Software Requirements Specification (SRS) for Even Up

Group - 5
Anshika Gupta - CS22BTECH11007
Yash Patel - CS22BTECH11047
Monish Asawa - MA22BTECH11014
Anirudh Saikrishnan - CS22BTECH11004

February 16, 2025

Contents

1 Introduction

1.1 Purpose

EvenUp is a user-friendly expense-management tool that takes the stress out of tracking and splitting bills among groups of friends or colleagues. Creating a group for each trip, event, or shared project allows you to add expenses in just a few taps, specify which individuals are responsible, and mark who paid. EvenUp instantly calculates each person's share, ensuring accuracy and preventing confusion.

1.2 Scope

Even Up allows users to:

- Sign up, log in, and manage their accounts securely.
- Create, edit, and delete individual and group expenses with descriptions and bill attachments.
- Automatically calculate and split costs among users, either equally or unequally based on user inputs.
- Manage various group types (e.g., One-Time Split, Grey Groups) for effective financial tracking.
- View balances, transaction history, and spending insights.
- Send payment reminders to group members for pending settlements.
- Simplify transactions within a group by optimizing settlements.
- Analyze expenses and track spending patterns.

1.3 Out of Scope

- The app does not support direct transactions between users through integrated payment.
- Users cannot split expenses in multiple currencies.
- The app cannot simplify transactions between users in different groups.

1.4 Acronyms

- OTS: One-Time Split

1.5 Definitions

- **Expense:** A financial transaction recorded in the system where money is spent by one or more users within a group.
- **Settle Up:** The process of clearing outstanding balances between two participants by making payments.

- **One-Time Split (OTS) group:** A type of group where users add all their expenses first and then confirm that they have added all their expenses. Once all participants have approved, the Settle Up option is enabled for the participants and the balances in the group are displayed, and the corresponding balances are updated in the friends section. This type of group is ideal for one-time events, such as trips or gatherings, as it consolidates all expenses before settlement, minimizing the number of transactions needed between participants.
- **Grey Group:** A type of group where users can only view expenses in which they are directly involved and can only see their own balances. This setup helps maintain confidentiality while still allowing users to split expenses easily. Additionally, it reduces clutter by displaying only relevant expenses, ensuring users focus solely on their own expenses.
- **Private Split:** A special type of expense split where a user can create a private expense record with another user without involving a larger group.
- **Friends Section:** A section in the application where a user can view balances involving their friends. This helps track financial interactions with individual users across different groups.

1.6 Overview

This SRS document is divided into the following sections:

- Section 2: Overall description of the system.
- Section 3: Specific requirements and use cases.
- Section 4: Future extensions.
- Section 5: Appendix including UI screens.

2 Overall Description

2.1 Product Perspective

Even Up is a mobile application designed to efficiently manage shared expenses and debts. This product is intended to be a mobile application designed to run on Android as well as on iOS devices.

2.2 Product Functions

Category	**Use Cases**	**Description of Use Cases**
System Authorization	Sign Up	Create a new account
	Login	Authenticate using username and password
	Logout	Log out from the system
	Forget Password	Change password from login page
	Reset Password	Reset password, if required
Group Management	Create a Private Split	Create a new private split with desired person
	Create Group	Create a new group for expense sharing
	Delete Group/ Private Split	Remove a group/ private-split when no longer needed
	Join group	Join an existing group
	Exit from Group	Exit from an existing group
Expense Management	Add Expense in Group	Log a new expense within a group
	Edit Expense	Modify an existing expense entry
	Delete Expense	Remove an expense record
Settlements	Settle Up	Clear outstanding balances among group members
	Give Reminder	Send reminders for pending payments
	Confirm Expense Submission in OTS Group	Confirm that all expenses have been added in OTS Group.
Financial Tracking	View Balances	Check the current balance of expenses
	View Expenditures Analysis	View detailed analysis of expenditures
	Show All Personal Transactions	Display all personal financial transactions
Interface Preferences	Night Mode	Toggle between Night mode and Light mode

Table 1: Use Case Table

2.3 User Characteristics

- Users should be familiar with mobile applications.
- Basic financial literacy is expected for managing expenses.
- User should have an email account.

- User has access to a good internet connection.

2.4 Operating Environment

- **Supported Platforms:** Android
- **Programming Languages:** Flutter (for front-end development), Go (for back-end services)
- **Database:** PostgreSQL

3 Specific Requirements

3.1 Functional Requirements

Below is a detailed description of each use case. The functionality remains identical to the original specification, but additional text has been added to elaborate on each use case.

3.1.1 Use Cases related to Authorization

Use Case 1: Sign Up

- **Primary Actor:** User (an individual who wants to create a new account for EvenUp).
- **Preconditions:**
  - The user does not have an existing account in the system.
  - The device has stable internet connectivity to communicate with the backend.
- **Main Flow:**
  1. The user selects the "Sign Up" option on the application's welcome or login screen.
  2. The system presents a form requesting essential account details such as email, password, and possibly other identifiable information (e.g., name).
  3. The user fills in the form and taps the "Sign Up" button.
  4. The system verifies the correctness and uniqueness of the provided details (e.g., checking if the email is already registered).
  5. Upon successful verification, the system creates the new account in its database and confirms the account creation to the user.
  6. The user is redirected to the main page or dashboard after the account setup is complete.
- **Alternate Flow:**
  1. If the provided email address is already associated with an existing account, the system notifies the user about the conflict and prompts them to try a different email or log in instead.

2. If the password entered does not meet security requirements (e.g., length, character complexity), the system displays an error message and requires the user to enter a stronger password before proceeding.
3. If there is a temporary network issue, the system may prompt the user to retry the registration once the connection is stable.

Use Case 2: Log In

- **Primary Actor:** User (someone who already possesses an EvenUp account).
- **Preconditions:**
  - The user has previously registered for an account.
  - The device must be connected to the internet to verify credentials with the backend.
- **Main Flow:**
  1. The user opens the app and is presented with the login screen.
  2. The user inputs the correct email and password into the corresponding text fields.
  3. The user selects the "Login" button to initiate the authentication process.
  4. The system checks the credentials against its stored records.
  5. Upon successful authentication, the system directs the user to the main dashboard or home screen, granting access to all authorized functionalities.
- **Alternate Flow:**
  1. If the email and password do not match any existing records, the system displays an error message indicating invalid credentials.
  2. If the user forgets the password, there is a "Forgot Password" link or button that leads to the relevant password recovery process (as detailed in Use Case 4).

Use Case 3: Log Out

- **Primary Actor:** User (anyone currently logged in to the system).
- **Preconditions:**
  - The user is already authenticated and using the application.
- **Main Flow:**
  1. The user selects the "Log Out" option found in the user profile menu.
  2. The application clears any cached session tokens and redirects the user to the login screen.
  3. The system ensures that protected data can no longer be accessed without re-authentication.

Use Case 4: Forgot Password

- **Primary Actor:** User (a registered user who has forgotten their password and is currently logged out).
- **Preconditions:**
  - The user's account exists in the system but user is logged out.
  - The user cannot remember the correct password.
- **Main Flow:**
  1. The user chooses the "Forgot password" option from the login page.
  2. The system directs the user to a dedicated "Forgot Password" screen.
  3. The system prompts the user to enter the registered email address for verification.
  4. After the user enters the email, the system sends a security token or password reset instruction via email.
  5. The user receives the email, obtains the one-time password or reset link, and enters the provided security information back in the app.
  6. The user is prompted to create a new, strong password that meets security criteria.
  7. Once successfully updated, the system confirms the password reset and returns the user to the login page.
- **Alternate Flow:**
  1. If the user enters an email address not found in the system, an error message appears, prompting the user to try again or check for any typos.
  2. If the user inputs an incorrect security token, the system rejects the reset attempt and requests the user to re-enter the correct information.
  3. If the new password does not meet security policies, the user is instructed to choose a stronger password before proceeding.

Use Case 5: Reset Password

- **Primary Actor:** User (already logged in and wishes to change the password).
- **Preconditions:**
  - The user is currently authenticated in the system and can access the profile page.
- **Main Flow:**
  1. The user navigates to the profile or settings page and selects "Reset Password."
  2. The system requests the current password for security purposes and prompts the user to enter it.

3. The system also asks the user for a new password and a confirmation of the new password.
4. The system verifies the old password's correctness and checks that the new password meets the security requirements.
5. Upon validation, the password is updated in the system, and the user is redirected to the main page.

- **Alternate Flow:**
  1. If the user fails to provide the correct current password, the system displays an error and prompts them to retry.
  2. If the new password is too weak or does not match its confirmation field, the system instructs the user to provide a valid new password.

3.1.2 Group Management

Use Case 6: Create Private Split

- **Primary Actor:** User (currently logged in, wishing to create a private expense record with one specific person).
- **Preconditions:**
  - The user is authenticated and has a stable internet connection.
- **Main Flow:**
  1. The user opens the application's interface and selects the "Create Private Split" option.
  2. The system displays a field where the user can input the username of the person with whom they intend to create the private split.
  3. The user enters the specific username and confirms the action to create a private split.
  4. The system validates the username to ensure it matches an existing user.
  5. The private split is then created, and both parties can begin adding or viewing their expenses within this private context.
- **Alternate Flow:**
  1. If the user enters a username that does not exist in the system, an error message is displayed, prompting the user to check the spelling or confirm the username before attempting again.
  2. If there is a temporary networking issue or other system constraints, the creation process may fail, and the user would be notified to try again later.

Use Case 7: Create Group

- **Primary Actor:** User (currently logged in and wanting to create a group for shared expenses).

- **Preconditions:**
  - The user's credentials are valid, and they are authenticated in the application.
- **Main Flow:**
  1. The user navigates to the "Create Group" feature.
  2. A dialogue prompts the user for the name of the group, and possibly an optional description or additional settings (such as selecting if the group is OTS, Grey, or a standard group).
  3. The user confirms the choices and submits the group creation request.
  4. The system registers the new group in the database and generates an invite link or code.
  5. The user can share this invite link with others, enabling them to join and collaborate on expense tracking.
- **Alternate Flow:**
  1. If the user leaves the name field empty or fails to provide any required information, the system rejects the creation attempt, prompting the user to fill all required fields.
  2. Any connectivity error or server-side issue would produce a notification prompting the user to retry once the issue is resolved.

Use Case 8: Delete Group/ Private-Split

- **Primary Actor:** User (logged in and currently a member of the group or the private-split in question).
- **Preconditions:**
  - The user is a part of the group.
- **Main Flow:**
  1. The user selects the relevant group or private-split from their list of groups.
  2. The user clicks on a "Delete Group" or "Delete Private Split" button.
  3. The system requests a final confirmation to avoid accidental deletions.
  4. If the user confirms, the group or private-split is removed from the system's database.
  5. All members are unable to access it subsequently, and no further expenses can be added or viewed.
- **Alternate Flow:**
  1. If there are outstanding debts not fully settled within the group or private-split, the system displays an error indicating that deletion is only possible after all balances have been settled.

2. The user may have to ensure all settlements are done, or contact group members to finalize balances before attempting deletion again.

Use Case 9: Join Group

- **Primary Actor:** User (wishing to become a member of an existing group).
- **Preconditions:**
  - The user has received an invite link or group code from a friend or group creator.
  - The user is logged in to the application.
- **Main Flow:**
  1. The user clicks the invite link or manually enters the group code in the app.
  2. The system identifies the relevant group and displays a prompt confirming that the user wants to join.
  3. The user confirms by selecting a “Join” or similar button.
  4. The system adds the user to the group’s member list, enabling them to view or add expenses as defined by the group’s rules (e.g., if it is a Grey Group, restricted viewing rules apply).
- **Alternate Flow:**
  1. If the user attempts to open the invite link while not logged in, the system redirects them to the login page. After a successful login, the user is taken back to the group join prompt.
  2. If the group is no longer active or has been deleted, the system notifies the user that the link is invalid.

Use Case 10: Exit from Group

- **Primary Actor:** User (currently a member of the group they wish to leave).
- **Preconditions:**
  - The user is logged in and a part of the group.
- **Main Flow:**
  1. The user opens the group’s details page and selects the “Exit Group” option.
  2. The system asks for a confirmation to prevent accidental removals.
  3. Once confirmed, the user is removed from the group’s member list and no longer has access to its expenses or data.
- **Alternate Flow:**
  1. If the user still has unsettled balances within the group (i.e., they owe money or are owed money), the system displays an error message and prevents the user from leaving until all debts are resolved.

2. The user may settle up any outstanding amounts and attempt the exit process again.

3.1.3 Expense Management

Use Case 11: Add Expense

- **Primary Actor:** User (currently logged in and a member of an existing group or private-split).
- **Preconditions:**
  - The user is part of the group or has a private-split with the person(s) for which the expense is being recorded.
- **Main Flow:**
  1. The user opens the specific group or private-split interface.
  2. The user selects "Add Expense," initiating a form where expense details are required.
  3. The form typically requests an expense description, a category tag (e.g., Travel, Food, Necessities, Others), and the total amount spent.
  4. The user specifies who contributed to the expense and how much each contributed
  5. The expense is then recorded, becoming visible to all relevant parties (except in a Grey Group, where it's only visible to participants involved in that particular expense).
- **Alternate Flow:**
  1. If a required field (e.g., total amount or description) is not filled in, the system displays an error message and does not add the expense until the information is complete.
  2. If the sum of the amounts contributed by individuals does not match the total expense or the shares owed do not match the total, the system notifies the user of the discrepancy. The user must correct these entries before finalizing.
  3. A temporary connection issue may cause the expense addition to fail, prompting the user to retry.

Use Case 12: Edit Expense

- **Primary Actor:** User (currently logged in and part of an existing group or private-split with at least one expense recorded).
- **Preconditions:**
  - The user is a part of the group/private split.
- **Main Flow:**
  1. The user locates the expense within the group or private-split expense list.

2. The user chooses the “Edit Expense” option, prompting the system to display the current details of the expense.
3. The user updates fields such as the description, category, contributors, or amounts.
4. The system validates the new data, ensuring that contributions and splits remain consistent.
5. Once confirmed, the edited expense is updated in the group or private-split record, and other members are notified of the change if necessary.
6. If it is a Grey Group, only the involved parties see and receive the updates.

- **Alternate Flow:**
  1. If any of the fields are left blank or the new amounts do not add up correctly, the system will refuse to save the changes, prompting the user to correct errors.
  2. If a user involved in the expense has left the group, the system will generate an error message because the user's share may no longer be valid. The expense can remain unchanged until the situation is addressed.
  3. If someone already partially settled a share of the expense prior to the edit, the system recalculates the new net owed amounts, showing the correct difference to reflect partial settlements.

Use Case 13: Delete Expense

- **Primary Actor:** User (currently logged in and part of an existing group or private-split that contains the expense to be removed).
- **Preconditions:**
  – The user is already logged in and part of an existing group/ private- split with at least one expense recorded.
- **Main Flow:**
  1. The user navigates to the expense they wish to remove.
  2. The user taps or clicks the “Delete Expense” option.
  3. The system displays a confirmation message to ensure the user truly intends to delete the expense.
  4. If confirmed, the expense record is removed from the database.
  5. Balances within the group or private-split are recalculated, reflecting that the expense no longer exists.
- **Alternate Flow:**
  1. If a participant involved in the expense has left the group, the system disallows the deletion
  2. If the user decides not to confirm deletion at the prompt, the expense remains intact.

3. If there is a system error or network failure, the expense may not be removed, prompting the user to retry once the issue is resolved.

3.1.4 Settlements

Use Case 14: Give Reminder

- **Primary Actor:** User (logged in and owed money by another participant).
- **Preconditions:**
  - The user has outstanding balances in which they are the creditor (another user owes them).
- **Main Flow:**

  Method 1: Reminder via the Group/Private-Split Balances

  1. The user opens the relevant group or private-split and navigates to its balance summary.
  2. The user locates the specific unsettled balance where they are owed money.
  3. The user taps the "Remind" button next to the unsettled balance.
  4. The system requests final confirmation.
  5. On confirmation, the system sends an email reminder to the participant who owes the money, detailing the amount and context of the expense.

  Method 2: Reminder via the Friends Section

  1. The user navigates to the "Friends" section, which summarizes all balances with different contacts.
  2. The user identifies a friend who owes them money and selects the "Remind" option corresponding to that unsettled balance.
  3. The system requests confirmation to ensure the reminder is intended.
  4. Once confirmed, an email reminder is dispatched to the owing party, listing the amount owed and relevant group or expense references.
- **Alternate Flow:**
  1. If the user opts to cancel at the confirmation prompt, no reminder is sent.
  2. If there is a system or network error at the time of sending, the reminder may not be delivered. The user may receive a notification to retry once the system is operational again.

  Note: The reminder option is only visible for balances where the user is owed money.

Use Case 15: Settle Up

- **Primary Actor:** User (logged in, seeking to mark a balance as settled once payment is made outside the app).
- **Preconditions:**
  - An existing unsettled balance exists between two or more users.
  - The user intends to clear the balance by acknowledging that real-world payment has been made.
- **Main Flow:**

  Method 1: Settle Up via the Group/Private-Split Balances

  1. The user opens the group or private-split view and locates the unsettled balance they wish to clear.
  2. They click or tap "Settle Up" for that specific balance.
  3. The system presents a dialog or screen asking for confirmation of the settle-up action.
  4. The user provides confirmation, indicating they have paid or received payment externally.
  5. Once the user's confirmation is logged, the system waits for the other party's confirmation.
  6. When both parties confirm, the expense is marked as settled, and the net balances are updated accordingly.

  Method 2: Settle Up via the Friends Section

  1. The user navigates to the "Friends" section, which lists all outstanding debts and credits with various contacts.
  2. The user selects the "Settle Up" option for the friend in question.
  3. The system presents a confirmation prompt, reminding the user that no actual payment is processed through the app.
  4. If the user proceeds, the system records that the user has performed their part of the settlement.
  5. The other involved party must also confirm or mark the payment as received to finalize the settlement.
  6. Once confirmed by both sides, the system updates or removes the unsettled balance, reflecting that the debt is cleared.
- **Alternate Flow:**
  1. If the user cancels the operation at the confirmation prompt, no changes are made, and the balance remains unsettled.
  2. If only one party confirms while the other does not, the system keeps the balance in a partially settled state until the second user also acts.

3. Network or server errors may require the user to retry the settle-up action later.

Note:

- The user is expected to pay the balance amount to the person he/she owes. (This payment cannot be done via the app.)
- Once a party has given confirmation from their end, it cannot be revoked.

Use Case 16: Confirm Expense Submission in OTS Group

- **Primary Actor:** User (logged in and participating in a One-Time Split (OTS) group).
- **Preconditions:**
  - The user has finished adding all relevant expenses for the event or trip in the OTS group.
- **Main Flow:**
  1. The user opens the OTS group and selects the "Added all expenses" or similar confirmation button.
  2. The system marks that the user has finalized their expense entries and no longer allows this user to add new expenses in the OTS group.
  3. A notification or status update indicates the user's readiness for final settlement.
  4. Once all group members have indicated they have added their expenses, the system unlocks the "Settle Up" feature for the group.
  5. The net balances become visible to all participants, showing how much each person owes or is owed.

  Note: After clicking on 'Added all expenses', no expenses can be added by the particular user, so it is recommended to add all the expenses first, then click on this button.

3.1.5 Financial Tracking

Use Case 17: View Balances

- **Primary Actor:** User (logged in and wanting to see current debts or credits).
- **Preconditions:**
  - The user has participated in at least one group or private-split with recorded expenses.
- **Main Flow:**

  Method 1: View Balances from the "Friends" Section

1. The user navigates to the “Friends” section within the app’s menu.
2. A consolidated list of all individuals with whom the user has unsettled balances is displayed.
3. The user can see how much they owe or are owed in total by each friend.

Method 2: View Balances for a Particular Group

1. The user selects a specific group from their groups list.
2. The group’s balance summary is shown at the top or on a dedicated “Balances” tab.
3. The user can see each member’s status—who owes money and who is owed.

Use Case 18: View Expenditure Analysis

- **Primary Actor:** User (logged in and looking to get detailed insights about group spending).
- **Preconditions:**
  - The user is part of a group with recorded expenses.
- **Main Flow:**
  1. The user opens the group for which they want a breakdown of expenses.
  2. The user taps or clicks on the “Show Analysis” button or tab dedicated to analytics.
  3. The system compiles and displays a detailed breakdown of expenses.
  4. The user can review these insights to understand spending patterns, top expense contributors, or other summary statistics.

Use Case 19: Show All Personal Transactions

- **Primary Actor:** User (authenticated and wanting a personal transaction history).
- **Preconditions:**
  - The user has completed or participated in one or more transactions (expenses or settlements) within the system.
- **Main Flow:**
  1. The user navigates to a “Transactions” or “History” section in the application.
  2. The system displays a chronological list of all personal transactions, including expenses paid, expenses shared, and settlements.
  3. Each entry typically includes a date, description, and the corresponding amount.
  4. The user can scroll to review past transactions and may filter or sort them for convenience.

Use Case 20: Night Mode

- **Primary Actor:** User (currently logged in).
- **Preconditions:**
  - The user is able to access the settings or interface preferences.
- **Main Flow:**
  1. The user locates the “Night Mode” or “Dark Theme” toggle within the application’s settings.
  2. The user activates the toggle, switching the interface to darker tones designed for low-light conditions or personal preference.
  3. All screens, including text, backgrounds, and controls, adapt to a darker palette.
- **Alternate Flow:**
  1. If the user toggles off “Night Mode,” the application reverts to the default light theme.

4 Non-Functional Requirements

- **Security:** The application must use secure user authentication, ensuring that user credentials are protected with encryption in transit and at rest.
- **Performance:** The system should respond to common requests, such as adding an expense or loading a group’s balances, within 2 seconds under normal network conditions.
- **Usability:** The user interface should be consistent and intuitive, following well-established design principles to minimize user confusion.
- **Maintainability:** Good programming practices must be followed, including clear documentation, meaningful naming conventions, and modular architecture, to ensure the system can be updated or extended with minimal effort.

5 Future Extensions

- Allow users to add expenses in different currencies.
- One user can send a notification (instead of an email) to another reminding them to settle the payment.
- Introduce recurring expenses for regularly repeated costs (e.g., monthly subscriptions, weekly groceries in a shared apartment).
- Share the balances via messaging platforms like WhatsApp.
- Add a chat feature for group discussions related to expenses.

6 Appendix

6.1 User Screens

Illustrations of key user interface screens.

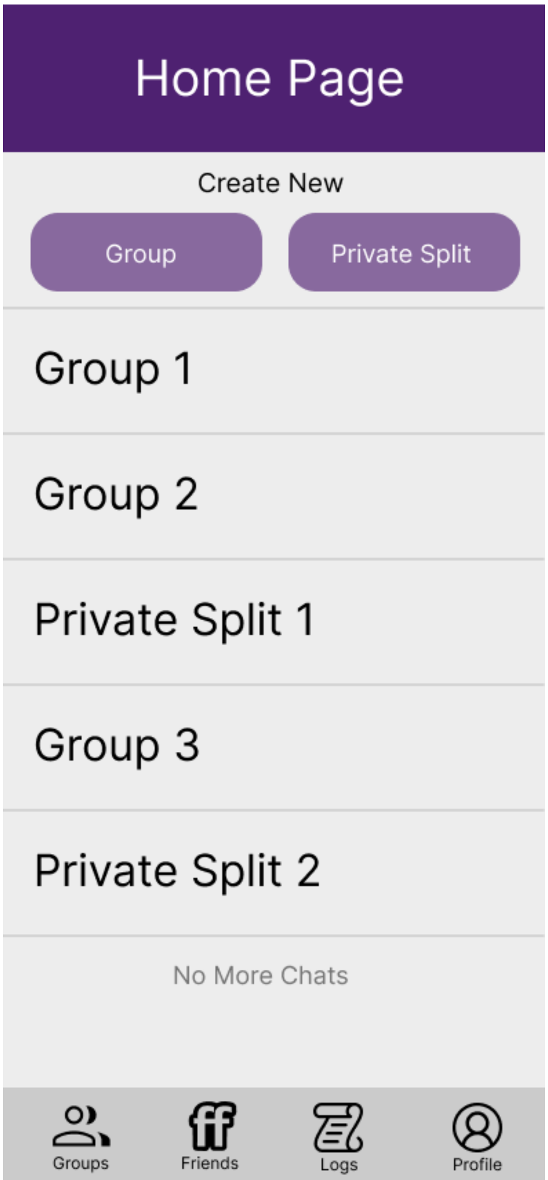


Figure 1: Home Page

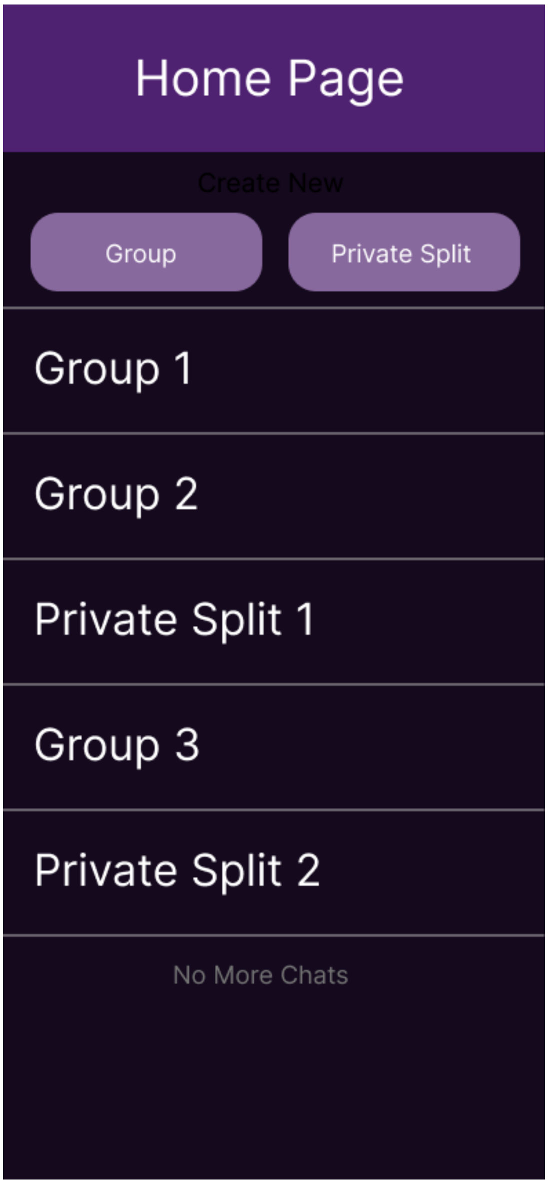
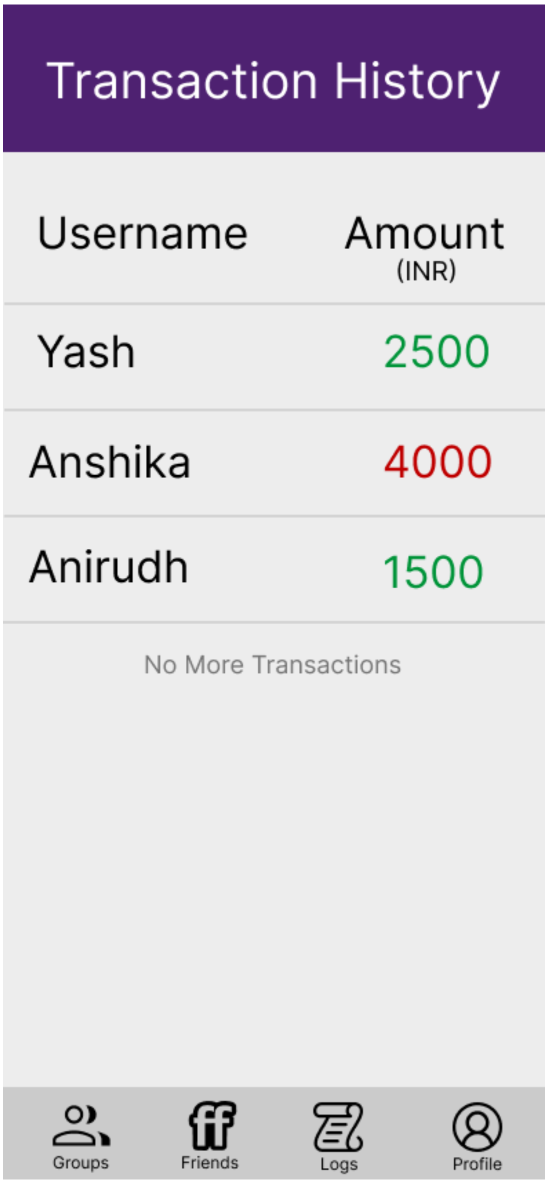


Figure 2: Home Page Night Mode

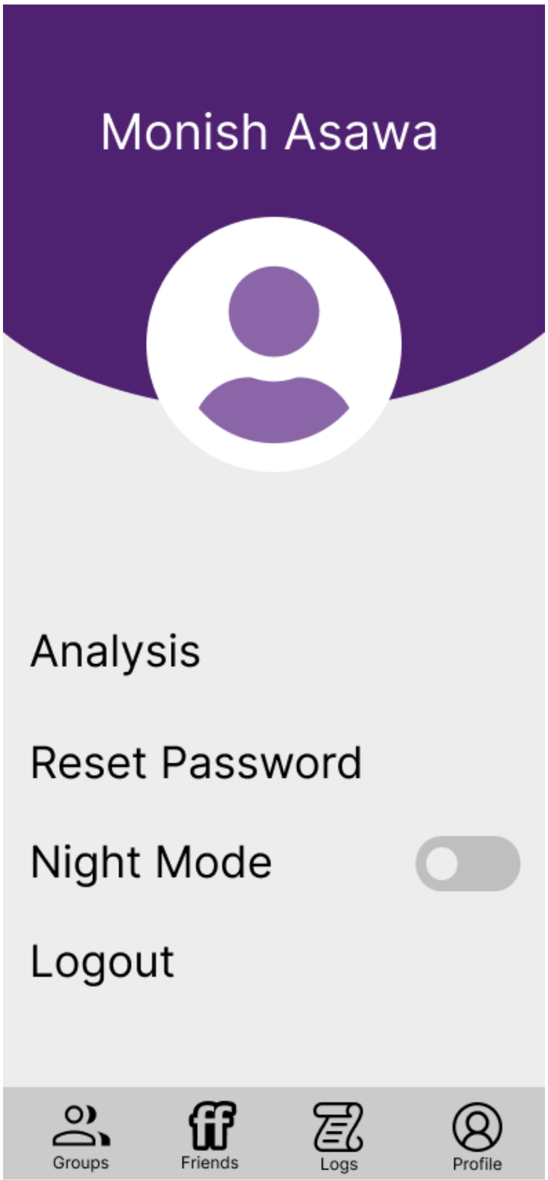


Transaction History

Username	Amount (INR)
Yash	2500
Anshika	4000
Anirudh	1500

No More Transactions

Figure 3: Transaction History



Monish Asawa

Analysis

Reset Password

Night Mode ○

Logout

Figure 4: User Profile

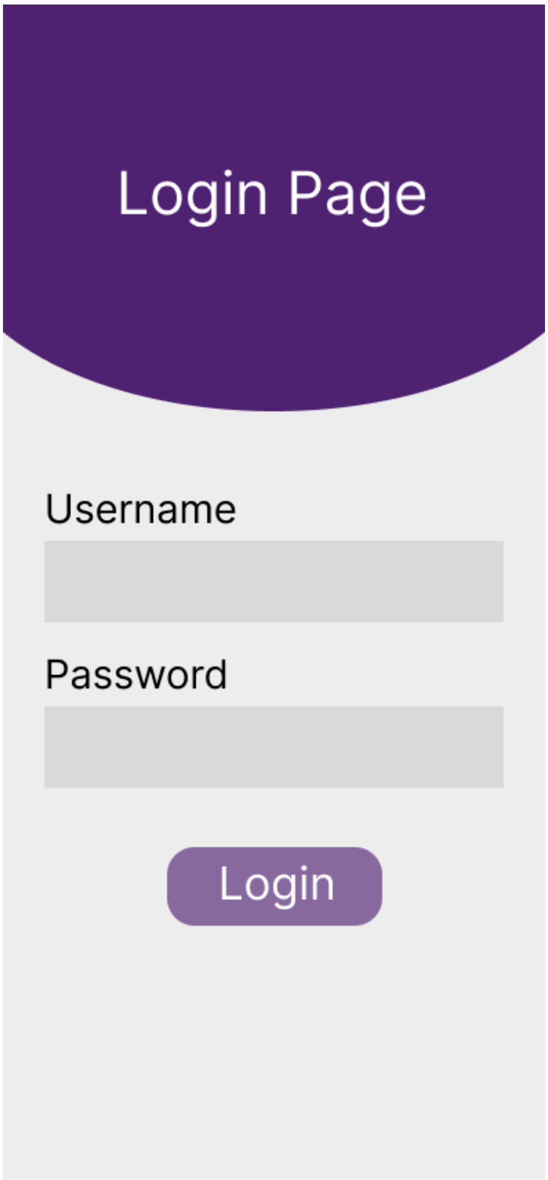


Figure 5: Login Page

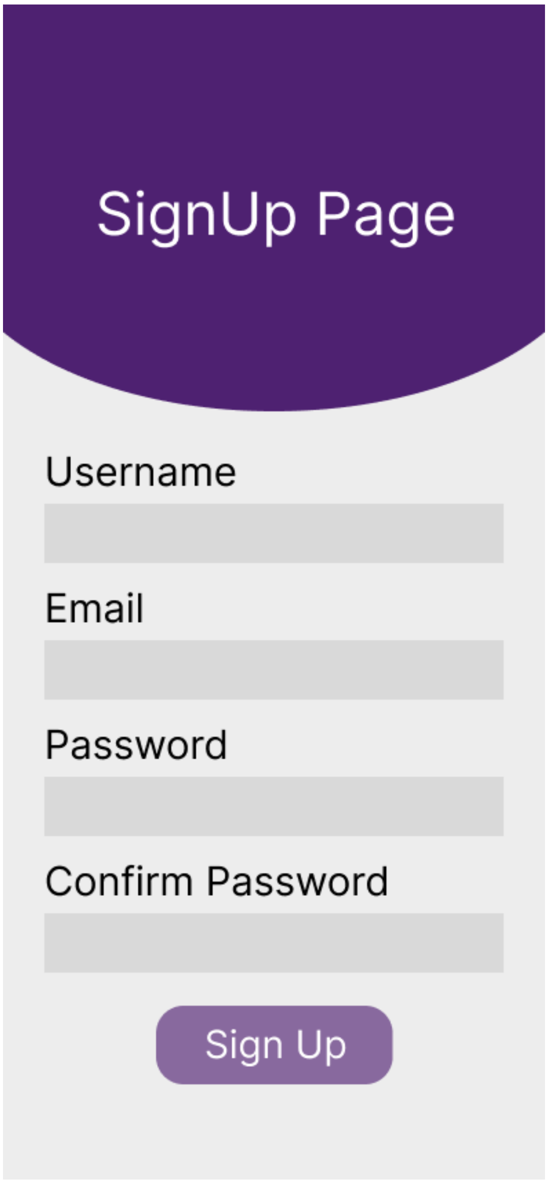
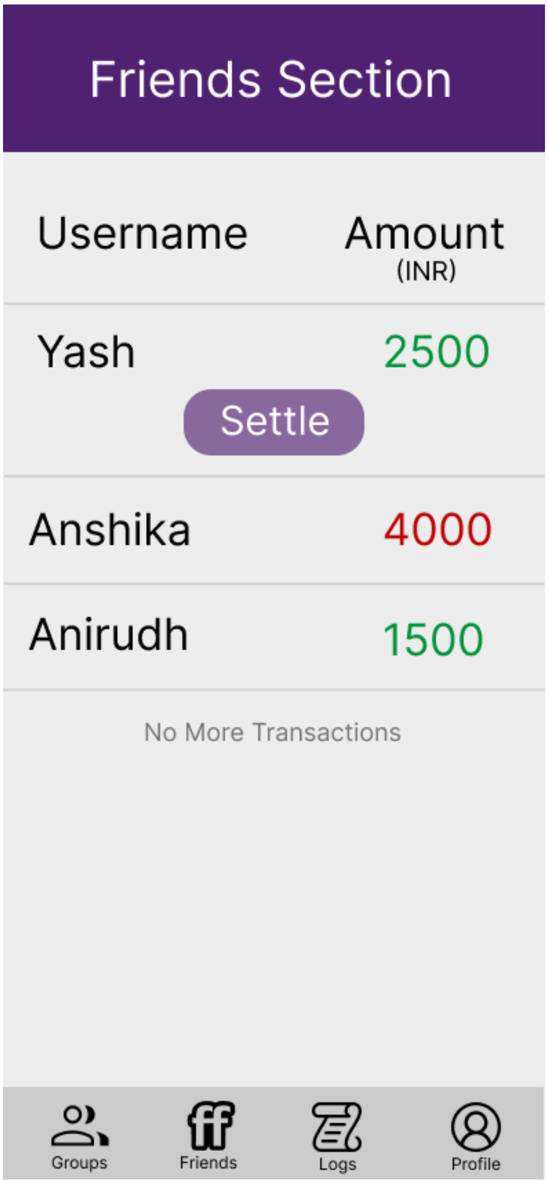


Figure 6: SignUp Page

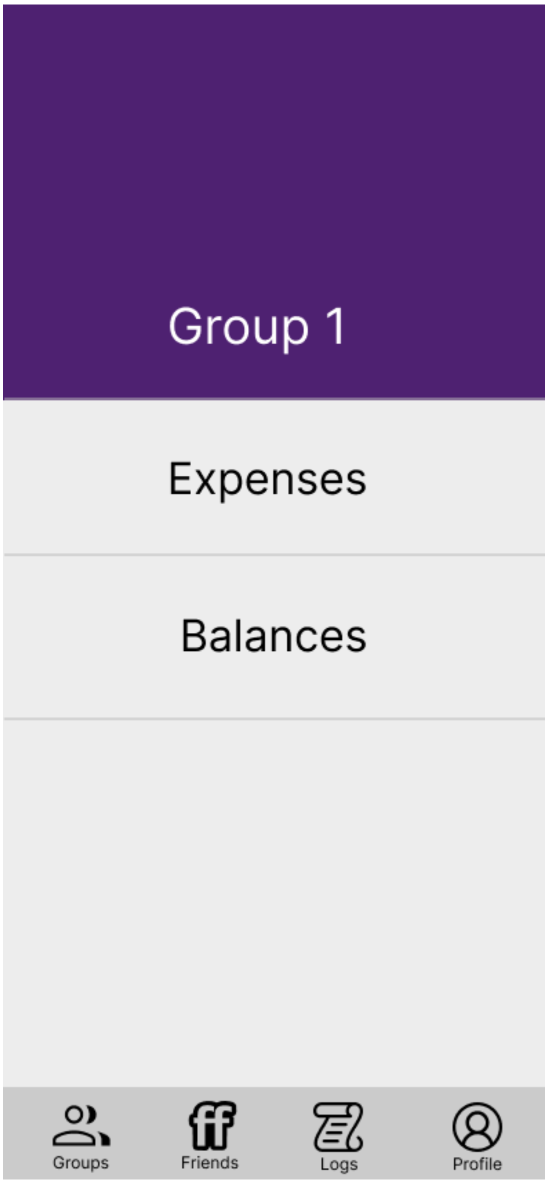


Friends Section

Username	Amount (INR)
Yash	2500
Anshika	4000
Anirudh	1500

No More Transactions

Figure 7: Friends Section

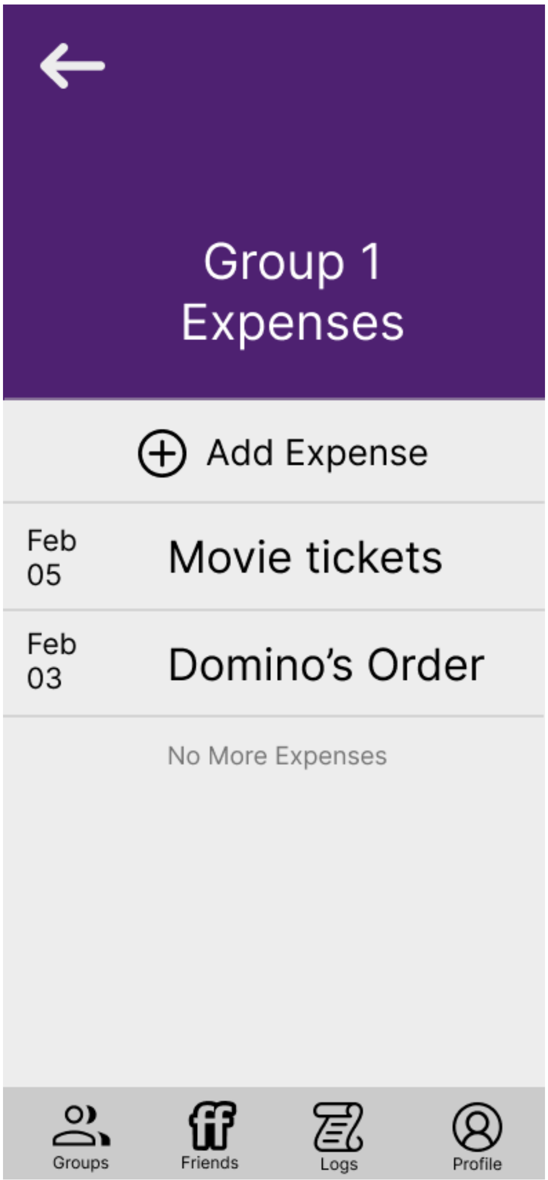


Group 1

Expenses

Balances

Figure 8: Group Interface

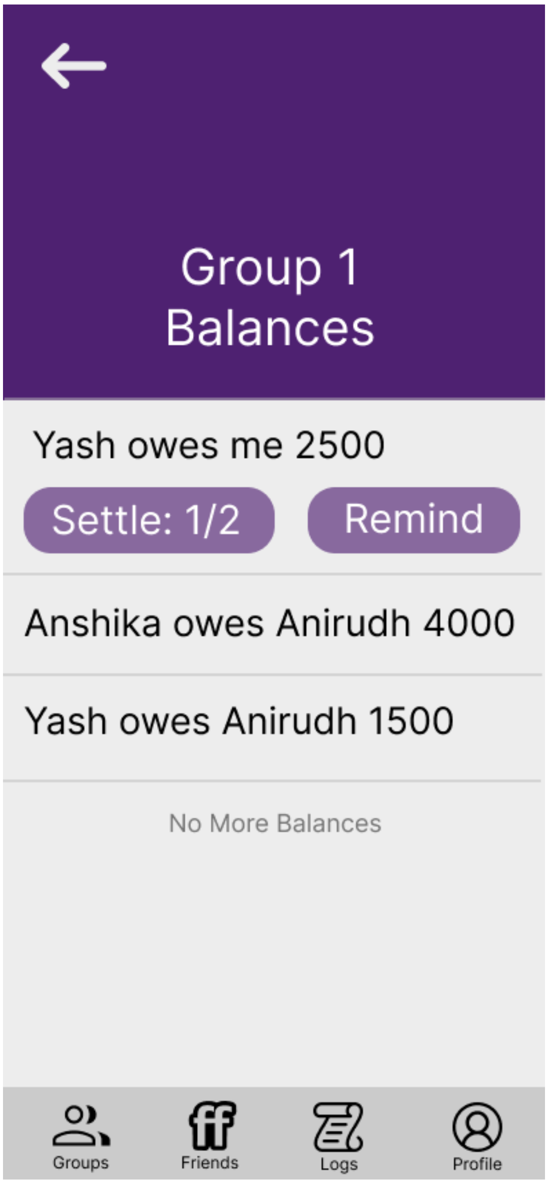


Feb 05 Movie tickets

Feb 03 Domino’s Order

No More Expenses

Figure 9: Group Expenses Page



Yash owes me 2500

Anshika owes Anirudh 4000

Yash owes Anirudh 1500

No More Balances

Figure 10: Group Balances Page