# College Capital

## Sprint 2 Planning Document

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

## Sprint Overview

Over the course of this sprint, our goal is to implement increased functionality that should be expected from a fully developed financial app while also refining the user stories we have already implemented to allow for a smooth user experience. Furthermore, while we designed the project to be scalable in Sprint 1, we need to continue to streamline our new React-Firebase setup in order to allow for quicker and easier development.

**Scrum Master:** Matthew Story
**Meeting Plan:** Monday, Wednesday, Friday 10:30 AM

## Risks and Challenges

The primary risk for this sprint comes from our current architecture. Last sprint, we had to restart our project from scratch on several occasions due to incompatibilities between React and Firebase. Often we would run into a situation in which our file structure was prohibiting further development, so our only choice was to start again. While we believe our current architecture will serve our needs, if we are incorrect, we run the risk of having to restart again sometime during this sprint. Therefore, before doing too much work, it is crucial we all fully understand our setup and its capabilities in order to negate this risk.

# Current Sprint Detail

**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 |

**Acceptance Criteria**

1. Given the algorithm for the visualizations page is correctly implemented, when creating expenditure visualizations, a pie graph should be generated that tracks and shows general expenses categorically for the current month.
2. Given the algorithm for the visualizations page is implemented correctly, when creating expenditure graphs, a line graph should be generated that tracks and represents expenses for each day of the current month.
3. Given the algorithm for the visualization page is implemented correctly, when creating expenditure graphs, a line graph should be generated that shows the expenditure for each week in the current month.
4. Given the manual tests are accurate, when testing, testers will manipulate financial values in order to guarantee that changes are correctly updated in the visualization.

**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 |

**Acceptance Criteria**

1. Given the algorithm for the visualization export is correctly implemented, a dropdown menu will be present on each individual visualization that allows the user to select their desired file format and download their visualization.
2. Given the algorithm for the visualization export is correctly implemented, the user will be able to export their visualization as a PNG, JPG, PDF, SVG, CSV, or an XLSX
3. Given the algorithm for the visualization export is correctly implemented, the user will be able to export their visualization and specify their desired file name to save the file under and desired file location to save the file in.
4. Given the manual tests are correct, when testing, testers will manipulate financial values in order to guarantee that changes to one's exported visualization are correct.
5. Given the manual tests are accurate, when testing, testers will manipulate export types in order to guarantee that file type is correctly updated in the exported file.

**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 |

**Acceptance Criteria**

1. Given the algorithm to populate the page with the original state of the transaction is correct, when inputting a new expenditure or transaction, the transaction should display the account that it draws money from.
2. Given the algorithm to retrieve the original state of the transactions is correct, when a user attempts to add a new expenditure, they must choose a financial account to draw from before inputting into the database.
3. Given the algorithm to update transactions in the database is correct, when a user adds a new transaction or expenditure, the financial account that the transaction draws from should be immediately updated to reflect the new account balance.
4. Given the manual tests are accurate, when testing, testers will manually add and update transactions with both valid and invalid input values to make sure input validation performs correctly.
5. Given the manual tests are accurate, when testing, testers will manually add and update transactions to make sure that both additions and updates are reflected correctly in the database.

6. Given the manual tests are accurate, when testing, testers will manually add and update transactions to make sure that both additions and updates are reflected correctly on the transactions page itself.

**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** |

**Acceptance Criteria**

1. Given the UI for viewing one's transaction history has been correctly implemented, when viewing expenditures for a specific account, transactions should be listed in the order in which they occurred.
2. Given the UI for viewing one's transaction history has been correctly implemented, when looking at a specific expenditure, transactions listed on the page should contain all relevant information (features such as date, amount, and merchant).
3. Given the UI for viewing one's transaction history has been correctly implemented, when looking at a specific expenditure, users should have the option of removing that transaction which will update their account total and not affect the other expenditures.
4. Given the manual tests are accurate, when running the unit tests, they will go back and forth between the page and the database to make sure that any updates to one's transaction history are correctly reflected on the page.
5. Given the manual tests are accurate, when testing, testers will go back and forth between the page and the database to make sure that any updates to one's transaction history are correctly reflected in the database.

**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 | 1 Hour | Charlie |

**Acceptance Criteria**

1. Given the algorithm for exporting transactions is implemented correctly, when retrieving their financial information, users should be able to export their transactions as a file type of their choosing.
2. Given the UI for exporting transactions is implemented correctly, when interacting with the page that will export their expenditures, users should be able to easily select what transactions they want to include in the generated report.
3. Given the manual tests are accurate, when testing, testers will attempt to retrieve data for export and validate against expected output.
4. Given the manual tests are accurate, when testing, testers will attempt to export transactions in CSV format.
5. Given the manual tests are accurate, when testing, testers will attempt to export transactions in JSON format.

**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 sort expense by category | 6 Hours | Justin |
| 4 | Create algorithm to apply sorting | 6 Hours | Matthew |
| 5 | Create tests to validate functionality | 1 Hour | Charlie |

**Acceptance Criteria**

1. Given the UI for specifying expense categories has been implemented correctly, when adding a new expenditure, user's should be able to easily see available options for expense categories, as well as where on the page to interact with to manipulate this information.
2. Given the algorithm to store expense categories in the database has been implemented correctly, when manipulating transactions, additions/changes of/to expenses should be correctly reflected in the database.
3. Given the algorithm for sorting expenses has been implemented correctly, when looking through their expenditures, users should be able to sort their expenses with any of the relevant columns provided.
4. Given the manual tests are accurate, when testing, testers will attempt to specify/modify expense categories such that those categories are correctly displayed on the page and updated in the database.
5. Given the manual tests are accurate, when testing, testers will make sure sorting works correctly by testing various columns to sort by.

**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 |

1. Given the UI for updating funds has been implemented correctly, when selecting a fund to edit, users should be able to see a selector for fund categories.
2. Given the UI for updating funds has been implemented correctly, when selecting a category to update funds, users should only be able to select from a list of valid categories.
3. Given the algorithm to update expense categories in the database has been implemented correctly, when updating fund categories, changes should reflect in the database in real time.
4. Given the manual tests are accurate, when testing, testers will attempt to specify fund categories such that those categories are correctly displayed on the page and updated in the database.
5. Given the manual tests are accurate, when testing, testers will attempt to specify blank fund categories and should fail to do so.

**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 |

**Acceptance Criteria**

1. Given the algorithm for the Daily Usage visualization is correctly implemented, when creating expenditure day by day visualizations, a line graph should be generated that tracks and shows general expenses over the current month day by day.
2. Given the algorithm to retrieve data for day by day visualizations is implemented correctly, when retrieving the data to generate expenditure graphs, only data from the chosen length of time should be included in the visualization.
3. Given the manual tests are accurate, when testing, testers will manipulate financial values in order to guarantee that changes to one’s financial information are correctly updated in the database.
4. Given the manual tests are accurate, when testing, testers will manipulate financial values in order to guarantee that changes are correctly updated in the visualization.

**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 |

**Acceptance Criteria**

1. Given the algorithm for the Weekly Usage visualization is correctly implemented, when creating expenditure day by day visualizations, a line graph should be generated that tracks and shows general expenses over the current month week by week.
2. Given the algorithm to retrieve data for week by week visualizations is implemented correctly, when retrieving the data to generate expenditure graphs, only data from the chosen length of time should be included in the visualization.
3. Given the manual tests are correct, when testing, testers will manipulate financial values in order to guarantee that changes to one’s financial information are correctly updated in the database.
4. Given the manual tests are accurate, when testing, testers will manipulate financial values in order to guarantee that changes are correctly updated in the visualization.

**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 |

**Acceptance Criteria**

1. Given the algorithm for the Monthly Usage visualization is correctly implemented, when creating expenditure Month by Month visualizations, a line graph should be generated that tracks and shows general expenses over the current year month by month.
2. Given the algorithm to retrieve data for month by month visualizations is implemented correctly, when retrieving the data to generate expenditure graphs, only data from the chosen length of time should be included in the visualization.
3. Given the manual tests are correct, when testing, testers will manipulate financial values in order to guarantee that changes to one’s financial information are correctly updated in the database.
4. Given the manual tests are accurate, when testing, testers will manipulate financial values in order to guarantee that changes are correctly updated in the visualization.

**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 |

**Acceptance Criteria:**

1. Given the algorithm for delivering messages is functioning properly, when a user opens a new ticket for Support, then the ticket will appear on the User's messaging module and the Support account will receive an email with the User's email and the ticket's content.
2. Given the UI for the Support Page is implemented correctly, when the user visits the Support page, then he should be able to view all active tickets sent to Support that are stored in the database.
3. Given there is integration between the messaging module and Firebase, when a message is sent from the user to Support, then the message should be stored in the Firebase database for future viewing.
4. Given the manual tests are accurate, when testing, testers will send messages through the support module to make sure they are sent correctly from User to Support.
5. Given the manual tests are accurate, when testing, testers will send messages through the support module to make sure they are saved correctly from User to Support.

6. Given the manual tests are accurate, when testing, testers will monitor where the messages are stored to ensure that encryption of the messages is done correctly.

**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 |

**Acceptance Criteria:**

1. Given the algorithm for delivering messages is properly implemented, when a user sends a message to Support, then it should be displayed on the Firebase database under their UID and username for administrators to view.
2. Given the UI for messaging Support is functioning, when an administrator views the tickets that users have sent, then he should be presented with the contents of the message as well as the email of the user it was sent by.
3. Given the algorithm for administrators messaging users is working, when an administrator messages a user to provide assistance, then the message should be delivered to the user from an email collegecapteam@gmail.com or support@collegecap.org.
4. Given the UI and algorithm to manage messages are implemented correctly, when closing a ticket, the ticket should be removed from the database and subsequently the User's messaging module.
5. Given the manual tests are accurate, when testing, testers will use the message module to test that message manipulation (marking as completed, deleting, etc.) is done correctly.

6. Given the manual tests are accurate, when testing, testers will attempt to send messages back to users to make sure that both sending and receiving as both a user and support personnel are done correctly.

**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 |

**Acceptance Criteria:**

1. Given the algorithm for storing messages is correctly implemented, when looking at messages offline users have sent, site administrators should be able to see the messages stored by specific users in our Firebase database.
2. Given the functionality to global message all users is implemented correctly, when opening their homepage, all Users should be able to see all the latest notifications from support when they log back on.
3. Given the security for the global messaging module is implemented, when any users who are not the support account attempt to make a new post, they should be redirected to the homepage with the support notifications.
4. Given the manual tests are accurate, when testing, testers will make sure that messages are saved correctly in such a way as to be accessible even when not online.
5. Given the manual tests are accurate, when testing, testers will make sure that any messages received while offline will prompt a notification upon the User logging back in.

# Remaining Backlog

### Functional Requirements (User Stories)

1. ~~As a user, I would like to be able to register for a College Capital account.~~
2. ~~As a user, I would like to be able to login to my College Capital account.~~
3. ~~As a user, I would like to be able to manage my College Capital account settings.~~
4. ~~As a user, I would like to be able to choose my username.~~
5. ~~As a user, I would like to be able to reset my password.~~
6. As a user, I would like to be able to split my account balance into multiple accounts.
7. ~~As a user, I would like to be able to monitor my current funds.~~
8. ~~As a user, I would like to be able to update my current funds.~~
9. ~~As a user, I would like to be able to visualize my current financial status.~~
10. As a user, I would like to be able to visualize past spending habits.
11. ~~As a user, I would like to be able to export my visualizations.~~
12. As a user, I would like to be able to view a forecast of my current spending habits.
13. ~~As a user, I would like to be able to update my transaction history.~~
14. ~~As a user, I would like to be able to view my transaction history.~~
15. ~~As a user, I would like to be able to export my transaction history.~~
16. ~~As a user, I would like to be able to categorize my expenses.~~
17. ~~As a user, I would like to be able to specify categories when updating my funds.~~
18. As a user, I would like to be able to add my own, custom spending categories.
19. As a user, I would like to be able to search my previous transactions.
20. As a user, I would like to be able to provide limits to overall spending.
21. As a user, I would like to be able to provide limits to specific spending categories.
22. As a user, I would like to be able to receive alerts when I exceed my spending limitations.
23. ~~As a user, I would like to be able to see a snapshot of my daily usage.~~
24. ~~As a user, I would like to be able to see a snapshot of my weekly usages.~~

25. ~~As a user, I would like to be able to see a snapshot of my monthly usages.~~
26. As a user, I would like to be able to be alerted of low account balances.
27. As a user, I would like to be able to be alerted of large single transaction purchases.
28. As a user, I would like to be able to schedule regular expected payments.
29. As a user, I would like to be able to see spending suggestions based on my expenditures.
30. As a user, I would like to be able to use different browsers.
31. As a user, I would like to be able to use different screen resolutions.
32. ~~As a user, I would like to be able to message support 24/7 when I need help with the app.~~
33. ~~As an administrator, I would like to be able to respond to support messages 24/7.~~
34. ~~As an administrator, I would like to be able to respond to support messages when the user is offline.~~
35. ~~As a user, I would like for my financial information to be encrypted at all times.~~
36. ~~As a user, I would like for my login credentials to be encrypted at all times.~~
37. ~~As an administrator, I would like to be able to decrypt user financial information.~~
38. ~~As an administrator, I would like to be able to decrypt user login credentials.~~
39. ~~As an administrator, I would like to be able to manipulate a user's account data for support purposes.~~
40. As a Purdue student, I would like to be able to view my dining dollars (if time permits).
41. As a Purdue student, I would like to be able to view my university dining hall meal swipes (if time permits).
42. As a Purdue student, I would like to be able to view my Purdue boiler express funds (if time permits).
43. As a student, I would like to input financial aid.
44. As a student, I would like for my account totals to be adjusted properly after inputting financial aid.
45. ~~As a user, I would like to be able to create memos for myself.~~

46. As a user, I would like to be able to view public financial information about companies I've invested in (if time permits).
47. As a user, I would like to be able to use PayPal features from within the app (if time permits).
48. As a user, I would like to be alerted of any suspicious logins.
49. As a user, I would like to be able to add authorized users that can access my account information.
50. As a user, I would like to be able to use two factor authentication (if time permits).
51. As a user, I would like to be able to create expense reports (if time permits).
52. As a user, I would like to be able to file taxes (if time permits).
53. As a user, I would like to be able to pay employees (if time permits).
54. As a user, I would like to be able to keep a log of the payments (if time permits).
55. As a user, I would like to be able to report any issues with payroll (if time permits).
56. As a user, I would like to be able to claim and transfer my money (if time permits).
57. As a user, I would like to be able to modify time entries (if time permits)
58. As a user, I would like to be able to authorize payments to employees (if time permits).
59. As a user, I would like to be able to share my financial history (if time permits).
60. As a user, I would like to be able to see my credit score using Fico services (if time permits).
61. As a user, I would like to be able to login using a Gmail account.
62. As a user, I would like to be able to cancel any unwanted transactions.

## Non Functional Requirements

1. As an administrator I would like the app to run on all browsers so users can have access to the website from any browser of their personal choice. Specific browsers that will definitely be accommodated will be Google Chrome, Mozilla Firefox, and Microsoft Edge.
2. The app should be functional on screens of different resolutions, allowing patrons to use the app on old as well as new machines. It will certainly support 720p, 1080p, 1440p, 2k, and 4k resolutions.
3. The UI must be clear, direct, and clean. The UI will be of minimalistic design so that it is easier for users to follow all the information and are able to interact with the web app with ease. The UI design should also be able to accomodate a relatively large number of modules for displaying financial information, up to around 15 modules.
4. The app should be easily scalable to accomodate a much larger user base, up to around 1000 users at any given time, and still be able to operate 24 hours a day, 7 days a week.
5. The app must be secure in all facets, but especially concerning user information in the database. As an administrator we would like to create a network which allows 1000's of users data to be stored in the database without any security threats which will run 24/7.