**Collaborators**

* Packages:
  + React-router-dom
* Front-End components:
  + Lobby.js
  + Game.js
  + LogIn.js
  + Registration.js

**Responsibilities**

* Holds the actual app
* Holds routes of the app that allows user to switch between log-in/registration, lobby, and in-game match

**App.js**

**Collaborators**

* ChessBoardController.java

**Responsibilities**

* Shows the current board that user is playing on
* Updates board when another player/user makes a move
* Allows player to make a move

**Game.js**

**Lobby.js**

**Responsibilities**

* Allow user to create a match
* Invite other people to that match

**Collaborators**

* CreateMatchController.java

**Responsibilities**

* Window that allows user to enter log-in credentials

**LogIn.js**

**Collaborators**

* LogInController.java

**Responsibilities**

* Window that allows user to enter information to register

**Registration.js**

**Collaborators**

* RegistrationController.java

**Responsibilities**

* Sends board information
* Update board based on the move

**ChessBoardController.java**

**Collaborators**

* ChessBoard.java

**Collaborators**

* Packages:
  + Database API

**Responsibilities**

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

**QuitMatchController.java**

**Responsibilities**

* Gets information on what other matches user is playing

**OngoingMatchController.java**

**Collaborators**

**Responsibilities**

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

**CreateMatchController.java**

**Collaborators**

* Packages:
  + Database API

**Responsibilities**

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

**LogInController.java**

**Collaborators**

* Packages:
  + Database API

**Responsibilities**

* Adds new log-in user to the database

**RegistrationController.java**

**Collaborators**

* Packages:
  + Database API

**Collaborators**

* ChessPiece.java
* ChessBoard.java

**Responsibilities**

* Extends from ChessPiece.java
* Valid Moves:
  + Can move 1 or 2 spaces vertically or horizontally
  + Can move 2 spaces diagonally

**Champion.java**

**Collaborators**

* ChessPiece.java
* ChessBoard.java

**Responsibilities**

* Extends from ChessPiece.java
* Valid Moves (weird one):
  + 1 space diagonally
  + 3 spaces horizontally then 1 space vertically
  + 3 spaces vertically then 1 space horizontally

**Wizard.java**

**Collaborators**

* ChessPiece.java
* ChessBoard.java

**Responsibilities**

* Extends from ChessPiece.java
* Valid Moves:
  + Can **ONLY** move diagonally
  + It can move anywhere diagonally

**Bishop.java**

**Collaborators**

* ChessPiece.java
* ChessBoard.java

**Responsibilities**

* Extends from ChessPiece.java
* Valid Moves:
  + Can **ONLY** move one space in any direction

**King.java**

**Collaborators**

* ChessPiece.java
* ChessBoard.java

**Responsibilities**

* Extends from ChessPiece.java
* Valid Moves:
  + Can move 2 spaces vertically then 1 space horizontally
  + Can move 1 space vertically then 2 space vertically

**Knight.java**

**Collaborators**

* ChessPiece.java
* ChessBoard.java

**Responsibilities**

* Extends from ChessPiece.java
* 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

**Pawn.java**

**Collaborators**

* ChessPiece.java
* ChessBoard.java

**Responsibilities**

* Extends from ChessPiece.java
* Valid Moves:
  + Can either move to any space diagonally, vertically, or horizontally

**Queen.java**

**Collaborators**

* ChessPiece.java
* ChessBoard.java

**Responsibilities**

* Extends from ChessPiece.java
* Valid Moves:
  + Can **ONLY** move anywhere vertically **OR** horizontally

**Rook.java**

**Collaborators**

* ChessBoard.java

**Responsibilities**

* Holds information on how the chess piece moves and where it is on the board
* Tells user what moves are legal and where it can be placed
* It set the position of the piece on the board

**ChessPiece.java {abstract}**

**Collaborators**

* ChessPiece.java and any object that is extended from this class
  + Bishop.java, King.java, Knight.java, Pawn.java, Queen.java, Rook.java

**Responsibilities**

* Keeps track of all the chess pieces

**ChessBoard.java**