**Das Finanzen**

Primary Goals:

1. Practice creating development documentation.
2. Practice using Unity before diving into my game.
3. Practice Test Driven Development.

Secondary Goals:

1. Build a functional Finance app for Google/Android phones.
2. Support “Day to Day” and “Reoccurring” expenses.
3. Display remaining allocated expenses for the month.
4. Include a Data/Graphs View.
5. Able to modify the Reoccurring and day to day without being forced to modify all.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**MAIN MENU**

|  |  |
| --- | --- |
| 1. BDAS = Before Daily Average Spend    1. Auto calculated your daily average spent so far from the start of the month to present. 2. Current Month    1. App should always auto load to this page on hard launch. 3. ADAS = After Daily Average Spend    1. Auto calculated from Total Remaining for the rest of the month divided by remaining number of days in the month. 4. Prev    1. Go back to previous months to view your historic data. Everything should be the same except Total Remaining will become “Total Saved”.    2. Data modification attempts should have a confirmation pop up appear. 5. Next    1. Move to the next months to pre-enter information. 6. Total Remaining    1. Tap on this to enter in the amount you are budgeting for. |  |

1. Day to Day
   1. Tap on any of the subcatagory names to go into the subcatagory menu.
   2. Total spent so far for the month will be displayed on the main page.
   3. “Remaining” can be tapped to toggle display to “Spent” and vice versa.
2. Reoccurring
   1. Tap on any of the subcatagory names to go into the subcatagory menu.
   2. Total spent so far for the month will be displayed on the main page.
   3. When Modifying Reoccurring items you will be presented with an option as to whether you intend to modify just this month or this + all future months.
   4. Spent is always displayed (No reason to show Remaining).
3. Go to Chart View

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SUB CATAGORY VIEW**

|  |  |
| --- | --- |
| 1. Back button to take you back to the main menu. 2. Tap here to adjust this catagories planned expenses for the month. 3. Tap on a date to modify. 4. Tap on a name to modify. 5. Tap on the Spent amount to modify. 6. Tap on to change the set color for this sub catagory. 7. Tap on to add a new Transaction to the list.    1. Date will be auto-populated. User must enter the other options. |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CHART VIEW**

|  |  |
| --- | --- |
| 1. Back button to take you back to the main menu. 2. $100 for each marker.    1. Not configurable. 3. Scrollable Field (left to right) 4. Legend can be scrolled down to view all colors. |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TECHNICAL DETAILS**

Unit Tests:

* Practice TDD with this project so there should be unit tests created for every class/method.
* Potentially look into end to end testing or integration tests.

Expenses:

* Keep it as only a Data Structure.
  + This can be saved to a file on the device.
* Decimal List Data Types.
* Reoccurring is the same as Day to Day but only with a Bool to determine the difference.
* Don’t create future months till the months start (it’s an opt in processes)
* Modifying a Reoccurring payment needs a method to trigger a confirmation box for occurrence vs series.

UI

* I think this is a good place to practice “events” in C#
  + Tap button -> Send event to UI manager
* Abstract Factory for the different Page types should be good here. However, I think each of them will need to be initialized upon hard launch even when they aren’t visible. Maybe I should practice having them hidden and/or initialized when called, that might be closer to how the game would work. I think the Flyweight design pattern is applicable here?
  + TODO - Look to see if the Flyweight pattern is worth practicing/using here.

Graph

* Surely this’ll take it’s own class.
* Need to look into drawing custom shapes. Then I just layer the shapes over each other.

Daily Average Spend (DAS)

* Might not even need to be a class. Could even be a static method as it is only a formula based on the expensed data.

Calendar

* There’s probably something built into .NET or Unity for this but I’ll have to research how to use the pre-built Calendar class.
  + The Sub-Catagory UI page needs access to this. Then I can save the Expenses data with the dates which should give me all the info I need to populate the other screens.
  + It’ll need to be called once during start up to get the current time.

Color Picker

* Unique UI class that won’t be like any other UI classes in this app. It’ll need to be very slimmed down with stripped capabilities. Just display a pop up box with a bunch of color squares for the user to select.

