Team 3

College Capital

Sprint 1 Retrospective

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

What Went Well

Throughout the duration of the sprint, our group really found its identity as each individual settled into their role. We began to get an idea of each person's strengths and weaknesses which enabled us to build a strong foundation for our application. This groundwork includes our base UI components as well as our database, both of which were set up in a manner such that they could communicate, allowing future development to happen quickly and easily.

User Stories 1, 4, and 5

- As a user, I would like to be able to register for a College Capital Account.
- As a user, I would like to be able to choose my username.
- As a user, I would like to be able to reset my password.

#	Description	Estimated Time	Owner
1	Create UI for user sign up page	7 Hours (Research)	Justin
2	Develop algorithm to store user information in database	6 Hours (Research)	Matthew
3	Debug user input using unit tests	4 Hours	Charlie, Muhammad
4	Create UI to reset password	3 Hours	Justin
5	Develop algorithm to confirm user identity before resetting password	5 Hours	Jeremy
6	Develop algorithm to update the database's account information	2 Hours	Matthew

Completed

The homepage for our app contains links to our sign up page as well as a reset password link. When a user signs up, they can input their own username as well as other personal information that is automatically sent to the database where it is stored in a new document unique to the user. Furthermore, when resetting a password specifically, an email is sent to the user to confirm their identity before any changes are made.

User Stories 35-39

- As a user, I would like for my financial information to be encrypted at all times.
- As a user, I would like for my login credentials to be encrypted at all times.
- As an administrator, I would like to be able to decrypt user financial information.
- As an administrator, I would like to be able to decrypt user login credentials.
- As an administrator, I would like to be able to manipulate a user's account data for support purposes.

#	Description	Estimated Time	Owner
1	**Develop algorithm to encrypt user login credentials**	**6 Hours (Research) (Each)**	**Matthew Jeremy**
2	**Debug encryption algorithms using unit tests**	**3 Hours (Each)**	**Matthew Jeremy**
3	**Develop decryption algorithm for support purposes**	**2 Hours (Each)**	**Matthew Jeremy**

Completed

Upon sign up, sensitive user credentials (such as passwords) are hidden from us in our database. We can retrieve them using special hash keys that Firebase generated for us upon initialization of the project. Other user information is protected through Firebase's secure read and write functions, however these details can be manually adjusted in our console.

User Story 2

- As a user, I would like to login to my College Capital account.

#	Description	Estimated Time	Owner
1	Create UI for user login page (viewing and input)	5 Hours (Research)	Justin
2	Develop algorithm to validate user information against the database	4 Hours	Jeremy
3	Develop algorithm in cooperation with front-end that will lock account if necessary	3 Hours	Matthew
4	Debug user input using unit tests	5 Hours	Justin
5	Create UI for homepage post login	5 Hours (Research)	Justin

Completed

After signing up, users can log in through the home page of our app. Upon submission, their information is checked against our database for accuracy, and upon a successful match, the page will redirect to the user homepage where memos from all users can be seen, similar to a message board.

User Stories 3 and 45

- As a user, I would like to be able to manage my College Capital account settings.
- As a user, I would like to be able to create memos for myself.

#	Description	Estimated Time	Owner
1	**Create UI for user profile page**	**8 Hours (Research)**	**Justin**
2	**Create UI to allow for updating personal information**	**6 Hours (Research)**	**Jethro**
3	**Create UI to allow for personal messages or memos**	**6 Hours (each) (Research)**	**Muhammad, Charlie**
4	**Develop algorithm to store personal information in database (name, phone number, etc)**	**4 Hour**	**Jeremy**
5	**Debug user input using unit tests**	**5 Hours**	**Muhammad, Charlie**

Completed

After logging on, the user is presented with new options on the nav bar: new post (memos) and profile. On the profile page, users have the option of updating their personal information, in which changes will be reflected in the database. Similarly, on the new post page, a user can create a new memo which will be posted on the user's homepage and saved to a new, unique doc in our storage solution. Changes to either document in our database will be reflected immediately.

User Stories 7 and 8

- As a user, I would like to be able to monitor my current funds.
- As a user, I would like to be able to update my current funds.

#	Description	Estimated Time	Owner
1	Create UI for financials page	5 Hours	Jethro
2	Create UI to allow users to input account (checkings, savings) totals	7 Hours	Jethro
3	Develop algorithm to store user's account balances in database	4 Hour	Jeremy
4	Connect database to UI for real-time information	4 Hours (Research)	Matthew
5	Develop algorithm to update the database's financial information	4 Hour	Jeremy
6	Perform unit tests on user inputs	5 Hours	Muhammad, Charlie

Completed

Similar to profile and new post, a financials link is available to the user after signing in. Here, the user can create a financial account by filling out the supplied form. As with the other pages, this data is written to the database in a document unique to both the user and the account. Once a new account is made, the page will pull that information from the database and display it immediately. The user then has the option to update those finances, with changes to the database being reflected in the UI instantaneously.

What Did Not Go Well

As mentioned in the planning document, a big challenge for this sprint was familiarizing ourselves with our primary technologies: Firebase and React. As we were creating our app, at several different points, we realized that we had fundamental flaws with our approach in integrating both APIs. This led to us restarting our project from scratch multiple times, cutting down on our available time and resources. As such, we couldn't incorporate all the transaction features we wanted to during this sprint.

Additionally, as a result of our having restarted the project multiple times, getting the project to a point where efficient testing would have been viable was difficult. Basic testing was done in the form of some rudimentary bug-fixing, but very little was done in the way of actual unit testing (i.e. edge cases).

User Stories 13, 14, 16 and 17

- As a user, I would like to be able to update my transaction history.
- As a user, I would like to be able to view my transaction history.
- As a user, I would like to be able to categorize my expenses.
- As a user, I would like to be able to specify categories when updating my funds.

#	Description	Estimated Time	Owner
1	Develop spending categories/classes in the backend	4 Hours	Matthew
2	Create UI to allow users to input transactions in specific categories	7 Hours	Jethro
3	Create UI to view transactions/transaction history in premade categories	5 Hours	Jethro
4	Develop algorithm to update account	2 Hours	Matthew

	totals in database		
5	**Debug database update algorithm using unit tests**	**5 Hours**	**Charlie, Muhammad**
6	**Perform unit tests on user inputs**	**5 Hours**	**Charlie, Muhammad**

Not Completed

The general layout and functionality for the financials page was completed, but we ran out of time to implement transactions that build off of those user-created finances. This will be a priority for us next sprint as we build off of the foundations we laid this sprint.

Ways We Can Improve

Over the duration of Sprint 1, our group excelled at communication and being flexible as we restarted our project several times over. However, most of our major discussions occurred over Slack - this is a byproduct of each of us having very different schedules. While most of our exchanges went well over the app, sometimes, when describing technical changes or issues, there were ideas that were simply difficult to express over text. This sometimes led to confusion and differing ideas about how we would implement a specific feature or user story. In order to combat these effects in Sprint 2, our group needs to make a greater effort to meet more than twice a week, especially when working on a new component of the project. This will allow us to keep everyone on the same page while maintaining less-crowded slack channels which will also make our communication easier.

Furthermore, these additional meetings will also allow us to have a greater understanding of the APIs we are implementing to create our project. As mentioned several times throughout this document, we had an enormous amount of difficulty integrating our tools (Firebase, React, and Redux) together. This led to constant resets of our code, which set us behind. In order to stay on schedule, over the course of the next sprint, before implementing further user stories, we need to ensure that each member of our team has an understanding of what they are implementing and how they are going to implement it. We also need to ensure that each of us have a strong fundamental understanding of how our project is operating now. This will make development much easier for all of us and hopefully prevent any further large-scale changes to our code.

Finally, as a result of those code resets, our work load became unbalanced. Different individuals had to re-implement their specific components, which took varying amounts of time. Again, by increasing our basic understanding of the project, we can ensure that we stick to our even load distribution as defined in the planning document.