Software Design Specification

for

Advanced Tic Tac Toe Game

VERSION 1.0

Prepared by:

- Abdelrahman Mohamed
- Mohamed Hisham Mohamed
- Salah Eldin Hassen
- Shahd Gamal
- Shahd Hamad

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this SDS document is to provide a detailed design specification for the Advanced Tic Tac Toe Game. This document outlines the system architecture, detailed design of components, database design, and user interface design.

.

1.2 Scope

This SDS document covers the design of the Advanced Tic Tac Toe Game, including its architectural structure, detailed component design, database schema, and user interface. It serves as a guide for the development team to implement the system.

1.3 Definitions, Acronyms, and Abbreviations

- **AI:** Artificial Intelligence
- **GUI:** Graphical User Interface
- **SDS:** Software Design Specification
- **CI/CD:** Continuous Integration/Continuous Deployment
- **ER Diagram:** Entity-Relationship Diagram

1.4 Overview

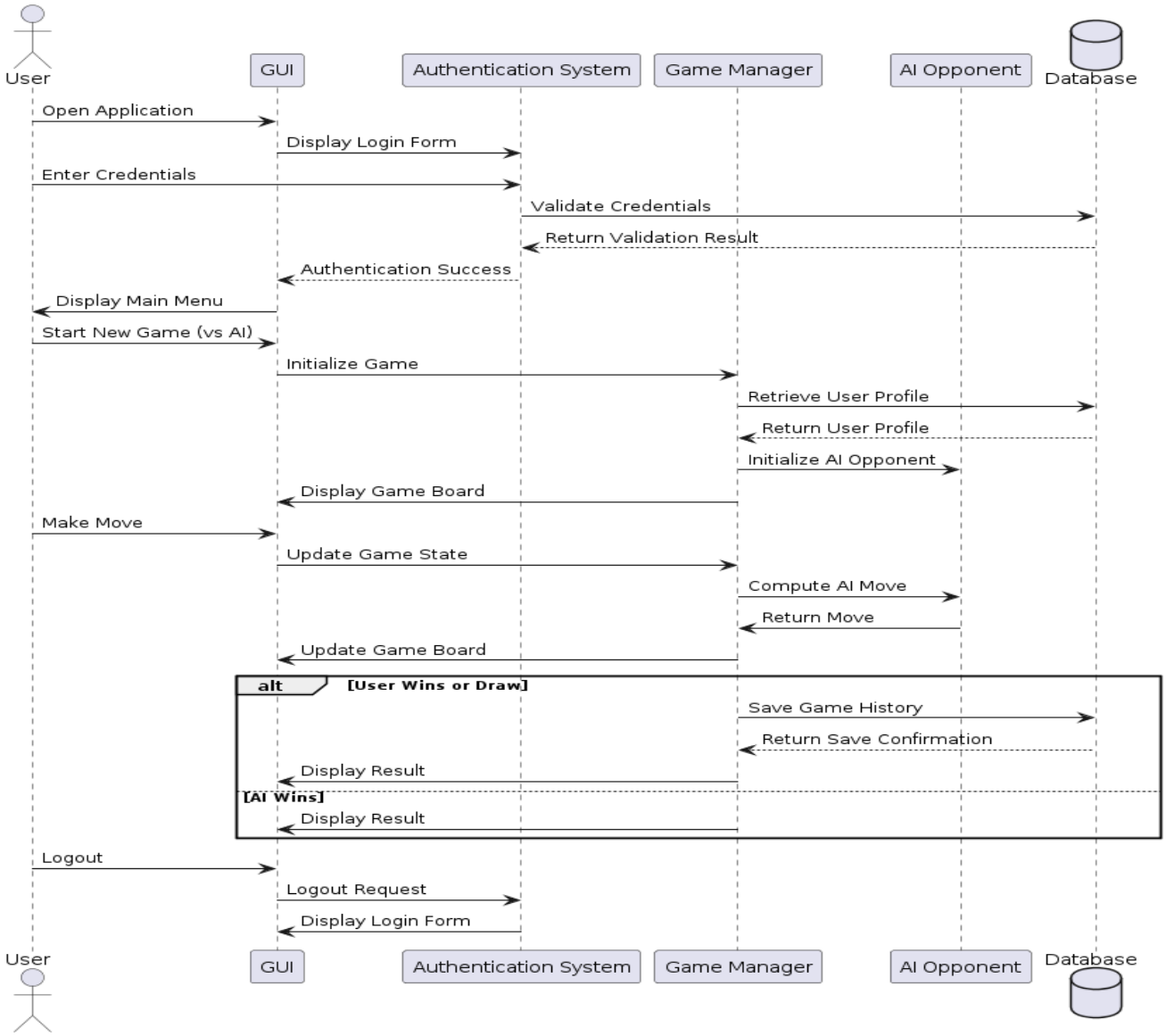
This document is organized into sections covering the system architecture, detailed design, database design, user interface design, and design considerations. Each section provides diagrams and descriptions to support the implementation of the Advanced Tic Tac Toe Game.

2. Detailed Design

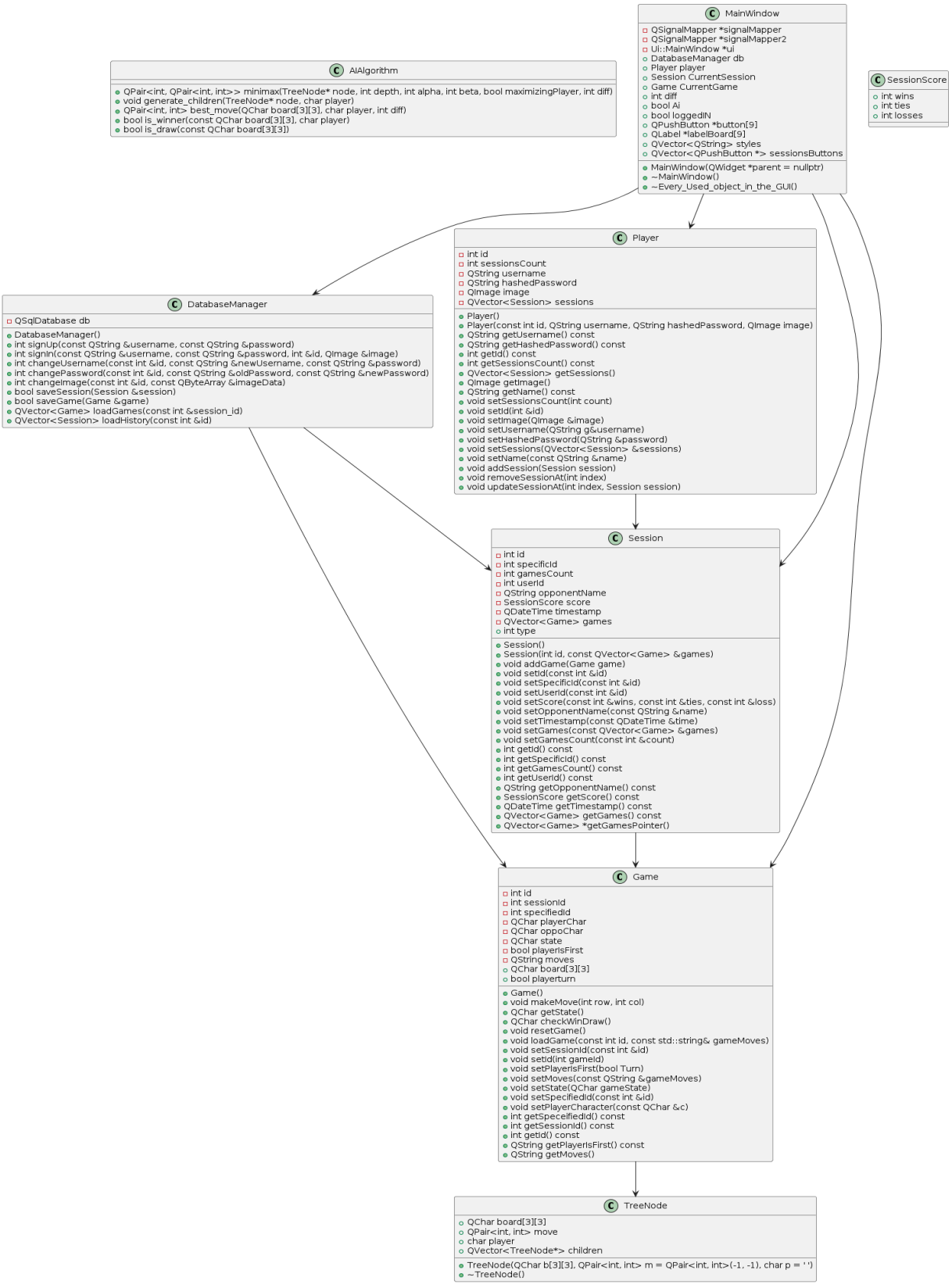
The system is designed using Object-Oriented Programming (OOP) principles. The main classes are:

1. **DataBaseManger:** Manges Data saved in database to separate interaction between main system and database functionalities.
2. **Game:** Manages the game state and logic for checking the score.
3. **Session:** Manages collected number of games that are played in same session
4. **Player:** Handles the logged in user information and authentication
5. **AI algorithm:** Implements the AI opponent using the minimax algorithm.
6. **Mainwindow:** Handles the functions and usage of each used object in the GUI.

2.1 Sequence Diagram

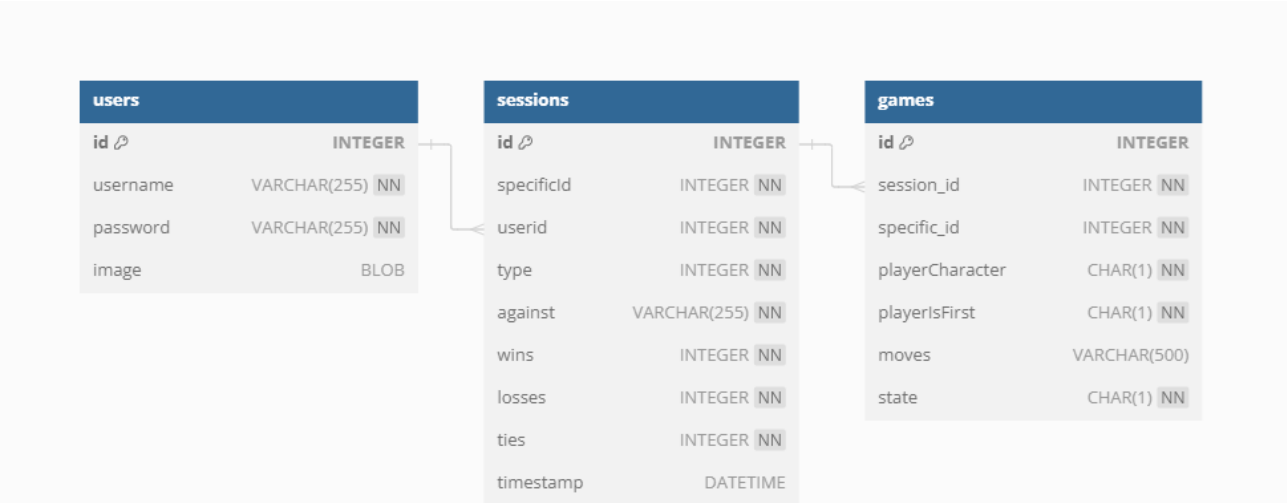


2.2 Classes Diagram



2.3 Database Design

This is a schema to visualize the Database design

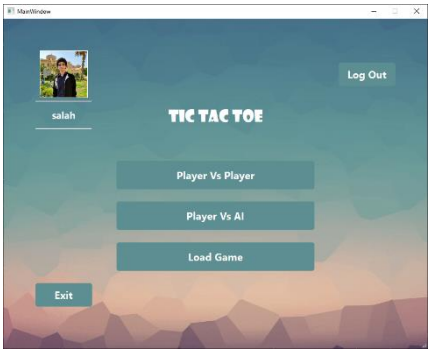


3. User Interface Design

3.1 Overview of the GUI

The GUI provides an interactive interface for users to play Tic Tac Toe, manage their profiles, and view game history. It is designed using Qt Framework for a consistent look, the design is based changeable widgets with only one window.

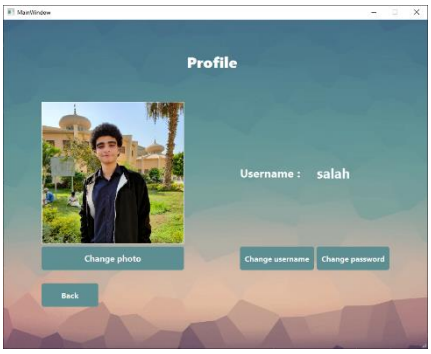
3.2 Widgets Design

3.2.1 Main Window: this is the first window that appears when you enter the game. (default at the beginning of the Program)	

3.2.2 Profile Window:

this window displays the information of the user, and it is only available if a user is logged in.

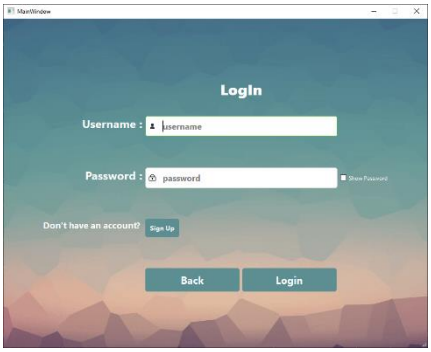
(Accessed from Main Window)



3.2.3 Sign In Window:

This is the window where the user could sign in to an existing account in the Database.

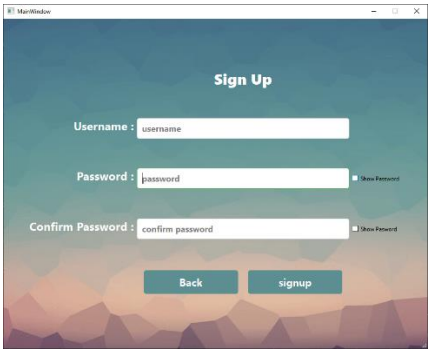
(Accessed from Main Window)



3.2.4 Sign Up Window

This window is accessed if the user want to create a new profile and save it to the Database

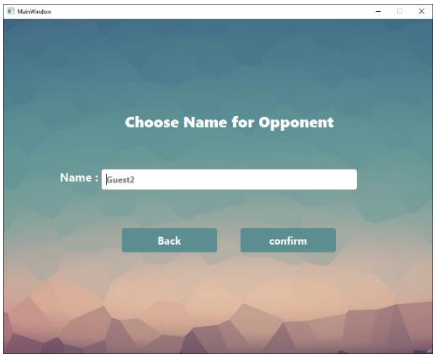
(Accessed from Sign in Window)



3.2.5 Player vs Player Window

This window is used to enter the opponent's name when the user is going to play against another player

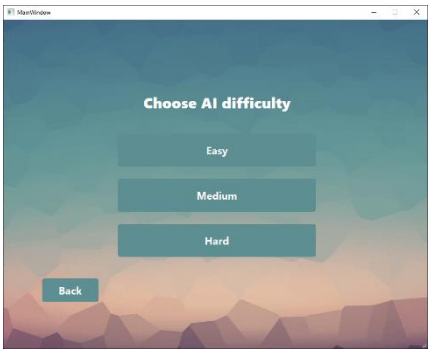
(Accessed from Main Window)



3.2.6 Player vs AI Window

This window is accessed when a user wants to play against an AI and want to choose the difficulty

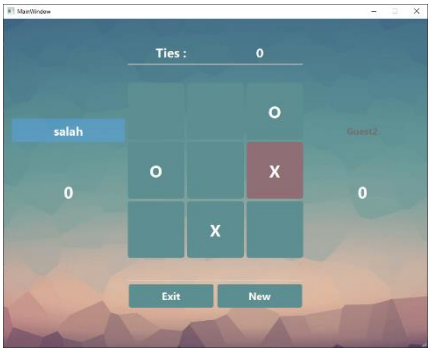
(Accessed from Main Window)



3.2.7 Board Window

This is the window where the game takes place either between user and AI or User and Player

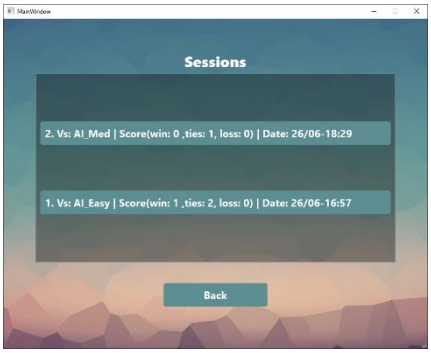
(Accessed from Player vs Ai Window
And Player vs Player Window)



3.2.8 Sessions Window

This is the window where you could access sessions played by a specific user and saved in Database and is only available if a user is logged in

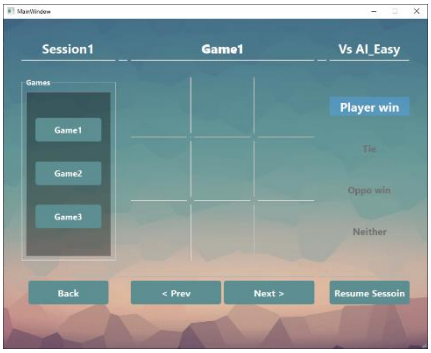
(Accessed from Main Window)



3.2.9 Loaded Games Window

This window is used to re load already played and saved session in a specific session.

(Accessed from Main Window)



4. Appendices

4.1 Glossary

- **Minimax Algorithm:** A recursive algorithm used for decision-making and game theory.
- **Alpha-Beta Pruning:** An optimization technique for the minimax algorithm that reduces the number of nodes evaluated.
- **Hashing:** converting the password into a fixed-length string of characters using a cryptographic hash function making it difficult to reverse-engineer the original password from its hash.

4.2 Acronyms and Abbreviations

- **AI:** Artificial Intelligence
- **CI/CD:** Continuous Integration/Continuous Deployment
- **GUI:** Graphical User Interface

4.3 References

- Google C++ Style Guide: Google C++ Style Guide - GitHub
- GitHub Actions Documentation: https://docs.github.com/en/actions
- SQLite Documentation: SQLite Documentation
- YAML Official Website: YAML Tutorial: Everything You Need to Get Started in Minutes - CloudBees