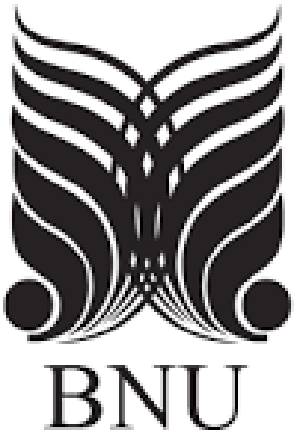
Object Oriented Programming



systems

Group Members:

Talha Ali (SUM24-0054)

Muhammad Shahroz Jajja (SUM24-0059)

Mujtaba Ahmed (SUM24-0081)

Salman Mazhar (SUM24-0053)

Project Hangman

1. Separation of Concerns

The code is structured with a clear separation of responsibilities:

• Game Class: Acts as an interface to ensure all games implement a consistent structure.

• Hangman Class: Implements the Game interface and encapsulates the logic for the Hangman puzzle, adhering to the single-responsibility principle.

• GameSystem Class: Manages user accounts, badges, and the overall game flow, decoupled from the specific game logic.

This modular design ensures that individual components can be developed, tested, and debugged independently.

2. File-Based Persistence

Rationale: Ensures user credentials persist between game sessions, allowing seamless reauthentication.

Implementation:

• The signUp.txt file stores usernames and passwords.

• loadUsersFromFile() and saveUserToFile() methods handle file-based I/O.

This approach ensures data integrity and prevents loss of user information between sessions.

3. Extensibility

The Game interface allows easy addition of new games.

Steps to add a new game:

• Create a new class that inherits from Game.

• Implement the play() method with the specific game logic.

• Add the new game instance to the games vector in the GameSystem constructor.

This flexibility supports future expansions of the application without major architectural changes.

4. User Feedback

The game ensures consistent user feedback to enhance the experience:

• Displays congratulatory messages on success.

• Reveals the correct word when the user fails the Hangman puzzle.

• Visual feedback in the form of a "hanging man" enhances user engagement and immersiveness.

5. Use of Standard Libraries

The code leverages C++ STL (map, vector, ifstream, ofstream) for efficient data handling:

• map is used for user management, providing fast lookup and insertion of user credentials.

• vector stores badges and words for the Hangman game, supporting dynamic resizing and efficient indexing.

Using standard libraries reduces development time and ensures robust and efficient data handling.

6. Object-Oriented Principles

The code demonstrates key object-oriented programming (OOP) principles:

• Encapsulation: Sensitive data, such as user credentials and badges, is encapsulated in the GameSystem class.

• Inheritance and Polymorphism: The Game interface and its implementation in Hangman demonstrate polymorphic behavior, allowing the system to handle different games uniformly.

7. Scalability

The design supports scalability in terms of:

• Adding new users without performance degradation, achieved through optimized file handling methods.

• Expanding the game library using the modular Game interface, ensuring that additional games can be incorporated seamlessly.

8. Maintainability

The modular design of the system makes it maintainable:

• Clear separation of logic ensures that changes in one component do not affect others.

• The use of interfaces and abstract classes simplifies future enhancements and debugging processes.

9. Error Handling

Basic error handling is incorporated to manage unexpected scenarios:

• Checks for empty word lists in Hangman prevent runtime errors.

• Duplicate username checks ensure data integrity during the signup process.

Future improvements could include enhanced exception handling and user input validation.

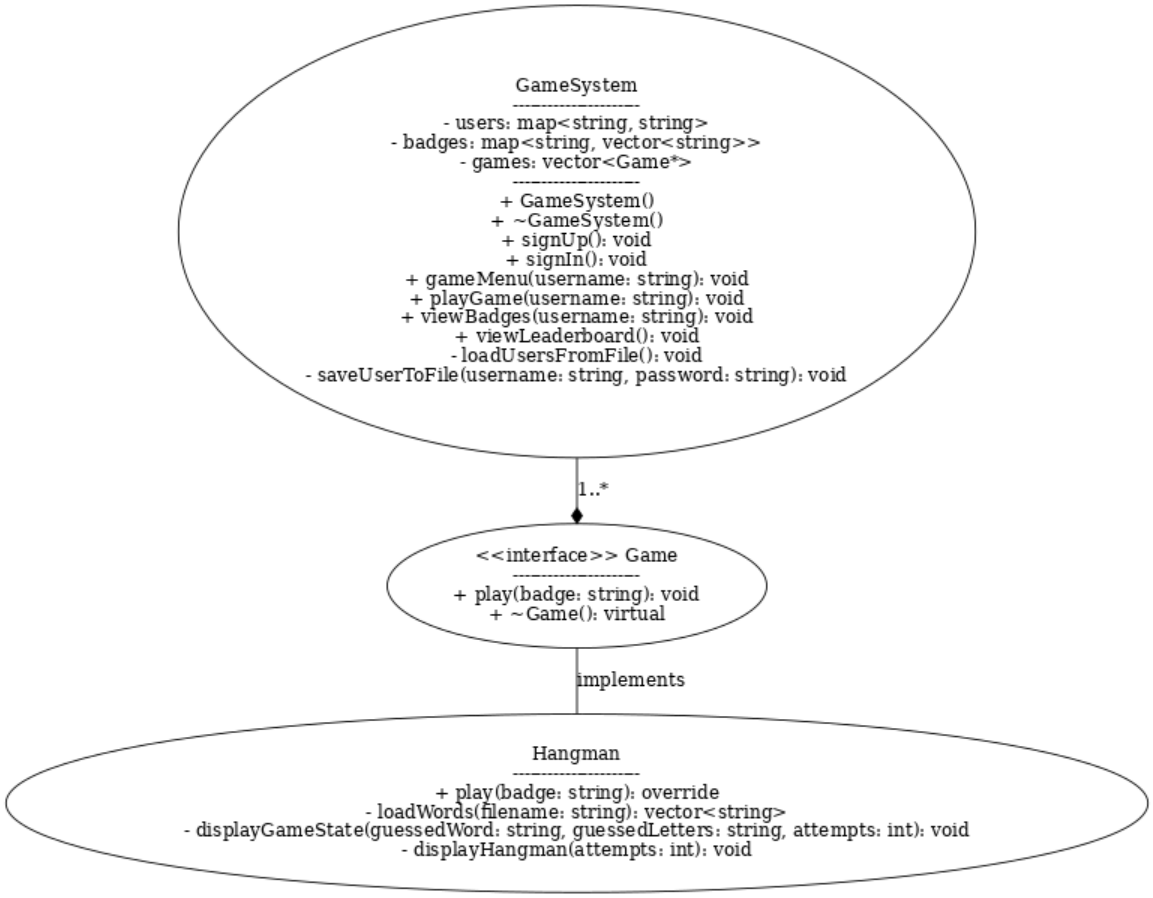
10. Visual Representation

The Hangman game incorporates a visual representation of the "hanging man":

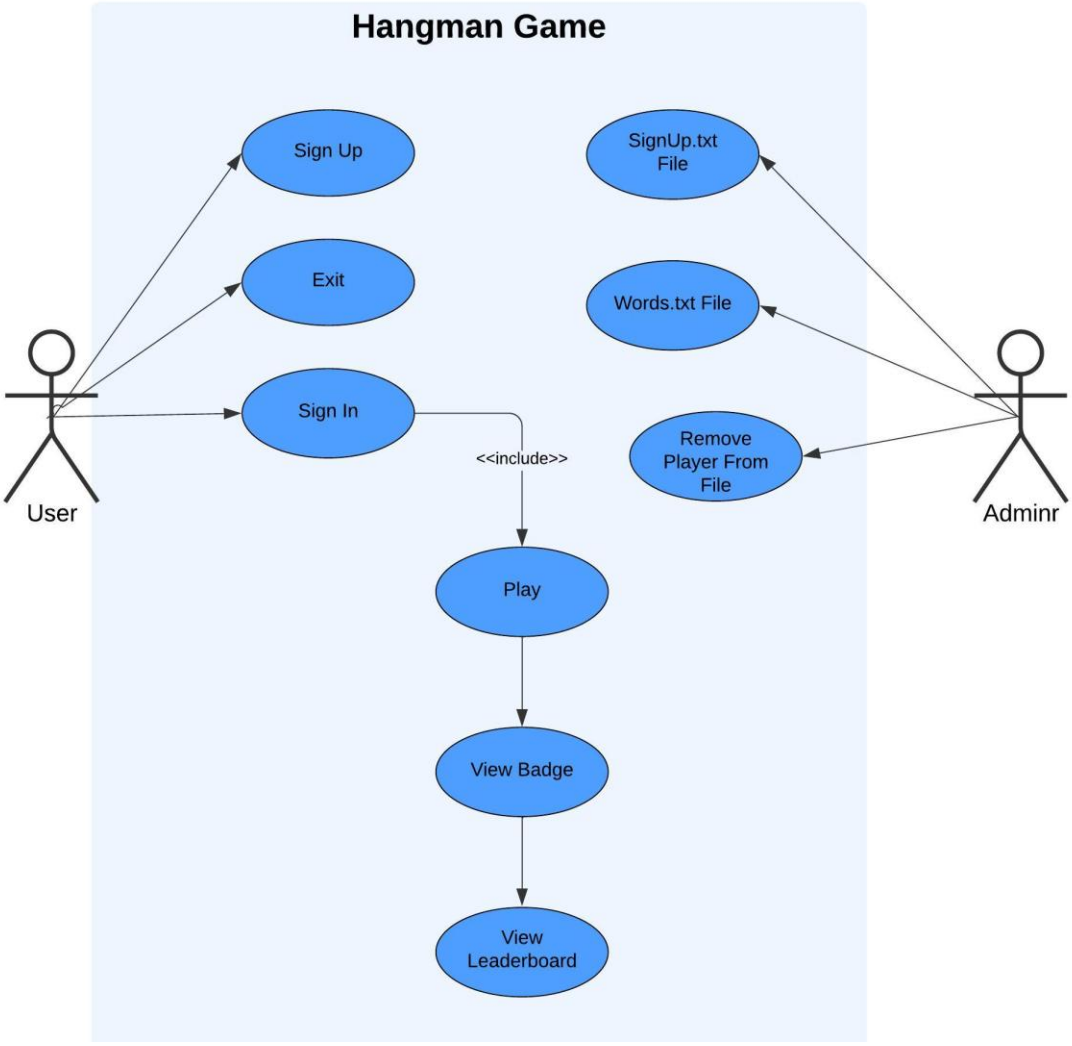
• Provides an engaging and intuitive gameplay experience.

• Clearly communicates the number of remaining attempts to the player.

UML Diagram:



Use Case Diagram:


Hangman Game
Sign Up
Exit
Sign In
<<include>>
Play
View Badge
View Leaderboard
SignUp.txt File
Words.txt File
Remove Player From File
User
Adminr

Helping Tools:

In our project, we extensively used AI tool ChatGPT to assist in building our game project, **Hangman: Guess The Word**. Additionally, we referred to a GitHub Hangman project to understand the class structures and some aspects of the logic to better design and implement our code.

Github link : https://github.com/jasmin-30/Hangman.git

Geeks of Geeks: Hangman Game in C++ - GeeksforGeeks

Stack overflow: string - C++ Simple hangman game - Stack Overflow