**Chess Board Game Software Requirements**

**Task Description**

1. Describe the requirements for the chosen system: functional and non-functional.
2. Design use cases for the system based on the requirements.
3. Identify objects, classes, and relationships in the system. Optionally, design CRC cards.
4. Design class diagrams picturing classes, their attributes, and relations in the system.
5. Optionally, convert diagrams into Java code.
6. Send the following results: requirements, use cases, CRCs (optional), class diagrams, Java code of the classes.

**Implementation**

**Functional Requirements**

1. The application must fully implement and simulate the process of playing a famous board game – Chess.
2. The application must have a playing board which contains figures from Chess Game.
3. The figures are movable according to the game rules.
4. The game must support at least two-players game mode.
5. The application must understand when the game is over and identify which player wins.

**Non-Functional Requirements**

1. The game should have a single-player game mode which should be implanted with help of an AI agent.
2. The game should show whose turn it is.
3. The game should show possible moves a selected figure on the board can go to.

**Use Cases**

Title: Make a move

Primary Actor: A Player whose turn it is

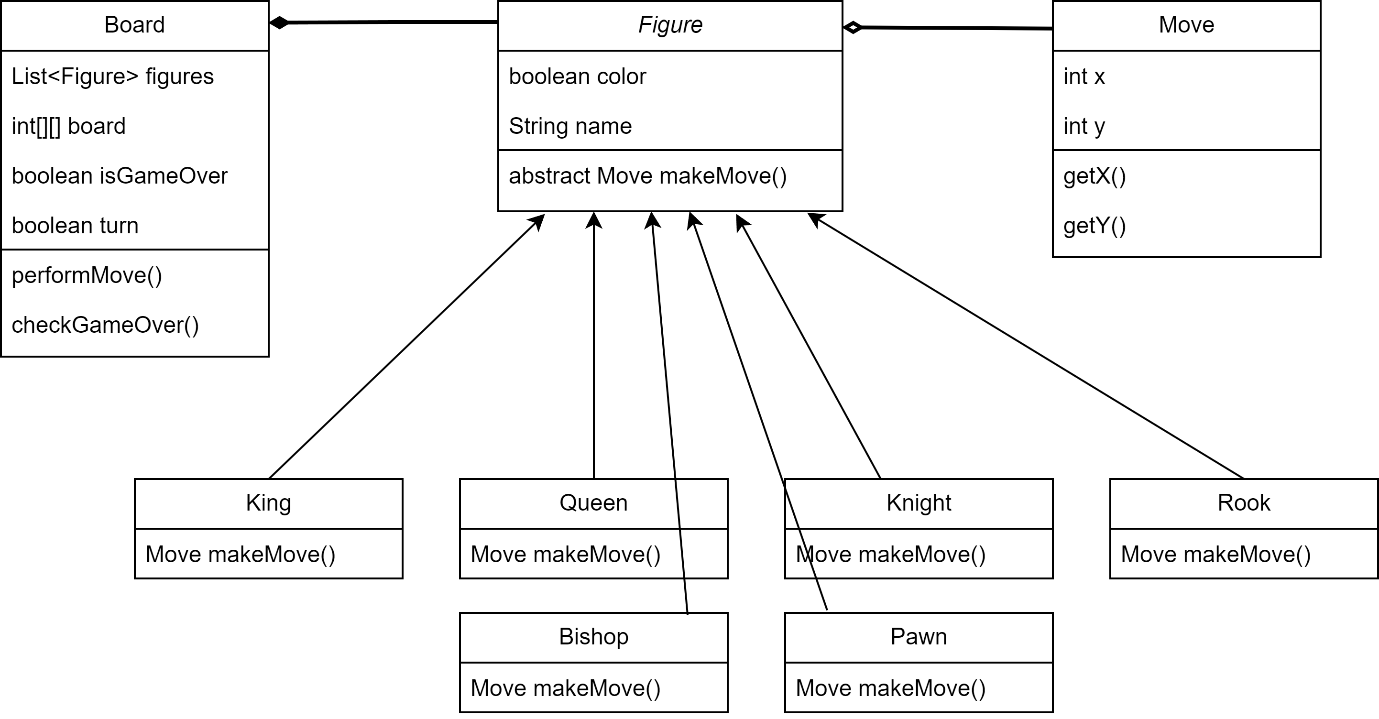
Success Scenario:

1. Select a figure on a board you want to make a move with.
2. See possible moves of the figure that it can make using game hints and make your choice.
3. Tap on the cell on the game board which corresponds to your decision and make a move.

**Creating a class diagram**

Defining objects: Board, Figure, Move, King, Queen, Knight, Rook, Bishop, Pawn.

**Entity-Relation Diagram**

****