Team 3

College Capital

Product Backlog

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

Problem Statement

College students today have more to manage than ever with rigorous course loads, hectic social lives, and an increasingly present sense of independence. Along with these increased responsibilities comes a larger financial burden, one which can cause stress and unnecessary chaos while attempting to juggle various sources of income and expense. College Capital alleviates these burdens by aggregating a student's financial information into one place, allowing for quick access to things like monthly tuition payments, weekly meal plans, and personal expenses. Unlike other financial services, College Capital distinguishes itself by providing features like predictive spending based on current habits and suggestions to help the user manage their finances in the future. As a final product, it will be a one-stop shop for a college student's financial needs, featuring seamless integration with their existing financial services - such as Dining Dollars - and providing a complete overview of their finances.

Background Information

The Problem: As a result of the immense suite of life changes thrust onto the average college student, they typically struggle with managing their personal finances. Reasons for this can include neglecting to create a basic budget, misusing their credit card and/or neglecting to plan ahead financially. Our platform helps college students manage and plan their finances and suggest to user's how they can create healthy financial habits.

The Audience: Modern college students deal with varying financial stressors, managing various social and professional accounts while also adhering to budgeting constraints. Current options to centralize these entities are poorly designed, have low functionality, and lack usability. While the primary audience is college students or post-grads who struggle with these issues, the application is accessible to all.

Domain/Other Applications and their Limitations: Other applications do not include financial sources unique to college students, such as Dining Dollars, Boiler Express, or Meal Plans (in the case of a Purdue student). In addition, other applications, depending on profit models, require purchases/subscriptions or sell user data. This application will function as a freeware during its lifetime sponsored by the Purdue University College of Computer Science. Furthermore, this app will provide additional functionality - compared to its competition - in predictive spending models and suggestions on how to budget while maintaining the same quality of life.

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 be able to use different browsers.
31. As a user, I would like to be able 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

Basic Performance and Architecture

The implementation of front-end and back-end are kept separate to simplify division of labor and for more consistent compatibility of different work. For front end we will be using HTML, CSS, and JavaScript, to maintain consistency throughout the many entities that the application will manage, boost browser compatibility with the application, and keep viewing options open for possible expansion to a mobile platform. For back end, we will be using Java and Firebase as our database to utilize industry-tested stability and handy features provided natively. Additional technology can be reviewed and added the project later if proven to be beneficial.

Security

Given the delicate nature of financial information, it is essential that both users and organizations relying on our app have their information soundly secured, 24/7. In our implementation, the project will use secure channel communication (https) to protect user data on-the-fly. Upon completion, a security audit will be conducted against the application and database upon project completion.

As a result of College Capital's deep integration with Firebase, we will be employing several of the security features that come with the API. These include a vast set of Realtime Database Rules that allow for custom user permissions, data validation, and user authentication. Furthermore, in terms of storage, we will be encrypting both user data and passwords to ensure private information does not fall into the wrong hands.

Usability

Given the multitude of features and entities that we plan to implement in the software, it will be vital that the user interface is kept tidy and well designed to promote the user viewing experience. Considering this, it will also be important that the design is compatible with different screen resolutions, screen proportions, and browser interfaces. Similar implementations separate the individual entities to their own pages or must be accessed with individual hyperlinks to a separate website. We wish to create a single viewing hub where all the desired information can be seen and analyzed easily.

Scalability

As a web app, the site should be accessible 24 hours a day, 365 days a year. We plan on being able to host at least 100 users concurrently, handling around the same number of requests per second, if not more. Speedwise, the website should be implemented in such a way as to be able to load within 300-400 ms. The backend is designed to hold 1 GB of user data initially. Should the application produce more data, Firebase can be easily upgraded to higher storage capacity.

Hosting and Deployment

Due to the backend and frontend being implemented separately, their changes and deployment can be updated separately as well. For now we plan to host the backend on a Google Firebase server for development and testing. For front end we would host it on a Github Pages Site, with the possibility of migrating to a custom Google firebase domain.