

## Criterion E: Evaluation

### Evaluation of Success Criteria

*Success Criteria (This is just for reference, so it will not be counted in the word count. See Criterion A for where it is originally written.):*

1. A page to calculate the most prevalent trends in a specific game map.
  - a. It will calculate trends based on each specific time matches have been played on a specific game map.
2. Transfers information from Google FireBase to a List and show on the “Match History” page
  - a. Match History page shows the matches from Google FireBase.
3. Allows for sorting the Match History page
  - a. User can use a switch on the page to sort Match History from newest to oldest or oldest to newest
4. Allows for game matches to be added to Google FireBases
  - a. Fields can be filled out
  - b. A switch button to determine whether the match is a win or loss
  - c. Button allowing a match to be uploaded to the FireBase
5. A user-friendly UI that is easy to follow.
  - a. All the buttons on the program are easy to follow with simple descriptions of what they do.
6. A working log-in system.
  - a. When the user’s username is entered, the user will be allowed into the program
  - b. (Error Handling) When anything else is entered and trying to log in, it will not let the user past the first page and an error message will show.
7. Saves and loads information for the app to Google FireBase
  - a. Uses FireBase with an implemented hierarchical system to store data that is accessed and edited through the app.
8. Displays game map descriptions.
  - a. When a map is entered, a description of the map is displayed on the page.

*Success Criteria Evaluation:*

Success Criteria	Evaluation
1	The trends calculated were useful to my client. My client said, “Wow that’s so cool. I always knew Jett was my agent on this map, and it is reassuring to see the stats back it up.”
2	The Match History list worked and had a functional UI. My client expressed his satisfaction by saying, “Love how clean [the match history list] is. I can just scroll through really easily.
3	The Match History is sorted effectively using a stack to reverse it. After clicking the switch the sort the list, my client said “This is fast and easy. I like it.”
4	The Create Match Page is simple and works well to add matches to FireBase. After testing it out, my client said, “That worked wonderfully. That is easy to use.”
5	The UI is easy-to-follow and has simple, understandable buttons. After using the app, my client was asked to rate the UI’s user-friendliness on a scale from one to ten, and he replied “It’s a ten. I didn’t have to ask you for any directions when using it. I like how easy it is to use.”
6	The Log-In Page works well by only allowing users go to the Home Page if the correct username is added. If an incorrect one is tried, an error message shows up. After using it, my client said, “The log-in works well.”
7	I used a hierarchical data structure on FireBase that stored data and was accessed from within the app to save. When discussing FireBase, my client said, “It’s important that my data’s fast, and FireBase does that.”
8	The app displays the corresponding description to the map inputted by the user. This was useful to my client to remind him of key features for each map. After entering in a map, my client said, “And check out the map description. That’s a cool feature.”

*(SEE APPENDIX FOR TRANSCRIPTS OF LAST MEETING WITH CLIENT AND ADVISOR)*

**Recommendations for Further Development**

My code is set up to further develop. The method names within classes are easily named and easy to follow: For example, fillMatchList() fills a LinkedList of Match objects that are from

transferred from Firebase. My code also has comments talking through what confusing code does: For example, comments talked through how it creates a match and adds it to Firebase. To further develop the app, I would add more interpretations to display in the “Advice” page. This page has extra space so that it can expand to other factors than just using the map and agents. It can take into account for kills, deaths, assists, and teammates played with. Another thing that could be done is make the program more visually appealing: Right now, it is clean and simple, using white and blue, but one possible extension is it could include images to display the characters in Valorant. Lastly, more information attached to matches could be added, such as their kills, deaths, and teammates played with. This could be done through adding String variables to the “Match” class and allow them to be added into Firebase as well. These areas for further development can address my client’s future needs, but the program at its current state works for what he communicated he needed done.

Word Count: 219