

# Software Requirement Specification (SRS)

## Documentation

Project Name: Chess-Simulator  
Date: 26-February-2025  
Version: 0.0.1  
By: Tejas Kayande and Sammed Burse

## 1 Introduction

### 1.1 Application Scope

Application focuses on creating a user friendly chess application that follows the UI standards of other leading chess applications. The focus is also on creating a highly performant and memory efficient application for the low end computers.

### 1.2 Intendent Audiance

The application is aimed at avid chess players that enjoy casual games of chess, and like to study opening theories and practice their strategies by playing with stranger and friends. This application also involves an AI engine to further enhance the studies, which also makes for a reletively strong openent to practice opening theories and tactics.

### 1.3 Intended Use

The application is well suited for chess player that are familiar with some basic chess terminologies and gameplay. Its intended use involves analysis of games and position evaluation with the help of built in AI engine.

## 2 Requirements

### 2.1 Functional Requirements

#### *Core Machanics*

Rendering – The application should display the 8x8 Chess Board on the screen on a window that is interactive and responds with very low latency.

Piece Movement – The user should be able to move the Chess Pieces on the board on any square.

Move Validation – The moves made by the user should be validated by checking them against the traditional rules of chess, and only allow the moves that are legal.

Special Moves – The application must supports special moves like castling and en-passant.

Game State Detection – The application should keep track of the position on the board and respect the traditional conclusions of a chess match, i.e, Checks, Checmate and Stalemate.

### *User Interaction & Interface*

Piece Selection and Movement – The user should be to select the pieces on the board and then drag them to the square to move the piece to that square.

Move Highlighting – The latest move on the board and valid moves for the currently selected piece should be visually indicated. Although the application should also have a way to turn this feature off in case any user finds in annoying.

Move History Tracking – The application must maintain and display a history of moves made during the game. It should also have a feature to continue the game from the history into different moves.

Visual and Audio Effects – The application must play sounds when a move is made and it should also have basic animation for dragging of piece for a smoother user experience.

### *Game Modes & Features*

Player-vs-Player Mode – The application must allow for 2 players to play each other on the same computer system where the turn is automatically altered after a move is made

AI Opponent – The application should include an option to play againts the built in AI engine or any external engines like stockfist

Multiplayer Support – The game should allow user to start a match online with other player and friends.

## 2.2 Non-Functional Requirements

### *Performance & Efficiency*

Smooth Graphics Rendering – The chessboard and pieces must be rendered efficiently using without noticeable lag or stuttering.

Low Resource Usage – The application should run with minimal CPU and memory consumption to ensure smooth performance on mid-range systems

Fast Move Processing – Moves should be validated and executed within milliseconds to maintain a seamless user experience.

### *Usability & Accessibility*

User-Friendly Interface – The game UI must be intuitive, allowing players to easily select and move pieces.

Clear Visual Cues – Highlighting of legal moves, selected pieces, and check/checkmate conditions must be easily distinguishable.

Configurable Settings – Players should be able to adjust board themes, piece styles, and other UI preferences.

Keyboard Shortcuts – Key bindings for actions like undo, redo, and board flip should be available.