Software Requirements Specification

for

Chess and Checkers

Version 1.0 approved

Prepared by <Group No. 5>

Section No. 4

10/14/2021

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 5

3. External Interface Requirements 5

3.1 User Interfaces 5

3.2 Hardware Interfaces 5

3.3 Software Interfaces 5

3.4 Communications Interfaces 5

4. System Features 6

6

4.1 Account Management 6

4.2 Play Game 12

4.3 In – Game Features 15

4.4 Pause Menu 17

4.5 Setting Menu 18

4.6 User Support 20

4.7 View Statistics 22

4.8 View Events 23

4.9 Administration 25

5. Other Nonfunctional Requirements 28

5.1 Performance Requirements 28

5.2 Safety Requirements 29

5.3 Security Requirements 29

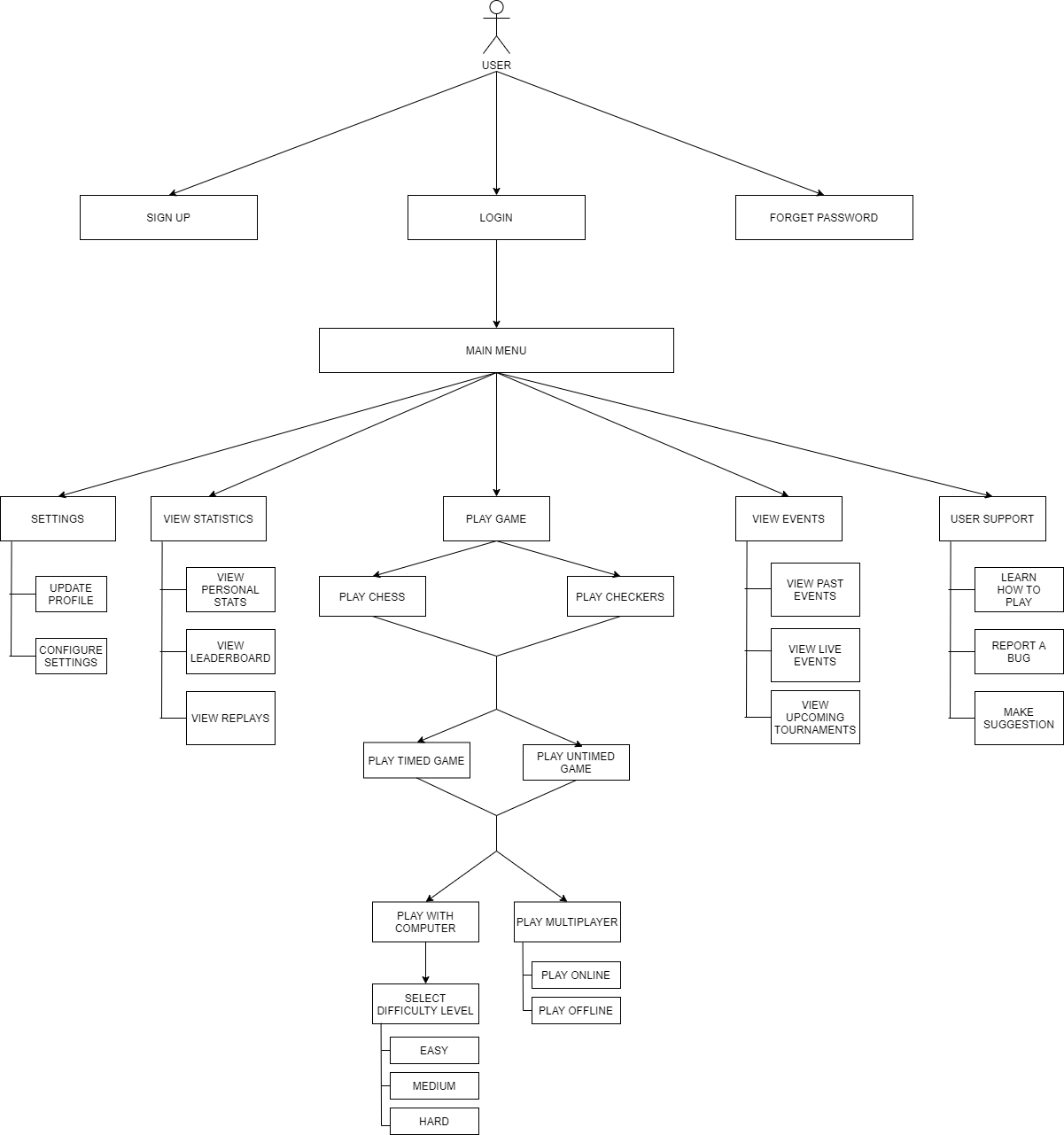
5.4 Software Quality Attributes 30

5.5 Business Rules 30

6. Other Requirements 31

Appendix A: Glossary 32

Appendix B: Analysis Models 2

 2

Appendix C: To Be Determined List 3

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This document contains detailed description of a Chess and checkers Game Application features and requirements, including but not limited to attributes, performance, functionality, constraints, usability, Interface, and the programming language/Engine used.

## Document Conventions

The body of this document is written in Times with the font size 11. Headings and Sub-headings have a font size of 14 and all Times New Roman have been highlighted to show significance.

## Intended Audience and Reading Suggestions

This document is intended for developers, project managers, and users only. For developers you should refer to section 3 and 4. Users may refer to section 2, and program managers can view the entire contents of this document. Section 3 and 4 are in most detail containing all technicalities of the software whereas section 2 provides an overview of the software. Section 5 and 6 refers to software integrity and how it should be used in a professional environment.

## Product Scope

The purpose of this game is to engage users in a single/multiplayer chess and checker’s game. With distinct difficulty levels and an engaging UI, this software allows users to compete versus other users online or on the same device or against AI. This software will allow developers to publish this game on various domains I.e., Google play store, App Store etc. And could generate revenue by placing 3rd party ads. Users will be able to login, play, track their progress and view their moves as this game will provide a feature to show the player their previous gameplays.

## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

<https://www.systemrequirementslab.com/cyri/requirements/pac-man-256/13819>

<https://www.gsmarena.com/samsung_i9300_galaxy_s_iii-4238.php>

<https://www.instructables.com/Playing-Chess/>

<https://www.chess.com/>

<https://www.ojp.gov/pdffiles1/nij/grants/249556.pdf>

<http://www.dgp.toronto.edu/~ppacheco/course/444/spec>

<http://www.sakkpalota.hu/index.php/en/chess/rules>

<https://usermanual.wiki/Pdf/ChessUserManual.1371972786/html>

<https://www.youtube.com/results?search_query=chess+tricks>

<https://lichess.org/>

# Overall Description

## Product Perspective

Chess and checkers being a vintage and a prestigious game holds its importance to its players. Users are generally tending to like strategy and tactical games. Chess and checkers allow users to compete with friends on a single device, eliminating the use of physical board. Since the constraint designs of this game are meant to follow all chess and checkers rules (time and other constraints depend on difficulty levels) there is 0 margin of any error, deception, or forgery.

## Product Functions

This game will provide the following functions:

**2.1.1 Account Management**

Signup

Users can sign up and create a new account to play the game, options available to sign up will be sign up via Gmail Facebook and Email address.

Login

Users will have to login to play the game, alternative options to log up via Facebook and Gmail are available as well.

Forget Password

Users can request to reset their password from an in-game option, an email will be sent on the registered email to reset the password

Delete Account

An option will be available to delete the account, this action will permanently delete the account.

**2.1.2 Play Game**

Chose Game (Chess or Checkers)

A choice between chess and checkers will be made available to the users

Set Difficulty Level

User can set the difficulty level according to their prefrence

Game Mode (Multiplayer or AI)

Users can also chose to play against AI or their friend(Online or on same device)

**2.1.3 User Support**

Report a bug

There will be an option to contact the developers to report bugs, Users will be asked to provide their name and email address to report bugs.

Game Feedback

Users will have an option to provide feedback to the developers.

Learn how to play

Users will be provided a manual/tutorial that will guide them with rules of the game.

**2.1.4 Settings**

Dark/Light Mode

Users will have an option to switch game UI to dark/light mode.

Sound Toggle

Users will have an option to toggle sound and music on/off.

Board color toggle

Users will have an option to change the color of the board.

Select Avatar

Users will have an option to change their profile picture avatar by choosing from our gallery or uploading their own image.

Language Toggle

Users will have an option to change the software language.

Select Piece color

Users will have an option to change the color of chess and checkers pieces.

**2.1.5 Events**

Local Chess tournaments

By using mobile location, the app will show users local chess tournaments.

Live tournaments

This app will show users all Live chess tournaments happening around the world.

Past tournaments

This app will also show results and details of all past chess tournaments happening around the world.

**2.1.6 Stats**

Win/Loss percentage

This app will have an option that will show users their stats as a win/loss percentage.

Total games played

This app will also show users their total games played.

Average game time

Average game duration will be made available to the user.

Replays

This game will record and will have an option that will make game replays available to the user.

Player Rating

Player rating will be calculated based on wins and will be made available to the user.

**2.1.7 Live features**

Hint

There will be an option to take help from AI as hints.

Undo (only AI)

Users will have an option to reverse their move, but this option will be available on games against AI.

Pause

Users will be able to pause the game for an indefinite period, but this option will be available on AI and single PC multiplayer game.

Save

Users can save their progress and resume it again, but this option will be available on AI and single PC multiplayer game.

## User Classes and Characteristics

The game should be useable by any users however basic knowledge of chess and checkers is required to play this game **efficiently**. There is an in-game interface that allows user to make decisions. A small chess and checkers tutorial will be enough for users to play a game.

## Operating Environment

The Minimum system requirement must be met to run the game

Desktop:

* **Processor:** Dual Core2.2 Ghz**.**
* **Memory:** 2 GB RAM.
* **Graphics:** Dedicated Video Ram 384 MB
* **Storage:** 1500 MB available space.
* **Sound Card:** DirectX sound device
* **OS:** The game will run on Windows 7 or higher with DirectX 9+

Mobile Device:

* **Processor:** Dual Core processor
* **RAM:** 1 GB RAM
* **Storage:** 200 MB (Package Size)
* **OS:**
* Android Version 6.0+
* iOS 10+

## Design and Implementation Constraints

The Developers should keep the following in mind when designing the software:

**Database:** A database should be implemented to store user information

**Network Server:** For a smooth matchmaking for online game and connect the software to database to synchronize user information

**Language requirements:** software must be multilingual, including the following languages: English, French, German, Spanish, Mandarin, Japanese, Arabic and Hindi.

**Graphic:** 3D models and patches may be graphic extensive on some devices.

**Memory and Power Management:** This game is not memory or CPU intensive, however CPU load might be increased as the game gets complex.

**Software/Engine:** Unity and Unreal will be required for all developers.

**Maintenance Break:** The game will be down for all users during updates or maintenance.

## User Documentation

YouTube link of a chess and checkers game tutorial will be made available in-game. Apart from this in-ap hints will be dynamic and be accessible at all points, which will not only give help related to the game context, but will provide support of the software I.e., Display adjustments, sound/Music etc.

## Assumptions and Dependencies

* Some features might not be available on Mobile application which include screen resolution adjustment.
* All test phase of this app will be taken place in controlled environment on maxed out settings and configurations which will be quite different from end user since the user will have their own settings.
* At the time of installation, the package size is 2 GB, but the size could increase as the game is progressed (cache, app data, save game files etc.)

**Assumptions for mobile phone is taken as following:**

* A mobile device takes input through touchscreen instead of a keyboard and mouse used in PC. Cell phones display is vertical by default however user may play this game by switching display orientation.
* This software is dependent on Unreal engine and Unity and will work on packages provided by the respective software’s.

# External Interface Requirements

## 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.>

## 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.>

## 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.>

## 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 Management

**4.1.1 Description and Priority**

Account management is an essential feature of the system that allows users to create, login, logout, delete their accounts and request a new password in case they forget their account password. It is a high priority feature through which the system provides the management of multiple user accounts.

**4.1.2 Stimulus/Response Sequences**

**Signup Case:** A new user opens the application, a form is displayed which will have two buttons, one for log-in and one for sign-up. User presses/clicks on the sign-up button, the system will return the sign-up form having multiple fields such as Name, Username, Email address and password. The information provided by the user will be checked by the system and then stored in the database. If the details provided meet the system criteria then user will be returned the home page as a logged-in user. Else, an error message appears with the reason of sign-up failure.

**Login Case:** A user opens the application, a form is displayed which will have two buttons, one for log-in and one for sign-up. User presses/clicks on the log-in button, the system will return the log-in form with username and password fields to be filled by an existing user. The system verifies the data with the existing user account information stored in the database and grants application access to the user. Else, an error message appears with the reason log-in failure.

**Logout Case:** User presses or clicks the logout button in the settings menu. System prompts the confirmation message. User confirms the logout request, system approves the log out request of the user. User is logged out and system returns the login/signup page.

**Forgot Password Case:** User clicks/taps forgot password link. A form appears with only one field requiring the email address of the user account. That email is then cross validated by the system’s database and if a match is found then the system sends the new password on user’s email address. The user enters the password and user is returned to login page.

**Delete Account Case**: User presses or clicks the Delete account button in the settings menu. System prompts the confirmation message. User confirms the account deletion, system approves the account deletion request of the user. User account is permanently deleted, and system returns the login/signup page.

**4.1.3** **Functional Requirements**

REQ-1: The system shall display the login screen every time the application starts.

REQ-2: User shall be able to login by clicking login button.

REQ-3: The system shall allow the user to login using Gmail Id.

REQ-4: The system shall allow the user to login using Facebook Id.

REQ-5: The system shall allow the user to login using email/username and password.

REQ-6: System shall cross verify the information given by user with the existing information in the database and grant access to the user.

REQ-7: System shall show an error if data isn’t matched.

REQ-8: System shall display the sign-up button at the login screen.

REQ-9: The user shall be able to register a new account.

REQ-10: User can sign-up using Gmail account.

REQ-11: User can sign-up using Facebook account.

REQ-12: User shall be asked to enter details such as name, username, email, and password.

REQ-13: System shall verify the email address provided and grant access to the user. Else, system will show an error message.

REQ-14: Logged-in User shall have an option to logout.

REQ-15: Before logging out, System should ask the user for confirmation.

REQ-16: Upon Confirmation, User shall be logged out of the account.

REQ-17: User shall have an option to request for an alternate password incase he/she forgets the password.

REQ-18: System shall send a new password on user’s email and user will be able to login using that password.

REQ-19: User shall have an option to delete the account.

REQ-20: System shall ask for the user confirmation before deleting the account.

REQ-21: The account and any relevant information shall be deleted from the database upon user confirmation.

|  |  |
| --- | --- |
| **Use case name** | **Log In** |
| **Related Requirements** | REQ 1  REQ 2  REQ 3  REQ 4  REQ 5  REQ 6  REQ 7 |
| **Goal in Context** | The existing user can login and access the application. |
| **Actor(s)** | |
| Primary | User |
| Secondary | Database Admin |
| **Pre-Conditions** | |
| 1.  2. | The Application is running.  User has an account. |
| **Post-Conditions** | |
| Success | User is logged in and the system displays a home page. |
| Failure | User is unable to log in for one or more reasons. |
| **Trigger** | User clicks or presses the login button. |
| **Included Cases** | **Check Identity** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2  3.    4.    5. Include::Check Identity  6.  7. | Application starts.  The user clicks/taps on the login button.  The System prompts the user for a username and password or register new account.  The user enters the username and password.    The system validates the entered username and password from database.  The user has signed in.  The system navigates the user to the home page. |
| **Exceptions** | **Login Failed:**  5.1 – The system doesn’t verify the user’s credentials.  5.2 – The user’s attempt to login is denied. |

|  |  |
| --- | --- |
| **Use case name** | **Signup** |
| **Related Requirements** | REQ 8  REQ 9  REQ 10  REQ 11  REQ 12  REQ 13 |
| **Goal in Context** | To create a new account. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1.  2. | The email should be valid and unique.  The password must meet the given format. |
| **Post-Conditions** | |
| Success | The user entered valid information and was returned to the home page as a Logged-In User. |
| Failure | The user entered the invalid information and was unable to sign up. |
| **Trigger** | User clicks or presses the signup button. |
| **Included Cases** | **Verify Username** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.    4.    5.  Include:: verify username  6.  7. | Application starts.  The user click/taps on the signup button.  The System prompts user for registration information such as Username, email address, password, etc.  The user enters in their information.    Database will check the entered username against the usernames stored in the  database.  If the username is unique, Account is created  If the username already exists, System prompts to choose a new Username. |
| **Exceptions** | **Invalid Information provided:**  5.1 - The user provided incorrect information or left fields blank.  5.2 – The user’s request for signup is rejected. |

|  |  |
| --- | --- |
| **Use case name** | **Log out** |
| **Related Requirements** | REQ 14  REQ 15  REQ 16 |
| **Goal in Context** | The user is able to logout from the system. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The user must be logged-in. |
| **Post-Conditions** | |
| Success | The user has logged out. |
| Failure | The attempt to logout failed. |
| **Trigger** | User clicks or presses the Logout button. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4.  5. | User Click or taps the logout button.  They System prompts the user for logout confirmation.  The user confirms the logout request.  The user has logged out.  The system displays the login page. |
| **Exception** | **Confirmation Rejected:**  3.1 – The user cancelled the confirmation.  3.2 – The user’s attempt to logout failed. |

|  |  |
| --- | --- |
| **Use case name** | **Forgot password** |
| **Related Requirements** | REQ 17  REQ 18 |
| **Goal in Context** | User requests an alternate password for their account. |
| **Actor(s)** | |
| Primary | User |
| Secondary | Database Admin |
| **Pre-Conditions** | |
| 1. | The user account must exist. |
| **Post-Conditions** | |
| Success | The user is able to log-in using new password. |
| Failure | The attempt to provide a new password failed for one or more reasons. |
| **Trigger** | User clicks or presses forgot password link. |
| **Included Cases** |  |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4.  5. | User clicks/taps forgot password link.  The system prompts the user to enter Email address.  The user enters the email address.  System verifies the email and sends the new password on user’s email.  User is able to log in using the new credentials. |
| **Exception** | **Invalid Email Address­:**  4.1 – The system doesn’t verify the email address.  4.2 – The user’s request for an alternate password is rejected. |

## 

|  |  |
| --- | --- |
| **Use case name** | **Delete Account** |
| **Related Requirements** | REQ 19  REQ 20  REQ 21 |
| **Goal in Context** | Logged-in User can delete the account. |
| **Actor(s)** | |
| ­­Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The user must be logged-in. |
| **Post-Conditions** | |
| Success | The user account has been permanently deleted. |
| Failure | The user is unable to delete the account. |
| **Trigger** | User clicks or presses the Delete account button. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4.  5. | A logged-in user navigates to the setting.  User clicks or presses the delete account button.  The system asks the user for confirmation before deleting the account.  The user confirms the account deletion.  The account and all the data has been deleted. |
| **Exception** | **Confirmation Declined:**  4.1 – The user cancelled the confirmation.  4.2 - The attempt to delete the account failed. |

## 

## Play Game

**4.2.1 Description and Priority**

Play game is an integral feature of this application, as this allows the user to choose the game, they wish to play with all the customized settings. It is a high priority feature which enables the user to play the Chess or Checkers game.

**4.2.2 Stimulus/Response Sequences**

User will be on the Home Screen where there will be a button (Play Game).

Once user clicks this button, user will be redirected to a screen where user will have the option to choose between playing Chess or Checkers by clicking the corresponding button.

**4.2.3** **Functional Requirements**

REQ-22: User shall be able to start the game by clicking start game button.

REQ-23: User shall have the option to choose between playing chess or checkers.

REQ-24: User shall have the option to play with Computer (AI) or Multiplayer.

REQ-25: User shall have the choice to select difficulty level (Easy, Medium, Hard) if playing with Computer

REQ-26: User shall have the choice to select between a timed game or untimed game.

REQ-27: User shall have the option to choose sides.

REQ-28: White moves first.

REQ-29: User shall have the choice to go back to the home page from any point.

|  |  |
| --- | --- |
| **Use case name** | **Start Game** |
| **Related Requirements** | REQ 22  REQ 23 |
| **Goal in Context** | The user can start the game they wish to play. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1.  2.  3.  4. | The Application is running.  User has an account.  User is logged in.  Application is at the Home Page. |
| **Post-Conditions** | |
| Success | System displays select page screen. |
| Failure | System unable to display select page screen. |
| **Trigger** | User clicks or presses the start game button. |
| **Included Cases** |  |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2  3.    4. | Application is at the home page.  The user clicks/taps on the start game button displayed at the home page.  The System takes the user to the new page where they can select which game to play.  User will have choice to select either Play Chess or Play Checkers. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use case name** | **Play Chess** |
| **Related Requirements** | REQ 24  REQ 25  REQ 26  REQ 27  REQ 28  REQ 29 |
| **Goal in Context** | To start a new Chess game. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1.  2.  3.  4. | The Application is running.  User has an account.  User is logged in.  User has clicked the Play Chess button. |
| **Post-Conditions** | |
| Success | Chess game begins with the user’s chosen settings. |
| Failure | In multiplayer(online) mode when there is no available player to match. |
| **Trigger** | User clicks the play chess button. |
| **Included Cases** |  |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4.    5.    6. | User clicks the Play Chess Button.  User is asked to choose between a Timed or Untimed game.  User is asked to choose between play with Computer (AI) or Multiplayer.  If user chooses to play with computer, User will have the choice to choose between difficulty level (Easy, Medium, Hard).  If user chooses to play with computer, User will have the choice to pick the side,  In multiplayer sides will be assigned randomly.  User will be redirected to the game board after choosing all these settings. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use case name** | **Play Checkers** |
| **Related Requirements** | REQ 24  REQ 25  REQ 26  REQ 27  REQ 28  REQ 29 |
| **Goal in Context** | To start a new Checkers game. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1.  2.  3.  4. | The Application is running.  User has an account.  User is logged in.  User has clicked the Play Checkers button. |
| **Post-Conditions** | |
| Success | Checkers game begins with the user’s chosen settings. |
| Failure | In multiplayer(online) mode when there is no available player to match. |
| **Trigger** | User clicks the Play Checkers button. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4.    5. | User clicks the Play Checkers Button.  User is asked to choose between a Timed or Untimed game.  User is asked to choose between play with Computer (AI) or Multiplayer.  If user chooses to play with computer, User will have the choice to choose between difficulty level (Easy, Medium, Hard).  User will be redirected to the game board after choosing all these settings. |
| **Exception** |  |

## 

## In – Game Features

### Description and Priority

In Game Features allows the user to undo previous move or to take a hint for current move. It is a low priority feature as the game can be functioned without these features as well.

### Stimulus/Response Sequences

User will have the icons to undo their move or to take a hint for any move while the user is playing the game.

**4.3.3 Functional Requirements**

REQ-30: User shall be able to undo their previous move.

REQ-31: User shall be able to take the hint for their current move.

|  |  |
| --- | --- |
| **Use case name** | **In-game Features** |
| **Related Requirements** | REQ 30  REQ 31 |
| **Goal in Context** | The user can undo or take a hint for their moves. |
| **Actor(s)** | |
| Primary | User |
| Secondary | Database |
| **Pre-Conditions** | |
| 1.  2.  3.  4. | The Application is running.  User has an account.  User is logged in.  User must be in the game. |
| **Post-Conditions** | |
| Success | User is able to undo or take a hint for their moves successfully. |
| Failure | User is unable to undo or take a hint for their move due to system/database error. |
| **Trigger** | User clicks the undo or take a hint icon. |
| **Included Cases** | None |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2  3.  4.  5.  6. | User is playing the game.  If user clicks on the Undo icon.  System will prompt to confirm Undo.  If user confirms, their previous move will be deleted.  If user clicks on the Take a Hint icon.  System will display the hint for the current move. |

## 

## Pause Menu

**4.4.1 Description and Priority**

The pause menu feature is of high priority, it allows the player to pause the game anytime during the game. The player will also be able to resume game, save game and quit game from the pause menu and because of the portable nature of the device, player comfort is essential, hence this feature must be provided.

**4.4.2 Stimulus/Response Sequences**

**Step 1;** An In-game user presses ESC or clicks the pause button.

**Step 2:** Game pauses and a menu is displayed having 3 options to choose from: Resume game, Quit game and Save game.

1. Press “Save Game” to save the game.
2. Press “Quit Game” to forfeit the game.
3. Press “Resume Game” to go back to the game.

**Step 3:** The player presses on of the button, triggering its respective function.

**4.4.3** **Functional Requirements**

REQ 32: “Pause Game” button shall allow the user to pause the game anytime during the game.

REQ 33: The pause menu should appear and the game must be paused.

REQ 34: “Resume Game” option shall allow the user to resume game exactly from the moment of the pause action without any changes.

REQ 35: “Quit Game” option must return the user to the Main Screen of the application.

REQ 36: The “Save Game” option shall allow the user to save the game progress at any point.

REQ 37: Saving the game must save the current progress overwriting the previous saved progress.

|  |  |
| --- | --- |
| **Use case name** | **Pause Menu** |
| **Related Requirements** | REQ 32  REQ 33  REQ 34  REQ 35  REQ 36  REQ 37 |
| **Goal in Context** | User can pause the game anytime. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The User must be in a match. |
| **Post-Conditions** | |
| Success | Game Paused. |
| Failure | Game didn’t pause. |
| **Trigger** | User presses ESC or the pause button. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4. | User is in the game.  User presses ESC or taps the pause button.  The game pauses.  The system shows the pause menu with three options: Resume Game, Save Game and Quit Game. |

## Setting Menu

**4.5.1 Description and Priority**

The setting menu is of low priority as it is not essential to accessing gameplay but constitute set of standard features hence this feature must be included. The setting menu is accessible from the Main Screen and allows the player to configure profile and display setting to suit his/her convenience.

**4.5.2 Stimulus/Response Sequences**

**Step 1:** the user presses or clicks the setting button from the Home Screen.

**Step 2:** User is navigated to the setting menu, from here the player gets to select from 3 options:

1. Select “Account settings” for configuring the user profile.
2. Select “Configure Display” for configuring the graphical interface.
3. Select “Return to Home Screen” to go back to the Home Screen.

**Step 3:** The player chooses one of the options, triggering its respective function.

**4.5.3** **Functional Requirements**

REQ 38: The system shall have a feature for the user to update their profile, i.e., Username, Password, and Avatar.

REQ 39: The user shall be able to update their avatar by uploading a picture.

REQ 40: The user shall be able to change the avatar by choosing from available avatars.

REQ 41: The user shall be able to change the board color by choosing from available color.

REQ 42: The user shall be able to change the piece’s style.

REQ 43: The system shall have an option for the user to switch between light and dark theme.

|  |  |
| --- | --- |
| **Use case name** | **Configure profile** |
| **Related Requirements** | REQ 38  REQ 39  REQ 40 |
| **Goal in Context** | User can update the profile. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The user is in the setting menu. |
| **Post-Conditions** | |
| Success | Changes made by the user are saved in the database. |
| Failure | The database is not updated. |
| **Trigger** | User presses the account setting option. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4. | User clicks/presses the account setting option in the setting menu.  The system displays the fields with user information.  The user updates any field with required information and presses save button.  Updates done are saved in the database. |
| **Exception** | **Update failed:**  3.1 – The user didn’t press the save button, hence changes made are not saved. |

­

|  |  |
| --- | --- |
| **Use case name** | **Configure Settings** |
| **Related Requirements** | REQ 41  REQ 42  REQ 43 |
| **Goal in Context** | User can change the graphical interface. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The user must be in the setting menu. |
| **Post-Conditions** | |
| Success | Changes made by the user are saved in the database. |
| Failure | The user is unable to change things for some reason. |
| **Trigger** | User press the configure display option. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.        3.  4. | User clicks/presses the configure display option in the setting menu.  The system displays all the following options.   1. Change Board colour 2. Change Piece style 3. Change Theme   The user changes anything and presses save button.  Changes made are saved. |
| **Exception** | **Update failed:**  3.1 – The user didn’t press the save button, hence changes made are not saved. |

­

## User Support

**4.6.1 Description and Priority**

User Support is an additional feature, included to enhance user experience with the application. It is a Medium priority feature to improve the user interaction.

**4.6.2 Stimulus/Response Sequences**

User will be initially on the Home Page Screen from where user will have the option to go to User Support Menu by clicking its button.

Once on the User Support Menu, User can click on the Report a Bug button or Learn How to Play Button. Pressing each button, will redirect the user to its respective pages.

**4.6.3** **Functional Requirements**

REQ-44: User shall be able to learn the rules of Chess.

REQ-45: User shall be able to learn the moves of each piece of Chess.

REQ-46: User shall be able to learn strategies of Chess e.g. openings, endgames etc.

REQ-47: User shall be able to learn the rules of Checkers.

REQ-48: User shall be able to report any bug in the application.

REQ-49: User shall be able to give any suggestion related to the application improvement or features enhancement.

|  |  |
| --- | --- |
| **Use case name** | **Learn How to Play** |
| **Related Requirements** | REQ 44  REQ 45  REQ 46  REQ 47 |
| **Goal in Context** | The user can learn the game of Chess or Checkers. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1.  2.  3.  4. | The Application is running.  User has an account.  User is logged in.  Application is at the User Support Page. |
| **Post-Conditions** | |
| Success | System displays the how to play features. |
| Failure | Application is unable to load how to play features. |
| **Trigger** | User clicks or presses the How to Play button. |
| **Included Cases** | None |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.    3.  4. | Application is at the User Support page.  The user clicks/taps on the How to Play button displayed at the User Support Menu.  The System takes the user to the How to Play features page.  This page displays all the features like (Learn Chess, Learn Checkers, Learn Chess piece moves, Learn Chess strategies). |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use case name** | **Report a Bug** |
| **Related Requirements** | REQ 48  REQ 49 |
| **Goal in Context** | User is able to report a bug or make a suggestion. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1.  2.  3.  4. | The Application is running.  User has an account.  User is logged in.  Application is at the User Support Page. |
| **Post-Conditions** | |
| Success | Bug is reported or suggestion is successfully passed on to the admin. |
| Failure | Bug is unable to get reported or suggestion cannot be successfully passed on to the admin. |
| **Trigger** | User clicks the Report a Bug button. |
| **Included Cases** | None |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.    3. | User clicks the Report a Bug Button.  User will have the choice to either type in a suggestion in Make a Suggestion text box or Report a Bug in its corresponding text box.  User will click the Submit button once the text box is filled. |
| **Exceptions** |  |

## View Statistics

**4.7.1 Description and Priority**

View Statistics will showcase the overall records of the user, their track record of all the games played, their success rate. User can also watch replays of their previous games. It is a Medium priority feature as user can keep a track of their performance.

**4.7.2 Stimulus/Response Sequences**

User will be initially on the Home Page Screen from where user will have the option to go to View Statistics page by clicking its button.

**4.7.3** **Functional Requirements**

REQ-50: User shall be able to view the Number of Games Played.

REQ-51: User shall be able to view the Number of Games Won.

REQ-52: User shall be able to view their Win Percentage.

REQ-53: User shall be able to view their Rating.

REQ-54: User shall be able to view his standing compared to other players.

REQ-55: User shall be able to view the replays of previous games played.

|  |  |
| --- | --- |
| **Use case name** | **View Stats** |
| **Related Requirements** | REQ 50  REQ 51  REQ 52  REQ 53  REQ 54  REQ 55 |
| **Goal in Context** | The user can view their track record of all games played. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1.  2.  3.  4. | The Application is running.  User has an account.  User is logged in.  Application is at the View Statistics Page. |
| **Post-Conditions** | |
| Success | System displays the user statistics. |
| Failure | Application is unable to display the replays. |
| **Trigger** | User clicks or presses the View Statistics button. |
| **Included Cases** | None |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4. | Application is at the Home page.  The user clicks/taps on the View Statistics button displayed at the Home Page.  The System takes the user to the View Statistics page.  This page displays all the Statistics and Replays. |
| **Exceptions** |  |

## View Events

**4.8.1 Description and Priority**

View Events is an additional feature will display the live/past events and upcoming tournaments. It is a Low priority feature as user can have an idea of previous and upcoming chess tournaments.

**4.8.2 Stimulus/Response Sequences**

User will be initially on the Home Page Screen from where user will have the option to go to View Events page by clicking its button.

**4.8.3** **Functional Requirements**

REQ-56: User shall be able to view the Live Events.

REQ-57: User shall be able to view the Past Events.

REQ-58: User shall be able to view the upcoming Chess Tournaments.

|  |  |
| --- | --- |
| **Use case name** | **Events** |
| **Related Requirements** | REQ 56  REQ 57  REQ 58 |
| **Goal in Context** | The user can view previous, live and upcoming events/tournaments. |
| **Actor(s)** | |
| Primary | User |
| Secondary | None |
| **Pre-Conditions** | |
| 1.  2.  3.  4. | The Application is running.  User has an account.  User is logged in.  Application is at the View Events Page. |
| **Post-Conditions** | |
| Success | System displays the previous, live and upcoming events/tournaments. |
| Failure | Application is unable to display the events/tournaments. |
| **Trigger** | User clicks or presses the View Events button. |
| **Included Cases** | None |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4. | Application is at the Home page.  The user clicks/taps on the View Events button displayed at the Home Page.  The System takes the user to the View Events page.  This page displays all the previous, live and upcoming events/tournaments. |
| **Exceptions** |  |

## Administration

**4.9.1 Description and Priority**

The administrator feature is of high priority, because it will serve as bridge between the users and the developers for maintaining the application once it is launched by addressing user concerns to the developers and responding to user feedback. It will also have the privilege to add accounts to the blacklist and also remove them.

**4.9.2 Stimulus/Response Sequences**

**Step 1;** Admin Logins to the dashboard.

**Step 2:** A screen appears with the three options in the front:

1. View user accounts
2. View user suggestions
3. View reported bugs

**Step 3:** The Admin selects one of the options, triggering its respective function.

**4.9.3 Functional Requirements**

REQ 59: The system shall allow the admin to block any user account.

REQ 60: The system shall allow the admin to unblock the user account.

REQ 61: The system shall allow the admin to view user feedback.

REQ 62: The system shall allow the admin to view the reported bugs.

|  |  |
| --- | --- |
| **Use case name** | **Block User** |
| **Related Requirements** | REQ 59 |
| **Goal in Context** | To block the user. |
| **Actor(s)** | |
| Primary | Admin |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The Admin must be logged-in. |
| **Post-Conditions** | |
| Success | User account is blocked. |
| Failure | User account is not blocked. |
| **Trigger** | Admin selects the user account and press/click Add to blacklist. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4. | The Admin has logged-in to the dashboard.  Admin goes to the user account list.  Admin selects the user account that needs to be blacklisted.  Admin presses Add to blacklist button. |

|  |  |
| --- | --- |
| **Use case name** | **Unblock User** |
| **Related Requirements** | REQ 60 |
| **Goal in Context** | To unblock the banned user account. |
| **Actor(s)** | |
| Primary | Admin |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The Admin must be logged in. |
| **Post-Conditions** | |
| Success | User account is removed from the blacklist. |
| Failure | The attempt to unblock the user account failed. |
| **Trigger** | Admin selects the user account and press/click Unblock. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3.  4. | The Admin has logged-in to the dashboard.  Admin views the list of blocked account.  Admin selects the user account that needs to be unblocked.  Admin presses unblock. |

|  |  |
| --- | --- |
| **Use case name** | **View Suggestions** |
| **Related Requirements** | REQ 61 |
| **Goal in Context** | To view the app suggestions made by the user. |
| **Actor(s)** | |
| Primary | Admin |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The Admin must be logged-in. |
| **Post-Conditions** | |
| Success | Admin viewed the user suggestions. |
| Failure | Admin is unable to view user suggestion due to some reason. |
| **Trigger** | Admin presses or clicks View suggestions button. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3. | The Admin has logged-in to the dashboard.  Admin clicks/taps the view suggestion button.  A list of all user suggestion appears with their respective username. |

|  |  |
| --- | --- |
| **Use case name** | **View Reported bugs** |
| **Related Requirements** | REQ 62 |
| **Goal in Context** | To view the application bugs reported by the user. |
| **Actor(s)** | |
| Primary | Admin |
| Secondary | None |
| **Pre-Conditions** | |
| 1. | The Admin must be logged in. |
| **Post-Conditions** | |
| Success | Admin viewed the list of reported bug by the user. |
| Failure | Admin is unable to view the reported bugs. |
| **Trigger** | Admin presses/clicks the view reported bugs button. |
| **Included Cases** | **None** |
| **Main Flow** | |
| **Step** | **Action** |
| 1.  2.  3. | The Admin has logged-in to the dashboard.  Admin clicks/taps the view Reported bugs button.  A list of all reported application bugs appear with the username who reported the bug. |

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

1. In order to play in an online mode with multiple users, users shall be connected to the Internet and have a stable Internet connection to play the game smoothly.

1. In multiplayer mode, the user will play against another user.

1. In single player/online multiplayer mode, the user will play against the bot.
2. If a user’s connection is dropped while playing the game, wait 30 seconds before setting their status as abandoned and set the opponent’s status to ‘Winner’.
3. If finding a match takes longer than 15 seconds, in the online multiplayer mode, cancel the connection request and either ask the user to try again or exit the game.
4. In case of a faulty hardware, there can be performance issues which might slow down the chess application.
5. The hardware in the case of desktop to make moves should be a mouse and a touchpad if the game is played in mobiles or tablets (touch screen hardware).
6. User shall not face any noticeable input lag. Since some users have a bad internet connection, a message should appear that the lag that they’re facing is due to an unstable connection from their side.
7. In order to minimize the load period to less than 3-4 minutes, database should use optimized queries.
8. The application should have smooth transitions from one state to the other when making moves.
9. The application should be supported on Android 6.0 and above.
10. To ensure smooth transitions between users, the app should have an order in which moves are made.
11. The application shall not crash and when a user is running an older version, a message shall be displayed on their screens asking them to update their version.
12. If a user is unable to find an opponent, the application shall provide an option to play with the computer (AI), in which case the user is able to play smoothly as if playing against another user.
13. Devices running older versions of the app shall not be allowed to interact with the app unless they update to the latest version.
14. To maintain performance of the server, the server should limit the interaction of any such IP Address that sends spam requests for 10 minutes.

## 5.2 Safety Requirements

1. The application will be memory efficient (It will not utilize large memory space).
2. Application will not cause memory loss of the hardware used by the player and should be able to run on all types of machines/hardware.
3. The application will not affect/harm user’s device and the applications in the device once the chess application is installed in the hardware.
4. The application should avoid the certain limit of overheating the hardware while the application is running.
5. To avoid headache from constant playing, application will pop up a message to take a break from playing in order to reduce strain on their eyes.

## 5.3 Security Requirements

The application requires to login with the existing username and password or to create a new account by using the signup option.

As it is a game and it does not involve any personal information of a player, therefore, security is not a risk in this case.

However, if a user wants to secure his/her account, there will be an option for a 2-step verification which user can have a choice to apply.

In 2-step verification, user will have an option to set up the 2-step verification via an email or phone number. Whenever user will try to login, a code will be sent via an email or phone number and to login into the application, the user needs to type in the code to login into the application.

Through this step, an added functionality for security is provided as someone else will not have that user’s email or phone number’s access.

## 5.4 Software Quality Attributes

**Flexibility:**

* Application responds to changes made from time to time to increase the product value. (Flexible to modify)
* Changes can be performed in a given time span.
* Changes can be performed in a way that minimum costs are required.

**Usability:**

* Easy for beginners to use and learn the application.
* User-friendly as consistent user interface provided like frequently used chess apps.
* Easy to prepare for developers and to operate for users with self-explanatory layouts.

**Testability:**

* Application is easy to test and find defects.
* App is tested for each user story.
* Easy to divide into modules for testing.

**Efficiency:**

* App is memory efficient and should not use huge memory/disk space.
* Utilizes processor’s capacity efficiently, in order to provide real-time experience.

## 5.5 Business Rules

1. Username and Email for each user should be unique for identification.
2. User can verify through email and set a new password in case forgets current password by getting a 4-digit code.
3. User has 2 options for playing: Playing against Computer (AI) or Multiplayer (Against a User).
4. A user can play one game at a time from the above two options.
5. In case of playing with a bot, user will have an option of 3 difficulty levels: Easy, Medium, and Hard.
6. Once logged in, a user can:
7. Learn How To Play or Report A Bug
8. Update Profile or Customize Settings
9. View his/her stats and watch replays.
10. User Statistics will be updated regularly in order to update users’ records.
11. To update profile, user needs to enter correct username and password.
12. In order to change the existing password, user needs to enter the old password. In case of 2-step verification rule, an added code also needs to get added in the field in app.

# 6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

1. If a user abandons 3 matches, they should be given a temporary cool down of 30 minutes.
2. The cool down time period shall be incremented by 30 minutes until a user receives 3 cooldowns, after which the user shall be banned.
3. The application should meet with the current web standards.
4. The avatars should be vibrant.
5. The avatars shall have characters from shows/books/comics/anime which are currently trending.
6. The cool down counter shall be reset every 3 weeks.
7. The app shall have a rustic feel to it.
8. The app should consist of a client, server, and database.
9. Color blind users shall be able to use the app.
10. The app should show the available moves to the user.
11. The app should meet the standards in terms of accessibility.

Appendix A: Glossary

**SRS:** System Requirements Specifications - A comprehensive document that lists down all the software requirements for the developers and stakeholders of that software.

**Admin:** A person that has privileged access to the software

**User:** The person who will interact with the app and play the game.

**UI:** User Interface - The design of the app that the user will interact with.

**AI:** Artificial Intelligence – Intelligent piece of software programmed to act as another chess player with no human input while playing as a single player on the app.

**Avatar:** User profile picture.

**Bug:** An error in a computer program that produces unexpected results.

**Game Feedback:**

**Constraints:** Limitations in the software.

**Toggle:** To activate or disable features.

**Database:** Structured tables sued to store data.

**Network Server:** A powerful computer that provides various shared resources to workstations and other servers on a network.

**Test phase:** The time period during which the software is tested to find out whether their code and programming work according to customer requirements.

**Unreal Engine:** Game Engine used to develop games.

**Unity:** Game Engine used to develop games.

**Packages:** Multiple applications or code modules that work together to meet various goals and objectives.

**Prompt:** Result of a user interaction.

**FTP:** File Transfer Protocol – Establish connection with database via internet.

**Cool down:** A period where the user is not allowed to interact with the app.

**Bishop:** one of two pieces of the same color that may be moved any number squares diagonally, as long as no other piece blocks its way. One piece always remains on White squares and the other always on Black.

**Castling:** to move the king two squares horizontally and bring the appropriate rook to the square the king has passed over.

**Check:** To make a move that puts the opponents King under direct attack.

**Checkmate:** a situation in which an opponent’s king is in check and it cannot avoid being captured. This then brings the game to a victorious result.

**King:** The main piece of the game, checkmating this piece is the object of the game. It can move 1 space in any direction.

**Knight:** This piece can move 1 space vertically and 2 spaces horizontally or 2 spaces vertically and 1

Space horizontally. This piece looks like a horse. This piece can also jump over other pieces.

**Pawn:** One of eight men of one color and of the lowest value usually moved one square at a time

Vertically and capturing diagonally.

**Queen:** This piece can move in any number of spaces in any direction as long as no other piece is in its way.

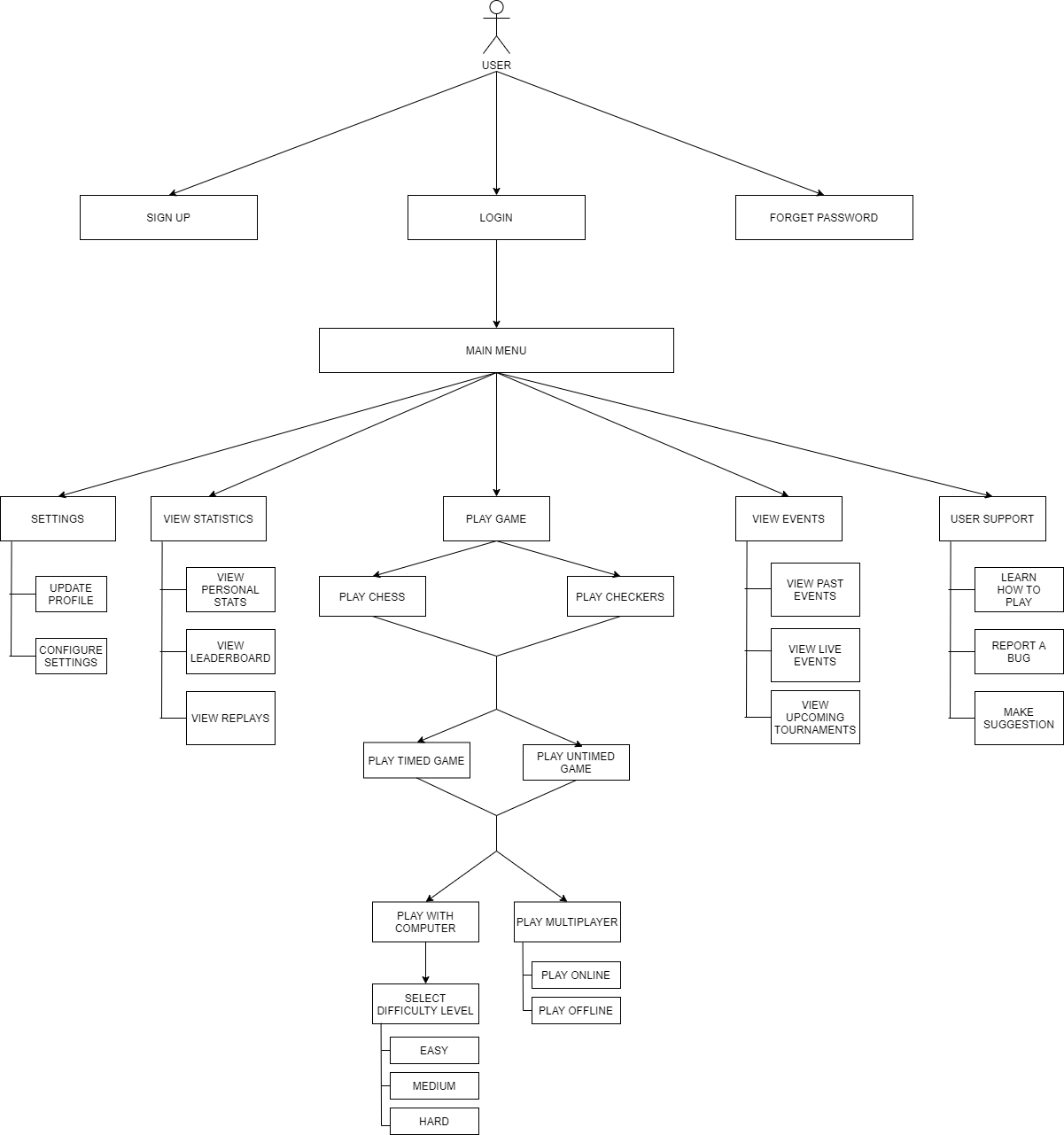
**Rook:** one of two pieces of the same color that may be moved any number squares horizontally or vertically, as long as no other piece blocks its way.

**System:** The environment of the application.

**VoIP:** Voice Over Internet Protocol – A set of rules that govern the exchange of audio over the internet.

**Power-Ups:** A special ability that grants users special powers.

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.>

1. The option to communicate via other players via chat.
2. The option to chat via audio.
3. The chat shall filter out any racial slurs and abuses that the user shall enter.
4. The user shall receive a warning each time they violate the application’s rules.
5. The option to start a tournament.
6. The ability to add/remove friends
7. A shopping section, where you can buy boards, power-ups and chess/checker pieces.