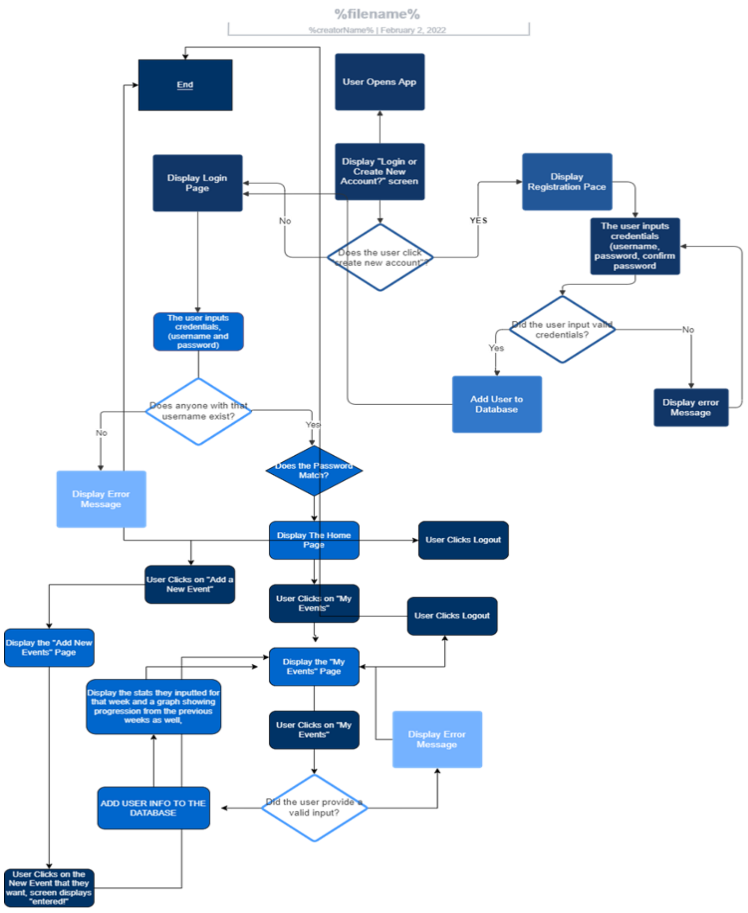
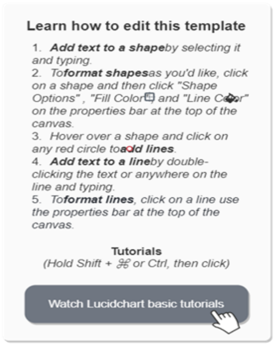
Criterion B: Design

Vidhya Narayanan
Pierson
Period 3 Computer Science
22 March 2022

Relevant Flowcharts & Diagrams:

Flowchart v1

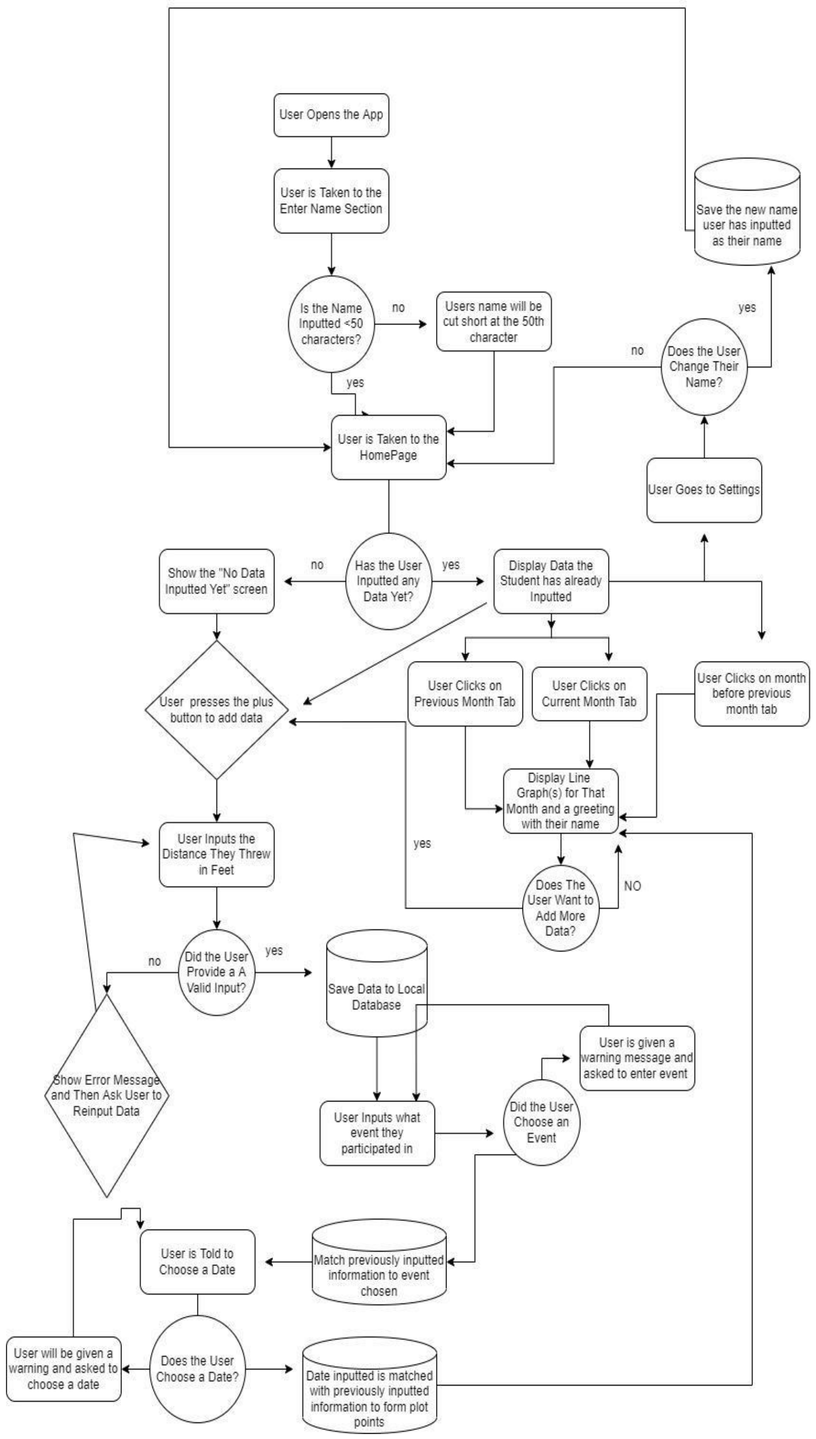

%filename%
%creatorName% | February 2, 2022
End
User Opens App
Display "Login or Create New Account?" screen
Display Login Page
Display Registration Pace
No
YES
Does the user click create new account?
The user inputs credentials (username, password, confirm password
Did the user input valid credentials?
Yes
No
The user inputs credentials, (username and password)
Add User to Database
Display error Message
Does anyone with that username exist?
No
Yes
Does the Password Match?
Display Error Message
Display The Home Page
User Clicks Logout
User Clicks on "Add a New Event"
User Clicks on "My Events"
User Clicks Logout
Display the "Add New Events" Page
Display the "My Events" Page
Display the stats they inputted for that week and a graph showing progression from the previous weeks as well.
User Clicks on "My Events"
Display Error Message
ADD USER INFO TO THE DATABASE
Did the user provide a valid input?
User Clicks on the New Event that they want, screen displays "entered!"


Learn how to edit this template

1. ***Add text to a shape*** by selecting it and typing.
2. *To* ***format shapes*** *as you'd like, click on a shape and then click "Shape Options" , "Fill Color" and "Line Color" on the properties bar at the top of the canvas.*
3. *Hover over a shape and click on any red circle to* ***add lines****.*
4. ***Add text to a line*** *by double-clicking the text or anywhere on the line and typing.*
5. *To* ***format lines****, click on a line use the properties bar at the top of the canvas.*

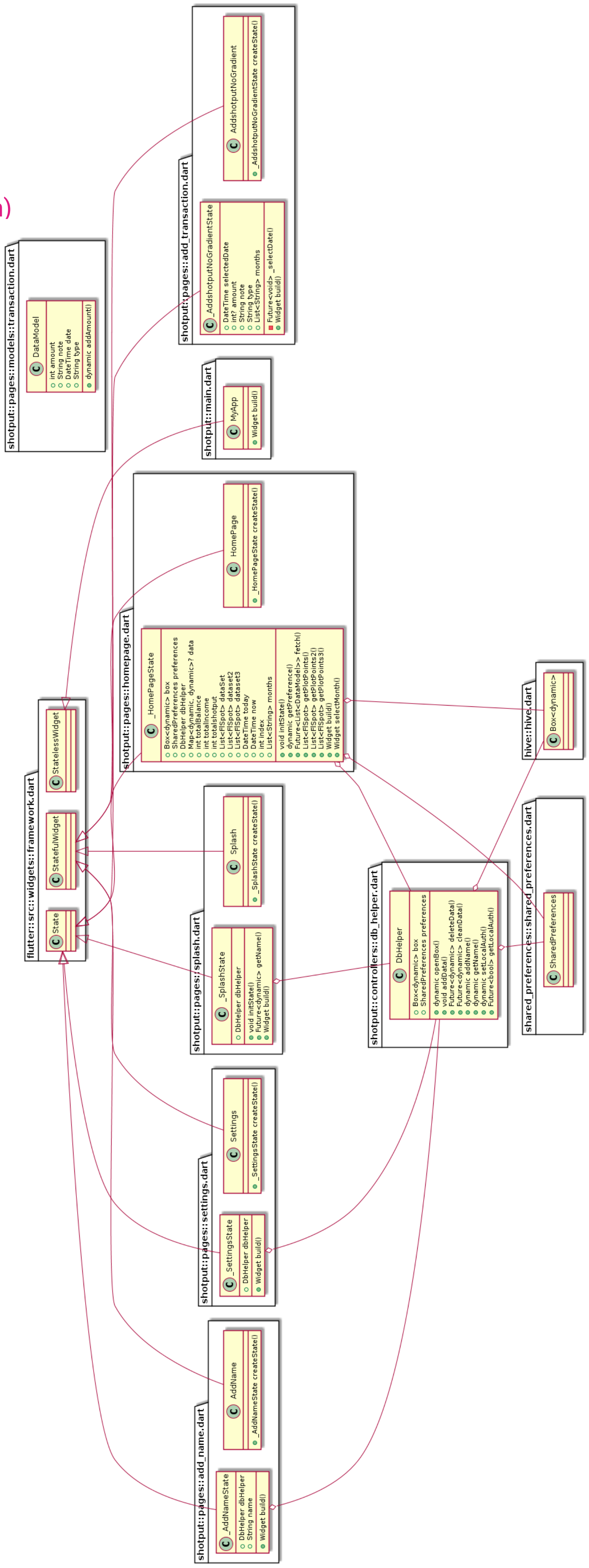
Tutorials
(Hold Shift + ⌘ or Ctrl, then click)

Flowchart v2


User Opens the App
User is Taken to the Enter Name Section
Is the Name Inputted <50 characters?
no
Users name will be cut short at the 50th character
yes
User is Taken to the HomePage
Save the new name user has inputted as their name
yes
no
Does the User Change Their Name?
User Goes to Settings
Show the "No Data Inputted Yet" screen
no
Has the User Inputted any Data Yet?
yes
Display Data the Student has already Inputted
User presses the plus button to add data
User Clicks on Previous Month Tab
User Clicks on Current Month Tab
User Clicks on month before previous month tab
Display Line Graph(s) for That Month and a greeting with their name
yes
NO
Does The User Want to Add More Data?
User Inputs the Distance They Threw in Feet
no
Did the User Provide a A Valid Input?
yes
Save Data to Local Database
Show Error Message and Then Ask User to Reinput Data
User is given a warning message and asked to enter event
User Inputs what event they participated in
Did the User Choose an Event
User is Told to Choose a Date
Match previously inputted information to event chosen
User will be given a warning and asked to choose a date
Does the User Choose a Date?
Date inputted is matched with previously inputted information to form plot points

UML Diagram:

(Chart is Easier to Read if You Zoom In)



UML Diagram + Algorithms:

In the UML diagram, we can see our “framework” package which operates at the higher-most level of the app. This package includes our “State” class, “Stateful Widget,” and “Stateless Widget.” In flutter, a State is information that can be read simultaneously while and after a widget is built and can experience change during a widget’s lifetime. In flutter, “widgets” are tools used towards constructing an app’s UI, or user interface. A Stateful Widget is one that can be updated or changed depending what the user does with the app, while a Stateless Widget is one that remains static.

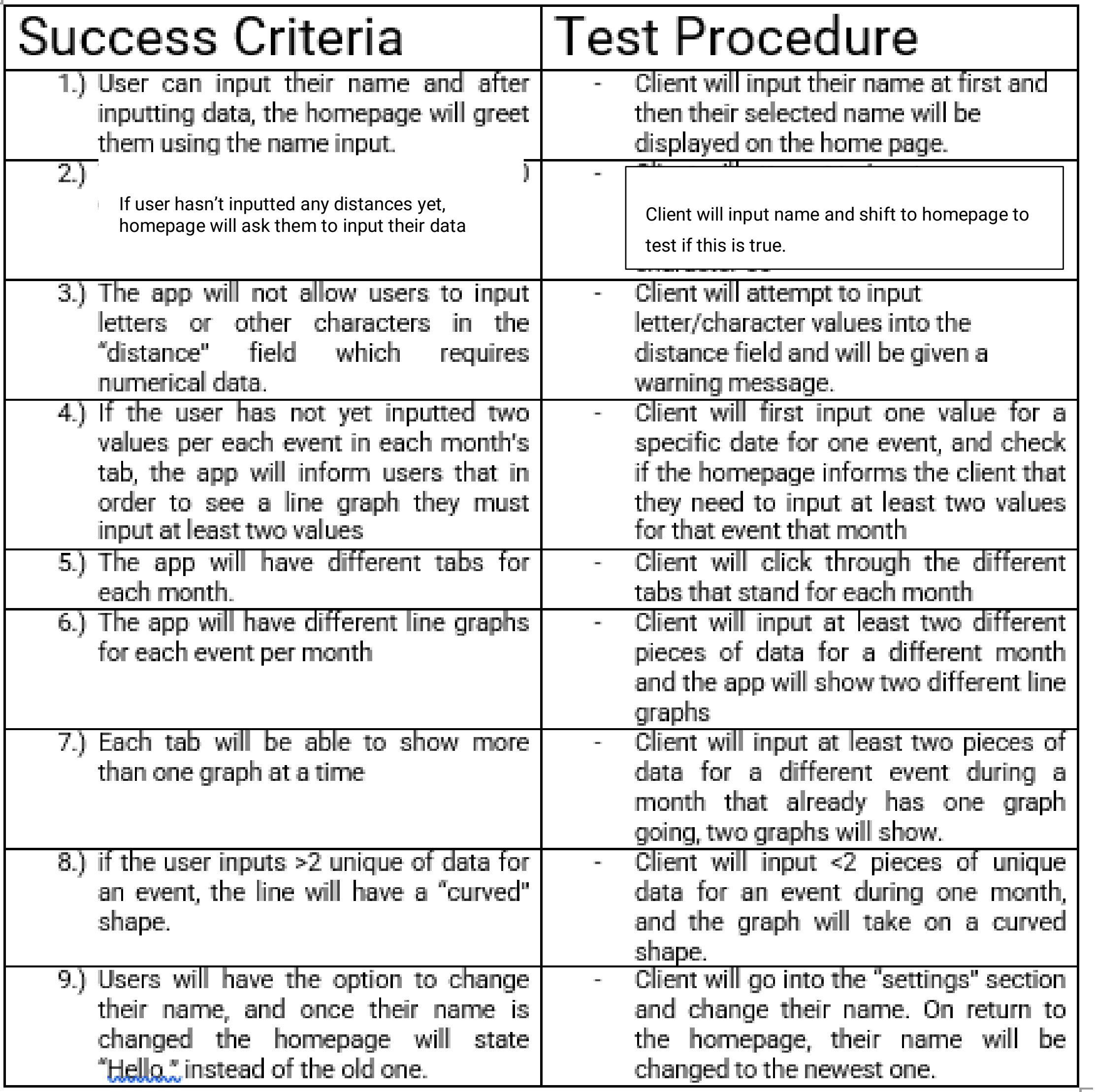
Our main class inherits from the Stateless Widget as main.dart’s function will be the same each time. When looking at classes that inherit from our Stateful widget, we can start off with our add_transaction/new_value (will probably change later) page, which includes the _Addshotputnogradient and Addshotput class, which are responsible for collecting information from the transaction screen, which changes because of it’s dependence on the live calendar function. There is also the _AddHomepageState class. One of the complex UI components of the Homepage was including tabs for each month which I did by combining a string array with month names through an Flspot that filters out event data from each day of the month by month selected The selectmonth() method created the button the users press to get to each month’s tab. Each month would be saved at either index 0, 1, or 2, and the months at each index would depend on which month of the year it is currently. The LineChart in each tab called information from the Flspot to create separate line graphs for each event’s results per month

On the tertiary level we have our database_helper page that includes the dbhelper class, which is designed to process the information the user inputs in order to be stored in the database (Hive Database.) This class inherits from the classes homepagestate, addnamestate, settingsstate, and splash state. Finally, at the fourth level we have our sharedpreferences class and our hive class, which both exist to store the user input in a dynamic box so that it can be stored in our Hive Database.

Data Structures:

Since the database I am using is Hive, all data will be stored in the HDFS path, or /user/hive/warehouse. As previously mentioned, data will be properly stored through the use of a database helper class that will store all information into boxes. The two information types the database will need to process in the case of my app would be integers and arrays.

Test Plan:



Success Criteria	Test Procedure
1.) User can input their name and after inputting data, the homepage will greet them using the name input.	- Client will input their name at first and then their selected name will be displayed on the home page.
2.) If user hasn't inputted any distances yet, homepage will ask them to input their data	- Client will input name and shift to homepage to test if this is true.
3.) The app will not allow users to input letters or other characters in the "distance" field which requires numerical data.	- Client will attempt to input letter/character values into the distance field and will be given a warning message.
4.) If the user has not yet inputted two values per each event in each month's tab, the app will inform users that in order to see a line graph they must input at least two values	- Client will first input one value for a specific date for one event, and check if the homepage informs the client that they need to input at least two values for that event that month
5.) The app will have different tabs for each month.	- Client will click through the different tabs that stand for each month
6.) The app will have different line graphs for each event per month	- Client will input at least two different pieces of data for a different month and the app will show two different line graphs
7.) Each tab will be able to show more than one graph at a time	- Client will input at least two pieces of data for a different event during a month that already has one graph going, two graphs will show.
8.) if the user inputs >2 unique of data for an event, the line will have a "curved" shape.	- Client will input <2 pieces of unique data for an event during one month, and the graph will take on a curved shape.
9.) Users will have the option to change their name, and once their name is changed the homepage will state "Hello." instead of the old one.	- Client will go into the "settings" section and change their name. On return to the homepage, their name will be changed to the newest one.

Word Count: 446