Assignment 5: Software Design

After you showed your friend Leeroy your *TourneyCalc* program, he suggested that you might be a good partner for his side business of running card e-gaming tournaments in a Woodruff lounge. You are to explore this idea by designing an application for managing gaming tournaments whose requirements are listed below. To do so, you should follow the same approach that we present in the P3L2 lesson, that is, analyze the requirements to identify and refine (1) classes, (2) attributes, (3) operations, and (4) relationships in your design. Just to be completely clear, **your task is to design the system, not to implement it**. Please note that not all requirements will necessarily affect your design in a direct way. For example, you do not have to represent the GUI in your design.

Your design should be expressed using a UML class diagram, and the level of detail of the design should be analogous to the level of detail we used in the P3L2 lesson. Basically, you have to provide enough details for the design to be self contained and to allow Leeroy to assess whether the design is suitably realizing the system requirements. To help with this, you must also provide a "design information" document in which you **concisely** describe, for each of the requirements listed below, how that requirement is either realized in your design or not considered because it does not affect it. For example:

…
2. The application must contain a list of items and corresponding item types.
To realize this requirement, I added to the design a class X with attributes Y and Z. Class X...
...
16. The User Interface (UI) must be intuitive and responsive.
Not considered because it does not affect the design directly.

Optionally, you can also provide in the document additional information about your design, such as assumptions or rationale for some design decisions.

You can use any UML tool to create your design. If you are not familiar with any specific tool, we recommend that you ask on Piazza for suggestions. In fact, on Piazza there is already some discussion about that.

Requirement

1. The tournament is organized as a single elimination tournament with third place playoff.
2. The application has two modes: *tournament manager* and *tournament player*. You can assume that the mode is selected when the application starts, with no authentication involved.
3. The tournament manager uses the system to (1) add players, (2) run tournaments, and (3) display prizes and profits.

4. The tournament players use the system to (1) view the match list and (2) view the total player prizes.
5. The app has an underlying database to save persistent information across runs (e.g., players in the system, status of ongoing tournaments).
6. A player in the system is characterized by the following information:
    a. *Name*
    b. Unique alphanumeric *username*
    c. Numeric *phone number*
    d. A *deck choice*, from a list of deck options
7. The tournament manager can add a player to and remove a player from the system.
8. There can only be one ongoing tournament at any given time.
9. To *start a tournament*:
    a. The tournament manager will enter the house cut.
    b. The tournament manager will enter the entry price.
    c. The tournament manager will enter all player usernames. **For simplicity, let's assume that there will be either 8 or 16 players in a tournament**.
    d. When the tournament manager has entered the above information, the system will display, in addition to the player names, the potential prizes and profit, as calculated in the *TourneyCalc* app that you developed for Assignment 4. The tournament manager will then be able to double check the information and start the tournament.
10. When there is no ongoing tournament, the player mode will show totals for every player in the system as a list sorted by total.
11. When a tournament is ongoing, the player mode will show a match list. The match list displays a list of players paired with other players representing ongoing matches, matches ready to be played, and results from completed matches.
12. When a tournament is ongoing, the manager mode will also show a match list. In this case, however, the tournament manager will be able to:
    a. Start a match ready to be played by selecting it from the list. The system will then mark the game between the specified players as started.
    b. End an ongoing match and specify a result for it.
    c. End the tournament. If the tournament is ended early, money is refunded.
13. After a tournament is completed, prizes for the winning player, the second place player, and the third place player (who wins the third place playoff) will be recorded in the underlying database .
14. After a tournament is completed, the house profit will also be recorded in the underlying database.
15. When there is no ongoing tournament, the tournament manager can:
    a. View totals for every player in the system as a list sorted by total. From there, the manager can also view a list of the player's individual prizes by selecting the player from the list.
    b. View the list of past house profits in chronological order and the total.
16. The User Interface (UI) must be intuitive and responsive.

To submit your assignment, you should do the following:

- Create a directory called Assignment5 in the usual **personal GitHub repository we assigned to you**.
- Save your UML class diagram in the Assignment5 directory as a PDF file named "`design.pdf`".
- Save the "design information" document in the same directory, in markdown format, and name it "
- `.md`".
- Commit and push your file(s) to your remote repository.
- Submit the commit ID for the files on T-Square.