**Introduction**

Team member roles:

1. AJ - Connecting PLAID to budgeting application
2. Brendan - Connecting PLAID to budgeting application
3. Cole - Backend
4. Ethan - Frontend
5. Laura - Frontend
6. Liam - Backend

Link to GIT repo: <https://github.com/colepp/Financial-app.git>

Communication Tools:

* Microsoft Teams
* Text message group

**Product Description**

Our app will track the users bank accounts using PLAID. You will be able to create budgets for yourself and track your spending.

Major features:

1. Customers will be able to view their bank account balance
2. Customers will be able to view bank account transaction history
3. Customers will be able to create custom budgets to track how they wish to spend their money
4. Customers will be able to track spending over time and view trends in their spending

Stretch Projects:

1. Implement a tool to make budgeting suggestions for users
2. Allow users to view their purchased cryptocurrency
3. Allow users to view and make changes to an investment portfolio

**Use Cases:**

View Account Balance - Cole

| Goal | View linked bank account balance |
| --- | --- |
| Primary Actor | Users with account in budgeting app |
| Scope | Personal finance/budgeting app |
| Level | User |
| Precondition | Successful installation of application and user is at login screen |
| Success end | User has successfully viewed the bank account balance of their linked bank account |
| Failure end | User cannot view the bank account balance of their linked bank account |
| Trigger | User clicks on the “Account Info” button in the navigation bar |
| Main success scenario | 1. User enters username and password 2. Application system verifies that the username and password entered are correct and logs the user in 3. User lands on the home screen 4. User clicks on the “Account info” button in the navigation bar 5. User lands on the account info screen 6. User scrolls down on page to view account balance of their linked bank account |
| Exceptions (error scenarios) | 2a. Username/Password is incorrect  2a-1. System returns user to login screen  2a-2. User attempts to input username and password again  6a. No bank account currently linked  6a-1. User clicks on settings in the navigation bar  6a-2. User lands on the settings page  6a-3. User clicks on the “Link Bank” button under bank settings section  6a-4. User follows steps of the “Link a Bank Account” use case below  6b. Bank account balance could not be fetched (PLAID API error)  6b-1. User reloads the page  6b-2. User attempts to view their bank account balance again |
| Variations (alternative scenarios) | Alternate Route 1:  3. User lands on the home screen  4. User clicks on the “Budgeting” button in the navigation bar  5. User lands on the budgeting page  6. User clicks on the “Create Budget” button at the top of the page  7. User lands on the budget creation page  8. User will be able to view their current account balance at the top of the page |

Create an Account - Liam

| Goal | Allow users to create an account that will give them access to the features of the app and that they can sign in to when they return to the app. |
| --- | --- |
| Primary Actor | First-time users |
| Scope | Personal finance/budgeting app |
| Level | User |
| Precondition | Successful installation |
| Success end | Users now have an account they can login to whenever they use the app. Account is associated with their name/email/phone number and the linking of financial accounts to the app account is optional. |
| Failure end | User does not create an account. |
| Trigger | User launches the app for the first time or presses ‘create account’ button at the login screen. |
| Main success scenario | 1. User opens app for the first time or selects ‘create account’ from login screen 2. User enters required info (Full name, email, phone #, username, new password, confirm new password) 3. Account is stored in database and user can now configure their profile(associate with bank/investment accounts) 4. Users can now access features of the app and log back in whenever. |
| Extensions (error scenarios) | 2a. Parts of necessary information are missing   * 2a.1. Highlight missing entries + prompt user to re-enter with proper info   2b. Email is not valid   * 2b.1. Prompt user to re-enter with proper email   2c. Username is taken   * 2c.1. Ask user to re-enter with new username (check before they hit enter if valid)   2d. Passwords do not match/fit security criteria   * 2d.1. Ask user to re-enter with proper passwords |

Link a Bank Account - AJ:

| Goal | Linking a bank account with a budgeting app account |
| --- | --- |
| Primary Actor | User of budgeting app with an account |
| Scope | Budgeting application with PLAID |
| Level | User |
| Precondition | User is at login screen |
| Success End | User successfully links bank account with budgeting app account |
| Failure End | User fails to link their bank account to their budgeting account |
| Trigger | User clicks “Link” button in settings to link a bank account to their account |

| Main Success Scenario | 1. User enters username and password 2. Application system verifies that the username and password entered are correct and logs the user in 3. User lands on the home screen and clicks on the settings button 4. User navigates through the settings to the “Link Bank” button and clicks it 5. PLAID pops up with a window and asks the user to select their bank 6. The user successfully selects their bank 7. A login screen pops up for the user to login to their bank 8. User successfully enters their login information 9. PLAID asks the user to select the account that they wish to link 10. User selects their account and clicks “Link” 11. PLAID informs the user that their account has been linked 12. PLAID window closes |
| --- | --- |
| Exceptions (error scenarios) | 2a. Username/Password is incorrect  2a-1. System returns user to login screen  2a-2. User attempts to input login information again  5a. PLAID API is currently experiencing difficulties  5a-1. PLAID pops up with a window informing the user that they are experiencing difficulties  5a-2. User clicks exit button to exit the window and returns to the settings screen  5a-3. User attempts to link bank account at a later time  6a. User selects the wrong bank  6a-1. PLAID brings user to the login screen for the incorrect bank  6a-2. User clicks the back button  6a-3. PLAID returns user to the list of banks  6a-4. User attempts to select the correct bank again  8a. User inputs incorrect login information  8a-1. Bank window returns user to the login screen of their selected bank  8a-2. User attempts to input login information for their bank again  9a. PLAID is unable to verify login information with the bank (not user error)  9a-1. PLAID returns user to login screen of their selected bank  9a-2. User attempts to input correct login information for their bank again  11a. PLAID experiences difficulties linking the selected account  11a-1. PLAID returns the user to the account selection screen  11a-2. User again selects the account that they wish to link  11a-3. User clicks the “Link” button |
| Variations (alternative scenarios) | Alternate Route 1:  3. User lands on home screen and clicks on the “Budgeting” button in the navigation bar  4. System navigates user to the budgeting screen  5. User clicks on the “Link an Account” button  6. User follows steps 5-11 in main success scenario |

Login – Laura

| Goal | Create a login page for a budgeting app account |
| --- | --- |
| Actor | Existing user of budgeting app |
| Precondition | User has an account and is at the login screen |
| Postcondition (success) | User successfully logs into the budgeting app |
| Trigger | User opening the app |
| Postcondition (success) steps | 1. User enters username and password 2. The system checks that the account exists and the correct password was input 3. The information is successfully verified, and the user is “let in” to the app (now associated with their account) |
| Exceptions (failure condition and scenario) | 1. Username or password are incorrect    1. User gets an error message letting them know the information provided is wrong/“does not match our records”    2. “Forgot my password” prompt is on screen below login prompts    3. User either 1) tries to input their information again or 2) clicks on “forgot my password”       1. If option 2, the user is asked to confirm their email       2. User receives an email with a one time code that they may input into the app to confirm they are the email account owner       3. In the app, the user is prompted to input the code they received on their email       4. User is redirected to the password reset page and resets password |
| Variation of success scenario | 1. User is asked if they are a new or returning user 2. If returning user, they are redirected to the login page 3. Postcondition (success) steps follow |

Create a Budget – Brendan

| Goal | The user creates a budget for themself |
| --- | --- |
| Primary Actor | Existing user of our app |
| Precondition | User has an account with a connected bank account and is logged in |
| Postcondition (success) | User successfully creates a budget using their bank account |
| Trigger | User presses a button to create a budget |
| Postcondition (success) steps | 1. Click a button to create a budget 2. Allow us to collect information about your income from PLAID 3. Allow us to collect information about your spending habits from PLAID 4. Enter your budget goals 5. Determine if you want notifications about your budget when you spend 6. Finalize Budget |
| Exceptions (failure condition and scenario) | The user does not allow to use their current account information to create a budget |
| Variation of Success Scenario | The user sets a budget for a set period such as a week, month, or year |

View Past Spending and Log Purchases - Ethan

| Goal | View past spending and log new purchases |
| --- | --- |
| Primary Actor | User of budgeting app with an account |
| Scope | Budgeting application with PLAID |
| Level | User |
| Precondition | User has an account and is at login screen |
| Success End | User has successfully logged purchases and can see updated past spending |
| Failure End | User has failed to log purchases and past spending is not updated |
| Trigger | User clicks the calendar button |
| Main Success Scenario | 1.User signs in and lands on homepage  2. User clicks the calendar icon  3. User is able to view purchases and log new purchases   1. To log new purchases, user clicks the “+” icon. User then describes the category of their purchase and enters the amount spent 2. To view purchases, the calendar can expand to visually show frequency of purchases and amount spent.   4. User now sees an updated calendar of purchases |
| Exceptions (error scenarios) | User has not made any purchases, inputs an incorrect value (negative, for example) |
| Variations (alternative scenarios) | More in-depth chart or graph of purchases is displayed, user may also be able to add purchases quickly through a shortcut on the homepage |

**Non-functional Requirements**

1. Encryption/Security - When storing personal bank info, we have to be diligent in our security processes.
2. Database response time - need the server/database to be able to respond in a reasonable amount of time to have a smooth user experience.
3. PLAID connection - Users have no control over if this works as long as they put in the right information (Need everything connected on our end)

**External Requirements**

1. The application will be able to handle errors such as invalid inputs into forms, invalid inputs into the PLAID windows, and any other common user errors that may occur
2. The application will be installable from the github that we have linked above. After installation users will be able to run the application with all features
3. In addition to the source code of the application being accessible to others on our github, we will provide documentation for the installation of the application and each feature of our application

**Team Process Description**

1. Toolset we intend to use:
   1. VScode - IDE we will all use
   2. Kanban Jira - Project tracking software to manage progress,task and bugs
   3. Github - Repository for sharing main project build
   4. Django - Backend Toolset for managing pages and database
2. Member Roles and Milestones:

* AJ - Connecting PLAID to budgeting application
  + Why: The PLAID API is essential to our application and needs specific team members to work on understanding the documentation of PLAID and connecting it to the budgeting application
  + Justification: Worked with APIs before in an internship. Have experience with Postman and testing APIs.
* Brendan - Connecting PLAID to budgeting application
  + Why: The PLAID API is essential to our application and needs specific team members to work on understanding the documentation of PLAID and connecting it to the budgeting application
  + Justification: Previous experience working with APIs. Know how to test and implement APIs.
* Cole - Backend
  + Why: Having not just a functional but also a working database that will work quickly and seamlessly also creating maximum security procedures for creating the data
  + Justification: I am currently working in and have worked in backend development in the past
* Ethan - Frontend
  + Why: Frontend and UX/UI design is crucial in making the application easy to use and accessible for users. A well-polished application inspires the user’s confidence in the company and reflects the company’s image and values.
  + Justification: Experience with UX design courses and practices, HTML, CSS and JavaScript
* Laura - Frontend
  + Why: Frontend and UX/UI design is crucial in making the application easy to use and accessible for users. A well-polished application inspires the user’s confidence in the company and reflects the company’s image and values.
  + Justification: Have experience with web design, HTML, CSS and JavaScript.
* Liam - Backend
  + Why: Having a working database and server linked to the app will be crucial to the performance and success of our application. Being able to store/access/protect the important user information and being able to connect the app to the internet is crucial to the operation of the app.
  + Justification: Currently working in back-end development at FMD

Milestones:

* Connecting PLAID:
  1. Implementation of a button that links PLAID quickstart
  2. Implementation of a button that allows the linking of a bank account to a user account
  3. Allow bank account balances to be viewable in the application
  4. Allow bank transactions to be viewable in the application
* Frontend:
  1. Have a barebones website that we can build upon
  2. Design a sleek and professional landing page
  3. Create pages for each banking feature
  4. Translate website into a mobile app
* Backend:
  1. Having remote access to a database/server
  2. Connect to the app and be able to store data/communicate with the server
  3. Meet response time/up-time requirements
  4. Make translation to support mobile application

1. Risks Preventing Completion:
   1. Running out of time to implement minimum features
   2. Being unable to connect PLAID to the application
   3. Being unable to implement features in frontend/backend
2. Because we will be implementing the prototyping model, feedback throughout the project will be important from outside sources as we develop our prototypes. However, during the latter half of the semester, feedback from outside sources will become especially important as we refine our prototypes and get closer to our final product. So, after the halfway point of the semester, we will conduct hallway testing to get feedback from peers as outside sources to determine what improvements we can make to our application.