**Нефункціональні вимоги**

**Підтримка операційних систем**

* **Windows:** Версії 10, 11.
* **Linux:** Основні дистрибутиви (Ubuntu, Fedora, Debian).

**Технології та середовище виконання**

* **.NET Core або .NET 6+:** Використання новітніх версій .NET гарантує максимальну продуктивність завдяки поліпшеній багатопоточності, кросплатформеності та покращеній взаємодії з апаратним забезпеченням.
* **ASP.NET Core:** Якщо є потреба у серверній стороні для мультиплеєрних ігор, ASP.NET Core забезпечить високопродуктивний веб-сервер з асинхронною обробкою запитів та хорошою масштабованістю.
* **EF Core (Entity Framework Core):** Для роботи з базами даних, цей ORM дозволяє ефективно працювати з реляційними базами даних, підтримуючи високі навантаження та забезпечуючи швидкий доступ до даних.
* SignalR: Для забезпечення реального часу комунікації між клієнтами (гравцями) і сервером під час ігрової сесії. SignalR дозволить легко реалізувати двосторонній зв’язок, такий як чат, оновлення статусу гри в режимі реального часу.

**Безпека**

* **OAuth або JWT:** Для забезпечення безпеки під час автентифікації та авторизації використання протоколів OAuth2 або JSON Web Tokens (JWT) є обов’язковим.
* **TLS/SSL:** Шифрування всього трафіку між сервером та клієнтами через TLS/SSL для захисту даних користувачів та конфіденційної інформації.

**Аутентифікація**

* **Традиційна форма:** Форма з полями для введення імені користувача та пароля. Якщо користувач новий, йому пропонується натиснути на "Зареєструватися", де він заповнить форму реєстрації з базовою інформацією (email, пароль, підтвердження пароля тощо).
* **Технології:** C++, C#, Java... для відображення форми та обробки введених даних. Для backend можна використовувати REST API або GraphQL.

**Клієнтська частина:**

* **Windows Presentation Foundation (WPF).**
* **Асинхронна обробка:** Для зменшення затримок у користувацькому інтерфейсі слід використовувати асинхронні виклики API або баз даних.

**Серверна частина**

* **Kestrel:** Вбудований веб-сервер у .NET Core для забезпечення швидкої обробки запитів. Для більшої надійності можна комбінувати Kestrel з проксі-сервером, таким як Nginx чи Apache.
* **Thread Pooling:** Використовувати можливості .NET для асинхронної обробки потоків і налаштування оптимального використання потоків під навантаженням.
* **Caching:** Використовувати вбудовані механізми кешування (наприклад, MemoryCache) для зменшення кількості запитів до бази даних і швидшої обробки повторних запитів.

**База даних**

* **SQL Server або PostgreSQL.**
* **Індексація таблиць:** Індексації критично важливих полів у таблицях бази даних дозволяє підвищити швидкість вибірки даних.

**Апаратне забезпечення:**

* **Процесор:** Intel i3 або еквівалентний AMD є достатньо продуктивними для карткової гри з помірною кількістю дій та обробки даних.
* **Оперативна пам'ять:** 4 ГБ RAM мінімально необхідні для безперебійної роботи, з урахуванням одночасного виконання інших програм (браузери, месенджери тощо).
* **Графічна карта:** Оскільки гра не є ресурсомісткою в плані графіки, достатньо вбудованої або дискретної карти з підтримкою OpenGL 3.2 або новіших версій.
* **Місце на диску:** 500 МБ – оптимальний мінімум для встановлення ігрових файлів.
* **Дозвіл екрану:** Мінімум 1280x720 пікселів для коректного відображення ігрового інтерфейсу.