Project Design

Due TBA

1. **Project Name:** YggFinance
2. **Group Members:** Alan Holman, Blake Hudson, and Austin Kerr
3. Temporary Design Information:
   - **YggFinance Web App**
     - The User will be able to **connect to** the YggFinance Web App through a URL in any modern Web Browser.
     - The YggFinance Web App will use the Material UI **style library** for displaying all of its elements to the User.
     - The YggFinance Web App will contain 4 Pages accessible through tabs located in a **Top Menu**: The **Welcome Page**, The **Savings Planner Page**, The **Monthly Budget Page**, and The **Net Worth Tracking Page**.
       1. The **Top Menu** should be a default Material UI TabPanel that will be available to the User at the top of the Web App at all times.
     - Upon connecting, the YggFinance Web App will open to the **Welcome Page**.
       1. The **Welcome Page** will contain a short description of each other page in the **Top Menu** with their respective uses and tools.
     - The YggFinance Web App will **store User Collected Input Data** on the user's machine as a cookie.
       1. No User Information will be stored on the Yggdrasil Server between sessions. (Server is Stateless)
   - **Savings Planner Page**
     - The Savings Planner Page will have an **initial investment field** for the user to input data into.
       1. The Initial investment field will be **labeled**: "Initial investment (USD)".
       2. The Initial investment field will **validate user input** to be only non-negative numbers in US currency form.
          a. If user input does not match the required format or the input is empty, the Web App will display a *Warning Message* (See Req. 5.1) to the user explaining such.
     - The Savings Planner Page will have an **average rate of return field** for the user to input data into.
       1. The average rate of return field will be **labeled**: "Average rate of return (%)".

2. The average rate of return field will **validate user input** to be only non-negative decimals in percentage form. (e.g. The user enters 7.02% as a percentage, this translates to the decimal form 0.0702 for the calculations).
    a. If user input does not match the required format or the input is empty, the Web App will display a *Warning Message* (See Req. 5.1) to the user explaining such.

- The Savings Planner Page will have a **monthly contributions field** for the user to input data into.
    1. The monthly contributions field will be **labeled**: "Monthly contributions (USD)".
    2. The monthly contributions field will contain a **default value** of 0.
    3. The monthly contributions field will **validate user input** to be only non-negative numbers in US currency form.
        a. If user input does not match the required format or the input is empty, the Web App will display a *Warning Message* (See Req. 5.1) to the user explaining such.

- The Savings Planner Page will have a **Planning Mode <u>Switch</u>** for the user to toggle.
    1. The Planning Mode Switch will **default** to the left position.
    2. The Planning Mode Switch should have **3 Labels**.
        a. The **Top Label** of the Planning Mode Switch should reflect the purpose of the switch.
            i. The **Top Label** will **display**: "Planning Mode".
        b. The **Side Labels** of the Planning Mode Switch should reflect the Type of Output that the Savings Mode Planner will give upon pressing the Calculate Button.
            i. The **Left Label** will **display**: "Time Frame".
            ii. The **Right Label** will **display**: "Savings Goal".

- The Savings Planner Page will have a **Time-Frame <u>Slider</u>** for the user to adjust.
    1. The Time-Frame Slider will be **labeled**: "Years to grow".
    2. The Time-Frame Slider will **default** to a value of 10 years.
    3. The Time-Frame Slider will have a **valid value range** beginning at 1 year and ending at 80 years.
    4. The Time-Frame Slider will **be enabled and visible** when the *Planning Mode Switch* is in the **Right** Position.
    5. The Time-Frame Slider will **be disabled and hidden** when the *Planning Mode Switch* is in the **Left** Position.

- The Savings Planner Page will have a **Savings Goal Text Field** for the user to input data into.
    1. The Savings Goal Text Field will be **labeled**: "Savings Goal (USD)".

2. The Savings Goal Text Field will **validate user input** to be only non-negative numbers in US currency form.
   a. If user input does not match the required format or the input is empty, the Web App will display a ***Warning Message*** (See Req. 5.1) to the user explaining such.
3. The Savings Goal Text Field will **be enabled and visible** when the *Planning Mode Switch* is in the **Left** Position.
4. The Savings Goal Text Field will **be disabled and hidden** when the *Planning Mode Switch* is in the **Right** Position.

■ The Savings Planner Page will have a **Calculate Button** for the user to press.
   1. The Calculate Button will be **labeled**: "Calculate".
   2. The Calculate Button will **be disabled** by **default**.
   3. The Calculate Button will **be enabled** when the following Text Fields contain **valid data**:
      a. The Initial Investment Text Field
      b. The Average Rate of Return Text Field
      c. (If Enabled and Visible): The Savings Goal Text Field
   4. When Pressed, The Calculate Button will **send a request** to the *Savings Planner Service (See Req. XXX)*.
      a. The **request** should contain all of the data that the user has input into the Savings Planner Page.
      b. When YggFinance **receives a response** from the *Savings Planner Service (See Req. XXX)*, YggFinance will update the *Results Table (See Req. 2.8)* with the data contained within the response.

■ The Savings Planner Page will have a **Results [Table](#)** to display results to the User.
   1. The Results Table will be **labeled**: "Results".
   2. The Results Table **Layout** will have No Header, 2 Columns, and 4 Rows.
      a. **Column 1, Row 1** will be **labeled**: "End Balance:".
      b. **Column 1, Row 2** will be **labeled**: "Starting Amount:".
      c. **Column 1, Row 3** will be **labeled**: "Total Contributions:".
      d. **Column 1, Row 4** will be **labeled**: "Total Interest:".
      e. **Column 2, Rows 1-4** will be **formatted** in US Currency form.
      f. **Each Row in Column 2** will **display data** associated with the label on the same Row in Column 1. (See Req. 2.7.4.2)

○ **Monthly Budget Page**
   ■ **Currently Displayed Month [Select](#)**
      1. Defaults to current month.
      2. Changes which month's data to be displayed in *Categories Collapsible Table* (See Req. 3.2).

- **Category Collapsible EditTable**
  1. 4 Columns
     a. Column 1: No Header
        i. Each Row: Collapse Component
           1. Default State: Not Hidden & Collapsed
           2. Inside Collapse Component: Transaction Table (See Req. 3.3)
     b. Column 2: "Category" Header
        i. Each Row: Name of Category (TextField)
           1. Default: "Unnamed"
           2. Validation: Cannot be Empty
     c. Column 3: "Spent" Header
        i. Each Row: Amount Spent in USD (TextField)
           1. Value: Adjusted Sum of all Entries in Column 4 of the Transaction EditTable (See Req. 3.3.1.5).
              a. Adjustment: For each Row, Column 4's number becomes negative if the entry in Column 3 is Checked.
           2. Validation: Cannot be Edited
           3. Format: USD Form (1.2345 -> $1.23)
     d. Column 4: "Planned" Header
        i. Each Row: Amount Planned in USD (TextField)
           1. Default: $0.00
           2. Validation: Must be a Non-negative Number
           3. Format: USD Form (1.2345 -> $1.23)
- **Transaction EditTable**
  1. 5 Data Columns
     a. Column 1: "Name" Header
        i. Each Row: Name of the Transaction (TextField)
           1. Default: "Unnamed"
           2. Validation: Cannot be Empty
     b. Column 2: "Mechant" Header
        i. Each Row: Name of the Merchant the transaction was paid to the order of. (TextField)
           1. Default: "Unknown"
           2. Validation: Cannot be Empty
     c. Column 3: "Income?" Header
        i. Each Row: Whether the Transaction is an income or not (Toggle)
           1. Default: Unchecked (an expense)
     d. Column 4: "Spent" Header
        i. Each Row: Amount spent on the transaction in USD (TextField)

1. Default: $0.00
2. Validation: Must be a Non-negative Number
3. Format: USD Form (1.2345 -> $1.23)
e. Column 5: "Date" Header
    i. Each Row: Date the Transaction Occurred (DatePicker)
        1. Default: Current Date
        2. Validation: Cannot be Empty
- **Reconcile Button**
    1. TODO
- **Reconcile** will be performed after the month the user chooses to reconcile is over and all transactions are entered.
    1. Reconciling the budget finalizes the month that was reconciled and closes it for editing.
    2. Reconciling means that all transactions are accurate per the user's bank statement
    3. The user will click the reconcile button in the bottom right of the monthly budget page.
        a. The reconcile button will be labeled: "Reconcile".
        b. Once clicked the reconcile button will bring up a new screen titled "Reconcile".
            i. In the reconcile screen the user will have the option to select from prior months to reconcile via dropdown menu
            ii. The user will have the option to do a manual reconciliation or csv upload.
                1. To select which option the user will have a clickable toggle switch next to each option.
                2. The option that is selected will be filled in.
        c. CSV reconciliation will ask the user a series of questions regarding their csv file.
            i. The user will select if the csv file has a header.
            ii. If there is not a header the user will select which columns aligns with the merchant, date, amount and amount
            iii. If there is a header, YggFinance will read in the headers and attempt to match them with the appropriate fields.
                1. The user will confirm if the headers YggFinance has selected are correct.
        d. A manual reconciliation will require the user to "check off" each transaction entered.

      i. YggFinance will display a list of all transactions entered for the selected month with an empty box next to each one

      ii. If that user can match the data in the transaction to their bank statement the box is checked.

      iii. Once all transactions are checked the reconcile button is clicked

        1. Clicking the submit button brings up a pop up box with the message: "Are you sure you want to reconcile this month's transactions?

        2. The two options are buttons, one labeled "yes" the other box "cancel".

        3. Clicking cancel returns the user to the reconcile screen

        4. Clicking yes saves the month that the reconciliation was performed.

- **Net Worth Page** will guide the user and receive input to calculate their net worth.
    - Once the user's net worth data has been initialized, it will be displayed.
        1. Net worth will be displayed in U.S. dollar form at the top of the net worth tab
        2. Net worth will be displayed as a pie chart
            a. The pie chart will be divided by a percentage of each factor that goes into net worth from a the user's total net worth
    - The fields required to calculate net worth will be located below the displayed data.
    - The user will be asked enter the following data via textbox in U.S. dollar form, each separate textbox will be labeled as follows:
        a. Fields will be in two columns aligned next to each other
        b. Fields 4.2.2 - 4.2.7 will be in the left column
        c. Fields 4.2.8-4.2.13 will be in the right column
        2. "Real estate value"
        3. "Checkings accounts value"
        4. "Savings account values"
        5. "Retirement account values"
        6. "Automobile values"
        7. "Other Assets"
        8. "Remaining mortgages balance"
        9. "Consumer debt"
        10. "Personal loans"
        11. "Auto loans"
        12. "Student loans"
        13. "Other liabilities"
    - Once the data is entered the submit button will be clicked

1. The submit button will be located at the bottom right of the list of fields.
      ■ Clicking submit will update the displayed information
○ **Other Components:**
      ■ The **Invalid Input Message** will be displayed as a Dialog
            1. The **Dialog** will contain a DialogTitle, Dialog Content, and DialogActions
                  a. The **DialogTitle** should display the text "Invalid Input"
                  b. The **DialogContent** should display the reason why the input is invalid.
                  c. The **DialogActions** should contain a button displaying the word "Okay"

4. Diagrams:
      ○