Software Requirements Specification

for

Chess & Checkers Game

**Version 1.0 approved**

**Prepared by Group No. 5**

**Section No. 3**

**14th October 2021**

**Table of Contents**

[Introduction](#_1fob9te) **1**

[Purpose](#_kwodr589cn02) **1**

[Document Conventions](#_lppwpncz50r8) 1

[Product Scope](#_3dy6vkm) 1

[References](#_1t3h5sf) 1

[Overall Description](#_4d34og8) **2**

[Product Perspective](#_2s8eyo1) 2

[Product Functions](#_17dp8vu) 2

[User Classes and Characteristics](#_3rdcrjn) 2

[Operating Environment](#_26in1rg) 3

[Design and Implementation Constraints](#_lnxbz9) 3

[User Documentation](#_35nkun2) 3

[Assumptions and Dependencies](#_1ksv4uv) 4

[External Interface Requirements](#_83wqf2vt2apy) **4**

[3.1 User Interfaces](#_mjst0r1ma5vs) 4

[3.2 Hardware Interfaces](#_g6avl79fep4) 4

[3.3 Software Interfaces](#_30b2oyr25dx) 4

[3.4 Communications Interfaces](#_bsr1b6tpfapz) 4

[System Features](#_shiryipt15fi) **5**

[Account Configuration](#_2xcytpi) 5

[Game Selection](#_3t6d5b7y1lff) 10

[Main Menu](#_tj29ssxnrrho) 12

[Settings](#_yfnpjlxtiin0) 17

[Online Multiplayer mode](#_d3etowxykkev) 23

[Offline Multiplayer mode](#_13pvsa87u4uf) 28

[Single Player mode](#_sv9m1jg5phe8) 31

[Game Play - ibrahim](#_a27xrlpoy9kl) 33

[Ingame Options - ahmad](#_8r9d84wwutzx) 40

[Other Nonfunctional Requirements](#_3whwml4) **45**

[Performance Requirements](#_2bn6wsx) 45

[Safety Requirements](#_qsh70q) 45

[Security Requirements](#_3as4poj) 45

[Software Quality Attributes](#_1pxezwc) 46

[Business Rules](#_49x2ik5) 46

[Other Requirements](#_2p2csry) **46**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Qiblatain | 14-10-21 | Document Creation | 1.0 |
|  |  |  |  |

# Introduction

# Purpose

This document specifies all the software requirements for the Chess/Checkers game software. These requirements relate to the functionality, constraints, performance, attributes, and the user interface.

This software certainly focuses on providing users with an interactive game experience. Goals of this software would be to provide online as well as offline multiplayer experience, along with an option to play against the bot in offline mode, meanwhile also enabling the users to customize their boards and select avatars of their choice.

## Document Conventions

The body of this document is written in Arial Font with a font size of 18 for the headings, 14 for the sub-headings and 11 for the main text.Bullet points are added to clarify the objectives and numbered bullets are added to quantify them.

## Intended Audience and Reading Suggestions

This SRS document is designed to help project managers and project testers to get an idea of the functionality of both of the games, of how they will work and be executed and what exactly can we expect from the system. The intended audience to play the games is literally everyone who wants to explore and play the two games, which means that it is meant for the public.

Section 2 provides an overall description and Section 4 contains use cases which provides a visual explanation of the applications. Meanwhile Section 5 and 6 contain Non Functional Requirements and all Other Requirements respectively.

## Product Scope

This application allows players to engage in a real-time game of chess and checkers. Players have the options of playing online and offline multiplayers and also with our advanced AI providing a realistic game experience. Each move of the player is administered and the constraints are checked accordingly in both of the games, making sure that the games function with only the legal moves. It also has features such as changing board themes and choosing different player avatars based on the game mode you are playing in.

## References

* Laws of Chess: https://handbook.fide.com/chapter/E012018
* Regulations for playing chess with Computers : https://handbook.fide.com/chapter/E03
* Online Chess Regulations : https://handbook.fide.com/chapter/OnlineChessRegulations
* Rules of Checkers : <https://www.ultraboardgames.com/checkers/game-rules.php>
* <https://www.slideshare.net/Axphey/the-complete-srs-documentation-of-our-developed-game>
* https://github.com/brandonalfred/ChessGame/blob/master/SRS.md

# Overall Description

## Product Perspective

Our product includes a chess and checkers simulator for the game as demanded by the client, based on the actual chess and checkers game. It is a self-contained product with no relation to any other product. Chess and checkers being two separate games with no similarities in between them except for the board layout, it is divided into two sub interfaces of chess and checkers. The user can switch between them through the menu provided at the start of the application. The external interfaces may include the main server that will be entertaining the requests coming through such as login authentication and handling API requests for the multiplayer game sessions. Besides these, the user will be interacting with the application either through a touch screen, a keyboard or a mouse. It is for those who have access to laptops/PCs/smartphones.

## Product Functions

* Signup/login for the user.
* The games: chess and checkers.
  + Single player and multiplayer.
  + Record each move of both the players in multiplayer and in single player.
  + A total timer.
  + Shows the user all the legal moves that are possible.
  + In each turn it checks for the winning conditions, the game either ends in a win for the user, loss for the user or win.
  + Allows the pieces to transform on specific conditions.

1. In Chess when the pawn reaches the end it can be transformed into any other piece.
2. In Checkers when the piece reaches the end, it becomes a double with extra powers.

* The users will be able to choose their own avatars from the given presets.
* The users will be able to change their board and pieces colors, from the given presets.
* A leaderboard will be provided based on user game rating.
* The user can choose from their previous games to play its replay.
* The application provides 3 difficulties to choose from, easy medium and hard.

## User Classes and Characteristics

There will be a total of two classes, Offline Player and Online Player. The functionality that the online player will be far greater as they will have access to the online server containing the database and multiplayer requests. They will be able to play online multiplayer, offline multiplayer, single player, and will be able to access replays and leaderboards. On the other hand, offline players will only have access to offline multiplayer and single player mode.

## Operating Environment

The software is needed to be a working application on a desktop as well as mobile phones. The desktop app can be implemented using java, it would be able to run on both windows and mac os using the java virtual machine. The android app can be developed on the native android sdk using the java language. The IOS application will be developed on swift.

Desktop Application, it should have

* Windows 7, Windows 8, Windows 8.1, Windows 10 or later
* macOS 10.1 or later

In the case of Android the device should have

* Android Lollipop 5.0

## Design and Implementation Constraints

The developers would have to cater the following features:

* A database system that is running side by side to manage user accounts and replays.
* A server that will receive, process and return requests by online users to sign up, login, authenticate credentials and manage online multiplayer. It should be able to do this for a maximum of 10000 concurrent users.
* Minimizing memory and processing power usage by optimizing algorithms and avoiding unnecessary complexities, in order to make it accessible to more users and be considered a lightweight, easy to use application.
* Synchronized graphics, input/ outputs on both server and client side.
* Ensuring that all required functions are error free and bug free via beta testing.
* All interfaces and game data should be written in English.

## User Documentation

* **Chess Ruleset:** It contains details about the components and rules of chess. It will describe the movement of each piece, such as the moveset for king, along with the win conditions of the game.
* **Checkers Ruleset:** this document will state all the details concerning the game of checkers. It will define the movement process of the piece, and identify all the winning conditions.
* **App manual:** It contains details about how to operate the app. The navigation through the app including the settings available to the user will be identified and explained.

## Assumptions and Dependencies

* The desktop application is made using the Java programming language which requires a Java virtual machine to run. It is therefore assumed that the user has the necessary software installed on their device.
* To play the online multiplayer mode and also sign up the user must have an internet connection throughout the game.

# External Interface Requirements

## 3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## 3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

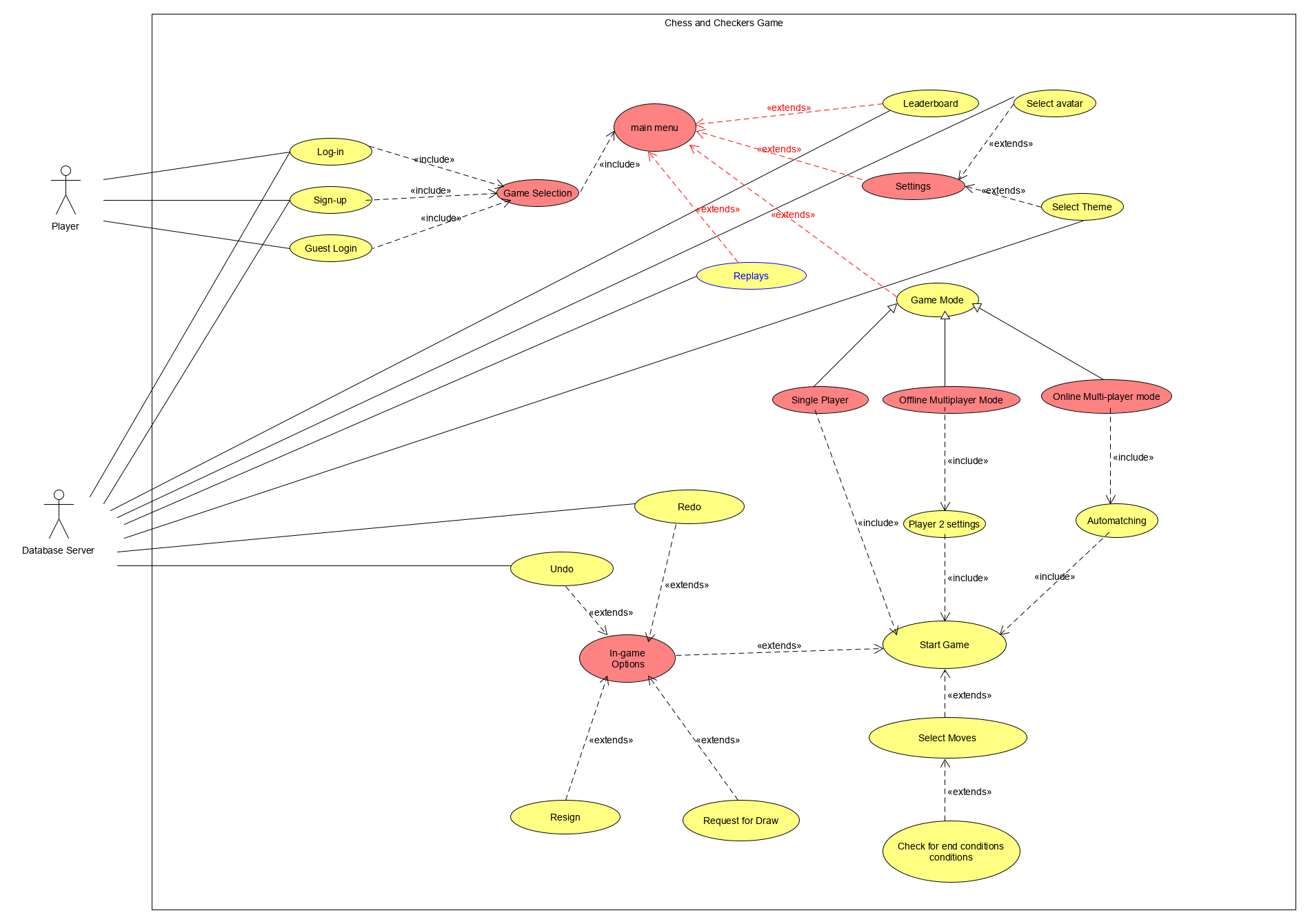
## 3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## 3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

**

## Account Configuration

**4.1.1 Description and Priority**

The feature provides a variety of login options to the user to access the application. This feature is the first page of the application when it is opened. It lets the user access the application by creating an account, using an existing account, or by using a guest account. This feature contains buttons and fields such as Login, Sign up, Guest login, Email address, and Password. It is a high priority feature as access to the application depends upon it.

**4.1.2 Stimulus/Response Sequences**

*Login Case:* When the user opens the application, they are presented with a login page. This page contains two fields: Email address and password which are needed to be filled by an existing user, and three buttons: Log in, Sign up, and Guest login. After entering the details, the user clicks on log in button. The system would verify the details from the server. Upon successful attempt, the system would return the main page of the application to the user.

When a user enters invalid information, the forgot password button appears. This provides the user an option to reset their password. When the user clicks this button a form with one field appears: Email address. When the existing user enters their Email address, it is verified by the server and upon successful verification, an email is sent to the user with the option to create a new password, user can then log in with a new password. Upon unsuccessful verification, the user is asked for a correct Email address.

*Sign-up case:* When the user opens the application, they are presented with a login page. This page contains two fields: Email address and password which are needed to be filled by the new user, three buttons: Log in, Sign up, and Guest login. After entering a correct Email address and password that matches the criteria, the user clicks on the Sign-up button. The system would create new user details on the server. Upon successful creation, the system would return the main page of the application to the user. Otherwise, an error message with the reason for sign-up failure would be displayed.

*Guest login:* When the user opens the application, they are presented with a login page. This page contains two fields: Email address and password, three buttons: Log in, Sign up, and Guest login. User can use the application without creating or using an existing account, just by clicking the Guest login option. This option would give the user a random guest id to use the application. After clicking the Guest login option, the system returns the main page of the application to the user with a guest id.

**4.1.3 Functional Requirements**

***REQ 1.1:*** User would have the option to log in

***REQ 1.2:*** User can login using Gmail account or Facebook account

***REQ 1.3:*** User can login using email address and password

***REQ 1.4:*** Upon entering the details, system will cross check the data with the server and if correct, system will give user the access to application

***REQ 1.5:*** System would show an error upon invalid details

***REQ 1.6:*** Forgot password button is displayed if login attempt fails.

***REQ 1.7:*** User would be given an option to request for new password

***REQ 1.8:*** System would send an email to user to reset password

***REQ 1.9:*** User would have the option to sign up

***REQ 1.10:*** Upon entering the details, system would verify the email address and would create an account of the user in the server

***REQ 1.11:*** Access to application would be granted after successful account creation. Otherwise, it will show an error with the reason of sign-up failure.

***REQ 1.12:*** User would have an option to log in as a guest

***REQ 1.13:*** Upon clicking guest login, user would be given random guest id to use the application

***REQ 1.14:*** User needs to be connected to the internet.

| **Use case name** | | **Log in** |
| --- | --- | --- |
| Related requirements | | REQ 1.1, REQ 1.2, REQ 1.3, REQ 1.4, REQ 1.5, REQ 1.6, REQ 1.7, REQ 1.8, REQ 1.14 |
| Goal in context | | An existing user can log in |
| Preconditions | | User must exist |
| Successful end condition | | User gets logged in |
| Failed end condition | | User couldn’t be authenticated |
| Primary actors | | User |
| Secondary actors | | Database server |
| Trigger | | An existing user asks to log in |
| Included cases | | Game selection |
| Main flow | step | Action |
|  | 1 | User enters their credentials |
|  | 2 | Clicks on log in button |
|  | 3 | Server checks and authenticates user details |
|  | 4 | User is successfully logged in on correct credentials |
|  | 5 | Log in is dismissed on wrong credentials |
|  | 6 | Forgot password button appears, user clicks to change the password |
|  | 7 | User enters their email address |
|  | 8 | Server checks and authenticates user email address |
|  | 9 | An email is sent by the system to the user with password reset option |
|  | 10 | User password is successfully changed |

| **Use case name** | | **Sign up** |
| --- | --- | --- |
| Related requirements | | REQ 1.9, REQ 1.10, REQ 1.11, REQ 1.14 |
| Goal in context | | A new user can sign up |
| preconditions | | User does not exist |
| Successful end condition | | New user created |
| Failed end condition | | New user not created |
| Primary actors | | User |
| Secondary actors | | Database server |
| trigger | | A new user clicks on sign up |
| Included cases | | Game selection |
| Main flow | step | Action |
|  | 1 | A new user asks to sign up |
|  | 2 | User enters his details |
|  | 3 | User details are created in the server if the user details don’t exist |
|  | 4 | New user is created |

| **Use case name** | | **Guest login** |
| --- | --- | --- |
| Related requirements | | REQ 1.12, REQ 1.13, REQ 6.2 |
| Goal in context | | User can use the application as a guest |
| preconditions | | User does not want to use an account |
| Successful end condition | | New guest id is created |
| Failed end condition | | Null |
| Primary actors | | User |
| Secondary actors | | Null |
| trigger | | User clicks on guest login |
| Included cases | | Game selection |
| Main flow | step | Action |
|  | 1 | User clicks on guest login |
|  | 2 | Random guest id is created |
|  | 3 | User can access the application with that guest id |

## Game Selection

**4.2.1 Description and Priority**

To choose between chess and checkers we need to give the user a choice to make a selection. This is a high priority task, as when the user selects their game choice, this selection will be the basis upon which all future selections are based.

**4.2.2 Stimulus/Response Sequences**

*Game selection case:* There will be a menu provided to the user to select between chess and checkers. It will lead to somewhat different main menus for the chess and checkers game.

**4.2.3 Functional Requirements**

***REQ 2.1:*** Game selection window shall be displayed after the player logs in.

***REQ 2.2:*** System shall display two options for the game:

1. Chess

2. Checkers

***REQ 2.3:*** The player shall be able to select the option of their choice.

***REQ 2.4:*** Main Menu shall be displayed when an option is selected

| **Use case name** | | **Game Selection** |
| --- | --- | --- |
| Related requirements | | REQ 2.1  REQ 2.2  REQ 2.3  REQ 2.4 |
| Goal in context | | The Player can select what game to play: Chess or Checkers |
| Preconditions | | The player must be logged in. |
| Successful end condition | | The Player has selected what game to play. |
| Failed end condition | | The player has not selected the game for different reasons and an error message is displayed |
| Primary actors | | Player |
| Secondary actors | | Null |
| Trigger | | Player has logged in successfully |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | After the user is logged in, a game selection window is displayed. |
|  | 2 | System displays two options:  1. Chess  2. Checkers |
|  | 3 | Player selects what game to play by clicking on the option. |
|  | 4 | After the player selects the game, the Main Menu page is displayed. |

## Main Menu

**4.3.1 Description and Priority**

The main menu provides a variety of options to the user about what they want to do in the application. It is where the user ends up after logging in, and the main window from which they can navigate to other features of the application. It contains icons and buttons which will direct the user to other windows such as Replays, Leaderboard, Settings and Game Mode. It is a high priority feature, since it would be very difficult to navigate to other parts of the application without the main menu.

**4.3.2 Stimulus/Response Sequences**

*Replays Case:* When the user clicks on the Replays option, a new window opens up. This window contains all the games played by the user of the current logged-in account, identified by a timestamp. The user may select a game to view the replay of all moves made during that game. A back button will be available on the Replays screen. Pressing the back button will redirect the player back to the main menu.

*Leaderboard Case:* When the user selects the Leaderboard option, they are directed to the Leaderboard window, where the current standings of all players are listed. The players with the greatest stats are shown at the top of the Leaderboard. When the player is ready to go back to the main menu, they can click the back button provided, which will redirect them to the main menu.

*Game Mode Case:* If the game mode option is clicked, the Game Mode window opens, which will specify all the game modes available to the player. If the player is currently connected to the internet, the options shown will be: Offline Multiplayer, Online Multiplayer, and Single Player. If the player is offline, the visible game mode options will be: Offline Multiplayer, and Single Player. A back option will be provided here. Pressing the back button will take the user back to the main menu window.

*Settings Case:* A settings option will be available to the users. When selected, the user will be directed to the settings window, where he can select an avatar and choose a board theme. Further details about this case, along with the use case table are present in Section 4.4.

**4.3.3 Functional Requirements**

***REQ 3.1:*** A “replays” icon must show up on the main menu screen.

***REQ 3.2:*** User shall be directed to replays screen upon pressing icon.

***REQ 3.3:*** There should be a list of all previous games on the replays screen.

***REQ 3.4:*** The user can select any of their previous games from the options in the list.

***REQ 3.5:*** Upon selection of a game, the user can view all moves made during the game.

***REQ 3.6:*** There should be a back option on the replays page

***REQ 3.7:*** When the back button is pressed, the user is taken back to the main menu page.

***REQ 3.8:*** A “Leaderboard” icon must show up on the main menu screen.

***REQ 3.9:*** User shall be directed to the leaderboard screen upon pressing the icon.

***REQ 3.10:*** There shall be a list of the current standings of all players on the leaderboard screen.

***REQ 3.11:*** There should be a back option on the Leaderboard page.

***REQ 3.12:*** A Game Modes option must show up on the main menu screen.

***REQ 3.13:*** Player can request to select the game mode by pressing the option.

***REQ 3.14:*** Player shall be directed to another screen with the available game modes, depending on their network connectivity.

***REQ 3.15:*** If the user is online, the options available are: Single Player, Offline Multiplayer, and Online Multiplayer.

***REQ 3.16:*** If the user is offline, the options available are Single Player and Offline Multiplayer.

***REQ 3.17:*** There should be a back option on the Game Modes page.

| **Use case name** | | **Replays** |
| --- | --- | --- |
| Related requirements | | REQ 3.1, REQ 3.2, REQ 3.3, REQ 3.4, REQ 3.5, REQ 3.6, REQ 3.7 |
| Goal in context | | Player can view a record of all moves made in previously played games. |
| Preconditions | | Player must be on the main menu and have pressed the replay icon. |
| Successful end condition | | User can view previous game logs. |
| Failed end condition | | Replays do not open, and an error message is shown. |
| Primary actors | | Player |
| Secondary actors | | Database server |
| Trigger | | User clicks on Replays option |
| Included cases | | None |
| Main flow | Step | Action |
|  | 1 | User clicks on the “*Replays*” option |
|  | 2 | A new page opens with timestamps of all previous games. |
|  | 3 | Upon selecting the timestamp of a previous game, a log of all moves made in the game will show up. |
|  | 4 | When the back button is pressed, the user is directed back to the main menu page. |

| **Use case name** | | **Leaderboard** |
| --- | --- | --- |
| Related requirements | | REQ 3.7, REQ 3.8, REQ 3.9, REQ 3.10, REQ 3.11, REQ 1.14 |
| Goal in context | | Player can view the current leaderboard of whatever game they have selected (chess/checkers). |
| Preconditions | | Player must be on the main menu and have pressed the leaderboards icon. |
| Successful end condition | | Player can view the leaderboard. |
| Failed end condition | | Leaderboard does not open, and an error message is shown. |
| Primary actors | | Player |
| Secondary actors | | Database server |
| Trigger | | User clicks on Leaderboards option |
| Included cases | | None |
| Main flow | Step | Action |
|  | 1 | User selects the “Leaderboards” option |
|  | 2 | A new page opens with the current standings of all users with an account. |
|  | 3 | When the back button is pressed, the user is directed back to the main menu page. |

| **Use case name** | | **Game Mode** |
| --- | --- | --- |
| Related requirements | | REQ 3.7, REQ 3.12, REQ 3.13, REQ 3.14, REQ 3.15, REQ 3.16, REQ 3.17 |
| Goal in context | | The Player can view all the game modes available in the application. |
| Preconditions | | Player must be on the main menu and have selected the game modes option. |
| Successful end condition | | User can view the game mode options. |
| Failed end condition | | Game Modes screen does not open, and an error message is shown. |
| Primary actors | | Player |
| Secondary actors | | None |
| Trigger | | User clicks on game modes option |
| Included cases | | None |
| Main flow | Step | Action |
|  | 1 | User clicks on the “Game Modes” option. |
|  | 2 | A new page opens with the available game mode options, based on the Player’s internet connectivity. |
|  | 3 | If the user is online, the options available are: Single Player, Offline Multiplayer, and Online Multiplayer. |
|  | 4 | If the user is offline, the options available are Single Player and Offline Multiplayer. |
|  | 5 | When the back button is pressed, the user is directed back to the main menu page. |

## Settings

**4.4.1 Description and Priority**

Settings is a system feature that enables players to change game settings. It allows the user to change their avatar and theme of the game. The player can select the option of their choice from a list provided. It is a low priority feature as it does not affect the functionality of the game.

**4.4.2 Stimulus/Response Sequences**

*Settings Case*: When the player clicks on Settings, a new screen is displayed. This screen will have two options: Select Avatar and Select Theme. Players can click on any option they want to change. The screen will also show a back button which when clicked will redirect the player to the main menu page.

*Select Avatar Case*: When a player clicks on the Select Avatar option, a new screen is displayed. This screen will list all available avatar options for the player to choose from. The player will click on an avatar of their choice. The system will update the value of avatar in the database. The selected avatar will be displayed during the game. The screen will also show a back button which when clicked will redirect the player to the settings page.

*Select Theme Case*: When a player clicks on the Select Theme option, a new screen is displayed. This screen will list all available theme options for the player to choose from. The player will click on a theme enabling them to choose a color for the board and pieces of their choice. The system will update the value of the theme in the database. The screen will also show a back button which when clicked will redirect the player to the settings page.

**4.4.3 Functional Requirements**

***REQ 4.1:*** The system must display the Settings button on the main menu.

***REQ 4.2:*** After the user taps on the Settings button, system must show the Settings page

***REQ 4.3:*** The Settings page must list all available settings.

***REQ 4.4:*** There should be a back option that redirects the user to the main menu when clicked.

***REQ 4.5:*** The system must display a Select Avatar button on the Settings page.

***REQ 4.6:*** After the user taps on the Select Avatar button, system must show the Select Avatar page

***REQ 4.7:*** The Select Avatar page must list all available avatar options for the user to choose from.

***REQ 4.8:*** The user must be able to click on the avatar of their choice

***REQ 4.9:*** The system must update the value of the avatar in the database

***REQ 4.10:*** The system must display a Select Theme button on the Settings page.

***REQ 4.11:*** After the user taps on the Select Theme button, system must show the Select Theme page

***REQ 4.12:*** The Select Theme page must list all available theme options for the user to choose from.

***REQ 4.13*:** The user must be able to click on the theme of their choice

***REQ 4.14:*** The system must update the value of the theme in the database

***REQ 4.15:*** There should be a back option that redirects the user to the settings screen when clicked.

| **Use case name** | | **Settings** |
| --- | --- | --- |
| Related requirements | | REQ 4.1  REQ 4.2  REQ 4.3  REQ 4.4 |
| Goal in context | | The user can view the different settings available. |
| Preconditions | | 1. The user must be logged in.  2. The setting option has been clicked. |
| Successful end condition | | The user can view different settings available. |
| Failed end condition | | Setting page does not open for different reasons and an error message is displayed |
| Primary actors | | Player |
| Secondary actors | | Null |
| Trigger | | User taps on Settings. |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | User taps on the Settings button and system returns a settings page. |
|  | 2 | System displays all available settings. |
|  | 3 | User is redirected to the Main Menu page when the Back button is clicked. |

| **Use case name** | | **Select Avatar** |
| --- | --- | --- |
| Related requirements | | REQ 4.5  REQ 4.6  REQ 4.7  REQ 4.8  REQ 4.9  REQ 4.15 |
| Goal in context | | The Player can select an avatar for their profile from a list of options. |
| Preconditions | | 1. The setting option has been clicked.  2. The Select Avatar option has been clicked. |
| Successful end condition | | The Player has selected an avatar of their choice. |
| Failed end condition | | Avatar is not selected for different reasons and an error message is displayed |
| Primary actors | | Player |
| Secondary actors | | Database server |
| Trigger | | Player taps on Select Avatar. |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | User taps on the Select Avatar button and a new page is returned. |
|  | 2 | System displays all available avatar options for the user to choose from. |
|  | 3 | User selects an avatar by clicking on it. |
|  | 4 | System updates the value of the theme in the database. |
|  | 5 | The user is redirected to the Settings page when the Back button is clicked. |

| **Use case name** | | **Select Theme** |
| --- | --- | --- |
| Related requirements | | REQ 4.10  REQ 4.11  REQ 4.12  REQ 4.13  REQ 4.14  REQ 4.15 |
| Goal in context | | The user can select a color theme for their game from a list of options. |
| Preconditions | | 1. The setting option has been clicked.  2. The Select Theme option has been clicked. |
| Successful end condition | | The user has selected a theme of their choice. |
| Failed end condition | | Theme is not selected for different reasons and an error message is displayed |
| Primary actors | | Player |
| Secondary actors | | Database server |
| Trigger | | User taps on Select Theme. |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | User taps on the Select Theme button and a new page is returned. |
|  | 2 | System displays all available color theme options for the user to choose from. |
|  | 3 | User selects a theme by clicking on it. |
|  | 4 | System updates the value of the theme in the database. |
|  | 5 | The user is redirected to the Settings page when the Back button is clicked. |

## Online Multiplayer mode

**4.5.1 Description and Priority**

This game mode allows a user to play an online game and match with players around the globe. Two players will be auto-matched according to their availability and when the players requested to join the multiplayer game. A queue will be utilised for this purpose, and the player at the front of the queue will be matched with our active player. An in-game settings button will also be available to provide additional features to the user during the game.

**4.5.2 Stimulus/Response Sequences**

*Online Multiplayer mode case:* as the user enters this mode his login and connection to the server is checked. A screen pops up to decide whether to play with a friend or auto-match.

*Auto-matching case:* players are matched according to their availability and rankings. A link is established while a user is displayed a window of searching animations.

*In-game Settings case:* as the user chooses settings during the game in which he can customize the boards and his avatar in the game too.

**4.5.3 Functional Requirements**

***REQ 5.1:*** User must be able to connect to the server.

***REQ 5.2:*** User must be logged in.

***REQ 5.3:*** Link must be established between two players.

***REQ 5.4:*** User must be allowed to view their previous moves during the game.

***REQ 5.5:*** User must be able exit the game.

***REQ 5.6:*** After selecting the settings button, the user must be able to access the different in-game options.

***REQ 5.7:*** One player can request a draw from the other player.

***REQ 5.8:*** the system must display an in-game settings button.

| **Use case name** | | **Online Multiplayer Mode** |
| --- | --- | --- |
| Related requirements | | REQ 5.1  REQ 5.2  REQ 5.3  REQ 5.4  REQ 1.14 |
| Goal in context | | This mode allows the user to compete against players all around the globe. |
| Preconditions | | 1. User must have clicked the Online Multiplayer Mode button.  2. User must be logged in  3. User must be connected to the server |
| Successful end condition | | User have successfully completed the match and had returned to main menu |
| Failed end condition | | 1. User may have exited the game before completing it  2. Users may not have or lost access to server connection.  3. User may not have been logged in |
| Primary actors | | Player 1 |
| Secondary actors | | Player 2 |
| Trigger | | User selects online multiplayer mode button |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | User selects online multiplayer mode |
|  | 2 | System checks whether the user has a connection to the server. |
|  | 3 | System checks whether the user has logged in |
|  | 4 | If previous conditions are true then server matches the player with some other player (i.e., player 2) which is also known as auto matching. |
|  | 5 | When both players have signed up for the game and established a connection successfully. The game has now started |
|  | 6 | An in game settings menu is displayed to provide other features during the game. |

| **Use case name** | | **Auto-matching** |
| --- | --- | --- |
| Related requirements | | REQ 5.1  REQ 5.3  REQ 1.14 |
| Goal in context | | Match players according to their position in the queue, so they can compete with each other. |
| Preconditions | | User must have selected to play online multiplayer mode |
| Successful end condition | | Two players have been matched and connection links between them are established. |
| Failed end condition | | 1. Server connection is lost  2. Player two have selected cancel button to terminate this process |
| Primary actors | | Player 1 |
| Secondary actors | | Player 2 |
| Trigger | | The Online Multiplayer option is selected. |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | User selects online multiplayer mode |
|  | 2 | The server tries to establish a connection. |
|  | 3 | Another player who also wants to play the game online is searched for. |
|  | 4 | Links are established between the two players |
|  | 5 | If the queue is empty, the player is given the option to play with a bot or exit. |
|  | 6 | Game has now started. |

| **Use case name** | | **In-game settings** |
| --- | --- | --- |
| Related requirements | | REQ 5.4  REQ 5.5  REQ 5.6  REQ 5.7  REQ 5.8 |
| Goal in context | | To provide other in-game facilities. |
| Preconditions | | The online multiplayer mode game should have been started, and user must have selected the in-game settings option. |
| Successful end condition | | The in-game settings menu is visible |
| Failed end condition | | 1. User has pressed the back button 2. User has exited the game. |
| Primary actors | | Player 1 |
| Secondary actors | | Player 2 |
| Trigger | | User has clicked the setting button. |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | User selects in-game settings |
|  | 2 | A menu showing the options - undo, redo, new game, and quit - pops up. |
|  | 3 | A user may then select any of these options. |
|  | 4 | If the user presses back, then the game screen is visible. |

## Offline Multiplayer mode

**4.6.1 Description and Priority**

This feature enables the user to play the game in an offline mode in which they can play with another player on the same device. The secondary player needs not to have an account, but they will be required to enter their name and select avatar in a later stage. This feature is of medium priority since it is one of many game modes being offered in the system. It benefits the user in such a way that they can easily play with someone else without having access to the internet.

After creating a new game in the offline multiplayer mode, the system enables the user to help identify the newly added player. The new player playing on the same device as Player 1 will be able to enter the username to identify themself and choose an avatar of their own. This setting feature is of low priority since it will not affect the functionality and the process of the game. It benefits the user in such a way that a player can be easily distinguished from another player, regardless of them playing on the same device. After configuring the settings, both the players can now play a 1v1 match.

**4.6.2 Stimulus/Response Sequences**

*Offline Multiplayer Case:* In this case, the player will be able to click on the offline multiplayer mode button and enter a customized game just for two players who can play on the same device simultaneously. The player will be able to proceed by creating a new game

*Player 2 Settings Case:* In this case, the player will be able to configure the settings for Player 2 by clicking the option before beginning the game. The player can then add a suitable username for Player 2 in the textbox. The player will be given the option to select a distinguishable avatar, just like Player 1. The player can exit the setting and start the game.

**4.6.3 Functional Requirements**

***REQ 6.1:*** User can choose offline mode from multiplayer option

***REQ 6.2:*** User needs not to be connected to the internet

***REQ 6.3:*** User shall play with another player on the same device

***REQ 6.4:*** User should select the new game button to be able to configure settings

***REQ 6.5:*** User should select the start game button to be able to actually start playing

***REQ 6.6:*** User must enter the username for Player 2 in the textbox

***REQ 6.7:*** User must select an avatar for Player 2 from the given options

***REQ 6.8:*** System should direct the user to previous page in case of an error

| **Use case name** | | **Offline Multiplayer Mode** |
| --- | --- | --- |
| Related requirements | | 6.1, 6.2, 6.3, 6.4, 6.8 |
| Goal in context | | User starts an offline multiplayer game |
| Preconditions | | Player may have an existing account or can be guest login, and must choose the offline multiplayer mode option from the game mode |
| Successful end condition | | New Offline game successfully starts |
| Failed end condition | | Player returns to the previous menu |
| Primary actors | | Player 1 |
| Secondary actors | | None |
| Trigger | | Player chooses offline multiplayer mode |
| Included cases | | Offline Multiplayer  Player 2 settings  Ingame settings |
| Main flow | Step | Action |
|  | 1 | Player 1 chooses the new offline multiplayer mode |
|  | 2 | Player 1 can click the button to create a new game |
|  | 3  Include: Player 2 Settings | Player 1 can configure the settings for Player 2 as per their choice |
|  | 4 | Player 1 starts a new game |
|  | 5 | Players can also access the in-game settings once the game has started. |

## 

| **Use case name** | | **Player 2 Settings** |
| --- | --- | --- |
| Related requirements | | REQ 6.6, REQ 6.7, REQ 6.8 |
| Goal in context | | The new player will be able to configure its name and avatar |
| preconditions | | Player must be in the offline multiplayer mode |
| Successful end condition | | The configurations for the new offline player are saved |
| Failed end condition | | Player returns to the previous menu |
| Primary actors | | Player 1 |
| Secondary actors | | None |
| Trigger | | Player chooses the player 2 settings option |
| Included cases | | None |
| Main flow | step | Action |
|  | 1 | The player selects the button to configure the settings |
|  | 2 | Player 2 adds their username in the text box |
|  | 3 | Player 2 selects the avatar of their choice |
|  | 4 | The configurations of Player 2 are saved temporarily for the new game |

## Single Player mode

**4.7.1 Description and Priority**

This feature enables the user to play with the computer. The user will be given a choice of difficulty level they wish to play in. This feature is of medium priority since it is one of many game modes being offered in the system. It benefits the user in such a way that they can easily play with the computer if they do not have another player to play with. It benefits the user in such a way that they can easily play with the computer without having access to the internet.

**4.7.2 Stimulus/Response Sequences**

*Single Player Case:* The player, after selecting the single player mode from the game modes, is given an option to select the difficulty level of the game to play with the computer. They can choose between an easy/medium/hard level and then start the game.

**4.7.3 Functional Requirements**

***REQ 7.1:*** User can choose single player mode from game modes.

***REQ 7.2:*** User should select the new game button to be able to configure settings

***REQ 7.3:*** User should select the start game button to be able to actually start playing

***REQ 7.4:*** User should select easy/medium/hard difficulty level

***REQ 7.5:*** System should direct the user to previous page in case of an error

| **Use case name** | | **Single Player** |
| --- | --- | --- |
| Related requirements | | REQ 7.1, REQ 6.2, REQ 7.2, REQ 7.3, REQ 7.4, REQ 7.5 |
| Goal in context | | The new player will be able to play with the computer |
| preconditions | | Player may have an existing account or can be guest login and must choose the single player mode option from the game modes |
| Successful end condition | | New Offline game successfully starts with the computer |
| Failed end condition | | Player returns to the previous menu |
| Primary actors | | Player 1 |
| Secondary actors | | None |
| Trigger | | Player chooses single player mode |
| Included cases | | Start Game |
| Main flow | step | Action |
|  | 1 | The player selects the single player mode |
|  | 2 | The player selects the difficulty level |
|  | 3 | The player can then start the game with the computer |

## Game Play

**4.8.1 Description and Priority**

This will be the most important aspect of the software, as the main game will be played via this feature. It encompasses everything from the beginning of a new game, to every move that can be made, to rulesets, timer, and in game options. This will also have to check whether or not the game is terminated at each turn. It should be kept in mind that the game rules and moves will work very differently for each game, and both will have very different termination conditions.

**4.8.2 Stimulus/Response Sequences**

Start game case: In this case the game will be initialized, this will be responsible for things like creating the board, displaying the timer, showing the player name alongside their avatar, selecting the color of each user i.e. which player will own the black/ white pieces. The person with the white pieces will be prompted to begin his first move. The timer will be set as soon as the first move takes place

Highlight Legal Options Case: The user will be shown the legal options of moving each piece when they select that specific piece so that they can consider only the relevant possible moves.

* In the case of chess these legal options will have to take into consideration things like first pawn move, en passant, pawn promotion and castling.
* In the case of chess these legal options must take into account the possibility of capturing an opponent's piece in which case the user must perform any possible capture.

*Select Move Case:*

*Ingame Settings Case:* This action will allow the user to alter certain settings or perform certain holistic actions while playing the game. The possible options will include redo, undo, resign, request for draw and (pause game in case of offline multiplayer / single player).

*Check For End Conditions Case:* After each move has occurred, the system will check whether certain end conditions have been met or not. If they have been met, the game will then display whether the match has resulted in a win or a loss and show the results to the players accordingly.

**4.8.3 Functional Requirements**

***REQ 8.1:*** User should be shown the board and pieces

***REQ 8.2:*** User should be shown the timer

***REQ 8.3:*** The game should prompt the user with the white pieces to select a move

***REQ 8.4:*** The game should start a timer that will expire if the first move is not selected in the timelimit

***REQ 8.5:*** The game should start the timer after the first move is done

***REQ 8.6:*** The game show allow the user to select a given piece.

***REQ 8.7:*** The game should show each possible legal move for a given piece when that piece is selected

--CHESS

***REQ 8.7.1.1:*** King piece can move one tile in any direction, unless that tile is under attack by an enemy piece.

***REQ 8.7.1.2:*** Queen piece is allowed to move in any direction, that is, diagonally, horizontally and vertically, for upto a maximum of 8 tiles, unless its movement is blocked by a piece.

***REQ 8.7.1.3:*** Bishop can move along any diagonal direction for a maximum of 8 tiles, unless its movement is blocked by a piece

***REQ 8.7.1.4:*** Bishop can move horizontally or vertically for a maximum of 8 tiles, unless its movement is blocked by a piece .

***REQ 8.7.1.5:*** Knights are able to move in L direction ie two steps horizontally or vertically, and then one step perpendicular to the first two.

***REQ 8.7.1.6:*** Pawns are able to move two steps forward if only it is their first move, else they can only move one step forward. They can attack other pieces in forward diagonal position only and that too one step only. If a pawn reaches the end of the board they are able to promote themselves to any other piece except for a king.

***REQ 8.7.1.7:*** Castling can happen if the king and the target rook for the player’s side have not moved since the beginning of the match, and will allow the king to move 2 steps towards the rook closer to it, or three steps towards the rook farther from it, and will place the rook adjacent to the king, such that it crosses over the king. If such a move leads to the king being threatened by an enemy piece, it will be considered illegal.

***REQ 8.7.1.8:*** En passant is a special capture that happens when one side moves their pawn two tiles forward, allowing an opposing pawn to capture it if would have been able to do so had the pawn only moved one tile forward.

***REQ 8.7.1.9:*** If the king is under threat by any enemy piece, then the player must perform a move that negates that threat.

-- CHECKERS

***REQ 8.7.2.1:*** A normal piece (called man) is allowed to move one tile diagonally forward, if the destination tile is unoccupied

***REQ 8.7.2.1:*** A man may capture an opposing piece by jumping over an enemy piece that is front, diagonally adjacent to it, onto an unoccupied tile which is another step in the same direction

***REQ 8.7.2.3:*** A man must capture an opposing man if it can, and is allowed to make multiple captures in the same turn as long as one capture directly allows the next one

***REQ 8.7.2.4:*** A man will be promoted into a king if it reaches the first row of the opponent's side

***REQ 8.7.2.5:*** A king functions as a man with the additional power of being able to move and capture diagonally backwards as well

***REQ 8.8:*** The game should highlight the piece that is selected (via clicking)

***REQ 8.9:*** The game should check whether the move performed is amongst the possible legal moves, and not validate the move if it is illegal (keep it at its original position)

***REQ 8.10:*** The game should move the selected piece to the selected destination tile

***REQ 8.11:*** For chess, the game should show a list of all moves that have been performed during the game

***REQ 8.12:*** When a pawn promotes, the game should show a pop up allowing the user to select the promotion

***REQ 8.13:*** The game should display the eliminated pieces of each side

***REQ 8.14:*** The game should have an in game menu that allows the user to redo, undo, resign or request for a draw

***REQ 8.14.1:*** Redo will allow the user to reverse an undo.

***REQ 8.14.2:*** Undo allows the user to revert the last performed move, transferring the turn to the player who last moved. multiple undos can be performed to repeat this process.

***REQ 8.14.3:*** Resign allows the user to end the game by resigning, resulting in a forfeit, and the other user winning.

***REQ 8.14.4:*** A user can request for a draw, which prompts the other user to either accept the draw or keep playing, with mutual consent the game can be terminated ending in a draw.

***REQ 8.15:*** The game should check at the end of each turn whether or not the game has ended, according to the rules of the game being played.

***REQ 8.15.1.1:*** If any player’s king is under threat o

***REQ 8.16:*** The game should display which side has won and lost or that the match resulted in a draw.

| **Use case name** | | **Start game** |
| --- | --- | --- |
| Related requirements | | REQ 8.1, REQ 8.2, REQ 8.3, REQ 8.4, REQ 8.5, REQ 8.11, REQ 8.12, REQ 8.13 |
| Goal in context | | The game of chess/ checkers would begin |
| preconditions | | The player(s) must have logged in and the matchmaking been successful in case of online multiplayer |
| Successful end condition | | The game is finished with the result being either a draw, win or loss |
| Failed end condition | | The internet connection is lost during multiplayer, or the application crashes during the game |
| Primary actors | | Player 1 |
| Secondary actors | | Player 2 (in case of online multiplayer) |
| Trigger | | Player selects the relevant options after choosing the selected game mode |
| Included cases | | ingame settings  select move  check for end conditions |
| Main flow | step | Action |
|  | 1 | As soon as the game starts a 1 minute timer starts |
|  | 2 | User is prompted to make a move |
|  | 3 | If the user does not make the first move then the game ends in a forfeit. |
|  | 4 | Otherwise, the game will continue as normal and the other player’s turn will start |

| **Use case name** | | **Select Move** |
| --- | --- | --- |
| Related requirements | | REQ 8.6, REQ 8.7, REQ 8.8, REQ 8.9, REQ 8.10 |
| Goal in context | | The user will be able to select a move, see the possible moves and select the destination tile |
| preconditions | | The game should have started |
| Successful end condition | | The user selects a destination tile that allows a legal move |
| Failed end condition | | none |
| Primary actors | | Player 1 |
| Secondary actors | | Player 2 |
| Trigger | | Player clicks on a piece |
| Included cases | | none |
| Main flow | step | Action |
|  | 1 | The player clicks on one of his pieces |
|  | 2 | The player will be shown the possible moves for that pieces |
|  | 3 | The player will select a destination tile |
|  | 4 | If the selection is valid the move will take place |
|  | 5 | The turn of the opposing player will begin |

| Use case name | | In game settings |
| --- | --- | --- |
| Related requirements | | REQ 8.14 |
| Goal in context | | Provides the user with the options available during the game. |
| preconditions | | The game should have begun |
| Successful end condition | | An action in the in game settings menu is selected |
| Failed end condition | | none |
| Primary actors | | Player 1 |
| Secondary actors | | none |
| Trigger | | Player selects the ingame options menu. |
| Included cases | | Redo, Undo, |
| Main flow | step | Action |
|  | 1 | The player selects the ingame options, a new window with options opens up |
|  | 2 | Player selects the required option |
|  | 3 |  |

| Use case name | | Check for ending conditions. |
| --- | --- | --- |
| Related requirements | | REQ 7.1, REQ 7.2, REQ 7.3, REQ 7.4, REQ 7.5, REQ 7.6 |
| Goal in context | | The new player will be able to play with the computer |
| preconditions | | Player may have an existing account or can be guest login and must choose the single player mode option from the game modes |
| Successful end condition | | New Offline game successfully starts with the computer |
| Failed end condition | | Player returns to the previous menu |
| Primary actors | | Player 1 |
| Secondary actors | | None |
| Trigger | | Player chooses single player mode |
| Included cases | | Single Player |
| Main flow | step | Action |
|  | 1 | The player selects the single player mode |
|  | 2 | The player selects the difficulty level |
|  | 3 | The player can then start the game with the computer |

## In-game Options

**4.9.1 Description and Priority**

This feature enables users to choose from different Ingame options such as redo,undo,resign, or request a draw from the game. These are all high priority features since all these reflect how the game turns out when being played.

**4.9.2 Stimulus/Response Sequences**

*Redo Case:* In this case the user is given the option to Redo the previous move he has used Undo on. This is allowed in Single Player Mode meanwhile in Multiplayer mode permission from the second player is required.

*Undo Case:* In this case the user is given the option to Undo the previous move after seeking permission from the second player when in Multiplayer mode. Otherwise move is allowed when in Single Player mode.

*Resign Case*: In this case the user gets the option to exit the game directly resulting in a loss and the win is given to the opponent.

*Request a Draw Case:* In this case the user is given the option to draw the game after both players agree to the draw. This is only available in Multiplayer mode.

**4.9.3 Functional Requirements**

***REQ 9.1 :*** The game should show an ingame options button.

***REQ 9.2 :*** When the user taps on ingame options the game shows all ingame options.

***REQ 9.3 :*** There should be an exit button to return back to the game.

***REQ 9.4 :*** The system should display the Redo option in ingame options.

***REQ 9.5 :*** When the user selects the Redo option, send a request to the second user to allow move.

***REQ 9.6 :*** If permission is granted then implement Redo the move otherwise continue the game.

***REQ 9.7 :*** The system should display the Undo option in ingame options.

***REQ 9.8 :*** When the user selects the Undo option, send a request to the second user to allow move.

***REQ 9.9 :*** If permission is granted then implement Undo the move otherwise continue the game.

***REQ 9.10 :*** The system should display the Resign option in ingame options.

***REQ 9.11 :*** When the user selects the Resign option the game should exit.

***REQ 9.12 :*** The system should display the Request for Draw option in ingame options.

***REQ 9.13 :*** When the user selects the Request for Draw option, send a request to the second user for a Draw.

***REQ 9.14 :*** If permission is granted then implement Draw the game otherwise continue the game.

| **Use case name** | | **Redo** |
| --- | --- | --- |
| Related requirements | | REQ 9.1, REQ 9.2, REQ 9.3, REQ 9.4, REQ 9.5, REQ 9.6 |
| Goal in context | | A use case to redo a move in game |
| Preconditions | | User is playing a game |
| Successful end condition | | Redo is applied to a move |
| Failed end condition | | Player 2 rejects move or Undo not applied |
| Primary actors | | Player 1 |
| Secondary actors | | Player 2 ( only in multiplayer game )  Database server |
| Trigger | | Player taps on Redo move button |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | Player 1 Selects Redo from In-game options |
|  | 2 | If Player 2 agrees then Redo is applied otherwise Redo is not allowed. ( Only for Multiplayer ) |
|  | 3 | Redo is applied to move in Single Player Mode |

| **Use case name** | | **Undo** |
| --- | --- | --- |
| Related requirements | | REQ 9.1, REQ 9.2, REQ 9.3, REQ 9.7, REQ 9.8, REQ 9.9 |
| Goal in context | | A use case to undo a move in game |
| Preconditions | | User is playing a game |
| Successful end condition | | Undo is applied to a move |
| Failed end condition | | Player 2 rejects move and first move of the game |
| Primary actors | | Player 1 |
| Secondary actors | | Player 2 ( only in multiplayer game )  Database server |
| Trigger | | Player taps on Undo move button |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | Player 1 Selects Undo from In-game options |
|  | 2 | If Player 2 agrees then Undo is applied otherwise Undo not allowed. ( Only for Multiplayer ) |
|  | 3 | Undo is applied to move in Single Player Game |

| **Use case name** | | **Resign** |
| --- | --- | --- |
| Related requirements | | REQ 9.1, REQ 9.2, REQ 9.3, REQ 9.10, REQ 9.11 |
| Goal in context | | A use case to Resign from a game in play |
| Preconditions | | User is playing a game |
| Successful end condition | | Player resigns from game |
| Failed end condition | | None |
| Primary actors | | Player |
| Secondary actors | | None |
| Trigger | | Player taps on Resign game button |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | Player Selects Resign from In-game options |
|  | 2 | The game ends |

| **Use case name** | | **Request for Draw** |
| --- | --- | --- |
| Related requirements | | 9.1,9.2,9.3,9.12,9.13,9.14 |
| Goal in context | | User wants to draw the game |
| Preconditions | | User is playing a multiplayer game |
| Successful end condition | | User Draws |
| Failed end condition | | Player 2 rejects draw, and game continues |
| Primary actors | | Player 1 |
| Secondary actors | | Player 2 |
| Trigger | | Player taps on Draw game button |
| Included cases | | Null |
| Main flow | Step | Action |
|  | 1 | Player 1 chooses to draw the current multiplayer game from In-game options |
|  | 2 | If Player 2 rejects, then game continues |
|  | 3 | If Player 2 accepts then game ends in Draw |

# Other Nonfunctional Requirements

## Performance Requirements

1. Player(s) shall be connected to the internet when playing in online multiplayer mode.
2. In case of network failure in the middle of the game, the system shall try to reconnect 2 times before displaying an error message.
3. In case another player is not available to play with, in the online multiplayer mode, the system will wait 2 minutes before giving them the option to either play with a bot offline or exit the game.
4. The application should not have any performance issues under normal conditions. However, if the device has a weaker hardware, it might face some difficulties or run slower.
5. The user shall be required to use a mousepad to make moves on the desktop application, and a touchpad in case of the mobile application.
6. There should be smooth animations when the players move their pieces during the game.
7. The applications should have fast loading times between turns in online multiplayer mode.
8. The application shall maintain an order of turns, making sure they switch smoothly between players.
9. When playing with an AI, the game should proceed at a pace that seems natural.
10. The application should not lag when checking for legal moves or winning conditions.

## Safety Requirements

The chess and checkers application will not cause any harm to the device or any other installed applications on the device. It will not be heavy enough in terms of memory and usage to cause overheating of the user's device, or any loss of memory.

## Security Requirements

The application simply requires you to login with your pre-existing credentials or sign up with a new account; the security of the user credentials in that case will be maintained. The application has no other security requirements concerned with private information that may compromise the security and privacy of the individuals playing the game.

## Software Quality Attributes

1. The application shall be responsive. It should work well on both desktop and mobile.
2. The User Interface shall be simple and intuitive, to ensure ease of use.
3. During game play, when a chess piece is selected, the tiles with legal positions shall be highlighted to guide users of possible moves.
4. The application shall maintain the integrity of the game by following the pre-defined set of rules.
5. The application should be easily testable manually by the developers each time an additional functionality is added.
6. The application shall be scalable in such a way that it can accommodate an increasing number of players in online multiplayer mode.
7. The probability that the system is available and operating properly whenever requested by the user should be high, with minimum risks and a lower mean failure time.
8. It should be easy to make changes in the application based on changing user requirements or changes in environment.
9. The code shall be maintainable - it should be written in such a way that it offers easy repair and improvement.
10. The system shall be robust in the sense that it provides crash-avoidance by dealing with exceptions or incorrect inputs.

## Business Rules

In case of any problems related to the application, the players will be assisted by contacting us via the email address provided.

# Other Requirements

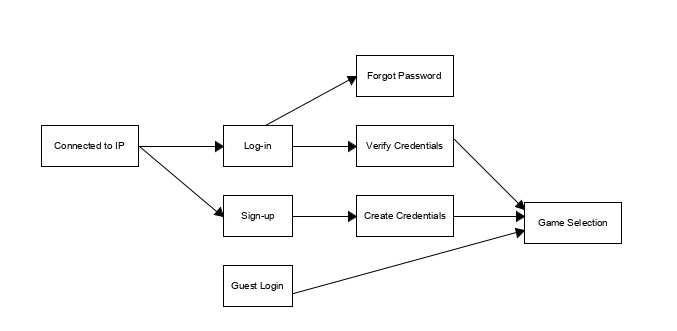
A database shall be created and maintained using the MySQL DBMS. This database will be used to keep a record of user data such as user login credentials, selected avatar, and selected theme. The database can also be used to keep a log of all games, along with the moves made by players during these games, so that replays may be fetched when requested by the user. Stats of all the players with an account will also be updated in the database, so that the Leaderboard can be updated periodically based on this data.

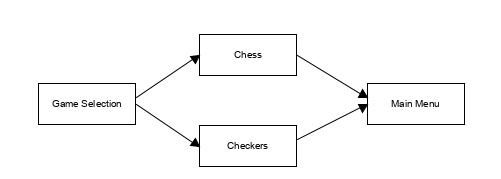
Special care will be taken to ensure that any external resource used is Open Source. No copyrighted software or resources will be used, to avoid licensing issues.

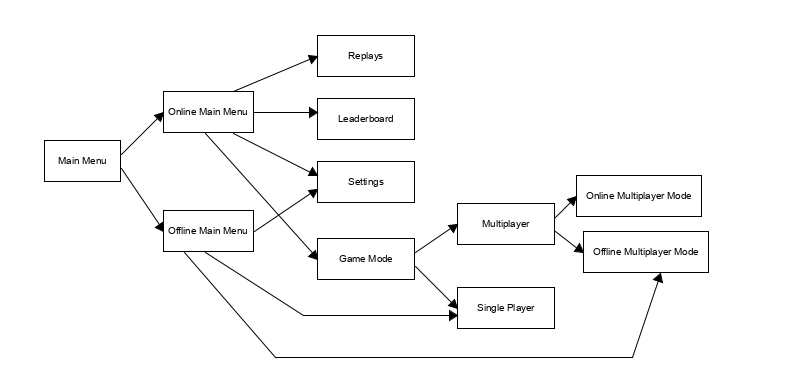
**Appendix A: Glossary**

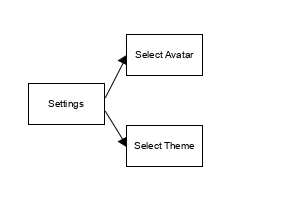
| **AI** | Artificial Intelligence. |
| --- | --- |
| **Angular** | A JavaScript-based open-source front-end web framework for developing single-page applications. |
| **API** | Application Programming Interface - a software intermediary that allows two applications to talk to each other. |
| **Beta Testing** | User testing or customer validation. |
| **Bot** | A software application that is programmed to do certain tasks. |
| **Bug** | A failure or a flaw in the software program. |
| **Client** | A system that uses remote services from a server. |
| **Database** | A system for storing and taking care of data. |
| **DBMS** | Database Management System. A software designed to store, retrieve, define, and manage data in a database. |
| **Functional Requirement** | A description of the service that the software must offer. |
| **Functions** | A set of instructions in a program, bundled together to achieve a specific outcome. |
| **Java** | An Object Oriented programming language and computing platform. |
| **Javascript** | An Object Oriented scripting or programming language that allows you to implement complex features on web pages. |
| **MySQL** | An open-source relational database management system. |
| **Network Failure** | The complete or partial failure of a component or components in a network because of malfunction or natural or human-caused disasters. |
| **Non-functional requirement** | A requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. |
| **Operating Environment** | The environment in which users run application software. |
| **PC** | Personal computer |
| **Processing Power** | The ability of a computer to manipulate data. Also known as CPU power or CPU cycles. |
| **Project Manager** | A professional playing the lead role in planning, executing, monitoring, controlling and closing projects. |
| **Project Tester** | Professionals tasked with executing and logging the tests run on a project, evaluating the results and documenting problems found. |
| **Queue** | A data structure which maintains a First-In-First-Out (FIFO) sequence of data. |
| **RAM** | Random Access Memory. A short term memory where data is stored as the processor needs it |
| **React** | A free and open-source front-end JavaScript library for building user interfaces or UI components. |
| **Server** | A system that provides services to other systems in its network. |
| **Stimulus/Response** | A change in the environment is the stimulus; the reaction of the software to it is the response. |
| **System Feature** | The “tools” you use within a system to complete a set of tasks or actions. |
| **Timestamp** | A digital record of the time of occurrence of a particular event. |
| **Use Case** | A use case is a list of actions or event steps typically defining the interactions between a role and a system to achieve a goal. |
| **User Interface** | The means by which the user and a computer system interact |
| **Webapp** | An application software that runs on a web server. |

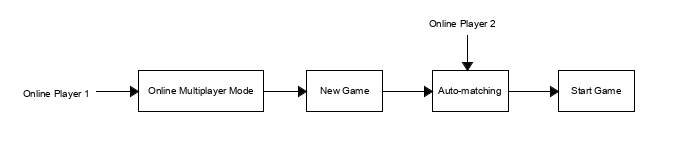
**Appendix B: Analysis Models**

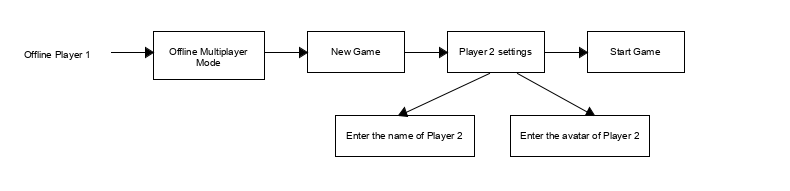
**

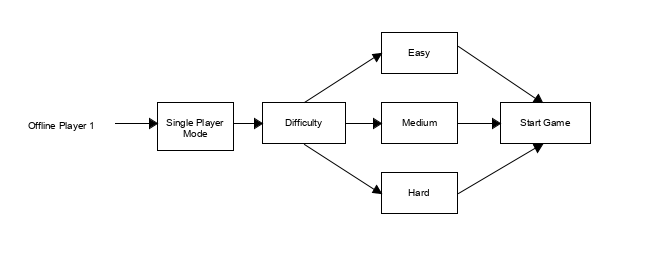
**

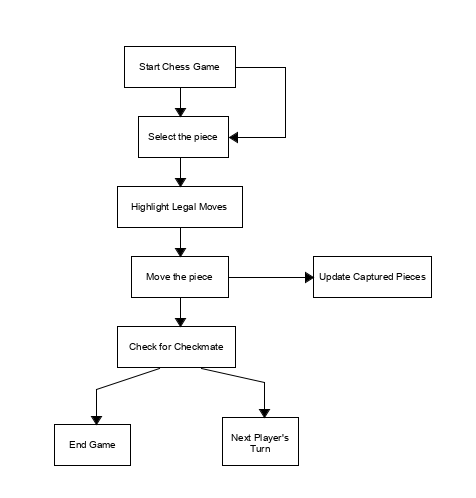
**

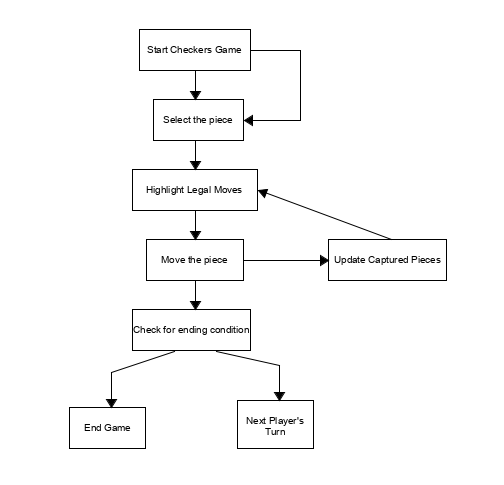
**

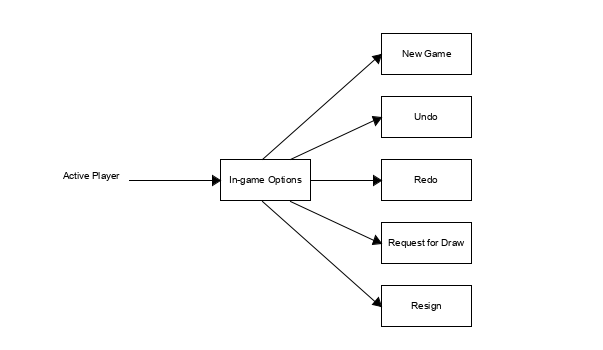
**

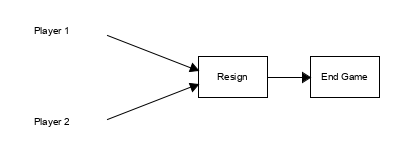
**

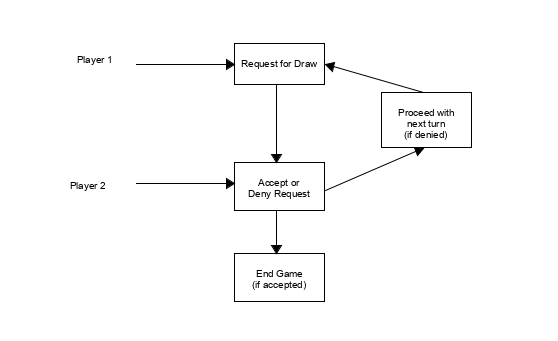
**

**

**

**

**

**

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*