AllinWallet

All-in-one Personal Finance App

Team 26 Sprint 1 Retrospective

Team Members

Junyi Zhang, Ray Chen, Rohith Sridharan, Shaan Hoda, Yufei Xu

What went well?

In this sprint, We are able to implement 8 out of 10 user stories completely. Also in this sprint, we have module testing for each user story so we can easily test each user story. In addition, we divide all the user story to separate module which can allow us to change each module individually without affect other parts in the future.

User Story #1

As a user, I would like to register for an AllinWallet account so that I can synchronize my profile and data across devices.

#	Description	Estimated Time	Owner
1	Set up back end server	3 Hrs	Yufei
2	Set up database	3 Hrs	Yufei
3	Build GUI for account registration	4 Hrs	Ray
4	Implement functionality to add a new account in the back end	4 Hrs	Shaan
5	Implement ability to create an account in the front end	4 Hrs	Ray
6	Test account registration functionalities	1 Hr	Rohith

Completed: We set up the project on the Firebase console (database). We have been working with Android Studio as the IDE to write code for the application primarily using Java and XML. Once the accounts are created, they are stored in the database, which are also linked to user data. It appears that registering an account is working correctly.

User Story #2

As a user, I would like to log in with a username and password.

#	Description	Estimated Time	Owner
1	Build GUI for login	4 Hrs	Junyi
2	Implement functionality to authenticate users in the back end	4 Hrs	Shaan
3	Implement ability to log in in the front end	4 Hrs	Junyi
4	Test Login functionalities	1 Hr	Rohith

Completed: Once the app is started, the user is able to login into their account with their credentials. The functionality seems to be working fine in terms of logging into an account, and the GUI for the login page is up and running. Perhaps during Sprint 3, we make some cosmetic improvements to it.

User Story #3

As a user, I would like to log out of the app so that I can prevent other people from accessing my profile.

#	Description	Estimated Time	Owner
1	Build GUI for start screen/landing page	4 Hrs	Rohith
2	Implement functionality to change user status in the back end	4 Hrs	Yufei
3	Implement ability to log out in the front end	4 Hrs	Rohith
4	Test Logout functionalities	1 Hr	Ray

Completed: The basic functionalities of the start screen/landing page is implemented and working properly. We have set up the Firebase Auth API to handle user status changes in the back end, and it is now functional. The logout button is located in the user profile page, and it logs out users correctly both in the front and back end.

User Story #4

As a user, I would like to add my email to my AllinWallet account so that I can reset my password through email.

#	Description	Estimated Time	Owner
1	Set up mailer to send emails to users	3 Hrs	Yufei
2	Add button to reset password	3 Hrs	Junyi
3	Implement functionality to update password in the back end	3 Hrs	Shaan
4	Implement ability to reset password in the front end	4 Hrs	Ray
5	Test password reset functionalities	1 Hr	Junyi

Completed: The front end functionalities of the password reset page have been implemented and are working properly. A user can enter this page from the login page and make a password reset request. We have set up the Firebase Auth API to handle most of the back end functionalities of this user story. Once the back end receives a password reset request, it will send out an email with a secure hyperlink for the user to reset their password.

User Story #5

As a user, I would like to enter the amount of money I have just spent and create a purchase in my account profile.

#	Description	Estimated Time	Owner
1	Build GUI for purchase creation	4 Hrs	Ray
2	Implement functionality to modify user data (add purchases) in the back end	3 Hrs	Shaan
3	Implement ability to add purchases in the front end	4 Hrs	Ray
4	Implement functionality to synchronize user data after adding a purchase	3 Hrs	Yufei
5	Test purchase creation functionalities	1 Hr	Junyi

Completed: When the user in the mainpage, there is a button. If the user click it. The user would go to the add purchase page. The GUI would ask the user to enter the amount of purchase and choose a category for the purchase. The user has the option to change the default title. After the user add the purchase, the newly added purchase would be displayed in the mainpage. In the following sprint, we need to add more functionalities in the add purchase page, like location based feature, etc and make GUI look better.

User Story #6

As a user, I would like to add a title to my purchases.

#	Description	Estimated Time	Owner
3	Implement ability to add a title to existing purchases in the front end	3 Hrs	Ray
4	Implement functionality to synchronize user data after modifying a purchase	2 Hrs	Yufei
5	Test purchase modification functionalities	1 Hr	Junyi

Completed: When the user in the add purchase page, the user can add the title to the purchase that the user created and the purchase with title can be successfully displayed in the mainpage. In the following sprints, we are going to build the GUI to editing the purchase and implement functionality to modify the existing purchase.

User Story #7

As a user, I would like to add categories to my newly created purchases.

#	Description	Estimated Time	Owner
1	Add to the GUI for editing existing purchases (adding visibility for categories)	3 Hrs	Ray
2	Implement functionality to modify user data (edit existing purchases) in the back end	3 Hrs	Shaan
3	Implement ability to add a category to existing purchases in the front end	2 Hrs	Ray
4	Implement functionality to synchronize user data after modifying a purchase	2 Hrs	Yufei
5	Test purchase modification functionalities	1 Hr	Junyi

Completed: Once you open the add purchase page there is a button to pop up some selection for the user to choose categories. And after choosing categories and click submit it will send the categories to database in the user collection uid document purchase collection. In the following sprint we might add gui for the pop up page to make more selections for the categories and also we want the program to remember the user choice and make the most recent use categories at first. One more functionality for the categories is that the user can create some categories by themselves that we do not have.

User Story #8

As a user, I would like to create a weekly, monthly, or annual budget.

#	Description	Estimated Time	Owner
3	Implement ability to produce a budget in the front end	4 Hrs	Rohith
4	Test the functionalities of the budgets	2 Hr	Ray

Completed: We create a module that can let the user enter their budget and the budget will store in database and also we show the spend money/budget in our main page.

User Story #9

As a user, I would like to see my weekly, monthly, or annual spend analysis report.

#	Description	Estimated Time	Owner
1	Build GUI for generating the analysis report	4 Hrs	Shaan
2	Implement functionality for getting purchase list in the back end	3 Hrs	Shaan
3	Implement ability for dealing the purchase data in the front end	3 Hrs	Ray
4	Implement ability to generate different type of report in the front end	3 Hrs	Ray
5	Test generating analysis report functionalities	1 Hr	Junyi

Completed: Created an analysis report which would allow the user to track users spending costs by weekly, monthly, yearly reports. So now once the user sets their budget, we can now create a report to show the cutoffs of their budgets for their corresponding choice of interest.

User Story #10

As a user, I would like the ability to know how many people are currently using this app.

#	Description	Estimated Time	Owner
1	Build GUI to see the number of active users of the application	3 Hrs	Junyi
2	Implement functionality for getting the number of active users of the application in the back end	3 Hrs	Shaan
3	Implement ability to generate the number of active users of the application in the front end	3 Hrs	Junyi
4	Test active users functionalities	1 Hr	Ray

Completed: Now when users want to see how many others are using this application they can conveniently check and it. Once they are in the main page, and choose to see the active users, it will now show how many active users the app currently has.

What did not go well?

We partially implemented the following 2 user stories. More work has to be done to these user stories. We also have to improve the GUI of the app.

User Story #6

As a user, I would like to add a title to my purchases.

#	Description	Estimated Time	Owner
1	Build the GUI to edit existing purchases.	3 Hrs	Ray
2	Implement functionality to modify user data (edit existing purchases) in the back end	3 Hrs	Shaan

Not Completed: When the user is in the add purchase page, the user can add the title to the purchase that the user created. The purchase with the respective title will be displayed in the mainpage, however, we are not able to edit the existing purchases. Due to the limited time, we did not implement the functionality to modify user data. In the following sprint, we will complete this user story. The purchase report can only show the monthly and annually report, but the weekly report is not finish yet.

User Story #8

As a user, I would like to create a weekly, monthly, or annual budget.

#	Description	Estimated Time	Owner
1	Build a GUI for the weekly/monthly/annual budget	4 Hrs	Rohith
2	Implement functionality to produce a weekly, monthly, and/or annual budget in the back end	4 Hrs	Shaan

Not Completed: We can only set a budget but don’t have the option to choose a weekly, monthly, or annually budget. We will finish this in the following sprint.

How should you improve?

Communication: We had the issue that because we divide our tasks, so most of time, we work individually. It would be hard for someone to catch up with others and understand what's going on right now. Also, if someone modify some data and make the project cannot run as perspective, it would waste a lot of time to debug for it. So in the following sprints, we need to consider work as 2 or 3 members in each group or have more group meetings than before. After dividing the work, members would work on tasks individually, at certain instances we had issues with multiple people working on the same task.

Testing: Although we accounted for module testing for each part, we did not manage our time properly therefore the testing does not fully test each module. There might be some unexpected bugs in the future development. In the next sprint we can spend more time on testing, or we can find some users to use our program to find bugs in our project.