KOÇ
ÜNİVERSİTESİ
COMPUTER ENGINEERING

COMP132: Advanced Programming
Programming Project Report

NBA Fantasy Game Project

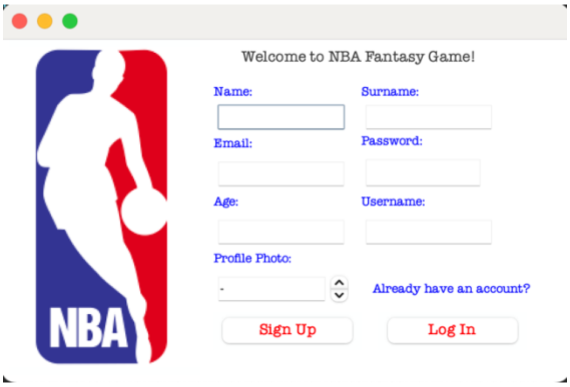
Faruk SARTIK - 72090

Fall 2023

Part 1

General Info & Application usage information:

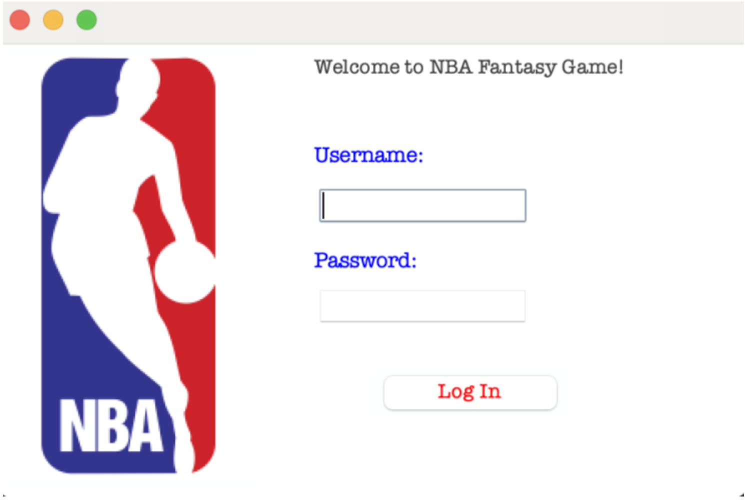
This NBA Fantasy game gives people a taste of a superficially enjoyable little NBA simulation. You must register for the game by entering your personal information such as name, surname, e-mail, password, age, username and profile photo selection on the login page.



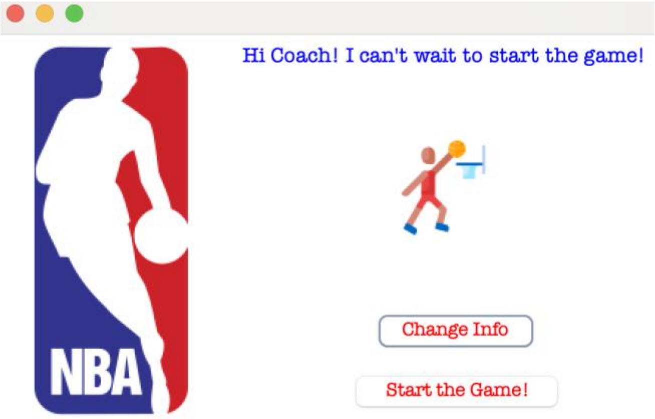
After successfully registering for the game, you can start the game by logging in to your account with your username and password.

These are the saved users to the game:

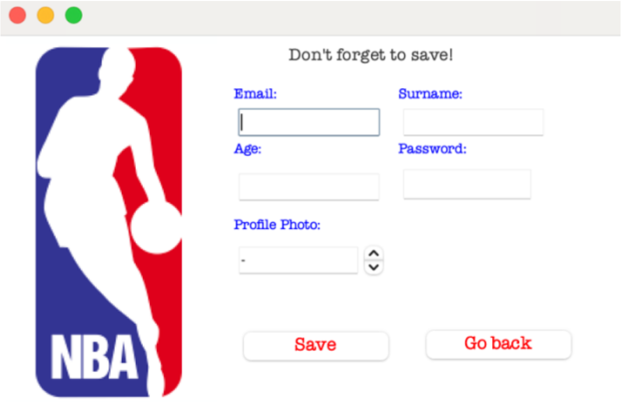
- Emre;Goksu;goksu@gmail.com;goksu123;24;goksu;/images/Erik Spoelstra.png
- Berkay;Tonbul;berkay@hotmail.com;berkay123;44;berkay;/images/Steve Kerr.png
- Faruk;Sartik;faruk@gmail.com;faruk123;33;faruk;/images/Michael Malone.png



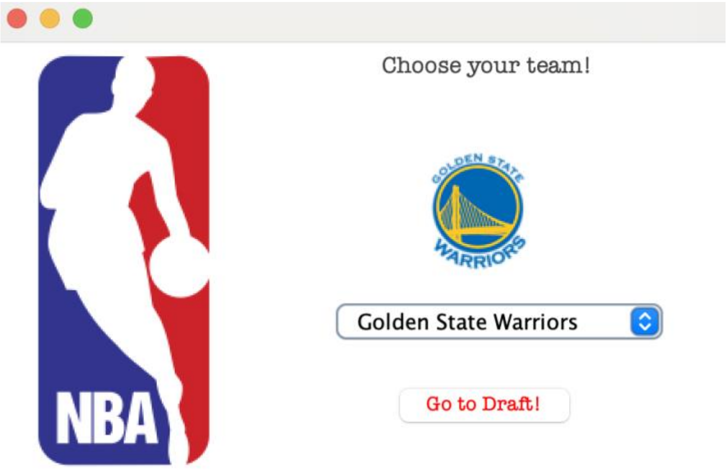
Before starting the game, the user encounters an intermediate page similar to the home page.


Hi Coach! I can't wait to start the game!

If the user who logs into his account wishes, he can update his individual information from the change information page.

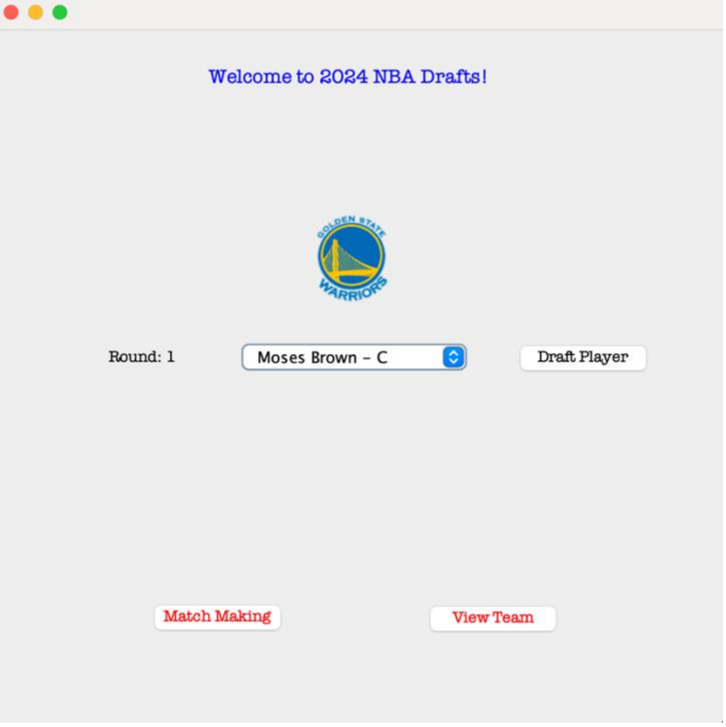

Don't forget to save!

Then, by clicking the Start Game button, you will come to the team selection page. He can choose whatever he wants among the 16 teams on the list.

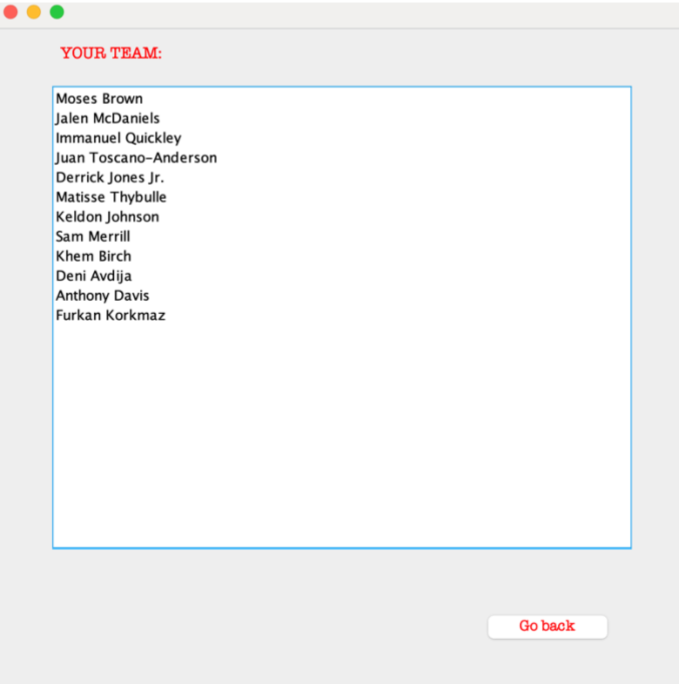

Choose your team!

Golden State Warriors

The user will then proceed to the draft page to create their roster. When it comes to the draft page, the draft will start automatically and when it is the user's turn, the system will wait for the user to select a player. In order to become a champion, the player must select the best players he can get from the draft screen according to the position distribution.

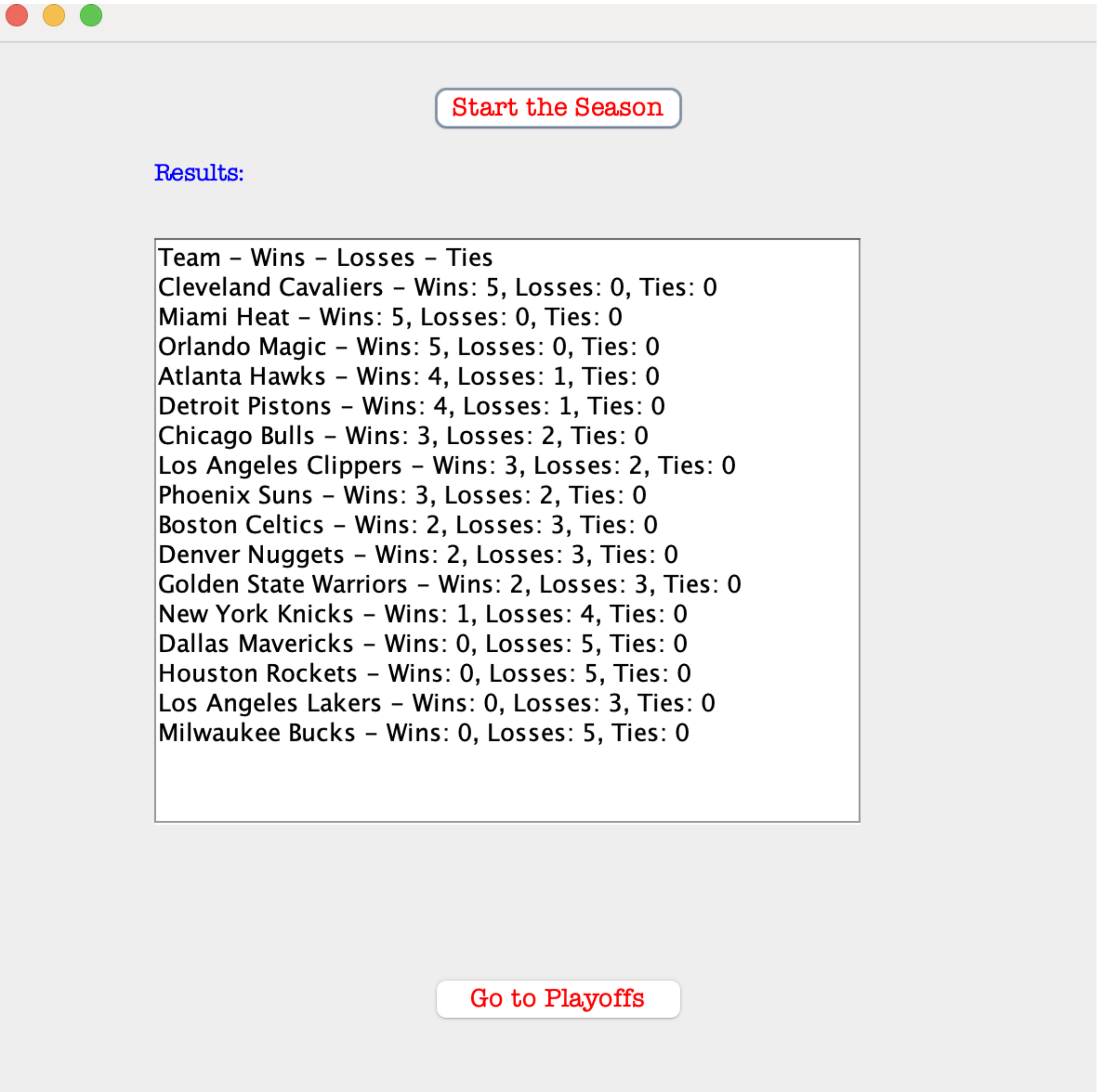


Once the selection of one player for each team is completed, the number of rounds will automatically increase by one step. When all 16 teams select a total of 12 players, the draft process will be completed and the user will receive a notification that the draft is over. After this notification, the user can go to the View Player page and examine his entire team and the annual statistical averages of the players in his team.


YOUR TEAM:

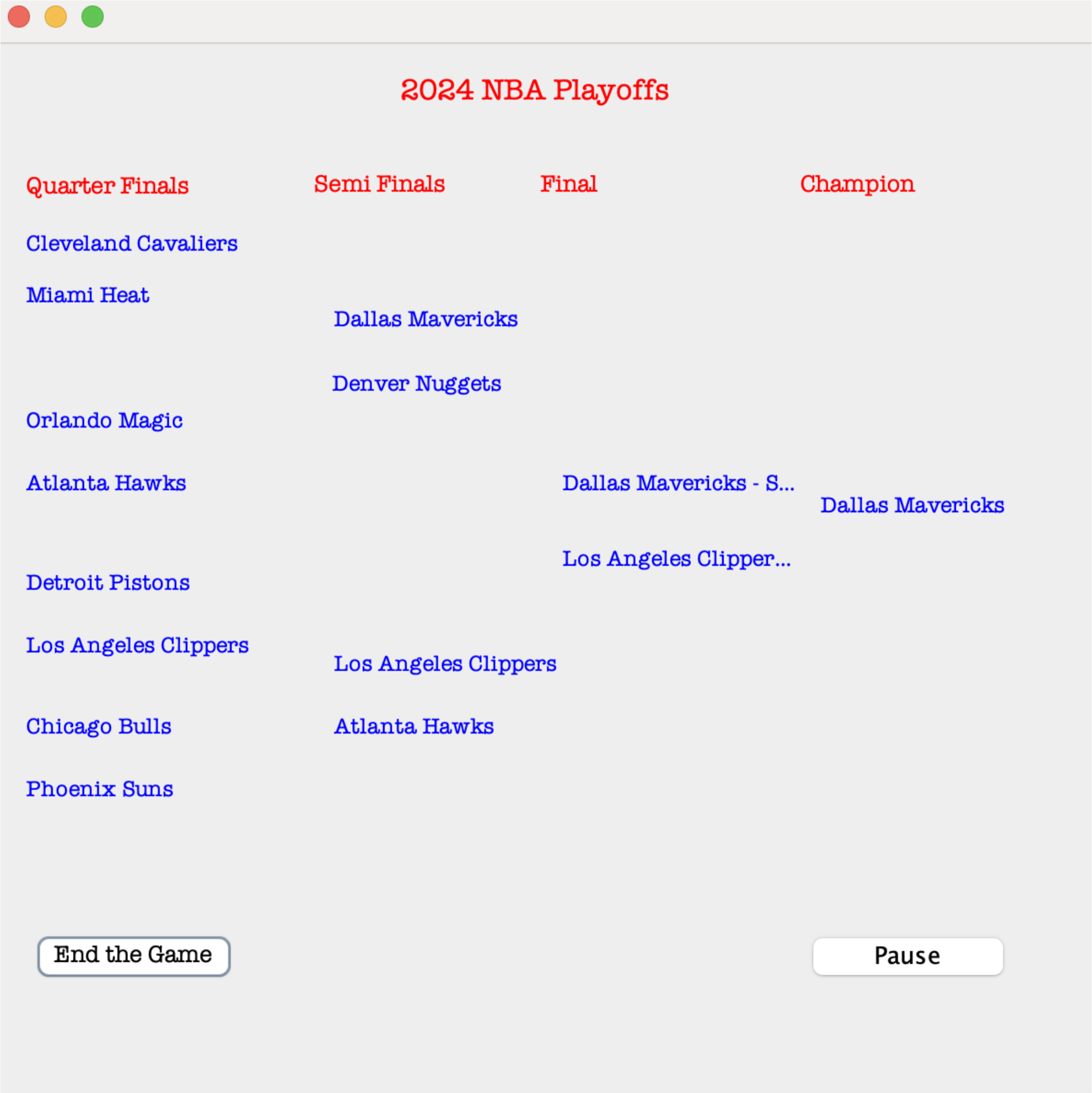
Moses Brown
Jalen McDaniels
Immanuel Quickley
Juan Toscano-Anderson
Derrick Jones Jr.
Matisse Thybulle
Keldon Johnson
Sam Merrill
Khem Birch
Deni Avdija
Anthony Davis
Furkan Korkmaz

To play a match, one must return to the draft page with the return button. Once the squads are established, the user can now start playing matches. When the user comes to the regular season page and presses the Start Season button, all teams in the league will have played 5 matches per team, for a total of 40 matches.



```
Team - Wins - Losses - Ties
Cleveland Cavaliers - Wins: 5, Losses: 0, Ties: 0
Miami Heat - Wins: 5, Losses: 0, Ties: 0
Orlando Magic - Wins: 5, Losses: 0, Ties: 0
Atlanta Hawks - Wins: 4, Losses: 1, Ties: 0
Detroit Pistons - Wins: 4, Losses: 1, Ties: 0
Chicago Bulls - Wins: 3, Losses: 2, Ties: 0
Los Angeles Clippers - Wins: 3, Losses: 2, Ties: 0
Phoenix Suns - Wins: 3, Losses: 2, Ties: 0
Boston Celtics - Wins: 2, Losses: 3, Ties: 0
Denver Nuggets - Wins: 2, Losses: 3, Ties: 0
Golden State Warriors - Wins: 2, Losses: 3, Ties: 0
New York Knicks - Wins: 1, Losses: 4, Ties: 0
Dallas Mavericks - Wins: 0, Losses: 5, Ties: 0
Houston Rockets - Wins: 0, Losses: 5, Ties: 0
Los Angeles Lakers - Wins: 0, Losses: 3, Ties: 0
Milwaukee Bucks - Wins: 0, Losses: 5, Ties: 0
```

After finishing the season, the Playoff journey will begin with the top 8 teams. When the user comes to the Playoff page, the playoff matches will automatically start between the top 8 teams and the winner of the single match advances to the next stage, all the way to the final. During the playoffs, players can pause and restart the match flow by pressing the Pause button. Once the champion is determined, the game can be ended.



2024 NBA Playoffs

Quarter Finals

Cleveland Cavaliers

Miami Heat

Orlando Magic

Atlanta Hawks

Detroit Pistons

Los Angeles Clippers

Chicago Bulls

Phoenix Suns

Semi Finals

Dallas Mavericks

Denver Nuggets

Los Angeles Clippers

Atlanta Hawks

Final

Dallas Mavericks - S...

Los Angeles Clipper...

Champion

Dallas Mavericks

Part 2

Project Design Description:

- Class relations.
- Inheritances, type hierarchies, interfaces, abstract classes.
- GUI components.
- CSV file processing details
- Your scoring algorithm
- Your drafting method
- Any additional design details that you want to mention

I must mention that one of the best things I understood throughout the project was that as the tasks got harder, I understood better why object oriented programming is necessary and makes a lot of sense (even though it is a simple game). I wrote the project in the order described in the project file. Basically, it consists of 4 packages: 1) User, 2) Player, 3) Team and 4)Match Making. In addition to these, there are txt or csv files that I create or use in my project file.

My coding process starts with the User Class in the User package. Here I have simply set up how we will get and use the information required for the user. I did a lot of validation using regex. Additionally, this class was where I worked on exception handling in detail throughout the program. I used a more practical structure in the following classes. My first GUI page starts with the NewUser Class. First of all, I learned the GUI both from the content shared in the course and by watching it on the internet. Then I designed almost all my GUI pages using WindowBuilder. In the NewUser Class, we obtain and save user information. After registering, we proceed to the login stage with the button here. With the LoginUser Class, we confirm the user's username and password and enable the user to log in if they are correct. StartOrChangeInfo Class can be considered as a transition page before starting the game. Any user can update their information by going to ChangeInfo Class. Or, he can proceed and start the game with the ChooseTeamGUI Class in the Teams Package.

Before moving on to the Teams Package, we can look at the Classes in the Player Package, where we edited the Csv file and calculated the players' scores. Here I created the Player Class as an abstract class. Thus, I was able to use the common features/objects of the players much more easily in the Classes I created for 5 different positions. I also enabled players to calculate their scores more easily with the Scorable Interface. I read this file with the ReadPlayerCSV class and extracted the necessary information. When calculating the players' scores, I initially gave different weights according to the positions, but this was not statistically realistic and I made a point-weighted distribution that was more representative of reality. Afterwards, I provided the opportunity for each score detail to change randomly within a certain range, and I set different lower limits to prevent it from falling below zero.

Coming to the Teams Package, first of all, in order to make my work easier in the future, I created the LoadFromCsv class, in which I filtered the Csv file and wrote the information I needed into a new Csv file. Thus, the file became simpler and I used it more easily. Then, I designed the place where team selection takes place with ChooseTeamGUI. One of the things I noticed from here on is that since I prefer a method of progressing by connecting different GUI pages in my code design, unfortunately there is not a single class that I can manage the general game flow with. After selecting the team, I listed all the players with their positions in the DraftingGUI class where the draft took place. When we came to this page, I automatically started the drafting process. When the draft started, all teams except the user's team were automatically managed by the computer. The draft order was determined randomly. The order was reversed each round. The distribution of players was ensured to include at least one person from each position. The draft ended with 12 players selected for each team. The season was designed with a total of 16 teams and 12 players each. From here, the entire team drafted with the ViewTeam class was shown to the user. By clicking on the player, the user can open the ViewPlayerDetails class and thus see the player's statistics in the csv file - I called it season average.

Unlike what I have done so far, I wanted to make the last MatchMaking package more compact by separating the GUI pages and the classes that operate in the background, but on the contrary, it became more complex. With the CurrentMatchResults class, I designed to create the different results of the teams when each game is simulated and write them to a txt file. Thus, as I ran this class, total score information for 16 teams would actually be created, and it would be enough for me to compare them when determining the winners of the matches. It achieved its goal at first, but as I encountered different errors, especially during the playoffs, I resorted to more various methods. The first GUI page in this package, SeasonGUI, was the page where I created 5 matches for each team and noted them. From this point on, a process took place in which I read the results of the matches in each playoff stage with both the MatchDataManager Class and the PlayoffMatchManager Class, then simulated the match again and then wrote the results to another txt file. I thought about improving the progress of the winners by reading and writing files through txt files, but a better structure could probably be created. Because of this system, the last Classes of my project were extremely complicated. Unfortunately, it was a little late when I realized this because all my GUI pages were loosely connected to each other rather than compactly. Finally, I achieved the desired stop and time gap using the Swing Timer in the PlayoffGUI class. Following a logic similar to the season, I showed the champions to the user by eliminating the teams.

Writing the game and being able to visualize it and see it in practice is a really enjoyable process, but I think more importantly, it improves our problem-solving skills and logic very well. I gave the details of the objects and methods I wrote in a much better way at the beginning of my code, but due to my illness of about the last 2 weeks, I could not spend the last few hours productively and some deficiencies were bound to occur. However, within the given time period, these excuses only go so far. It was a nice experience. Thanks.

References

Give the list of references you used during the project design and development. All the references should be cited within the text of the report.

1) https://www.youtube.com/playlist?list=PL3D7046DF2257751F
2) https://sites.google.com/ku.edu.tr/comp132/lecture-notes?authuser=0
3) https://www.youtube.com/@alexlorenlee
4) https://www.geeksforgeeks.org
5) https://stackoverflow.com