

# College Capital

## Sprint 1 Planning Document

---

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

### Sprint Overview

Over the course of this sprint, we hope to complete fundamental components of the front end and back end of the web application while gaining familiarity with the languages and platforms that we will be utilizing throughout the semester. Our focus will mainly lie with basic features such as a user account system, input of financial data, storing that information in our database, and a basic security system. Furthermore, while labor is divided between front end and back end, we hope to bring all components together to have a basic application by the end of the sprint rather than several individual pieces.

**Scrum Master:** Matthew Story

**Meeting Plan:** Wednesday's 7:30 PM, Friday's 5:30 PM

### Risks and Challenges

The two primary APIs for this project are React and Firebase. While we have already started working with them, the main challenge for this sprint will be learning how to use both effectively in our work. Furthermore, because our app handles sensitive information, security features need to be implemented right away. This will be another challenge as we can't devote all of our time to security, but what we do implement needs to be robust and secure.

## Current Sprint Detail

### 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,Muh ammad
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

### Acceptance Criteria:

1. Given the UI is correctly implemented, when attempting to fill out fields such as username, password, email, etc., the corresponding fields should only accept certain characters appropriate for each field.
2. Given the UI is correctly implemented, when the user attempts to submit invalid fields for personal information, then the page should reload with an error message specifying which field is incorrect.

3. Given the database algorithm is implemented correctly, when an administrator attempts to view a user's login information, then there should be a clear one-to-one relationship between username and password.
4. Given the identity-confirmation algorithm is correct, when a user attempts to reset their password, then they should receive an email with a unique passcode to confirm that they own the account.
5. Given the database is correctly implemented, when a user attempts to create an account with an email or username that already exists, the page should reload stating that the information is already in use.

**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

**Acceptance Criteria:**

1. Given the username encryption algorithm is implemented correctly, when an administrator views the raw information in the database, then they should not be able to understand the raw data.
2. Given the password encryption algorithm is implemented correctly, when an administrator views the raw information in the database, then they should not be able to tell what the passwords are.
3. Given the encryption algorithms are implemented correctly, when user information is stored in the database, then the information should not be able to be retrieved by users other than the one who submitted it.

4. Given the encryption algorithms are correctly implemented, when user information is stored in the database, the information should not be easily decrypted except by administrators.
5. Given the decryption algorithm is implemented correctly, when an administrator needs to retrieve a particular username or password, they should be able to decrypt the encrypted information to ensure correctness.

**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

**Acceptance Criteria:**

1. Given the algorithm to validate a user's information is correct, when a user attempts to login with the incorrect username or password, the login page should reload displaying an error message.
2. Given the UI for the login page is correct, when a user attempts to login to an account 6 or more times incorrectly, the page should reload stating the account has been locked and to contact support.
3. Given the algorithm that locks accounts is correct, when an account is locked after several incorrect login attempts, the email associated with the account should receive a notification.
4. Given that the algorithm to validate a user's information is implemented correctly, when a user provides the correct account username and password, the page should reload into the user's homepage.

5. Given the UI for the homepage is correctly implemented, when a user attempts to view the page without being logged in, the page should redirect to the login page without loading any information from the database.

**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

**Acceptance Criteria:**

1. Given the algorithm to authenticate a user's identity is correct, when a user logs in for the first time, the page should reload to a personal information upload page.
2. Given the algorithm to store personal information in the database is correct, when a user adds new information or memos to their account, that information should be encrypted using the same algorithms that encrypt username and passwords.
3. Given the UI to update personal information is correct, when a user changes the fields with their personal information, those changes should not be updated in the database until the user confirms their changes.



4. Given the UI for the user profile page is correct, when a user is not logged in to their account, the profile page should reload and redirect to the user login screen.

**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

**Acceptance Criteria:**

1. Given the UI for the financials page is correct, after a user's first successful login, every subsequent login should redirect the user to the financials page with a prompt to setup a monetary account if they don't have one already.
2. Given the algorithm to store personal information in the database is correct, when a user adds new financial accounts or other information to their account, that information should be encrypted using the same algorithms that encrypt username and passwords.
3. Given the algorithm to validate a user's input is correct, when a user enters information that contradicts with field validation rule, the page should prompt the user for the correct format.

4. Given the algorithm to validate a user's input is correct, when a user enters valid information, the page should give indication that the new input has been stored.
5. Given the algorithm to synchronize UI with database is correct, when a user enters a new valid input, the page should reload dependent fields with updated information.

**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 totals in database	2 Hours	Matthew
5	Debug database update algorithm using unit tests	5 Hours	Charlie, Muhammad
6	Perform unit tests on user inputs	5 Hours	Charlie, Muhammad

**Acceptance Criteria:**

1. Given the algorithm to authenticate a user's identity is correct, when a user logs in with correct credentials, the page should reload and display the user's transaction history.
2. Given the UI to create/edit transaction is correct, when a user creates/edits a transaction, the user should be able to assign a specific category to the transaction.

3. Given the UI to view transaction history, when a user selects a category to view, the page should reload with all the transactions in the selected category.
4. Given the algorithm to authenticate a user's input is correct, when a user tries to save a transaction with invalid inputs, the page should highlight the invalid inputs and prompt the correct format.
5. Given the algorithm to update account totals is correct, when a user creates/edits a transaction, the changes should be uploaded to database and update account totals.

## 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 authorise 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.