READ.ME

We'll focus on the following set of requirements while designing the game of chess:

1. The system should support two online players to play a game of chess.
2. All rules of international chess will be followed.
3. Each player will be randomly assigned a side, black or white.
4. Both players will play their moves one after the other. The white side plays the first move.
5. Players can't cancel or roll back their moves.
6. The system should maintain a log of all moves by both players.
7. Each side will start with 5 pawns
8. The game can finish either in a checkmate from one side, forfeit or stalemate (a draw), or resignation.

We have two actors in our system:

* **Player:** A registered account in the system, who will play the game. The player will play chess moves.
* **Admin:** To ban/modify players.

Here are the top use cases for chess:

* **Player moves a piece:** To make a valid move of any chess piece.
* **Resign or forfeit a game:** A player resigns from/forfeits the game.
* **Register new account/Cancel membership:** To add a new member or cancel an existing member.
* **Update game log:** To add a move to the game log.

Here are the main classes for chess:

Player: Player class represents one of the participants playing the game. It keeps track of which side (black or white) the player is playing.
Account: We'll have two types of accounts in the system: one will be a player, and the other will be an admin.
Game: This class controls the flow of a game. It keeps track of all the game moves, which player has the current turn, and the final result of the game.
Box: A box represents one block of the 8x8 grid and an optional piece.
Board: Board is an 8x8 set of boxes containing all active chess pieces.
Piece: The basic building block of the system, every piece will be placed on a box. This class contains the color the piece represents and the status of the piece (that is, if the piece is currently in play or not). This would be an abstract class and all game pieces will extend it.

Move: Represents a game move, containing the starting and ending box. The Move class will also keep track of the player who made the move, if it is a castling move, or if the move resulted in the capture of a piece.
GameController: Player class uses GameController to make moves.
GameView: Game class updates the GameView to show changes to the players.

Here is the code for the top use cases