Team 3

College Capital

Sprint 2 Retrospective

Muhammad Bokhari, Jeremy Chen, Justin Chen, Charlie Newell, Matthew Story, Jethro Zhou

What Went Well

In general, the workflow for this sprint went incredibly smoothly compared to the last sprint. As a whole, it felt like every member of our group had a much more solid understanding of our technologies which allowed us to build off of the groundwork that we had previously laid. As a result, we got to develop some of our more intricate user stories, giving our app features such as transactions, visualizations, and a support contact form.

User Story #9

As a user, I would like to be able to visualize my current financial status.

#	Description	Estimated Time	Owner
1	**Create UI for visualization page**	**4 Hours**	**Jethro**
2	**Create algorithm to generate visualization from data**	**6 Hours**	**Jethro**
3	**Create algorithm to retrieve data for visualization**	**6 Hours**	**Jethro**
4	**Create tests to validate functionality**	**1 Hour**	**Justin**

Completed

Our visualizations page includes a pie graph and several line graphs that demonstrate fiscally current expenditures by day, month, and year. These graphs are based on the data input into our database on the transactions page, updating in real time immediately after the user inputs their data. Further, the users have the option to export the individual visualizations in a variety of formats.

User Story #11

As a user I would like to be able to export my visualizations

#	Description	Estimated Time	Owner
1	Create algorithm to export visualizations in a variety of formats	3 Hours	Jethro
2	Create Ui for exporting visualizations	2 Hour	Jethro
3	Create algorithm to retrieve visualization for export	2 Hours	Jethro
4	Create tests to validate functionality	1 Hour	Charlie

Completed

On every visualization on the visualizations page, there is a dropdown menu that allows the user to select their desired format, PNG, JPG, PDF, SVG, CSV, or XLSX and once a choice is made a popup is generated where the user can save their exported visualization as the chosen file type and specify the name they want to save the file with and the location they wish to save the file in.

User Story #13

As a user, I would like to be able to update my transaction history.

#	Description	Estimated Time	Owner
1	Create UI for create/update transaction	8 Hours (each)	Charlie, Muhammad
2	Create algorithm to retrieve original state of transactions	4 Hours	Jeremy
3	Create algorithm to populate page with original state of transactions	4 Hours	Jeremy
4	Create algorithm to update transaction in database	4 Hours (each)	Jeremy, Muhammad
5	Create tests to validate functionality	1.5 Hours	Charlie

Completed

Our transactions page includes a table displaying all of the previously input expenditures, as well as two buttons for creating an additional transaction and exporting what is already in the table. Creating a new transaction redirects the user to another page with a form to create a new expenditure, including required fields for all relevant data.

User Story #14
As a user, I would like to be able to view my transaction history.

#	Description	Estimated Time	Owner
1	Create UI for view transaction	8 Hours	Charlie
2	Create algorithm to retrieve transaction history	6 Hours	Jeremy
3	Create tests to validate functionality	1 Hour	Charlie, Muhammad

Completed
The transaction history is displayed immediately upon navigating to the transactions page. It is presented in a table that allows the user to highlight a specific transaction for ease of use. Further, information such as date, vendor, and amount is shown for a complete expenditure understanding.

User Story #15

As a user, I would like to be able to export my transaction history.

#	Description	Estimated Time	Owner
1	Create UI for exporting transaction	7 Hours	Charlie
2	Create algorithm to retrieve transaction to export	6 Hours	Jeremy
3	Create algorithm to export transactions	6 Hours (each)	Jeremy, Muhammad
4	Create tests to validate functionality	2 Hour	Charlie

Completed

On the transactions page, below the transactions table, there is an export button that allows the user to download the table in both JSON and CSV formats. These formats allow for easy data storage and manipulation in financial tools such as Excel.

User Story #16

As a user, I would like to be able to categorize my expenses.

#	Description	Estimated Time	Owner
1	Create UI for specifying expense category	6 Hours	Justin
2	Create algorithm to store expense category in database	4 Hours	Matthew
3	Create UI to filter expense by category	6 Hours	Justin
4	Create algorithm to apply filter	6 Hours	Matthew
5	Create tests to validate functionality	1 Hour	Charlie

Completed

When creating a new transaction, users are presented with a variety of common categories such as dining, travel, tuition, grocery, bar & coffee shop, as well as general fees. These categories will help us in later user stories, but they also provide additional information to the user to help manage their finances.

User Story #17

As a user, I would like to be able to specify categories when updating my funds.

#	Description	Estimated Time	Owner
1	Create UI for specifying fund category	2 Hours	Jeremy
2	Create algorithm to store updated category in database	2 Hours	Jeremy
5	Create tests to validate functionality	1 Hour	Jeremy

Completed

When creating a new fund, users are able to choose from a variety of categories for the source such as PayPal, Dining Dollars, Boiler Express, Financial Aid, and bank. This allows our users greater control and understanding of their finances, especially when relating it to our expenditures module.

User Story #23

As a user I would like to be able to see a snapshot of my daily usage.

#	Description	Estimated Time	Owner
1	Create algorithm to parse current months data from database into day by day format	7 Hours	Muhammed
2	Create algorithm to visualize data	2 Hour	Jethro
3	Create algorithm to retrieve data for visualization	2 Hours	Muhammed
4	Create tests to validate functionality	0.5 Hours	Charlie

Completed

When the visualization page is opened, a line graph (with an x axis that measures time in days of the current month and a y axis that measures spending in dollars) is generated from the current user’s transaction data retrieved from the database. Further, the users have the option to export the individual visualization in a variety of formats.

User Story #24

As a user I would like to be able to see a snapshot of my weekly usage.

#	Description	Estimated Time	Owner
1	Create algorithm to parse current months data from database into week by week format	7 Hours	Muhammed
2	Create algorithm to visualize data	2 Hour	Jethro
3	Create algorithm to retrieve data for visualization	2 Hours	Muhammed
4	Create tests to validate functionality	0.5 Hours	Charlie

Completed

When the visualization page is opened, a line graph (with an x axis that measures time in weeks of the current month and a y axis that measures spending in dollars) is generated from the current user's transaction data retrieved from the database. Further, the users have the option to export the individual visualization in a variety of formats.

User Story #25

As a user I would like to be able to see a snapshot of my monthly usage.

#	Description	Estimated Time	Owner
1	Create algorithm to parse current years data from database into month by month format	7 Hours	Muhammed
2	Create algorithm to visualize data	2 Hour	Jethro
3	Create algorithm to retrieve data for visualization	2 Hours	Muhammed
4	Create tests to validate functionality	0.5 Hours	Charlie

Completed

When the visualization page is opened, a line graph (with an x axis that measures time in months of the current year and a y axis that measures spending in dollars) is generated from the current users transaction data retrieved from the database. Further, the users have the option to export the individual visualization in a variety of formats.

User Story #32

As a user, I would like to be able to message support 24/7 when I need help with the app.

#	Description	Estimated Time	Owner
1	Create UI for Support Page	7 Hours	Muhammad
2	Create UI for messaging module from User to Support	6 Hours	Jethro
3	Create algorithm for delivering message from User to Support	5 Hours	Jethro
4	Add encryption for messages	5 Hours	Jethro
5	Create tests to validate functionality	1 Hour	Charlie

Completed

On the support page, users are presented with a simple input that allows the user to describe their issue and then message support. This stores the message in the database, and then sends the support account email a message with the user's new "ticket". Further, their currently open ticket appears on the user's page, with the option to close it should they resolve their issue on their own.

User Story #33

As an administrator, I would like to be able to respond to support messages 24/7.

#	Description	Estimated Time	Owner
1	Create UI to view messages	5 Hours	Justin
2	Create UI to manage messages (mark as completed and delete)	4 Hours	Justin
4	Create user roles in database	4 Hours	Muhammad
4	Create algorithm to retrieve messages	4 Hours	Matthew
5	Create tests to validate functionality	1 Hour	Charlie

Completed

By implementing the email functionality of support messages, both users and administrators can respond to support tickets on their own time, 24 hours a day, 7 days a week. This allows for ease of use and communication between both the user and support team.

User Story #34

As an administrator, I would like to be able to respond to support messages when the user is offline.

#	Description	Estimated Time	Owner
1	Add implementation for storing messages into database	7 Hours	Matthew
2	Add security for global messaging module	1 Hour	Matthew
3	Add functionality for global messages from Support to all users	5 Hours	Matthew
4	Create tests to validate functionality	1 Hour	Charlie

Completed

In addition to the email functionality, messages are stored separately in our database, allowing for viewing offline on a user by user basis. Furthermore, regarding global messages, our users can see notifications sent by support to all users.

What Did Not Go Well

Compared to the last sprint, our communication on the technical aspects of our project has gotten worse. While we were able to complete all of our user stories and everyone contributed fairly, the implementations were not what all members expected, leading to some compatibility and functionality issues. Further, since transitioning to distance learning, we’ve had some schedule issues making working together difficult at times.

Ways We Can Improve

Due to current affairs as well as challenges we experienced over the duration of Sprint 2, over the course of the next sprint, it will be incredibly important for us to maintain high levels of communication using the technology we have available to us. We need to incorporate things such as screen sharing, video chat, and frequent scheduled meetings using software such as Discord in order to ensure each member has a technical understanding of our implementations. As with Sprint 1 and 2, Slack is our first and primary line of communication, so it will be important that each of us are readily available and responsive on that application. These measures will not only address the facets of this sprint that did not go well, but will allow us to develop a more robust app.

In addition to ensuring that we are all communicating frequently, openly, and clearly, we need to work on our overall workload pacing. We did well with fixing workload distribution from last sprint, but now we need to focus on spreading our work out evenly over the duration of the upcoming sprint - it seems we rely too heavily on “crunch time” in the last week or so leading up to the review. This causes a small degree of chaos, which does not help with our communication. Therefore, by pacing ourselves, we will not only improve our technical understandings of the implementation of our user stories, but we will also be able to achieve our goals with much less stress and much more correctness.