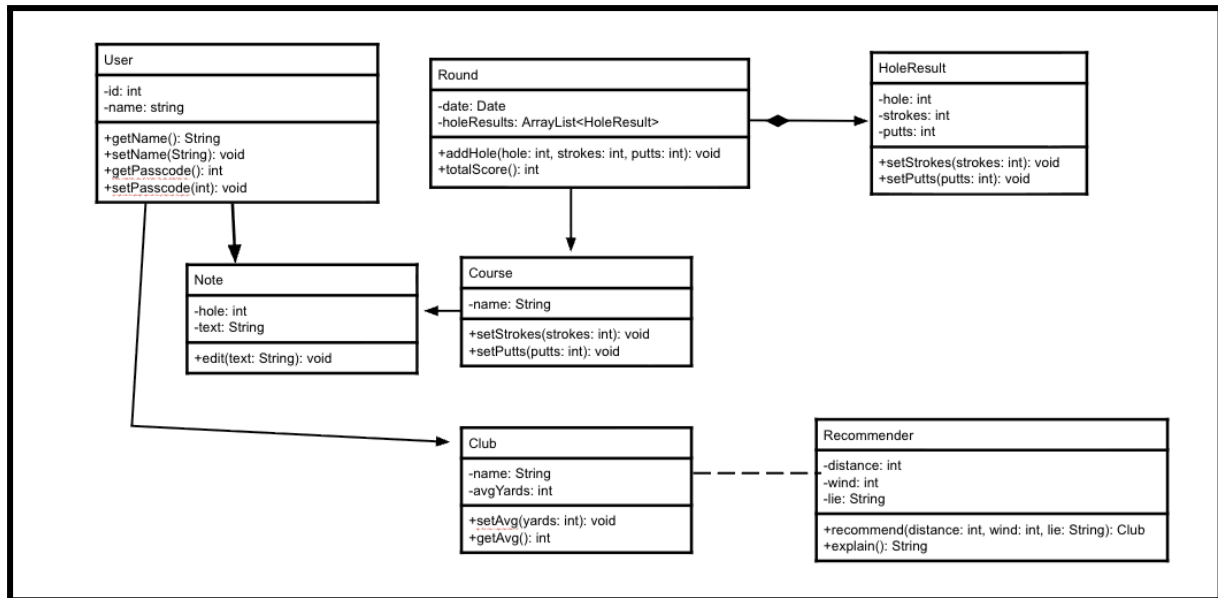
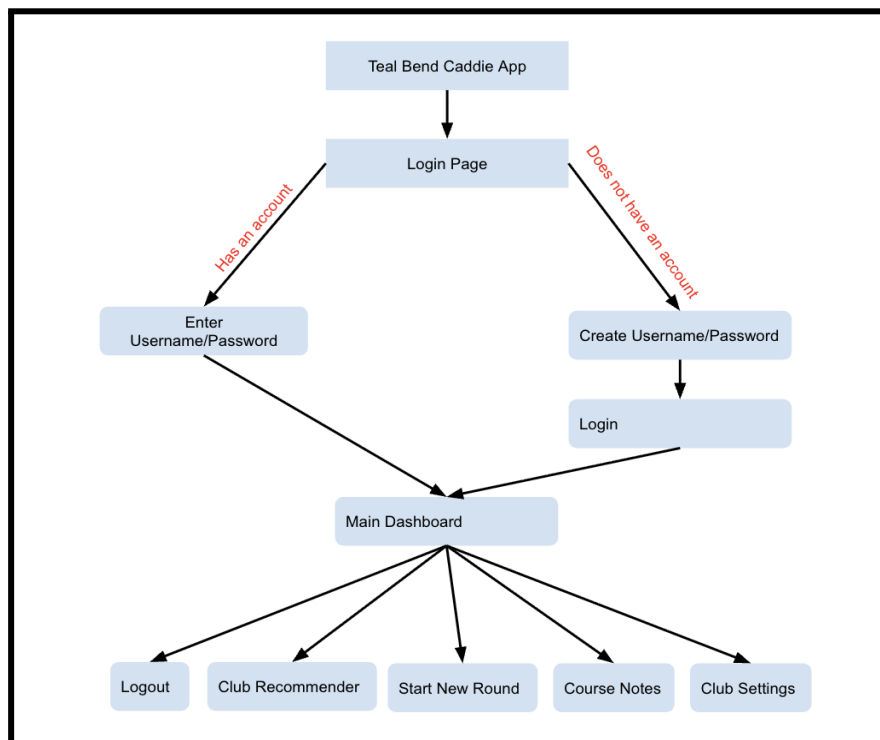


Criterion B: Design



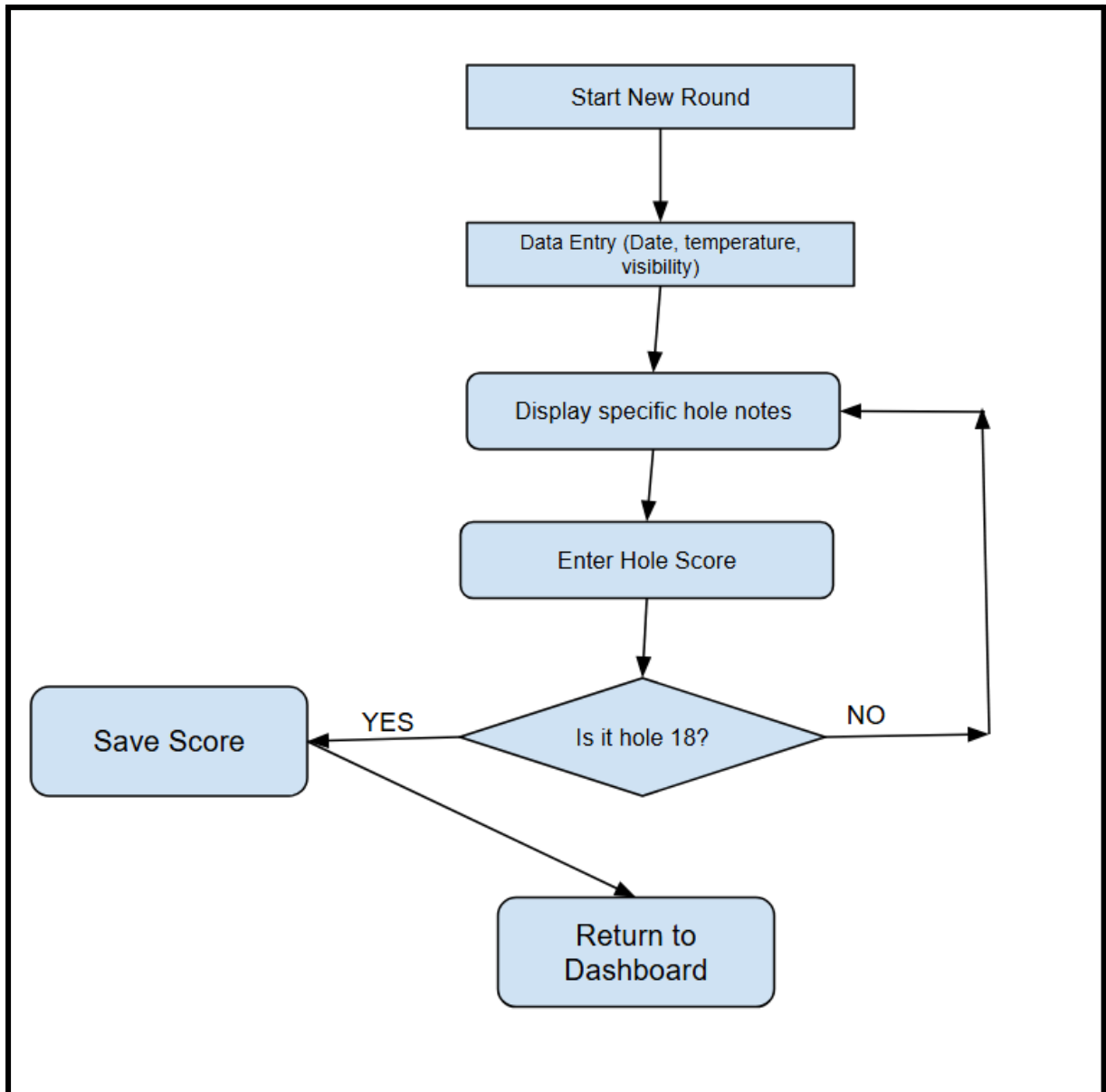
Program Outline:

- This is a UML diagram that outlines my entire program. It includes all my classes and shows how each class is related within my program.



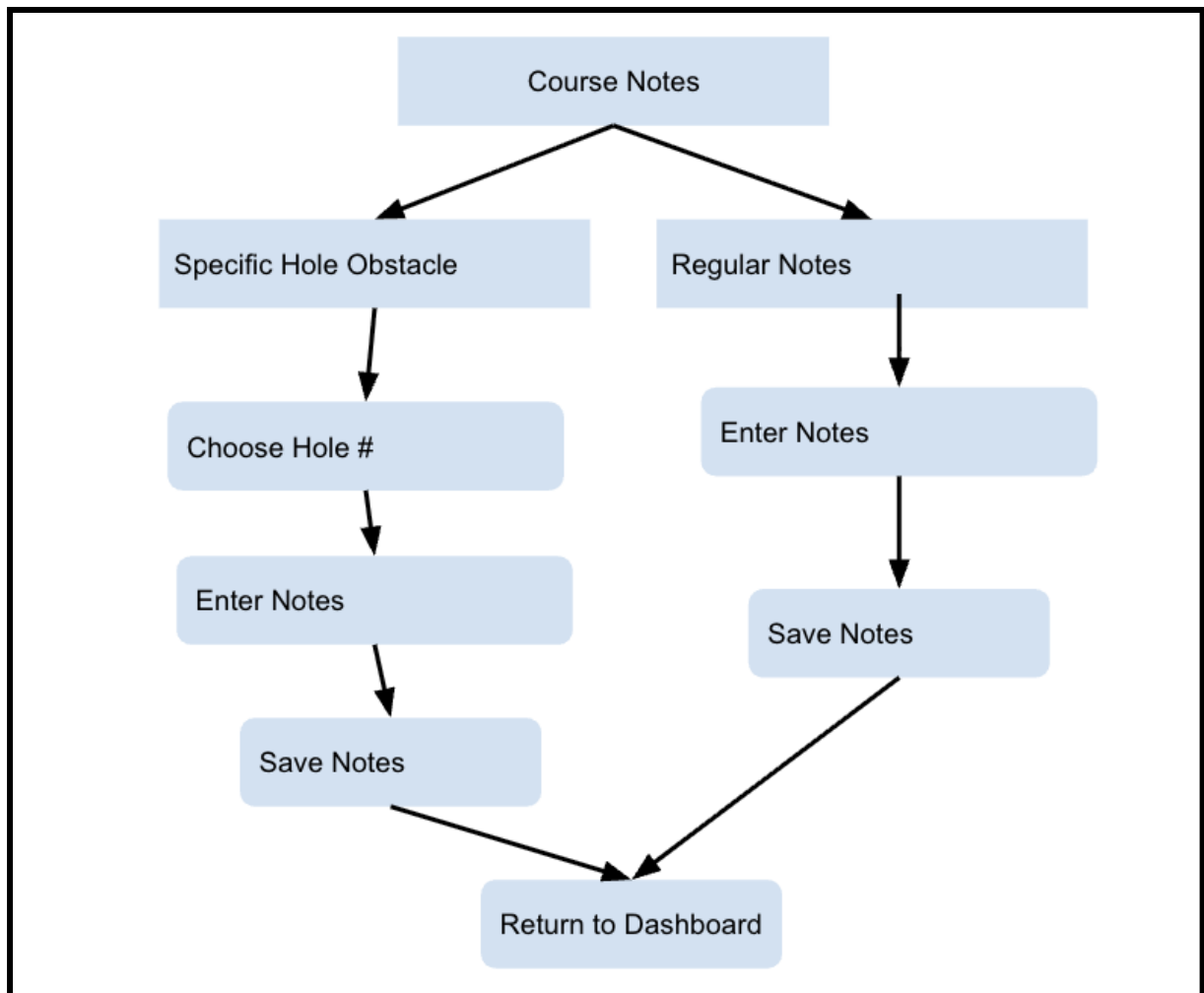
Program Navigation Flowchart:

- This is a flowchart that shows how the user will view my program.



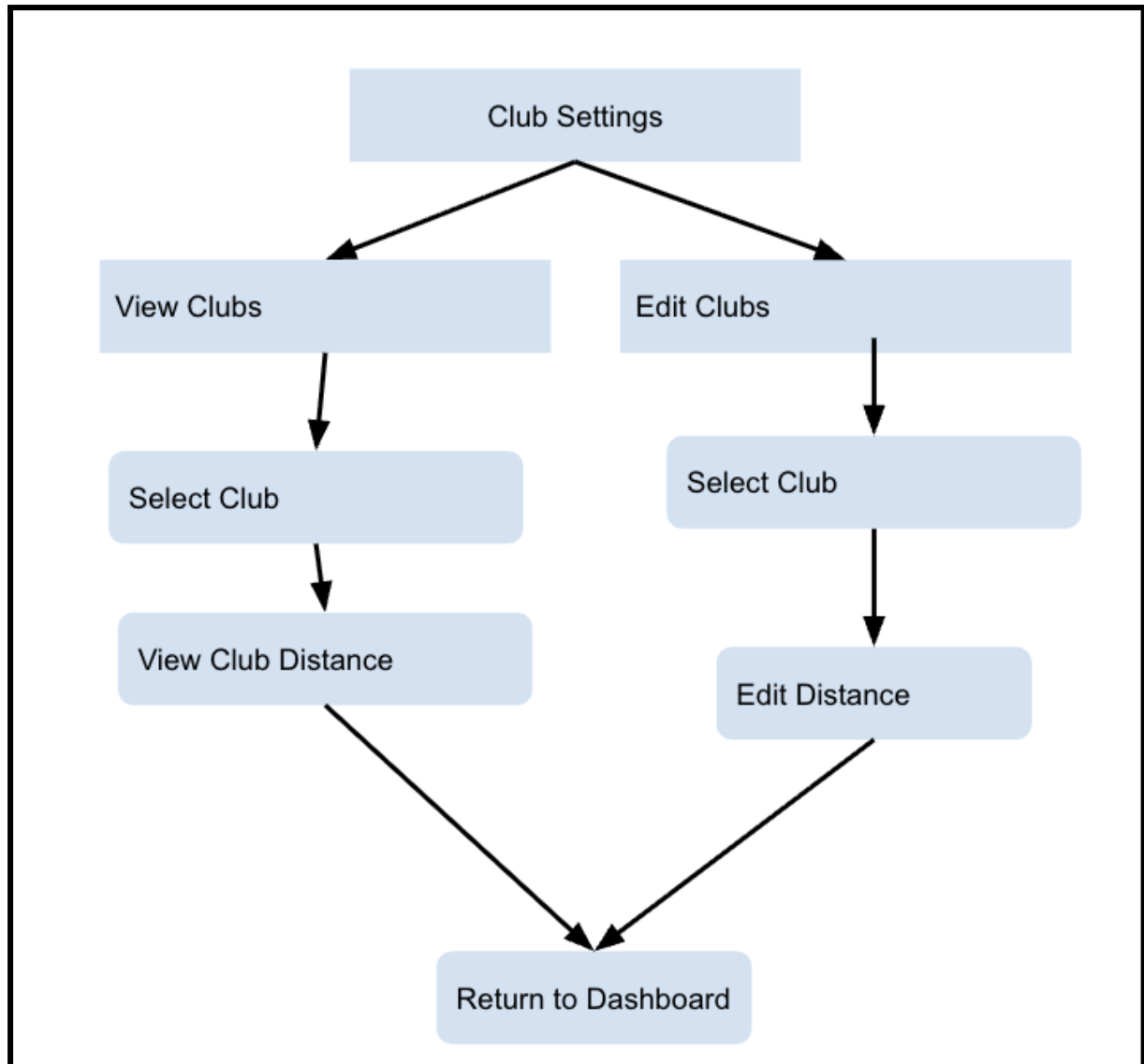
Start New Round Flowchart:

- This flowchart shows how the user will be able to navigate through a golf round
- The user will keep viewing their notes and recording their scores until they are done with their 18 hole round



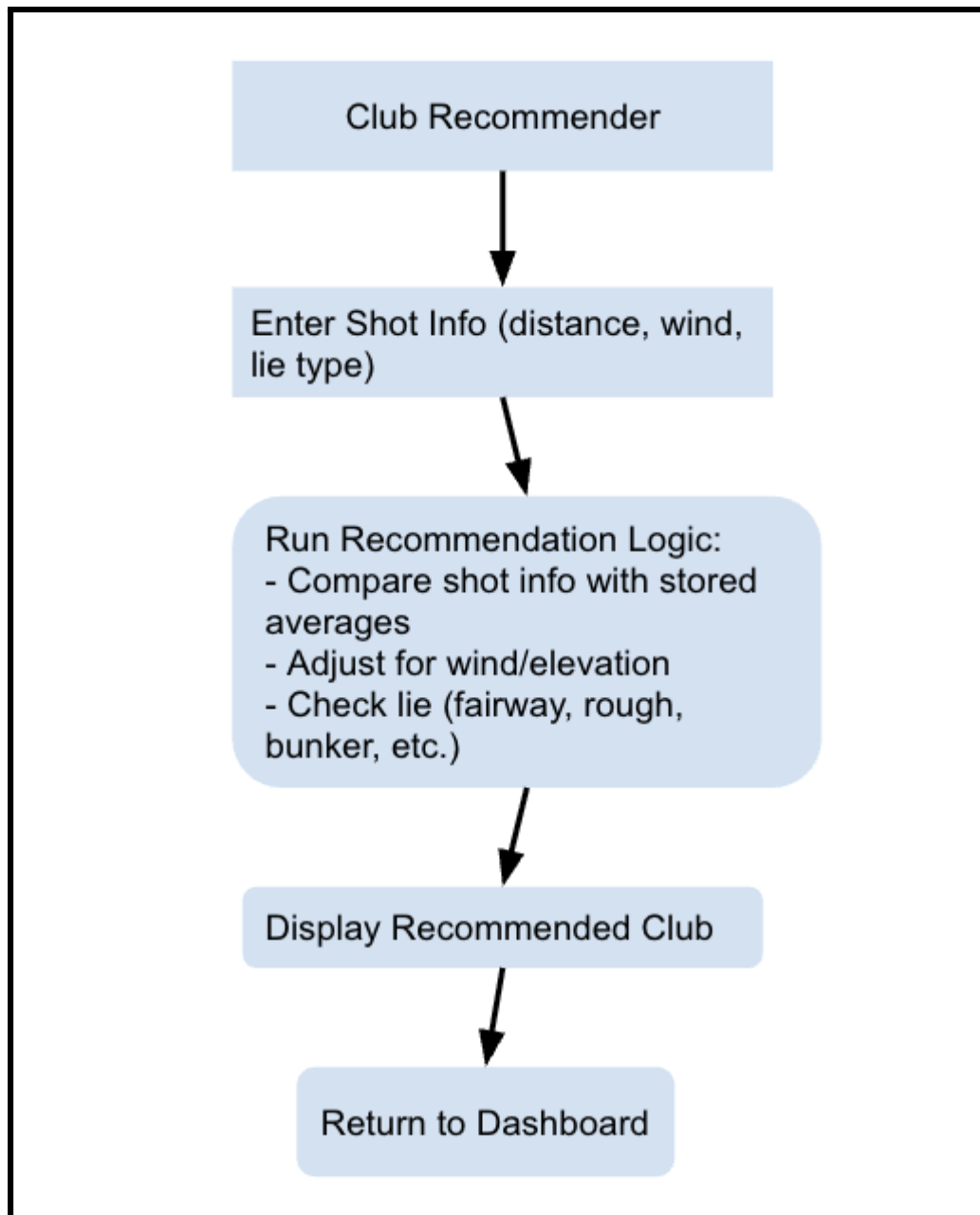
Course Note Feature Flowchart

- This is the logic behind the note taking system in my caddie app
- You can either have hole-specific notes or general notes
- These notes will be saved for later use during a real round



Club Settings Flowchart

- This is the logic behind the club settings in my program
- A user can view their clubs and edit their stock distances as they improve their golf game



Club Recommendation System Flowchart

- This is how the recommendation system will appeal to a user
- The user will be asked to enter information and then they will be provided a recommended club based on a calculation



UI Annotations

- **Club Settings**
 - On this screen, you can see your average distances for each club. You are also able to edit your average distances based on whether you improve or get worse. Your updates would be saved to your profile.
- **Club Recommender**
 - On this screen, the user can input distance, wind speed, lie type, etc. Then, an algorithm will recommend a club based on the given inputs and the users average distances.
- **Teal Bend Caddie Home Screen**
 - On this screen a user can start a new round by pressing the Start Round button. The user will be able to input their stats for each hole and also review notes they left on each hole. The navigation bar at the bottom allows you to quickly access the home page, remember, notes, and club settings.
- **Course Notes**
 - On this screen, the user can choose a hole that they want to save specific notes for. Once written, the user can click save to save their notes directly to their profile. These notes will appear when they get to that hole during their round.

Club Recommender Pseudocode:

```
Start

  Input Distance

  Input windSpeed

  Input lieType

  bestClub = null

  effectiveDistance = distance

  IF lieType = "Fairway"

    effectiveDistance *= 1.05

  IF lieType = "Light Rough"

    effectiveDistance *= 1.10

  IF lieType = "Heavy Rough"

    effectiveDistance *= 1.15

  IF lieType = "Sand"

    effectiveDistance *= 0.5

  IF lieType = "Tee"

    effectiveDistance *= 1.05

  FOR every club in userClubs

    if club.avgDistance is the closest to effectiveDistance

      bestClub = club

OUTPUT bestClub
```

Test Plan

Criteria Tested	Test Input/Action	Example	Expected Output
1 - The system will allow the user to input their average distances, which must be saved correctly in a CSV file.	When first using the program, the user enters his stock yardages	Driver: 250, 7-iron: 160, Pitching Wedge: 90, etc.	The program correctly saves the users stock yardages in a CSV file.
	The user views their clubs and stock yardages.		The CSV file displays every club in the user's bag along with the user's current stock yardages.

2 - The system will allow a user to update their average distances per club and save those to the CSV file.	The user chooses a single club from their clubset and changes the yardage.	Original–Driver: 250 Updated–Driver: 290	The program saves the users new yardage for the club in the clubset CSV file.
	The user views their clubs and stock yardages.		The CSV file correctly displays the most up to date stock yardages in the clubset.
3 - The club recommender algorithm will use different factors (like wind, distance, and terrain) to determine an optimal club to use.	The user will input different factors in the club recommender feature of the program.	Wind: 25 mph Distance: 146 yards Terrain: Rough	The correct club (most likely a 6 or 7 iron) will be recommended to the user based on the algorithm.
4 - The system will simulate an 18-hole round and save all historical scores in a CSV file.	The user will start a 18 hole round and input their score for each hole.	Score for hole 7: 4 Score for hole 8: 3	The scores will be added up and the total score will be provided to the user at the end of the round.
	The user views their historic scores to see their improvement.		The CSV file correctly displays the user's score history.
5 - The system will allow a user to edit general notes and hole-specific notes.	The user adds a note on a specific hole after they notice a mistake they keep making on it.	“Keep aiming over the pond; don’t go around it”	The note will save for that hole and will be shown to the user every time they play that specific hole.
6 - The system will display notes for each hole during the round so the user can be reminded.	The user is playing a round and reads their notes for hole 7 before they play it.	“““Keep aiming over the pond; don’t go around it”	The user is reminded of this mistake and tries to aim over the pond to score better.
7 - The system will have a simple UI interface that works efficiently without crashing.	The user navigates through all the pages of the program.		Methods to navigate through the program are clearly labeled making it an efficient process. UI buttons

			are clear to read and understand for all.
8 - The system will keep track of each user's data (notes, club distances, historical rounds) through CSV files.	The user checks their notes, club distances, and historical rounds.		All the data for the user is accurately saved on CSV files and correctly implemented into other parts of the program.