1. Review each of your objectives and explain if you met that objective and how you met that objective or if you did not meet that objective and why.

**Primary Objectives:**

1. **Europe Champion league championship class:**

**All planned functionalities in primary goal for Championship class were implemented successfully with some changes.**

**1. This class contains the game\_date LinkedList object. The Game\_Date class was implemented by importing the GregorianCalender package.**

**2. The Game status object is actually not in the championship class. Instead, it is an object in the GameController class which has the same functionality as I described in my project plan which is to control the game flow process.**

**3. Instead of result of each game, the winner club is evaluated by methods of championship class and saved as an object in this championship class. The score results are instead store in each Club object.**

**4. The Top Scorers list are stored in the championship class. What I did is add the scorer list from the match event object to the championship class after each match simulation and using higher order function sort the player object by their number of goals.**

**5. Transfer window status is store in the championship class. What I did is to create a Boolean static Boolean variable which is initially set to true. Once the transfer window is closed, I just change the static Boolean variable to false and vice versa.**

**6. match schedule are stored in the championship class. The match schedule includes group events and round events. Group events store all group match event objects as LinkedList, and round events include all round match event object also as LinkedList.**

**2. Soccer Club Class:**

**Most of the primary objectives are achieved and many more variables and objects are storage in this Club class.**

1. **Club name is stored in club class. Club name is an argument for initiate the club object.**
2. **A stadium object (composition) is stored in this club class. I create a stadium object to store the stadium information. The stadium object is also an argument for initiate the club object**
3. **Employee interface was implemented, and an Abstract Employee class was also implemented. Both player class and manager class extend and implemented these two classes. This club class store a linked List of player objects and a manager object.**
4. **This club class also contains an account object.**
5. **This club class also store all score result in different game stage such as group stage and round stage( round of 16, round of 8, semi-final, final).**
6. **A charm level variable is stored in club object. This is a argument in the club constructor. This value will affect the possibility of acceptance of transfer request. The higher the charm level, the higher possibility user can successful transfer the player they want to buy.**

**3. Hierarchy structure of Club Employee:**

**I did not use hierarchy employee tree structure for manager and player because the employee of a club is not that complicated. So, it is not necessary to implement the hierarchy tree structure. The employee structure is simpler. What I did is to implement a employee interface and abstract employee abstract class with some importance method and let player and manager class to implements and extend them.**

**4. Finance Account class:**

**This class was successfully created and called Account. It is very similar to the Account ICE. It includes some basic functionalities such as increase and decrease balance. The balance will be updated by the method in club class (updateBalance\_spent) after spent money either by paying salary or make transfer.**

**method to gain money to increase the balance:**

**All primary goals were achieved. There are only four ways to earn money which are:**

1. **Club make next round can earn specific amount of money**
2. **Each home match can earn ticket fund and a bit live stream fund and**
3. **away match can only earn some live stream fund.**
4. **Earn money after reaching Club CEO’s season objective.**

**What I did is to use the getter and setter of the Account object to change club balance after each match event.**

**5. Player class:**

**All primary objectives were achieved.**

**The player class store following variables which was originally planned:**

1. Name
2. Age
3. Nationality
4. Weekly (this changed to annual pay)
5. Shooting skills
6. Dribble skills
7. Passing skills
8. Intercept skills
9. Stealing skills
10. Goalkeeping skills

**Additional variable added are:**

1. Jersy number
2. position
3. player value

**Since there are about 20 players in each club and there are 32 clubs in total, I downloaded player csv dataset from the link below:**

<https://www.kaggle.com/stefanoleone992/fifa-21-complete-player-dataset>

**I chose the use the dataset of FIFA 2020 and I use excel and python to process the data to make it in a specific format. Then, I load the csv into Java and save the data into a 2D array. Then, I use iteration to create Player objects and add them into the player linked list object of club object for each of 32 clubs.**

**6. Class for simulating the game:**

**My original plan is to create 2 class which are user match simulator class and non\_user match simulator class.**

**Due to time limitation, I only finished the non\_user match simulator class which can simulate game result based on starting lineup player’s associated attributes and user selected tactic.**

**Same as the primary goal, user can select the following tactics:**

1. Park the bus (extremely high possibility to make intercept and steal, but extremely low possibility to make passing and score)
2. Counter-attack (higher possibility to make intercept and steal, and lower possibility to make passing and score)
3. Balance (equal possibility of attacking and defensing)
4. Attacking (lower possibility to make intercept and steal, and higher possibility to make passing and score)
5. All-out (extremely low possibility to make intercept and steal, and extremely high possibility to make passing and score)

**Each tactic has an associated factor. For example, the intercept factor of all-out tactics is 1 and the passing factor of tiki\_taka tactics is 40. When a player with tiki\_taka tactics want to pass and opponent player with all-out tactics want to intercept the pass. The result will be determined as following:**

**(int)Passing\_player\_result= (int)player\_passing\_skill + (int)random number of tiki\_taka tactics passing factor + (int)uncertainty factor;**

**(int)Intercept\_player\_result = (int)player\_intercept\_skill + (int)random number of all-out tactics intercept factor + (int)uncertainty factor;**

**if the Passing\_player\_result > Intercept\_player\_result , the first player successfully make the pass. Otherwise, the second player successfully intercept the pass.**

**All other actions such as shooting, goalkeeping and dribble and stealing are all following the similar method.**

**Also, at this point, due to time limitation, I cannot implement the code to allow user adjusts the formation. User at his point must use the default formation of the selected team.**

**Secondary Objectives:**

**Finance Account class:**

Objective: The account could be used to pay for loaning a player from other teams

**Player Loan functionality was not implemented due to time limitation**

**Player class:**

Objective: More attribute variable can be added into player class

1. competitive status: which will affect player’s performance
2. loyalty to the club: possibility to be transferred to another team

**None of these two features were implemented due to time limitation**

**Class for simulating the game:**

Objective: Adding Penalty kick feature to the game. The result will be based on player’s shooting skill and goalkeeper’s Goalkeeping skills

This is sort of implemented because if the result is draw, the result will be determined based on player’s shooting skill and opponent’s goalkeeper skill.

**Tertiary Objectives:**

1. Allowing user to save the progress and load the save files
2. Adding another championship such as England Premier league and Spanish La Liga which can be overlapped with the Europe Champion league
3. Team that not controlled by users can randomly transfer their players
4. Making the simulation more complicated by allowing player entering another field. For example, a midfielder can temporarily become attacker during the game.
5. Adding injury mechanism to the player class
6. Adding red card and yellow card mechanism to the game simulation

None of the Tertiary objectives above were completed because time limitation and need some artificial intelligence(AI) Knowledge.

2. If you had more time, what else would you do?

If I have more time I would first implement the user match simulator class which can let user see the whole match process in text and adjust their tactic during the match.

Then, I will try to achieve my secondary and Tertiary objectives. I will also do some research about game AI which can make this game more fun.

3. What did you learn from this project?

I have learned a lot about how to use GUI and composition, interface, Enum and abstract class. I also learn how to make MVC architecture even though my MVC architecture is not that good. This project teaches me a license which can make me create a better MVC architecture. Also, I really enjoy the process of development of this game.

4. Do you feel you would have done better or would you have preferred a structured project instead? Explain.

I think I could have done better next big project because my MVC architecture is not very SOLID. I think my MVC’s cohesion is relatively low, and coupling is high.

When I was coding this project, I was sometimes confused and cannot clearly differentiate the model and controller and view.

5. What grade do you feel you deserve on this project? Explain.

I feel I will get A on this project. I really put tons of effort on this game. I have spent between 100 – 150 hours on this project. My schedule in the recent 3 weeks is like sleeping, eating, finishing assignment and quiz, join lectures and labs and I put rest of times to working on this project.