Architecture, Design and UI

Team Star

October 1, 2024

1 Architecture

The architecture that we are planning on using for our project is web-based MVC, which will enable us to utilize software to allow for ease of use and moderation when considering a widely accessible application and vast user base. With this, we also plan on using a top-down class structure with an emphasis on isolated classes and functions, dissecting each individual data structure down to its base elements and keeping those elements purely isolated, with as few dependencies as feasible when developing an interconnected card game with multiple users such as this. Our only concern when considering this approach is how to best mesh the inevitable dependency when considering the lobby and server systems.

2 Technology

The technology we will be utilizing for this project will be primarily based inside of Java, using Java adjacent plugins, tools, and server hosting that were prominently displayed in the onboarding in the first individual assignment. The more notable technologies we can name aside from any independent choices that may occur during individual coding sessions will be focused on Firebase and the server database interactions and how we will have data persistence in regard to our game. For our testing frameworks, we have decided to go with JUnit as we feel it has the most readily available documentation alongside our prior experience with it and feel comfortable including it in our project.

3 Persistent Data

We have determined that all of our long-term data will be stored inside of a database utilizing Firebase, as a few of our members feel comfortable using and manipulating the data and structures surrounding it. We plan on having a collection of data packaged around each individual username which we use as a key to find and compare all data together when authenticating passwords or comparing prior and current data. We plan to include in persistent data

information such as name, age, login email, game stats, and avatar choice. Also since roles matter, such as admin, we will try to include that in the persistent data as well.

4 Coding Standards

Our team has decided to follow a very strict coding standard in regards to committing code and upholding certain degrees of readability as well as functionality, such as the commitment to never push code that does not work, but to rather only push commented-out code if pushing it for posterity is absolutely necessary, alongside using singular names and conventions, ie not using the same name multiple times with vague meaning around it, but using an entirely unique variable name if it is absolutely necessary to create another independent variable. We have also made the decision to state the entire purpose of every commit with thorough documentation of our purpose and functional changes when committing to GitHub, to ensure no one is left in the dark when functions or classes are changed by others.