SYMBOL-HOLIC

T1A3 - TERMINAL APPLICATION

I WAS GETTING BAD PAIN IN MY HANDS WHEN TYPING

Symbol-holic

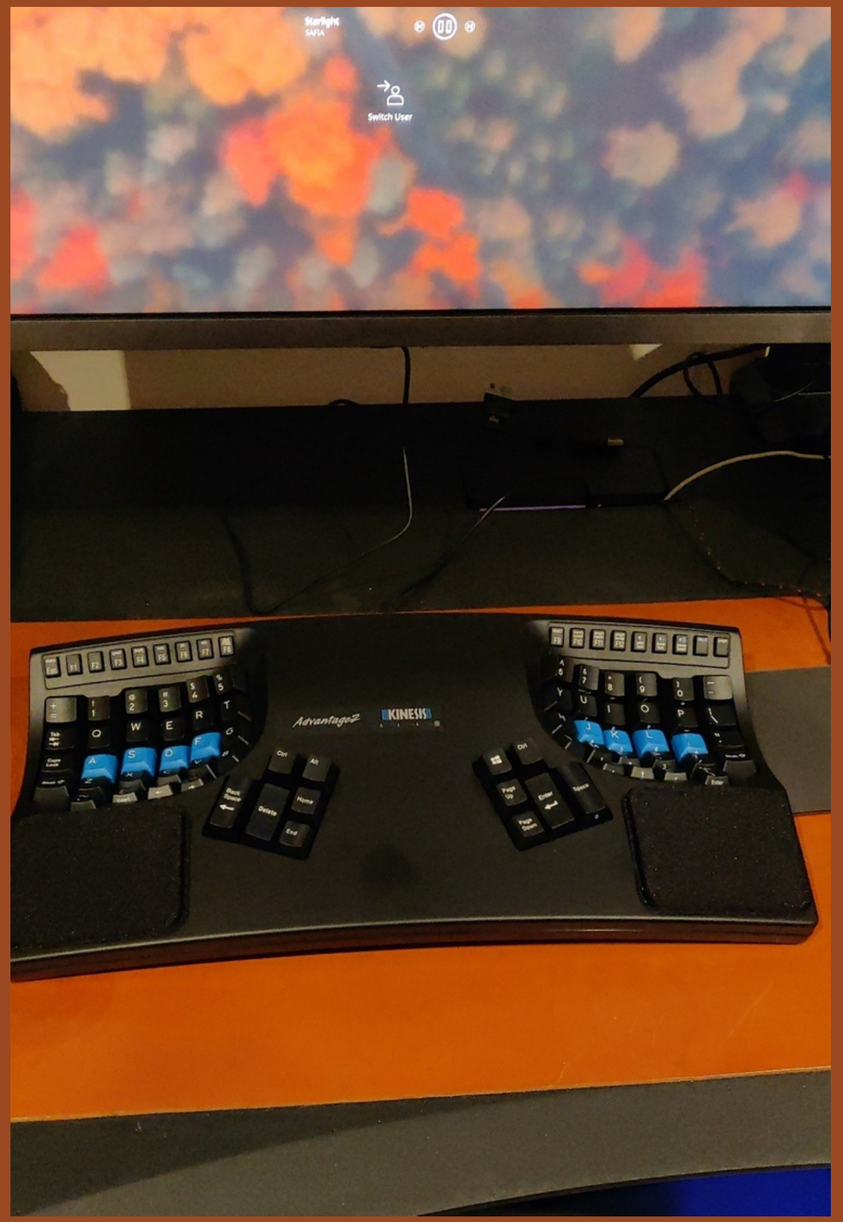
Sore right hand

Over-reliance on using my mouse, not using enough keybindings

QWERTY

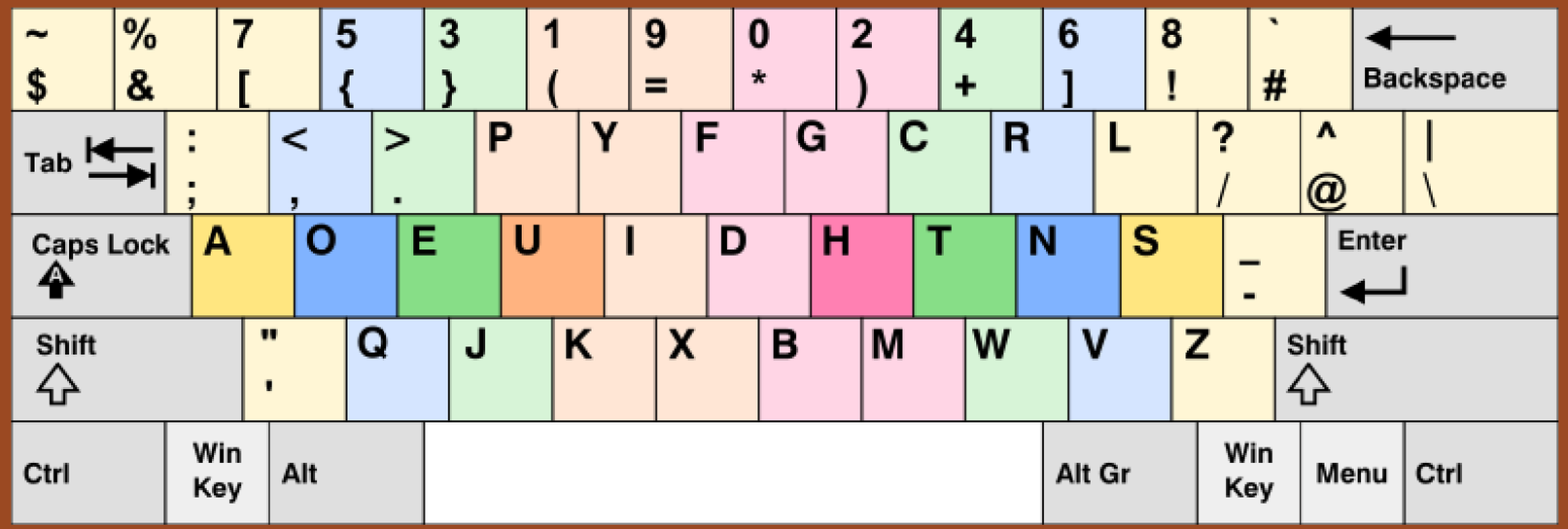
Although I could type quick on qwerty, it was hurting my hand, as it is a sub-optimal layout

SO I BOUGHT THIS



AND
TRANSITIONED TO
THIS

PROGRAMMERS DVORAK



ALPHABET CHARACTERS WERE EASY TO LEARN....

SYMBOLS WERE NOT.

SYMBOL-HOLIC WAS BORN

A simple yet effective way to train muscle memory when typing symbols.

Step 1

Flow Chart

Step 2

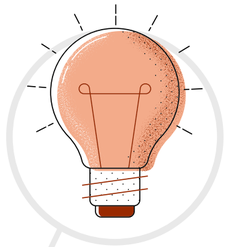
Trello

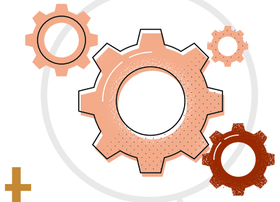
Step 3

Development plan

Step 4

Coding





Next steps

MY GOALS

For this assignment

SIMPLE + SPECIFIC

In design and code

REVIEW

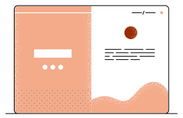
Keep the app simple, to allow time to do deep review and refactor

LEARN

Read docs and object oriented book

FEATURES

of symbol-holic



Two typing game modes

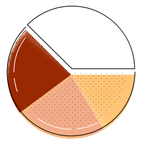
Randomized and targeted symbols

Store, modify and reset persistent data

Manipulating JSON file

Display menu for user

Utilizing TTY-Prompt



Display sorted persistent data

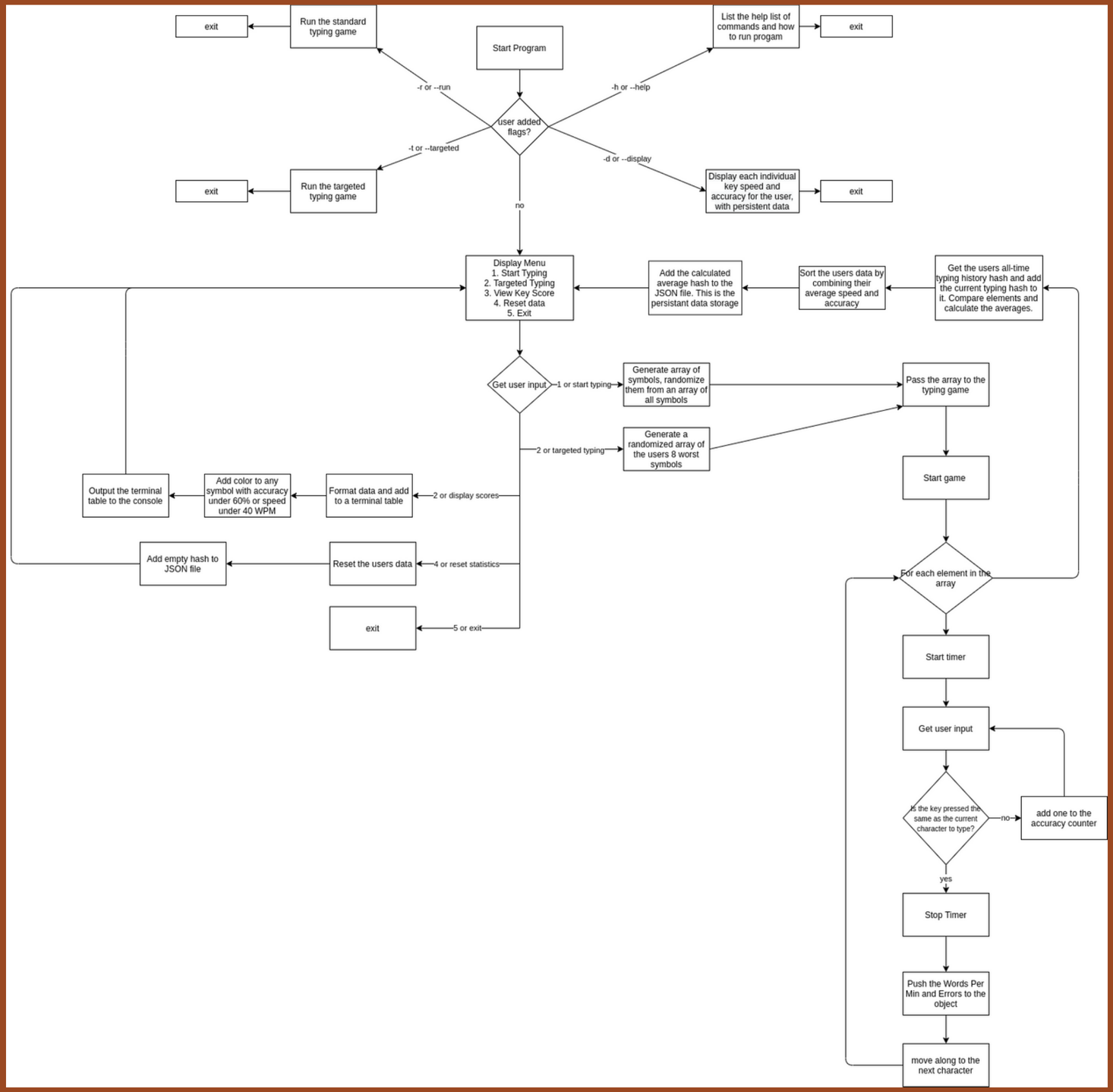
Utilizing terminal-table + colorize

STRUCTURE

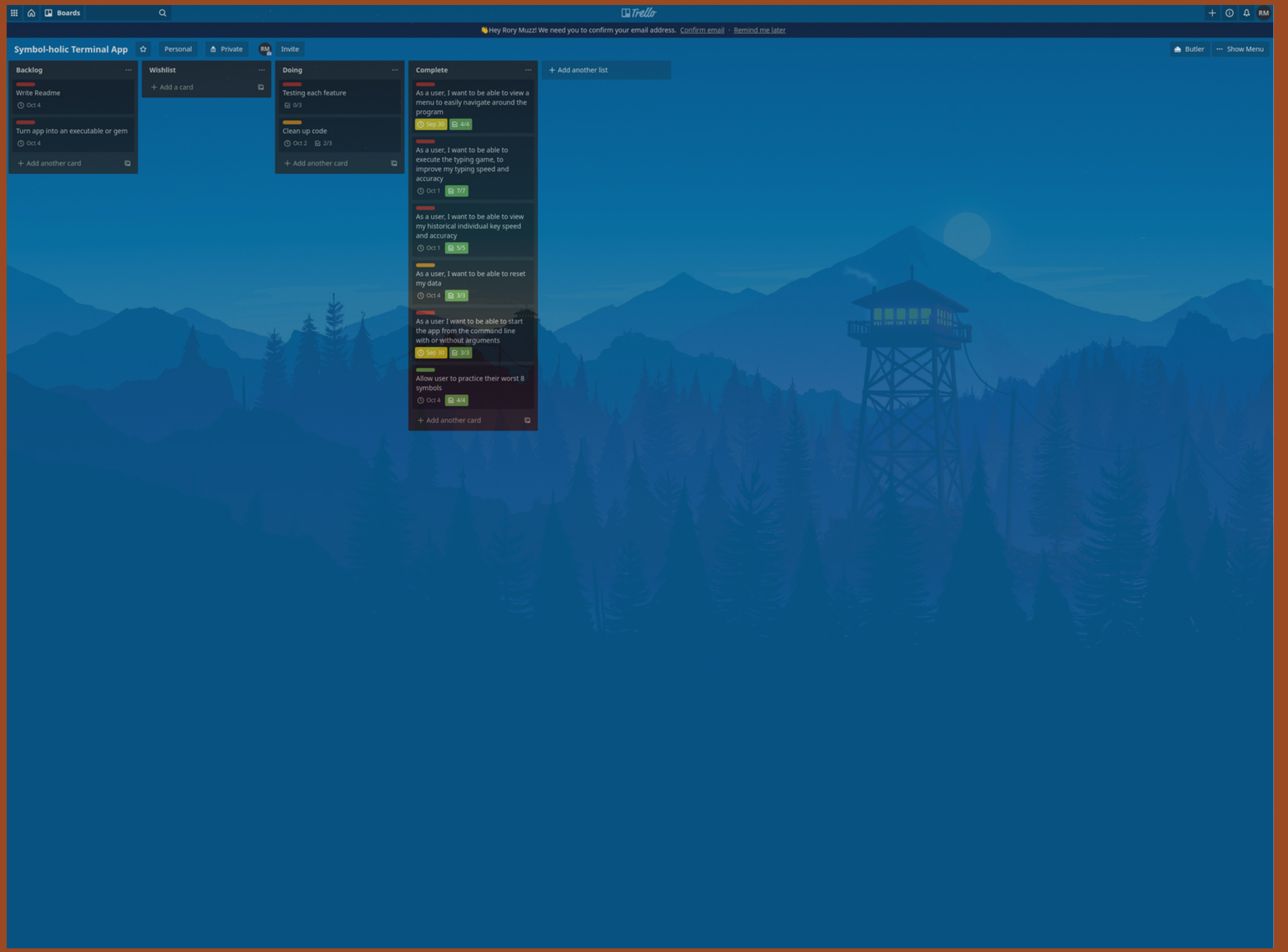
- Menu Class
- Statistics Class
- TypingGame Class

- Statistics Module
- Json file for storage

FLOW CHART


Start Program
user added flags?
-r or --run
Run the standard typing game
exit
-h or --help
List the help list of commands and how to run progam
exit
-t or --targeted
Run the targeted typing game
exit
-d or --display
Display each individual key speed and accuracy for the user, with persistent data
exit
no
Display Menu
1. Start Typing
2. Targeted Typing
3. View Key Score
4. Reset data
5. Exit
Add the calculated average hash to the JSON file. This is the persistant data storage
Sort the users data by combining their average speed and accuracy
Get the users all-time typing history hash and add the current typing hash to it. Compare elements and calculate the averages.
Get user input
1 or start typing
Generate array of symbols, randomize them from an array of all symbols
Pass the array to the typing game
2 or targeted typing
Generate a randomized array of the users 8 worst symbols
Start game
2 or display scores
Format data and add to a terminal table
Add color to any symbol with accuracy under 60% or speed under 40 WPM
Output the terminal table to the console
4 or reset statistics
Reset the users data
Add empty hash to JSON file
5 or exit
exit
For each element in the array
Start timer
Get user input
Is the key pressed the same as the current character to type?
no
add one to the accuracy counter
yes
Stop Timer
Push the Words Per Min and Errors to the object
move along to the next character

TRELLO


Boards
Trello
RM
Hey Rory Muzz! We need you to confirm your email address. Confirm email · Remind me later
Symbol-holic Terminal App
Personal
Private
RM
Invite
Butler
Show Menu
Backlog
Write Readme
Oct 4
Turn app into an executable or gem
Oct 4
Add another card
Wishlist
Add a card
Doing
Testing each feature
0/3
Clean up code
Oct 2
2/3
Add another card
Complete
As a user, I want to be able to view a menu to easily navigate around the program
Sep 30
4/4
As a user, I want to be able to execute the typing game, to improve my typing speed and accuracy
Oct 1
7/7
As a user, I want to be able to view my historical individual key speed and accuracy
Oct 1
5/5
As a user, I want to be able to reset my data
Oct 4
3/3
As a user I want to be able to start the app from the command line with or without arguments
Sep 30
3/3
Allow user to practice their worst 8 symbols
Oct 4
4/4
Add another card
Add another list

REVIEW

Fun Scale

★★★★★/★★★★★

Confusion of classes

I need to continue to read up about using classes. Most of my classes had no attributes, they are pretty much a container for methods.

Debugging in ruby is hard

Compared to a little bit of Javascript I have done, Ruby was challenging to debug. I never really got on with byebug, it's not visual enough for me. I used rdebug-ide instead which helped a lot, but it is a bit buggy and crashes terminal a bit.

Starting with flowchart was worth the effort

Instead of just jumping straight in, putting my thoughts into a flow chart helped design my whole app. Although it veered a bit from the first design, majority was the same.

LOGIC

Walkthrough