

Operating Systems Project

2019

Queli Rentao Pereira Pinto Serano

G00361469

## Project

To start my project, I created different classes for the clubs, players and player managers. These are then used to create objects throughout the application making my life simpler when dealing with players, clubs or agents.

I choose to use integers to deal with the majority of the number variables required, the only variables that I was considering using a double or float was the players valuation and the clubs transfer funds but I decided to go with integers because the clubs usually deal with big amounts and no decimal points.

I then setup a simple server to listen for connections of clients and created a client class that would be used to connect to the server using the server’s Ip and port. After getting that to work I proceeded to create a worker thread that would be responsible to handle the interactions with the users connected. I also decided to create an extra class called Transfer Market to deal with most of the file reading and writing so that we can store the data created permanently. In the worker thread I declared an instance of the Transfer Market class so I would be able to use the methods from that class on the thread.

In the worker thread I created some local variables to help me handle the interactions with the client and 3 volatile array lists that would keep the records from the server being shared between the clients connected to the server.

For the Register functionality I created to synchronized methods that call other methods that I wrote to check if the user trying to register already exist on the system, if they already exist an error message is returned, if not the user will be added to the records and a success message will be passed.

Then I proceeded to create another two synchronized methods to authenticate clubs or agents trying to login into the system, the methods check if the username and id match the ones registered and return if the user was successfully authenticated or not. Also, if a user fails to enter the correct details for 3 times in a row the program will alert the user and close its connection to the server.

After getting the registration and login working, I created a menu for each of the user types, that would be displayed depending on the user logged in. For the clubs the first functionality was searching for players based on their position so I created a method to display the players by position that takes in the position required by the client and creates a new array list of players that is populated by a loop that adds only the players with the position that the club is looking for and then returns the list to be displayed for the client. The next functionally for the clubs was to search for players selling in the club. For this I created a similar method to the one that searches for players by position but changed it to check if the players belong to the club that is searching and if their status is set to selling.

For the updating status feature, I created a method that would take in the agent and player’s id’s and the new status of the player. If the details are found in the database, the specific player will be removed and added again with the updated status. And the market variable updated with the new values.

For the player agent I first started to create a method to add a player do the database, the method takes in a player object and verifies if the player already exists In the database, if the player does not exist it adds the player to the database else it will return an error message.

Then I created another method to update the valuation of a player that is registered with that agent. The method takes in the players and agents Id together with the new valuation of the player, if the player Id and agent Id are correct the valuation for that player will be updated else an error message will be displayed. And finally for the last functionality I created a method like the update valuation to update the status of the player. This method takes in the player and agent id’s and the new status of the player, the status is then validated and if it is successfully validated it will update the status of the player else an error message will be displayed.