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	Hours	Status
Bill Splitter:			
1	As a user, I would like to create an event.	5	Finished
2	As an event creator, I would like to set the type(public or private) of the event	2	Finished
3	As an event creator, I would like to set a category(restaurant, hotel, flight, ets.) of the event.	2	Finished
4	As an event creator, I would like to customize a category of an event.	3	
5	As an event creator, I would like to input the total amount of the group spends.	2	Finished
6	As an event creator, I would like to modify the event's name.	2	Finished
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	3	Planned for sprint 2
8	As an event creator, I would like to have options to choose how to split a bill(either evenly or separately)	2	Finished
9	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).	4	Finished
10	As a developer, I would like to correctly calculate the split expenses for each member in an event	4	Finished
11	As an event creator, I would like to be able to delete a created event.	3	Finished
12	As an event creator, I would like to modify the member list.	3	Finished

13	As a user, I would like to invite friends to join the event.	4	Finished
14	As a user, I would like to be able to accept an invitation.	3	Planned for sprint 2
15	As a user, I would like to be able to decline an invitation.	3	Planned for sprint 2
16	As a user, I would like see the members in the events.	3	Finished
17	As a user, I would like to input my spends into the event.	4	Finished
18	As a user, I would like to see how much I owe or how much people owe me after the bill has been calculated.	5	Finished
19	As a user, I would like to be able to see the expenditure details of all members in an event.	4	Finished
20	As a user, I would like to receive a notification to remind me to pay to the others.	4	
21	As a user, I would like to be able to send reminder to people who owe me money.	4	
22	As a user, I would like to be able to send notification to the person after I pay off the money I owe.	4	
23	As a user, I would like to be able to confirm that if I received money from people who owe me.	3	
24	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)	4	Planned for sprint 2
25	As a user, I would like to view history of all events	4	Planned for sprint 2
26	As a user, I would like to view all the events where I owe others money	2	
27	As a user, I would like to view all the events where people owe me money	2	
Finance Manager:			
28	As a user, I would like to save my personal expenses by category	3	*Planned for sprint 2*
29	As a user, I would like to have all the general expense categories to choose	3	

30	As a user, I would like to create customized expense category	3	*Planned for sprint 2*
31	As a user, I would like to modify existing personal expenses	3	*Planned for sprint 2*
32	As a user, I would like to delete existing personal expenses	2	*Planned for sprint 2*
33	As a user, I would like to view my personal expenses by date in current month	4	
34	As a user, I would like to view my personal expenses in an ascending/descending order by price in current month	5	
35	As a user, I would like to be able to add notes to my specific expense	3	*Planned for sprint 2*
36	As a user, I would like to view total amount of money I spent in current period of time(current month)	3	*Planned for sprint 2*
37	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	4	
38	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	6	
39	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)	5	
40	As a user, I would like to view the average daily expenses in current period of time (in current month)	3	*Planned for sprint 2*
41	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.	4	
App Administrator:			
42	As a user, I would like to be able to sign up an account	8	finished
43	As a user, I would like to be able to log into my account.	6	finished
44	As a user, I would like to be able to logout my account.	4	finished

45	As a user, I would like to reset my password if I forget it.	6	partially finished, the rest will moved onto sprint 2
46	As a user, I would like to be able to update my profile.	5	Planned for sprint 2

- **Non-Functional Requirements**

1. As a user, I would like to have fast response time.
2. As a user, I would like to have a user friendly application.
3. As a developer, I would like the server to be secure in the way that non-authorized requests are denied.
4. As a developer, I would like the server to be able to handle large amount of users.
5. As a developer, I would like to make our application robust.
6. As a developer, I would like to modularize our code.
7. 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 event name, set event type and category	
4. Click ‘create’ button to confirm the event	5. Event setup page disappears, the system add the event information into database

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

Action:	System Response
1. Click type drop-down menu	2. Drop-down menu appears, there are “public” and “private” in the menu.
3. Choose a type	3. Selected type shows. (database will be updated after the event is created)

Case: set a category

Action	System Response
1. Click category drop-down menu	2. Drop-down menu appears, there are “restaurant”, “flight” and “hotel” in the menu
3. Choose a category	4. Selected category shows. (database will be updated after the event is created)

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 created	5. Category page disappears. Return to the creat event page. The new category will show in the category menu. The system will add the new category to the users in database. (database will be updated after the event is created)

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. The “edit” and “view” buttons appear
3. click “edit” button	4. Input box appear
5. delete the original name	
6. type the new name	

Action	System Response
7. click “done” button to confirm the change	8. Input box disappears. The new event’s name will shows on the page. The system will update the event’s name in database.

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. input the amount of money(>$0) that user owe on “input” button	
	2. if invalid (number doesn’t match total amount) . if valid, back end runs split function and amount of money that people owe to each other will show on summary.

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. event status changes correspondingly

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(based on each member’s expense)

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 box appears
5. click ‘confirm delete’ button to confirm	5. confirm delete message box disappears
	6. event drop down menu disappears
	7. Event gets removed from database

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. If adding, input box for email address appears
5. confirm add or remove — 6. member lists gets updated

Case: invite friends(registered users) 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. Notification of decline sent to inviter
3. Check the notification in notification center

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 spending 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. Individual split expense of each member 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. As an event creator, click “send reminder” button besides a member in the member list in an event	2. The corresponding member should be able to receive the reminder

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. After an event split	2. User's account balance number updated correctly by the system

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 up
7. Click "Save" button	
	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 an expense(any day) 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:	System Response:
1. Add/update an expense record	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:	System Response:
1. Add/update an expense record	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:	System Response:
1. Add/update an expense record	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:	System Response:
1. Add/update an expense record	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

Case: As a user, I would like to be able to sign up an account

Action:	System Response:
1. Click "sign up" button	2. Sign up page appears
3. Input the email address and the password	
4. Input name	
5. Click "submit" button	6. If the email address is valid, sign up page disappear and return to the sign-in page. If the email address is not valid, then return to the sign up page with a warning text next to the email address field.

Case: As a user, I would like to be able to log into my account.

Action:	System Response:
1. Input the email address and password (our main page include the sign in window)	
2. Click “sign-in” button	3. If the email address and the password match, the main page appears. If the email address and the password does not match, the red warning text “incorrect email address or password” shows.

Case: As a user, I would like to reset my password if I forget it

Action:	System Response:
1. Click “forget password” button	2. Forget password page appears
3. Input the email address and click “send verification code”	4. The system send temporary password to the email address.
5. Enter the verification code and click “verify” button	6. If the temporary password is correct, then the system new password input box appears. If the the temporary password is incorrect, then the warning text “incorrect verification code”.
7. Enter new password and retype new password. Click “done” button	8. If the passwords entered are the same, reset password successfully. The system update the database. If the password entered are different, warning text “password unmatched” appears.

Case: As a user, I would like to be able to logout my account

Action:	System Response:
1. Click “log out” button	2. The system close session. The sign-in page appears.

Case: As a user, I would like to be able to update my profile

Action:	System Response:
1. Click “profile” tab	2. The profile page shows up
2. Click “edit” button	4. Edit profile page shows up
5. Enter the information you want to modify and click “done” button	6. Return to the profile page, the profile page displays new information. The system update the new information on database.