

Flowey

Kangwei Ling (kl3076) Zefeng Liu (zl2715) Kunyan Han (kh2931) Xi Yang (xy2390)

	Server-Side	Client-Side
Deployment Platform	Linux	iOS
Language	Python 3	Swift

Background

“Emmm, where does my money go?”, “How much should I pay you? So we’ve spent 56.25, you paid 21, and Jiang paid 35.25, let’s figure that out”

We find that an intelligent expense tracker is badly needed among various people and small groups of people, especially among us students. This is often the case that someone pays the bill after meals, since many restaurants do not support money splitting. Or when a group of people would like to hang out, go shopping, and even hold parties, they eventually need to split the expenses with friends. However, when a lot of people are involved, the debt relation could be too complicated to figure out in a short time. Meanwhile, people cannot track their everyday expenses under such circumstances, lead to inconvenience in their money management.

Proposal

Flowey plans to provide people with better experience on expense tracking, easy bill splitting and debt management. Complicated bills and multiple debts are reduced to simple money flows between users, allowing them to track expenses more intelligently and clearly, and pay off debts easily.

We’ll use a **Client-Server architecture**. The server is a **RESTful** service provider with access to a **MySQL** database, where stores data of users, transactions, etc. Also, At first iteration, all logic and computations will be done on server.

For the client side application on iOS, Internet access is needed, thereby the client app will be just a collect data and display content. Later, if time permits, offline features with online synchronization may be supported.

User Stories

1. As a user, I want to **register** an account and then use this account to **log in** so that I can use this application.
2. As a user, I want to **add transactions** for my personal bills so that I can **keep track of my personal expenses**. These transactions must be **viewable** in some place like a dashboard in the application.

3. As a user, I want to be able to use a friend panel to **send friend request** to other users and **view debt status** between me and my friends. Other users have the right to reject the friend request and stay unconnected. With friends, I want to be able to add bills that should be **paid by other users or by shared payment**. Bills shared with others need to **be approved** before becoming effective.
4. As a user A, I want to **create/join a group** that also includes user B, C.... I can post bills that is shared among all group member with **equal** or **customized amount**, the bills must be approved by all group member in order to take effect. Different from just sharing bills with friends, groups have **independent expense tracker** that can be viewed by all group member.
5. As a user, I want to have all group bills and personal bills reduced to simplest form of **money flow between two people**. The **number of transfers** needed to clear all debts must be minimized. The due money flows can be cleared by issue a request for clear from either side of the flow.
6. As a user, I want to see the statistics over different time ranges of my expenses tracker or group expenses tracker so that I understand the spending trends.

additional (if time permits)

7. As a user, I want to add tags to some of the expenses so that I can track and analyze the expenses in some specified tags.
8. As a user, I want to have some account management utilities, including changing name and password, setting a avatar and personal profile.
9. As a user, I want to have new expense entries automatically added to my account when I agree that another user had paid for me.

Acceptance Test Planning

1. Register & Login

ID	GIVEN	WHEN	THEN
01	No user account exists with username ACC	User registers a new account with Username ACC and password PWD	New account created in our database
02	Account ACC already exists	User registers a new account with Username ACC and password PWD	Register rejected
03	User has an account ACC	User enters correct username and password	User logs in successfully
	User has an	User tries to log in	User cannot log

04	account ACC	with a wrong password	into the application
----	-------------	-----------------------	----------------------

2. Basic expense tracking (Given user has logged in)

ID	GIVEN	WHEN	THEN
01	There are records of expenses for this user in the database	User select the 'expenses' tab to view personal expenses	The screen will show list of expenses.
02	Currently there are no records of expenses for current user	User select the 'expenses' tab to view personal expenses	The screen show empty list.
03	Server and database are accessible	User clicked the 'add' button to add personal expense with detail input (amount, comment)	New expenses for current user will be added to database
04	Current user has the specified friends in the friend list	User clicked the 'add' button and choose to add expense related to friend (amount, comment, choose friend(s))	Server will accept this expense and send request to friend, wait for response
05	Current user has the specified group in the group list	User clicked the 'add' button and choose to add expense related to group (amount, comment, choose group)	Server will accept this expense and send request to member in group, wait for response

3. Money flow between friends

ID	GIVEN	WHEN	THEN
01	User A and user B are not friends	User A sends a friend request to B	User A waits for B to respond
02	User A and user B are not friends	User B receives a friend request from A, and chooses to	User A and user B become friends

		accept this request	
o3	User A and user B are not friends	User B receives a friend request from A, and chooses to decline this request	User A and user B stay unconnected
o4	User A has friends B, C, D and he / she owes B \$50, owes C \$75	User A splits his / her bill of \$90 and chooses B to request \$30 and C to request \$20	User A waits user B and user C to respond
o5	User A has friends B, C, D and he / she owes B \$50, owes C \$75	User B and C received requests of \$30, \$20 from user A and both accept this request	User A owes user B \$20, owes user C \$55
o6	User A has friends B, C, D and he / she owes B \$50, owes C \$75	User B and C received requests of \$30, \$20 from user A and but only B accepts this request and C does not respond	User A owes user B \$50, owes user C \$75; but user A stages this expenses
o7	User A has friends B, C, D and he / she owes B \$50, owes C \$75	User B and C received requests of \$30, \$20 from user A and but only B accepts this request but C refused	The bill becomes invalid and user A owes user B \$50, owes user C \$75
o8	User A has friends B, C, D	User A checks his / her debt status	User A sees the debt status between A and B, C, D

4. Groups

ID	GIVEN	WHEN	THEN
o1	No group with name G exists in user A's saved groups	User A creates a group G	Group G is created

	Group G exists in user A's saved groups, user A has friends B, C, D, E, F ...	User A invites users B, D, F ... into group G	Invitation to join the group G sent to users B, D, F ...
	User B received an invitation to join group G	User B accepted invitation to join group G	User B joined group G
	Group G exists	User A added a expense and splitted the bill equally or with customized amounts among members in group G	User A sent bill requests with corresponding amounts to other users in G (B, D, F ...) Other users in G (B, D, F ...) received the corresponding request from user A for the splitted bill
	User B received a bill request from A, which resulted from the newly added expense	User B approved the request	A new borrowing transaction turns active in A's account A new owing transaction turns active in B's account
	Group G exists and user A is a member of G	User A checks group G's expense tracker	User A sees all expenses recorded under group G and their statistics
	Group G exists and user A is a member of G	User A checks group G's flows	User A sees the flow between all users in group G which happened inside the group G

5. Simplify Money Flow

ID	GIVEN	WHEN	THEN
01	User A owes User B	Either user looks	B now owes A \$20

	\$10, and B also owes A \$30	up the money flow	
02	User A owes User B \$10, and B also owes A \$30	Either user click the detailed information about the money flow	All the flows will be displayed: User A owes User B \$10; B also owes A \$30;
03	User A owes User B \$10, and B also owes A \$30; After that, User B has paid A \$20 in some way	User B request to clear the flows between B and A	The server receives the request, and it will send a message to A for confirmation
04	User A owes User B \$10, and B also owes A \$30; After that, User B has paid A \$20 in some way, and B has requested to clear the flow information	User A refused to accept the flow clearing request	Do nothing
05	User A owes User B \$10, and B also owes A \$30; After that, User B has paid A \$20 in some way, and B has requested to clear the flow information	User A confirms that the flow information can be cleared	The server receives the request, and clear all the current flows between A and B
06	User A owes User B \$10, and B also owes A \$30; After that, User B has paid A \$20 in some way	User A request to clear the flows between A and B	The server receives the request, and clear all the current flows between A and B

6. Statistics

ID	GIVEN	WHEN	THEN
01	There are records	User clicked to	Screen show bar

	of current user's expenses	view personal expense tracker statistics	chart for expenses
02	There are records of users' expenses in the group	Any user clicked to view group expense tracker statistics	Screen show bar chart for group expense