CS-M24 Group Project, Part II A Specifications + Design Document

For this assignment, you will have to write a report (possibly 10 pages) summarizing the user requirements specifications and the design of your *connect 4* application, following the description of the requirements as given here after.

You may want to include in this document (but not limited to):

- Two specific sections related to the *User Requirements Specifications* and *a Preliminary/conceptual/architectural Software Design*.
- Any additional requirements/specifications needed on top of the original requirements.
    - Should the client requirements be inconsistent, fixes will be provided as new requirements.
- A strong focus on the OO technology employed, as well as a focus on the user interface. Class diagrams will have some importance there.
- UML Diagrams whenever suitable to the project (State, Use Case, Activity, Class Diagrams, etc...).
- A list of the components/modules used in your application and for each class a description of the class members. Test cases should be provided as well.
- A section on your collaboration and progress as a team, including minutes of the meetings you will hold.
- The required Appendix documenting your individual effort. This appendix **must** be included with the rest of the report.
- A list requirements ordered by difficulty if possible.
- Some evidence that you have been able to setup a good collaborative project, with individual tasks well defined and separate work well distributed among group members.

Project Requirements: A chess interface

The project is about creating a chess interface so that two players can play chess together on a board displayed on the screen. Please note at this stage that it could be difficult for you to implement all the requirements enounced here due to time constraints. However, you should be able to display a good and coherent management of the work, and show a concrete encapsulation of the different components of the project.

Here are the requirements as emitted by the client:

- The application should be runnable on phones, tablets and standard computers.
- The chess interface has several components, including a 2D board, a move list section, an accessible menu, names of players, and a position score available as an option.
    - Have a look at http://database.chessbase.com/js/apps/database/ for getting an idea of such an interface.
- White side always start the game.
- The application must store at least 8 player profiles, and be able to reload them when starting.
- The menu selecting the two players has a quick access link to players' profile.
- There are three types of players: Computer, Human, Network.
    - Use of OO programming is required in general, and particularly for this part. It will demonstrate the Team's ability to do proper OO software engineering!
- When someone calls the menu item "New game", the interface asks for the two types of players. There are three choices for white, and three choices for black obviously.
    - The computer network may have additional options like difficulty or the chess engine type.
- The application should allow finding a network player from a given IP.
    - See Network protocol.
- The application opens port 4567 when starting and closes it when exiting.
- A maximum of three other network applications can connect to your application (running).
- The application can be implemented in any languages, but java is possibly the most sensible choice.
- The computer player shall be as strong as possible. You may want to use your own engine, or re-use existing ones.
    - Code should allow plugging in new engines in a manageable way. There exist some free and open source chess engines that are the best engines in the world. Please have a look at the following chess engine ranking (http://www.computerchess.org.uk/ccrl/4040/) and at stockfish in particular (https://stockfishchess.org/).
    - Chess engines are often programs running separately where communication is made through the UCI protocol. (http://computer-chess.org/doku.php?id=computer_chess:wiki:lists:gui_protocol_support_list)
- For computer/computer type of games (this include network computers), there should be an optional game repetition value specifying how many games the two players will play. The total number of wins, losses and draws must be displayed at the end.
    - Sides are swapped after each game.
- A network player must be played by an application that has opened port 4567 on the network.
    - The host application must be given the player' (network) address.

    - The network communication can follow a given protocol, like with UCI engines.
- A human player must be able to drag and drop pieces, with the possible destination squares highlighted when dragging.
- Number of wins and losses for a stored profile should be recorded.