**Responsibilities**

Knows:

* Username and Password

Actions:

* Can move chess pieces of their own color
* Invite players to match
* Create matches
* Can switch between ongoing matches
* Can accept invites from other players

**Collaborators**

* Lobby
* Move
* Login
* Registration
* Game
* Chess Board
* Chess Pieces

frontend

**Players**

frontend

**Move**

**Responsibilities**

* Allow players to move a piece
* Check if the move is valid

**Collaborators**

* Chess Piece
* Chess Board

**Player Profile**

frontend

**Responsibilities**

* Display any player’s (including the user) statistics and player information

**Collaborators**

frontend

**Create Match**

**Collaborators**

* Players

**Responsibilities**

* Window that allows players to create a new match
* Invite other players to that new match

frontend

**Concurrent Matches**

**Responsibilities**

* Displays all concurrent matches that user is involved in

**Collaborators**

* Player

frontend

**Responsibilities**

* Allow user to log in into their existing account

**Login**

**Collaborators**

* Players

frontend

**Responsibilities**

* Allow user to register a new account
* Allow user to deregister

**Registration**

**Collaborators**

* Players

**Collaborators**

* Match

frontend

**Responsibilities**

* Allow user to create a match
* Allow user to invite existing players to a match
* Allow user to check their game history and statistics
* Allow user to deregister their account
* Allow user to view other people’s profile

**Lobby**

frontend

**Collaborators**

* Players

**Responsibilities**

* Allow 2 players to play Omega chess
* Decides whether game is finished along with the results for each player
* Allow users to view each other’s profiles
* Allow user to quit match
* Allow one user to play at a time

**Match**

frontend

**Responsibilities**

* Displays the board
* Updates the board when a player makes a move

**Chess Board**

**Collaborators**

java.cs414.teamd

**ChessBoardController**

**Collaborators**

* ChessBoard

**Responsibilities**

Actions:

* Sends board information
* Update board based on the move

java.cs414.teamd

**QuitMatchController**

**Collaborators**

* Packages:
  + Database API

**Responsibilities**

Actions:

* Updates database on the current match with information on the outcome of the match

java.cs414.teamd

**OngoingMatchController**

**Responsibilities**

Actions:

* Gets information on what other matches user is playing

**Collaborators**

* Packages:
  + Database API

java.cs414.teamd

**CreateMatchController**

**Responsibilities**

Actions:

* Updates database on a new match
* Creates an initial board state

**Collaborators**

* Packages:
  + Database API
* Classes:
  + ChessBoard

**Collaborators**

* Packages:
  + Database API

java.cs414.teamd

**LogInController**

**Responsibilities**

Actions:

* Verifies that the log-in credentials are in the database

**Collaborators**

* Packages:
  + Database API

java.cs414.teamd

**RegistrationController**

**Responsibilities**

* Receives information about registering new user

Actions:

* Insert to database for new user

**Responsibilities**

Knows:

* Which moves are valid
* Color of that piece that indicates which player controls it
* What position it is in

Actions:

* Can set its own position on the board

**Collaborators**

* Chess Board

java.cs414.teamd

**Chess Piece**

**Responsibilities**

* Extends from Chess Piece

Knows:

* Valid Moves:
  + Can move 1 or 2 spaces vertically or horizontally
  + Can move 2 spaces diagonally

**Collaborators**

* Chess Piece
* Chess Board

java.cs414.teamd

**Champion**

**Responsibilities**

* Extends from Chess Piece

Knows:

* Valid Moves (weird one):
  + 1 space diagonally
  + 3 spaces horizontally then 1 space vertically
  + 3 spaces vertically then 1 space horizontally

**Collaborators**

* Chess Piece
* Chess Board

java.cs414.teamd

**Wizard**

java.cs414.teamd

**Collaborators**

* Chess Piece
* Chess Board

**Responsibilities**

* Extends from Chess Piece

Knows:

* Valid Moves:
  + Can **ONLY** move diagonally
  + It can move anywhere diagonally

**Bishop**

java.cs414.teamd

**Collaborators**

* Chess Piece
* Chess Board

**Responsibilities**

* Extends from Chess Piece

Knows:

* Valid Moves:
  + Can **ONLY** move one space in any direction

**King**

**Responsibilities**

* Extends from Chess Piece

Knows

* Valid Moves:
  + Can move 2 spaces vertically then 1 space horizontally
  + Can move 1 space vertically then 2 space vertically

**Collaborators**

* Chess Piece
* Chess Board

java.cs414.teamd

**Knight**

**Responsibilities**

* Extends from Chess Piece

Knows:

* Valid Moves:
  + Can **ONLY** move 1 space vertically towards the opponent’s side
* Exceptions:
  + Can move 1, 2, **OR** 3 spaces from the start
  + En Passante Rules – ask dakota
  + Can change into any other piece (except king) when it reaches the other end of the board

**Collaborators**

* Chess Piece
* Chess Board

java.cs414.teamd

**Pawn**

java.cs414.teamd

**Collaborators**

* Chess Piece
* Chess Board

**Responsibilities**

* Extends from Chess Piece

Knows:

* Valid Moves:
  + Can either move to any space diagonally, vertically, or horizontally

**Queen**

java.cs414.teamd

**Collaborators**

* Chess Piece
* Chess Board

**Responsibilities**

* Extends from Chess Piece

Knows:

* Valid Moves:
  + Can **ONLY** move anywhere vertically **OR** horizontally

**Rook**