. Other members included in the bill (including the proposer) are notified of the acceptance 7. User is taken back to the dashboard <p><i>Payment:</i></p> <ol style="list-style-type: none"> 1. User selects the proposed payment on the dashboard/bills page 2. System displays the details of the payment (amount, who's paying, which bill the payment is for, etc.) 3. User selects the "Accept" option 4. Payment is registered on the bill 5. If the payment is not in full for that user, the amount the payer owes to the bill is adjusted accordingly 6. Balance of the payer is adjusted in accordance with the payment 7. If the accepted payment is in full and was the last one on the bill, the bill is marked as resolved <ol style="list-style-type: none"> a. Else, the bill's balance is updated but the bill is still unresolved and requires the other members to pay 8. Other members included in the bill (including the payer) are notified that a payment went through 9. User is taken back to the dashboard
Exceptions	3 (for both scenarios). User may instead select "Deny" and the proposal is rejected
Error scenarios	<p>Any: System displays timeout message and redirects to login. User may log in again and restart process</p> <p>4 (for both scenarios). Database connection fails, server error, or timeout occurs while registering acceptance on bill. The system displays a generic message and the acceptance is not recorded. The option to accept the proposal is still available.</p> <p>6 (<i>Bill</i>) & 8 (<i>Payment</i>). Notification delivery fails. Bill or payment acceptance still goes through, but the system displays a message indicating which notifications failed. The option to resend is provided. The user may retry or continue without resending.</p>