Product Backlog for WeCheck

Team 3: Dongji Cui, Moon sun Hwang, Mengxue Luo, Xiaohua Shi, Yiyang Zhou

Project Statement

In daily life, situation that one person pays for multiple individuals happens quite frequently. However, calculating individual expense for each person with taxes, tips, etc. could be painful. There is an app called Splitwise that can only split the bill evenly. In other cases, personal finance management, such as recording and organizing one's daily expense could be cumbersome. There are apps doing such tasks, for example, Mint. However, none of those apps are capable of doing both. We now offer a web application that can greatly help to solve those problems and provide data analysis.

Background Information:

College students tend to have many group events, such as a group of students going out to have dinner at a restaurant, watching movies together, or ordering pizza for classmates, etc. Calculating each person's expense at the end of the event is always painful, and sometimes can cause unnecessary disputes. Although currently there are certain apps that do bill splitting for the users, they merely are simple calculators that can do division, and they generally require a fair amount of human effort to input the prices and numbers of persons. We want to design an app that is not just a calculator, instead, it has a social aspect in which event holder can start an event, invite members to the event, or dismiss the event. The bill splitting will take part in that certain event, on the other hand, our app will provide extra option for splitting the bill other than just splitting the bill evenly within the event.

Since we are already building a bill splitter, it would be wise to also have a personal finance management feature added into the app. It is convenient to combine these two features together. In addition, college students are in a crucial stage of their lives where they are preparing to step into the real world. In order to do so, managing personal finance will be an essential skill. With the data analysis and visualization provided by our app, a user can better understand how the money is spent and pick up smart recommendations generated based on past data to save some money.

Requirements

- **Functional Requirements**

Backlog ID	Functional Requirement
Bill Splitter:	
1	As a user, I would like to create an event.
2	As an event creator, I would like to set the type(public or private) of the event
3	As an event creator, I would like to set a category(restaurant, hotel, flight, ets.) of the event.
4	As an event creator, I would like to customize a category of an event.
5	As an event creator, I would like to input the total amount of the group spends.
6	As an event creator, I would like to modify the event's name.
7	As an event creator, I would like to update the status of an event(in process or complete) after the event's status is valid
	As an event creator, I would like to have options to choose how to split a bill(either evenly or separately)
8	As a developer, I would like to check if the event inputs are correct (Such as after everyone input their expenses, the system would check if all inputs are valid and update its corresponding status).
	As a developer, I would like to correctly calculate the split expenses for each member in an event
9	As an event creator, I would like to be able to delete a created event.
10	As an event creator, I would like to modify the member list.
11	As a user, I would like to invite friends to join the event.
12	As a user, I would like to be able to accept an invitation.
13	As a user, I would like to be able to decline an invitation.
14	As a user, I would like see the members in the events.
15	As a user, I would like to input my spends into the event.
17	As a user, I would like to see how much I owe or how much people owe me after the bill has been calculated.
18	As a user, I would like to be able to see the expenditure details of all members in an event.

19	As a user, I would like to receive a notification to remind me to pay to the others.
20	As a user, I would like to be able to send reminder to people who owe me money.
21	As a user, I would like to be able to send notification to the person after I pay off the money I owe.
22	As a user, I would like to be able to confirm that if I received money from people who owe me.
23	As a developer, I would like to be able to update user's account balance (the money I owe others and the amount I am owed)
24	As a user, I would like to view history of all events
25	As a user, I would like to view all the events where I owe others money
26	As a user, I would like to view all the events where people owe me money
Finance Manager:	
27	As a user, I would like to save my personal expenses by category
28	As a user, I would like to have all the general expense categories to choose
29	As a user, I would like to create customized expense category
30	As a user, I would like to modify existing personal expenses
31	As a user, I would like to delete existing personal expenses
32	As a user, I would like to view my personal expenses by date in current month
33	As a user, I would like to view my personal expenses in an ascending/descending order by price in current month
34	As a user, I would like to be able to add notes to my specific expense
35	As a user, I would like to view total amount of money I spent in current period of time(current month)
36	As a user, I would like to be able to view the comparison by percentage of total amount money I spent in current period of time with same period of time in the last month
37	As a user, I would like to be able to view the comparison by percentage of top 3 category expenses in current month with same category expenses within same period of time in the last month
38	As a user, I would like to visualize my personal expense data in form of pie charts in current period of time (in current month)

39	As a user, I would like to view the average daily expenses in current period of time (in current month)
40	As a user, I would like to be able to view the comparison by percentage of average daily expenses in current period of time with in last month.

- **Non-Functional Requirements**

1. As a user, I would like to be able to sign up an account.
2. As a user, I would like to be able to log into my account.
3. As a user, I would like to be able to logout my account.
4. As a user, I would like to reset my password if I forget it.
5. As a user, I would like to be able to update my profile.
6. As a user, I would like to have fast response time.
7. As a user, I would like to have a user friendly application.
8. As a developer, I would like the server to be secure in the way that non-authorized requests are denied.
9. As a developer, I would like the server to be able to handle large amount of users.
10. As a developer, I would like to make our application robust.
11. As a developer, I would like to modularize our code.
12. As a developer, I would like to make our application well maintainable.

Use Case:

Case: create an event

Action	System Response
1. choose ‘create an event’ button	2. event setup page appears
3. type necessary information	
4. click ‘create’ button to confirm the event	5. event setup page disappears

Case: set the type(public or private) of the event

Action:	System Response
1. choose type from drop down menu	2. drop down menu appears
3. click ‘set’ button to confirm the choice	4. drop down menu disappear

Case: set a category

Action	System Response
1. choose ‘category’ button	2. category list appears
3. choose the existing category or type by your own	
4. click ‘add’ button to confirm the category	5. category list disappears

Case: customize a category of an event.

Action:	System Response
1. click “+” button	2. create category page appears
3. type the category name	
4. click “add” button to confirm the category create	5.category page disappears

Case: input the total amount of the group spends. (when category is ‘restaurant’)

Action:	System Response
1. click ‘enter the amount’ button	2. input page appears
3. type total amount of spend	
4. type the tax amount	
5. type the tip amount	
6. click “done” button to confirm the amount	7. input page disappears

Case: modify the event’s name.

Action:	System Response
1. click the event you want to modify	2. event drop down menu appears
3. click “edit” button	4. input box appear
5. delete the original name	
6. type the new name	
7. click “done” button to confirm the change	8.input box disappears

Case: check if the event input is correct (Such as after everyone input their expenses, the app would check if all inputs are valid and update its corresponding status)

Action:	System Response
1. check if the input are valid	2. if invalid, prompt every users to enter again. if valid, the system save the data.

Case: to have options to choose how to split a bill

Action:	System Response
1. Choose “even/separate” from drop down menu	2. Update and save event creator’s choice in database

Case: update the status of an event(in process or complete) Such as after everyone input their expenses, the app would check if all inputs are valid and update its corresponding status

Action:	System Response
1. click the event you want to update	2. event drop down menu appears
3. click “status” button	4. status drop down menu appears
5. choose status (in process or complete)	6. status drop down menu disappears
7. confirm the status	8. create category page disappears

Case: correctly calculate the split bill for each member in an event

Action:	System Response
1. Click on “split” button	2. Calculate the each member’s expenses based on the split way either evenly or separately

Case: delete a created event.

Action:	System Response
1. click the event you want to delete	2. event drop down menu appears
3. click ‘delete’ button	4. delete message box appears
5. click ‘confirm delete’ button to confirm	5. confirm delete message box disappears
	6. event drop down menu disappears

Case: modify the member list

Action:	System Response
1. click “member list” button	2. member lists page appears
3. modify the member list by using remove or add button	
4. confirm add or remove	4. member lists page disappears

Case: invite friends to join the event

Action:	System Response
1. click “Invite Friends” button	2. Invite Friends page appears
3. input friend’s email address	
4. confirm email address	5. invitation message is sent to the friend with that address.

Case: accept an invitation from friend

Action:	System Response
1. click “accept invitation link” in email	2. user is confirmed and is put inviter’s event list.
	3. event page pops out to user

Case: to decline an invitation

Action:	System Response
1. click “decline invitation link” in email	2. inviter is notified the declination from his/her friend.

Case: see the members in the events.

Action:	System Response
1. click “member” button	2. members lists page appears
3. user clicks exits button	4. members lists page disappears

Case: input my spends into the event

Action:	System Response
1. click “spends” button	2. spends pop box appears
3. user inputs his/her expenses related to the event	
4. user confirms his/her amounts	5. spends pop box disappears

Case: to confirm that the subtotal amount matches after all the members input their spends

Action:	System Response
1. click “spends match” button	2. it runs on backend with matching algorithm.if expense is matched with the bill, a happy face emoji with “ok” button dialog pops out to notify the user. Or if matching fails, a sad face emoji sends to user with “ok” button and “rematching” button.
3. user clicks “ok” button or “rematching” button	4. if user clicks “ok”, emoji dialog disappears. Or if user clicks “rematching”, input spends request dialog will send to each user.

Case: see the expenditure details of all members in an event.

Action:	System Response
1. Click “Expenditure” button	2. Expenditure details page appears
3. Choose an event to view expenditure details	4. Expenditure details of members appear

Case: receive a notification to remind me to pay to the others.

Action:	System Response
1. Click “Notification” button	2. Notification page appears
3. Choose “Received Notification” button	4. Full list of all received notifications
5. Select specific notification	6. Notification page with details appears

Case: send reminder to people who owe me money.

Action:	System Response
1. Click “Notification” button	2. Notification page appears
3. Choose “Send Reminder” button	4. List of all saved people appears
5. Select a specific person to send a reminder	6. List of events with the selected person
7. Choose the specific event to send the reminder	8. Confirm message box appears
9. Click “Send Reminder” button	10. Send reminder to the person

Case: confirm if I received money from people who owe me.

Action:	System Response
1. Click “All Events” button	2. Page of all events appears
3. Choose a specific event	4. List of all people for the event appears
5. Click “Confirm Reception”	6. Check mark is added next to the name

Case: update user’s account balance (the money I owe others and the amount I am owed)

Action:	System Response
1. Click “My Account Balance” button	2. Account balance page appears
3. Click “Amount I owe” / “am owned” option menu	4. All events, with who I owe to appears
5. Click “Update balance” menu	6. Prompt user to type in updated balance
7. Type in the updated balance	8. Confirm message box appears
9. Click “Confirm” button	10. User’s account balance is updated

Case: view history of all events

Action:	System Response
1. Click “All Events” button	2. Page with a list of all events appears
3. Choose a specific event	4. Event details appears

Case: view all the events where I owe others money

Action:	System Response
1. Click “All Events” button	2. Page with a list of all events appears
3. Click “Uncomplete Events” button	4. Change to page contains user’s incomplete events.
5. Click “Owe Others” button	6. Change to page contains events a user have to pay

Case: view all the event in which people owe me money

Action:	System Response
1. Click “All Events” button	2. Change to page contains all events related to the user.
3. Click “Uncomplete Events” button	4. Change to page contains user’s incomplete events.
5. Click “Owe Me” button	6. Change to page contains event from which user can get money

Case: save personal expense by category

Action:	System Response
1.Input expense amount	2. Check for validity of user input
	3. If input is valid, prompt user to select category
	4. If input is invalid, prompt user to re-type amount
5. Select one category	6. Confirmation window pops
7. Confirm the operation	
	8. Store amount and its category into database

Case: choose category in which a certain expense will be saved

Action:	System response:
1.Input amount of expense and click “Save”	2. Display default categories

Case: customize category when saving an expense

Action:	System response:
1. Input amount of expense and click “Save”	2. Display categories and “Customize” button
3. Click “Customize” button	4. Prompt user to input category name
5. Input category name	6. Check input validity, ask user to confirm
7. Confirm operation	8. Store data into database

Case: modify existing personal expenses

Action:	System Response:
1. Click “My expenses” button	2. Display user’s expenses
3. Select the expense to modify	4. Display detail information of that expense
5. Modify	
6. Click “Save modification” button	7. Update data in database

Case: delete a personal expenses

Action:	System Response:
1. Click “My Expenses” button	2. Display user’s expenses
3. Select expense to delete	4. Prompt user to confirm action
5. Confirm action	6. Remove data from database

Case: view personal expense by date

Action:	System Response
1. Click “View expense by date” button	2. Display expense by date

Case: view personal expenses in ascending/descending order by price

Action:	System Response:
1. Click “view” option button	2. Collapse the option list
3. Choose ascending/descending button	4. Display personal expenses in expected order

Case: add notes to a expense

Action:	System Response:
1. Input personal notes in an input box when creating an expense record	2. Enable user to take notes in the input box
3. Click “save” button to save the notes	4. Save user’s notes in database

Case: view total amount of money spent in current period of time(current month)

Action:	System Response:
1. Add/update an expense record	2. Update the total amount of money spent and display on the top of the page

Case: view comparison by percentage of total amount of money spent

Action:	System Response:
1. Add/update an expense record	2. Update the total amount of money spent in current period of time(current month)
	3. Get the total amount of money from the

same period of time in last month
4. Compare these two number and do the calculation
5. Display the calculated result on the page

Case: view comparison by percentage of top 3 categories

Action:
1. Add/update an expense record

System Response:
2. Update total amount of money spent in the corresponding category
3. Sort out the top 3 category with most expenses in current month
4. Compare the amount of found top 3 categories with corresponding category in last month by percentage
5. Display the calculated result on the page

Case: visualize personal expenses by category in form of pie chart

Action:
1. Add/update an expense record

System Response:
2. Update total amount in corresponding category
3. Generate the data visualization in pie chart
4. Display the visualization on the page

Case: view average daily expenses in current period of time(in current month)

Action:
1. Add/update an expense record

System Response:
2. Get total amount of money spent in current period of time from database
3. Divide the total amount by number of days passed in current month
4. Display the calculated result on the page

Case: view comparison by percentage of average daily expense

Action:
1. Add/update an expense record

System Response:
2. Update total amount of money spent in current period of time
3. Calculate the average daily expenses in current period of time(in current month)
4. Calculate the average daily expenses in same period of time in last month
5. Compare the two calculated results by percentage
6. Display the compared result on the page