MonopolyUA

“DevTeam”, 3 members

Dukhota Ivan, Skryahin Oleg, Sharaiev Bohdan

**Software Requirements Specification**

**Document**

**Version: 1 Date: (09/06/2025)**

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. The Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communications Interfaces

2.1.6 Memory Constraints

2.1.7 Operations

2.1.8 Site Adaptation Requirements

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

2.6 Apportioning of Requirements

3. Specific Requirements

3.1 External interfaces

3.2 Functions

3.3 Performance Requirements

3.4 Logical Database Requirements

3.5 Design Constraints

3.5.1 Standards Compliance

3.6 Software System Attributes

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

3.6.4 Maintainability

3.6.5 Portability

3.7 Organizing the Specific Requirements

3.7.1 System Mode

3.7.2 User Class

3.7.3 Objects

3.7.4 Feature

3.7.5 Stimulus

3.7.6 Response

3.7.7 Functional Hierarchy

3.8 Additional Comments

4. Change Management Process

5. Document Approvals

6. Supporting Information

**1. Introduction**

**1.1 Purpose**

Цей документ є специфікацією вимог до програми (SRS) для веб-застосунку «MonopolyUA». Головна ціль документу це визначення, що повинна виконувати система, щоб розробники могли перейти до проєктування та реалізації.

Аудиторія:

* розробники backend частини;
* розробники frontend частини.

**1.2 Scope**

Продукт: веб-застосунок «MonopolyUA».

Що робить:

* реєстрація/авторизація користувачів (JWT, Google Auth);
* головна сторінка, профілю, маркету, створення гри, дошки для гри;
* рейтингові таблиці та топ-гравці;
* створення/приєднання до ігрового лобі;
* відображення динамічної дошки гри з ходами, оплатами та купівлею.
* чат у реальному часі;
* обробка онлайн-статусів та AFK.

Що не робить:

* не націлена на мобільні застосунки;
* не підтримує інтеграцію реальних грошей.

Бізнес-мета: створити інтерактивну онлайн-версію «Монополії» з соціальними компонентами та транзакціями між гравцями.

**1.3 Definitions, Acronyms and Abbreviations**

* HTML / CSS / JS - Технології фронтенду;
* Django - Python фреймворк;
* DRF - Django REST Framework;
* Channels - Django Channels для WebSocket;
* Daphne - ASGI‑сервер;
* Redis - in‑memory datastore;
* PostgreSQL - Реляційна СУБД;
* Asyncio - Асинхронна бібліотека Python;
* UUID - Унікальний ідентифікатор об’єкта;
* JWT - JSON Web Token;
* CORS - Cross-Origin Resource Sharing;
* Simple JWT - Django‑бібліотека для JWT;
* django-redis - Бекенд кешування через Redis.

**1.4 References**

* RFC 7519 — JSON Web Token.
* Django 5.1 documentation.
* Django REST Framework docs.
* Django Channels docs + ASGI/Redis integration.
* PostgreSQL 15 docs.
* Relevant articles on WebSocket-based real-time games.

**1.5 Overview**

* Розділ 2: загальний опис системи та її контекст;
* Розділ 3: функціональні та нефункціональні вимоги;
* Розділ 4: сценарії використання;
* Розділ 5-6: зовнішні інтерфейси, інші вимоги.

**2 Overall Description**

**2.1 Product Perspective**

Веб-застосунок «MonopolyUA» є незалежною програмною системою, що працює автономно у вигляді повнофункціонального веб-клієнта та серверної частини з підтримкою реального часу. Його основне призначення — забезпечити користувачам інтерактивне середовище для гри у цифрову версію настільної гри «Монополія», орієнтовану на український культурний контекст. Програма не є компонентом більшого корпоративного рішення, однак вона побудована з використанням стандартних веб-протоколів і може масштабуватись або інтегруватись у ширші системи (наприклад, з платформами рейтингів чи сторонніми сервісами для зберігання профілів).

Застосунок включає фронтенд-частину, реалізовану засобами HTML, CSS та JavaScript без використання фреймворків, та бекенд-частину, розроблену на базі Django 5.1 із поділом на синхронні REST API та асинхронну логіку на WebSocket через Django Channels. Основним джерелом збереження даних виступає PostgreSQL, а Redis відіграє роль як кеш-сервера, так і каналу повідомлень. Завдяки цьому архітектура дозволяє реалізувати повноцінну багатокористувацьку гру в реальному часі з інтегрованим чатом та постійною передачею станів гри.

**2.1.1 System Interfaces**

Система взаємодіє з кількома зовнішніми службами та протоколами, необхідними для її роботи. Однією з ключових інтеграцій є Google OAuth 2.0, що використовується для авторизації користувачів за допомогою облікових записів Google. Це забезпечує безпечний і зручний доступ без потреби вводити пароль. Додатково, для відновлення пароля через email використовується SMTP-сервер, налаштований для надсилання листів із посиланнями на зміну пароля.

Іще одним важливим елементом є Redis, який, окрім кешування, використовується для організації обміну повідомленнями між клієнтом і сервером за допомогою Pub/Sub моделі. Цей механізм критично важливий для роботи чату, ігрових ходів і системи статусів онлайн/офлайн. Сама система також може бути інтегрована з моніторинговими сервісами або CI/CD пайплайнами для автоматизації розгортання.

**2.1.2 Interfaces**

Інтерфейс користувача системи представлений у вигляді традиційного багатосторінкового веб-сайту, де всі елементи візуального відображення реалізовані засобами HTML, CSS та JavaScript. Головна мета інтерфейсу — забезпечити простоту навігації, інтуїтивне керування та візуальну ідентичність, притаманну грі «Монополія». Усі основні розділи — профіль, лобі, ігрове поле, магазин — доступні з головного меню, а реакції на дії користувача реалізовані без перевантаження сторінки, з допомогою динамічної взаємодії з API. Інтерфейс також враховує основи доступності: використання контрастних кольорів, альтернативного тексту для зображень.

**2.1.3 Hardware Interfaces**

Веб-застосунок «MonopolyUA» не має прямої взаємодії з апаратним забезпеченням. Усі процеси виконуються у віртуальному або хмарному середовищі, тому вимоги до апаратної частини мінімальні та стандартні для типових серверів. Система не контролює зовнішні пристрої, як-от сканери, сенсори чи інші периферійні модулі. Усі компоненти взаємодіють через програмні інтерфейси, і будь-яке апаратне забезпечення розглядається лише як середовище виконання.

**2.1.4 Software Interfaces**

Програмна система «MonopolyUA» побудована на низці зовнішніх бібліотек та платформ. Бекенд реалізовано з використанням фреймворку Django (версія 5.1), що забезпечує базову структуру, маршрутизацію та взаємодію з базою даних PostgreSQL. Для створення REST API використовується Django REST Framework. Авторизація реалізована з допомогою бібліотеки Simple JWT, яка надає функціональність створення, оновлення та перевірки токенів доступу.

Асинхронна частина системи базується на Django Channels, що працює поверх ASGI-сервера Daphne, а Redis служить як кеш і транспортний канал. Частина клієнтської частини системи взаємодіє з сервером через WebSocket-протокол, що забезпечує двосторонню комунікацію в реальному часі, необхідну для гри, чату та статусів гравців.

**2.1.5 Communications Interfaces**

Застосунок використовує кілька типів комунікаційних інтерфейсів. Основна взаємодія між клієнтом і сервером здійснюється через HTTP, що забезпечує захищене з'єднання з REST API. Для обміну повідомленнями в режимі реального часу система використовує WebSocket-з’єднання, яке працює через ASGI та Redis. Крім цього, система підтримує CORS, що дозволяє безпечний доступ до API з фронтенду, розгорнутого на окремому домені.

**2.1.6 Memory Constraints**

У системи немає жорстко встановлених обмежень на об’єм оперативної пам’яті, проте з огляду на використання Redis та розподіленого оброблення даних, мінімальна рекомендована конфігурація для сервера передбачає не менше 2 GB RAM. Це дозволяє ефективно підтримувати декілька одночасних ігрових сесій, чат та кешування без значного навантаження. Під час масштабування рекомендується дотримуватись балансу між доступною пам’яттю Redis і кількістю активних WebSocket-підключень.

**2.1.7 Operations**

Система працює у постійному інтерактивному режимі. Більшість дій користувачів відбуваються у реальному часі, з мінімальним затримками та без необхідності оновлення сторінки. Адміністративна панель передбачає базові функції моніторингу, а також можливість автоматичного перезапуску сесій гри у разі збоїв. Для збереження стабільної роботи системи рекомендовано збереження даних про завершені ігри.

**2.1.8 Site Adaptation Requirements**

Перед початком роботи з системою необхідно виконати кілька підготовчих кроків, зокрема розгортання інфраструктури з встановленням Redis, PostgreSQL, Django-серверу та необхідних залежностей. У випадку хостингу в хмарному середовищі (наприклад, AWS або DigitalOcean), також слід враховувати налаштування SSL-сертифікатів і проксісерверів. Система може бути адаптована до конкретних умов інсталяції шляхом налаштування конфігураційних файлів середовища, а також вибору відповідного плану масштабування залежно від кількості користувачів.

**2.2 Product Functions**

Основне функціональне призначення веб-застосунку «MonopolyUA» полягає у забезпеченні гравцям повноцінного онлайн-досвіду, наближеного до класичної настільної гри «Монополія», із розширеннями, адаптованими до цифрового формату. Програма повинна надавати користувачу повний цикл взаємодії — від реєстрації та входу до участі в динамічному ігровому процесі з іншими гравцями, управління власним профілем, обміну предметами та спілкування у реальному часі.

Першочерговою функцією системи є створення безпечного механізму автентифікації, що реалізується за допомогою власного інтерфейсу логіна з JWT-токенами та авторизації через Google. Сюди ж входить функція відновлення паролю, яка здійснюється через email. Після входу користувач має змогу керувати власним профілем: змінювати особисті дані, нікнейм, аватарку, переглядати ігрову статистику та керувати друзями.

Другою ключовою підсистемою є особистий кабінет гравця. Він включає інвентар з предметами (скінами), які можна застосовувати в грі для персоналізації, продавати та у деяких випадках відкривати, якщо тип предмету – кейс. Також існує система друзів, яка дозволяє додавати, підтверджувати або видаляти. Це посилює соціальний аспект проєкту.

Третій функціональний блок — це ігровий маркет, що дозволяє купівлю предметів інших користувачів. Усі транзакції супроводжуються перевіркою балансу, зняттям коштів та оновленням інвентаря. Додатково реалізована система кейсів із випадковими винагородами, яка стимулює активність у грі.

Найважливішим ядром застосунку є сам ігровий процес. Користувачі можуть створювати або приєднуватись до ігрових лобі, в яких вони налаштовують параметри майбутньої гри. Кожна сесія містить повноцінне ігрове поле з елементами «Монополії»: гральними кубиками, об’єктами, грошима, штрафами та ін. Ігрова логіка реалізована через WebSocket, що дозволяє всім гравцям бачити зміни в реальному часі, включаючи чат, таймери та зміни стану гри.

Крім того, система виконує функції збору статистики, підрахунку перемог і формування рейтингових списків гравців. Ці рейтинги впливають на підбір суперників у майбутніх матчах, а також дозволяють реалізовувати змагання та нагороди для найактивніших учасників.

Усі ці функції пов’язані між собою логічно та реалізовані через взаємодію фронтенд-компонентів і бекенд-служб у режимі реального часу або через API-запити, що забезпечує динамічну і плавну роботу застосунку.

**2.3 User Characteristics**

Цільова аудиторія застосунку «MonopolyUA» складається з широкого кола користувачів, основну частину яких формують підлітки, молодь та дорослі віком від 14 до 35 років. Більшість користувачів мають базовий або середній рівень комп’ютерної грамотності, тому система повинна бути простою, інтуїтивно зрозумілою та доступною без потреби проходження навчання або читання інструкцій.

Користувачі знайомі з класичними браузерними іграми, соціальними мережами та мобільними додатками, що формує очікування до швидкого відгуку, інтерактивності інтерфейсу та соціальної взаємодії. З огляду на це, інтерфейс застосунку повинен мати мінімалістичний, адаптивний дизайн, бути зрозумілим навіть без текстових підказок і працювати стабільно на більшості сучасних браузерів.

З огляду на наявність функцій для гри в реальному часі, чатів і рейтингових систем, очікується, що користувачі мають досвід участі в багатокористувацьких онлайн-іграх або соціальних платформах. Це дозволяє припустити достатній рівень комфортності з механіками типу «додати у друзі», «запросити в лобі», «вийти з гри», «перевірити інвентар» тощо.

Водночас розробка враховує й менш досвідчених користувачів, тому реалізована система підказок, автоматичних повідомлень про хід гри, а також механізм обробки AFK-ситуацій, щоб уникнути збоїв у ігровому процесі. Усі ці аспекти враховуються в дизайні інтерфейсів і логіці бекенду з метою створення комфортного та збалансованого досвіду для всіх користувачів.

**2.4 Constraints**

В процесі розробки веб-гри «MonopolyUA» необхідно врахувати низку обмежень, що визначають технічні, організаційні та регуляторні рамки проєкту:

а) регуляторні політики та захист даних – система обробляє персональні дані користувачів (електронна пошта, аватар, історія ігор), тому розробка та експлуатація повинні відповідати вимогам GDPR (ЄС), Закону «Про захист персональних даних» (Україна) та аналогічних регуляцій. Надсилання паролів і повідомлень через електронну пошту реалізується з використанням TLS-з’єднання та захищеного SMTP-серверу;

б) інтеграція з зовнішніми інтерфейсами:

1) OAuth 2.0 для Google Sign-In;

2) SMTP-сервер для відновлення паролю та системних сповіщень.

Всі API-виклики до сторонніх сервісів повинні відбуватися через захищені HTTP-канали з перевіркою сертифікату.

в) паралельна робота та висока доступність – система підтримує одночасні ігрові сесії та кілька WebSocket-з’єднань на одного користувача. Використання Redis для розподіленого зберігання станів лобі та ігор забезпечує горизонтальне масштабування й синхронну роботу кількох інстансів сервера.

г) аудит і логування - усі ключові події (створення лобі, приєднання/вихід гравця, хід гравця, фінансові транзакції, банкрутство тощо) фіксуються в хронологічних лістах Redis та можуть бути експортовані для подальшого аналізу. Журнали також відіграють роль розробницького та експлуатаційного аудиту.

д) контрольні та безпекові функції:

1) CSRF-захист для REST-інтерфейсу;

2) валідація та очищення всіх вхідних даних, включно з повідомленнями чату;

3) захист WebSocket-каналів через JWT-мідлвера із обмеженим часом життя токена.

е) критичність додатку – хоча «MonopolyUA» не є системою із життєво критичними операціями, користувачі очікують безперебійної та чуйної роботи гри. Будь-яка втрата стану партії може призвести до розчарування та відтоку гравців;

ж) мовні та платформові вимоги – розробка виконана на Python 3.8+ з використанням Django 5.1 і Django Channels. Для клієнтської частини застосовано сучасні версії JavaScript ES6+. Будь-які сторонні бібліотеки повинні бути з відкритим кодом та підтримуватися спільнотою.

Ці обмеження визначають рамки вибору інструментів, архітектурних рішень та процедур розробки, усуваючи ризики невідповідності регламентам, гарантуючи безпеку й стабільність системи.

**2.5 Assumption and Dependencies**

Реалізація функціоналу веб-застосунку «MonopolyUA» ґрунтується на низці припущень і зовнішніх залежностей, які безпосередньо впливають на виконання вимог, описаних у цьому документі. У разі зміни цих припущень — вимоги можуть потребувати перегляду.

* Вся логіка зберігання стану гри, таймерів, хронології подій та менеджера ходів залежить від стабільної роботи Redis-сервера. У разі зміни архітектури з Redis на інше in-memory сховище (наприклад, Memcached або PostgreSQL Pub/Sub), вимоги щодо продуктивності та синхронізації повинні бути оновлені.
* Передбачається, що серверна частина продовжуватиме функціонувати в середовищі Django + Channels. Заміна цих технологій (наприклад, на FastAPI або Node.js) потребуватиме переосмислення архітектури WebSocket-комунікацій, мідлверів, а також механізмів автентифікації.
* Уся ідентифікація користувачів у WebSocket-з’єднаннях базується на JWT-токенах. Якщо зміниться метод автентифікації (наприклад, перехід на session-based або OAuth-only), буде потрібно оновити логіку обробки scope у WS-підключеннях.
* Механізм відновлення паролю залежить від можливості відправити лист користувачу з новим згенерованим паролем. У разі недоступності поштового сервісу функціональність скидання пароля буде порушено, що вимагає зміни SRS (наприклад, через перехід до верифікаційних токенів).
* Реалізація сторонньої автентифікації залежить від працездатності Google API. У разі зміни політики OAuth або недоступності Google Identity Platform — необхідна адаптація системи (наприклад, через підтримку інших провайдерів або fallback на внутрішню автентифікацію).
* Система продажу/купівлі скінів ґрунтується на внутрішньому ігровому балансі користувача. Припускається, що не планується підключення реальних платіжних систем. Якщо таке припущення зміниться, SRS доведеться переглянути з урахуванням фінансових і юридичних вимог (зокрема, підтримки реальних грошей, податків і безпеки транзакцій).
* Передбачається, що користувачі взаємодіють із грою через сучасні браузери, які підтримують WebSocket, localStorage та інші сучасні веб-технології. У разі зміни цільової платформи (наприклад, підтримка мобільного додатку або застарілих браузерів), інтерфейс та архітектура клієнта мають бути адаптовані.

Ці припущення є фундаментом для визначення поточних вимог. У разі їх зміни функціональні або нефункціональні вимоги до системи можуть потребувати перегляду, адаптації або переосмислення.

**2.6 Apportioning of Requirements**

У процесі планування розробки системи «MonopolyUA» було виявлено, що деякі вимоги, хоча й важливі, можуть бути реалізовані на подальших етапах через обмеження часу, ресурсів або складність реалізації. Враховуючи обсяг проекту, розробка розбивається на кілька ітерацій, де пріоритетними є функції, критичні для базової роботи системи.

До першої ітерації включено:

* Реалізація основної гри з класичними правилами монополії.
* Підтримка WebSocket-з’єднання для гри в реальному часі.
* Механізм черги ходів із таймерами.
* Базова система логів і чату.
* Авторизація (звичайна + Google OAuth).
* Профіль користувача з переглядом статистики та редагуванням даних.
* Базова система друзів і повідомлень.
* Система інвентарю та торгівлі скінів за внутрішню валюту.
* Застосування скінів до придбаних власностей під час гри.

До наступних ітерацій (можуть бути реалізовані в майбутньому):

* Розширена система анімацій на полі гри.
* Візуальна мапа всіх власностей і скінів гравців у поточній сесії.
* Повноцінна мобільна версія інтерфейсу (адаптація для touch-інтерфейсів).
* Доопрацювання торгової системи з можливістю створення аукціонів або обміну між гравцями.
* Введення нагород, досягнень та бойових пропусків.
* Створення системи репортів/скарг на гравців.
* Статистика по іграх друзів.
* Публічні/приватні лобі з паролем і фільтрами.
* Можливість створювати кастомні правила гри (варіативність правил).

Пріоритетність реалізації функцій у майбутніх ітераціях визначатиметься відповідно до відгуків користувачів, технічних можливостей та стратегічних цілей проєкту. Рішення про відкладення реалізації певних вимог приймалося у співпраці з замовником і командою розробки.

**3 Specific Requirements**

**3.1 External Interfaces**

**3.1.1 Інтерфейс автентифікації користувача:**

а) опис: Забезпечує реєстрацію, вхід, авторизацію через Google (OAuth) та відновлення паролю;

б) джерело введення: Користувач через форму входу/реєстрації;

в) призначення виводу: Сервер автентифікації;

г) допустимі значення:

1) електронна пошта: у форматі, що відповідає стандарту RFC 5322;

2) пароль: від 6 до 128 символів, щонайменше одна літера та одна цифра.

д) одиниці виміру: Текстові поля (рядки);

е) часова характеристика: Під час взаємодії з формою входу або реєстрації;

ж) зв’язки з іншими інтерфейсами: Профіль користувача, сесії гри.

з) формат вікна: Окремі форми входу та реєстрації;

и) формат даних: JSON;

к) фінальні повідомлення: “Вхід успішний”, “Невірний пароль”, “Користувача не знайдено”.

**3.1.2 Інтерфейс профілю користувача:**

а) опис: Дозволяє переглядати та редагувати нікнейм, аватар, пароль і переглядати статистику ігор;

б) джерело введення: Користувач на сторінці профілю;

в) призначення виводу: База даних користувачів;

г) допустимі значення:

1) нікнейм: 3–20 символів, тільки латинські літери, цифри, символи "\_", "-";

2) Зображення: .png, .jpg, розмір до 5MB.

д) одиниці виміру: Рядки, зображення;

е) часова характеристика: Після входу до профілю;

ж) зв’язки з іншими інтерфейсами: Аутентифікація;

з) формат вікна: Вкладка "Профіль";

и) формат даних: JSON або multipart/form-data;

к) фінальні повідомлення: “Профіль оновлено”, “Файл занадто великий”.

**3.1.3 Інтерфейс лобі гри:**

а) опис: Дозволяє створювати ігрові кімнати або приєднуватися до них;

б) джерело введення: Користувач із головного меню або за посиланням-запрошенням;

в) призначення виводу: Менеджер сесій гри;

г) допустимі значення:

1) назва кімнати: 3–30 символів;

2) кількість гравців: від 2 до 4.

д) одиниці виміру: Рядки, числа;

е) часова характеристика: До початку гри;

ж) зв’язки з іншими інтерфейсами: Список друзів, повідомлення;

з) формат вікна: Список доступних кімнат або вікно створення;

и) формат даних: JSON;

к) фінальні повідомлення: “Кімната створена”, “Гравець приєднався”, “Гру розпочато”.

**3.1.4 Інтерфейс самої гри:**

а) опис: Головне ігрове поле, чат, лог ходів, панель гравця, менеджер ходу;

б) джерело введення: Дії користувача під час гри;

в) призначення виводу: Сервер гри через WebSocket;

г) допустимі значення: Команди: "кинути кубики", "купити", "пропустити", "передати", "взяти заставу" тощо.

д) одиниці виміру: Команди, повідомлення;

е) часова характеристика: У режимі реального часу;

ж) зв’язки з іншими інтерфейсами: Менеджер черги, чат, профіль;

з) формат вікна: Ігрове поле з боковою панеллю дій;

и) формат даних: WebSocket (JSON);

к) фінальні повідомлення: “Хід завершено”, “Карточку куплено”, “Гравець банкрут”.

**3.1.5 Інтерфейс торгової платформи:**

а) опис: Купівля та продаж скінів за внутрішню валюту;

б) джерело введення: Користувач через сторінку ринку;

в) призначення виводу: Сервер торгівлі;

г) допустимі значення:

1) ціна: не менше 1 одиниці внутрішньої валюти;

2) кількість: не менше 1.

д) одиниці виміру: Числові та текстові фільтри;

е) часова характеристика: Доступна будь-коли;

ж) зв’язки з іншими інтерфейсами: Інвентар, база предметів;

з) формат вікна: Таблиця з фільтрами, кнопки дій;

и) формат даних: JSON;

к) фінальні повідомлення: “Предмет куплено”, “Товар знято з продажу”.

**3.1.6 Інтерфейс інвентарю:**

а) опис: Перегляд наявних скінів, вибір для гри, виставлення на продаж;

б) джерело введення: Користувач із вкладки інвентарю;

в) призначення виводу: Сервер інвентарю / торгівлі;

г) допустимі значення: Тільки ті скіни, що є у власності;

д) одиниці виміру: Назва, тип, кількість;

е) часова характеристика: У будь-який момент;

ж) зв’язки з іншими інтерфейсами: Гра, торгова платформа;

з) формат вікна: Плиткове відображення предметів;

и) формат даних: JSON;

к) фінальні повідомлення: “Предмет виставлено на продажу”.

**3.1.7 Інтерфейс повідомлень:**

а) опис: Обмін системними повідомленнями, запрошення в гру, запити в друзі;

б) джерело введення: Користувач або система;

в) призначення виводу: Інший користувач або сервер повідомлень;

г) допустимі значення: Повідомлення до 256 символів;

д) одиниці виміру: Текст;

е) часова характеристика: У режимі реального часу;

ж) зв’язки з іншими інтерфейсами: Друзі, лобі гри, профіль;

з) формат вікна: Панель сповіщень або вкладка повідомлень;

и) формат даних: JSON, WebSocket;

к) фінальні повідомлення: “Запрошення надіслано”, “Запит в друзі прийнято”.

**3.2 Functions**

**3.2.1 Реєстрація та авторизація**

3.2.1.1 Система повинна перевіряти коректність введеної електронної пошти за встановленим шаблоном.

3.2.1.2 Система повинна перевіряти, щоб пароль містив щонайменше одну цифру та одну літеру.

3.2.1.3 Система повинна надавати користувачу можливість входу через Google OAuth 2.0.

3.2.1.4 Система повинна надсилати лист із новим згенерованим паролем у разі запиту на відновлення доступу.

**3.2.2 Керування профілем користувача**

3.2.2.1 Система повинна дозволяти користувачу змінювати нікнейм, аватар та пароль.

3.2.2.2 Система повинна перевіряти унікальність нікнейму під час збереження.

3.2.2.3 Система повинна відображати статистику завершених ігор користувача (кількість ігор, перемоги, поразки тощо).

3.2.2.4 Система повинна перевіряти розмір та формат файлу аватару.

**3.2.3 Ігрове лобі та менеджер сесій**

3.2.3.1 Система повинна дозволяти створення нової ігрової кімнати з унікальним ідентифікатором.

3.2.3.2 Система повинна надавати можливість приєднання до існуючої гри за посиланням або зі списку доступних.

3.2.3.3 Система повинна перевіряти, що кількість гравців не перевищує 6 і не менше 2.

3.2.3.4 Система повинна автоматично запускати гру після досягнення мінімального числа гравців і підтвердження усіх учасників.

**3.2.4 Основна ігрова логіка**

3.2.4.1 Система повинна дозволяти користувачу кидати кубики та рухати фішку відповідно до результату.

3.2.4.2 Система повинна обробляти події при потраплянні на певну клітинку (купівля, оренда, податки, тюрма тощо).

3.2.4.3 Система повинна автоматично змінювати вигляд картки на полі, якщо у гравця є відповідний скін.

3.2.4.4 Система повинна виводити лог подій гри в режимі реального часу.

3.2.4.5 Система повинна обробляти перемогу, банкрутство та завершення гри згідно з правилами.

**3.2.5 Інвентар і скіни**

3.2.5.1 Система повинна дозволяти гравцю переглядати список наявних скінів.

3.2.5.2 Система повинна дозволяти вибирати активні скіни для гри.

3.2.5.3 Система повинна перевіряти наявність обраного скіна у гравця до його застосування в грі.

**3.2.6 Торгова платформа**

3.2.6.1 Система повинна дозволяти виставити предмет на продаж, вказавши кількість і ціну у внутрішній валюті.

3.2.6.2 Система повинна дозволяти переглядати доступні товари з фільтрацією за параметрами (тип, рідкість, ціна).

3.2.6.3 Система повинна перевіряти, чи дійсно користувач володіє відповідною кількістю товару перед виставленням.

3.2.6.4 Система повинна знімати предмет з торгівлі на запит користувача.

3.2.6.5 Система повинна зменшувати кількість валюти користувача при покупці та збільшувати її у продавця.

**3.2.7 Повідомлення та запрошення**

3.2.7.1 Система повинна дозволяти надсилання запрошення в гру іншим користувачам.

3.2.7.2 Система повинна дозволяти надсилання запитів у друзі.

3.2.7.3 Система повинна повідомляти користувача про отримання нових повідомлень у режимі реального часу.

**3.2.8 Обробка помилок та відновлення**

3.2.8.1 Система повинна інформувати користувача про неправильні або відсутні поля введення.

3.2.8.2 Система повинна зберігати стан гри при розриві з’єднання та дозволяти повторне приєднання.

3.2.8.3 Система повинна автоматично завершувати хід, якщо користувач не діє протягом визначеного часу.

**3.2.9 Канали зв’язку**

3.2.9.1 Система повинна підтримувати постійне WebSocket-з'єднання для обміну ігровими подіями в реальному часі.

3.2.9.2 Система повинна падати у fallback-режим при втраті WebSocket-з’єднання.

**3.3 Performance Requirements**

**3.3.1 Статичні вимоги**

3.3.1.1 Програмний засіб повинен забезпечувати підтримку щонайменше 500 одночасно авторизованих користувачів без деградації продуктивності.

3.3.1.2 Програмний засіб повинен підтримувати до 50 активних ігрових сесій одночасно на сервері з рекомендованими характеристиками.

3.3.1.3 Програмний засіб повинен обробляти до 1000 внутрішньоігрових транзакцій на хвилину.

3.3.1.4 Кількість одночасно відкритих з’єднань WebSocket не повинна перевищувати 1000 для одного інстансу сервера.

3.3.1.5 Кожен користувач може мати до 500 об’єктів (скінів) у власному інвентарі.

**3.3.2 Динамічні вимоги**

3.3.2.1 95% усіх ігрових дій (купівля, продаж, хід, обмін) повинні оброблятися протягом менше ніж 1 секунди після підтвердження користувачем.

3.3.2.2 Повідомлення через WebSocket повинні доставлятися з затримкою не більше 300 мс у межах однієї ігрової сесії.

3.3.2.3 Час від моменту запуску гри до повного завантаження ігрового поля на клієнтському інтерфейсі не повинен перевищувати 2 секунд.

3.3.2.4 Відкриття торгової платформи з фільтрами та переліком товарів має відбуватися за менше ніж 1.5 секунд для користувача із середнім інтернет-з'єднанням (10 Мбіт/с).

3.3.2.5 Пошук по торговій платформі повинен повертати результати за менше ніж 500 мс при вибірці до 1000 елементів.

**3.3.3 Продуктивність при пікових навантаженнях**

3.3.3.1 Під час пікових навантажень (до 500 користувачів онлайн), програмний засіб повинен зберігати не нижче 90% від заявленої базової швидкодії для обробки ігрових подій.

3.3.3.2 Програмний засіб повинен виконувати балансування навантаження при розгортанні на кількох інстансах для підтримки горизонтального масштабування.

**3.4 Logical Database Requirements**

**3.4.1 Типи інформації**

До основних типів даних, що використовуються в системі, належать:

* Облікові записи користувачів (e-mail, хеш пароля, нікнейм, аватар, ID, Google OAuth-токени);
* Друзі та запити в друзі;
* Повідомлення (тип, відправник, одержувач, час надсилання, статус);
* Ігрові сесії (учасники, стан гри, ігрове поле, лог ходів, чат);
* Статистика користувача (кількість перемог, поразок, зіграних ігор, середня тривалість гри);
* Інвентар (список скінів, кількість, ID користувача);
* Торгівельна платформа (активні лоти, ціна, кількість, продавець);
* Скіни (ID, тип, належність до типу властивості на полі);
* Ігрова валюта (баланс користувача, історія транзакцій).

**3.4.2 Частота використання**

* Дані профілю користувача: кожен сеанс входу / перегляду профілю.
* Ігрові сесії: активне читання/запис під час гри.
* Повідомлення: в режимі реального часу (висока частота доступу).
* Торгові операції: помірна частота доступу.
* Інвентар: висока частота у гравців, що часто змінюють скіни або торгують.
* Статистика: обробка після завершення кожної гри, доступ для перегляду в профілі.

**3.4.3 Можливості доступу**

* Дані користувача: доступ тільки авторизованого користувача або адміністратора.
* Ігрові сесії: доступ лише учасників гри.
* Повідомлення: доступ для відправника/одержувача.
* Торгова платформа: публічний перегляд, редагування — лише власник лоту.
* Інвентар: доступ користувача до власного інвентаря.

**3.4.4 Сутності та зв’язки**

Основні сутності:

* User — має інвентар, скіни, статистику, список друзів, ігрову валюту.
* GameSession — містить багатьох User, Chat, Log.
* Skin — належить одному User, відображається на об’єктах GameSession.
* Inventory — належить User, зберігає Skin з кількістю.
* TradeLot — створений User, містить Skin, кількість, ціну.
* Message — містить відправника та одержувача.
* FriendRequest — створюється User і спрямовується іншому User.

Зв’язки реалізуються через зовнішні ключі, де це можливо.

**3.4.5 Обмеження цілісності**

* Унікальність email та нікнейму користувача.
* Неможливість продажу більшої кількості скінів, ніж є в інвентарі.
* Неможливість участі користувача в декількох активних іграх одночасно.
* Валідація цінових значень на торговій платформі (додатні числа).
* Захист від SQL-інʼєкцій та некоректного вводу даних.

**3.4.6 Вимоги до збереження даних**

* Дані користувача та інвентарю мають зберігатися постійно, навіть після тривалого періоду неактивності.
* Історія ігор та статистика зберігається не менше 1 року.
* Повідомлення видаляються після 6 місяців, якщо не прочитані.
* Ігрові сесії очищуються через 24 години після завершення гри.
* Лоти з торгової платформи видаляються через 30 днів, якщо не реалізовані.

**3.5 Design Constraints**

**3.5.1 Standards Compliance**

У процесі розробки веб-застосунку "MonopolyUA" враховуються обмеження, пов'язані з дотриманням сучасних веб-стандартів, особливостями використовуваного середовища розгортання та регламентами обліку даних. Всі звіти та повідомлення, що створюються системою, повинні мати зрозумілий формат, відповідний сучасним вимогам до веб-інтерфейсів. Назви полів і змінних даних повинні відповідати зрозумілим іменам, прийнятим у галузі веб-розробки, із використанням camelCase або snake\_case відповідно до призначення. У системі буде реалізовано механізм журналювання критичних змін, зокрема купівлі, продажу або відкриття кейсів, що дозволить створити повну трасу змін стану гравця та інвентарю для перевірки цілісності транзакцій. Дані мають зберігатися відповідно до політик безпечного обміну та захисту персональних даних, що особливо важливо при авторизації через сторонні сервіси, як-от Google API.

**3.6 Software System Attributes**

**3.6.1 Reliability**

Система повинна бути достатньо надійною для стабільної роботи під навантаженням до 500 одночасних користувачів. Усі критичні компоненти, як-от обробка авторизації, ігрового процесу та транзакцій у маркеті, повинні бути протестовані на стійкість до збоїв. Надійність перевірятиметься шляхом багатократного відтворення основних сценаріїв взаємодії з користувачем із вимірюванням кількості помилок на 1000 транзакцій. Очікувана середня безвідмовна тривалість роботи має становити не менше 200 годин.

**3.6.2 Availability**

Програмна система має бути доступною цілодобово, оскільки передбачає багатокористувацький онлайн-доступ до сервісу. У разі збоїв користувач повинен мати можливість поновити сесію або гру із втратою не більше 5 секунд ігрових даних. Для забезпечення цього буде використовуватись збереження проміжного стану гри та асинхронна обробка запитів.

**3.6.3 Security**

Система повинна захищати користувацькі дані від несанкціонованого доступу, використовуючи JWT для автентифікації. Критичні дані (наприклад, паролі та токени) мають зберігатися у зашифрованому вигляді. Має бути реалізовано контроль за правильністю передачі повідомлень у WebSocket-каналах.

**3.6.4 Maintainability**

Проєкт побудований за принципом клієнт-серверної архітектури, де фронтенд і бекенд реалізовані як окремі незалежні додатки. Такий підхід дозволяє розвивати інтерфейс користувача та серверну логіку автономно, що спрощує підтримку й модернізацію системи.

Бекенд реалізовано з використанням Django REST Framework, що забезпечує модульну організацію коду. Основна логіка поділена на функціональні сегменти: автентифікація, управління профілем, маркет, ігрові механіки та чат. Це дає змогу легко змінювати або розширювати окремі частини системи без впливу на інші компоненти.

**3.6.5 Portability**

Портативність веб-застосунку «MonopolyUA» забезпечується використанням платформонезалежних технологій: Django для бекенду та стандартних технологі HTML/CSS/JS для фронтенду. Усі системозалежні налаштування (наприклад, шляхи до файлів, змінні середовища) винесені у конфігураційні файли, що дозволяє легко адаптувати систему під інші хост-машини або ОС. Використання перевірених інструментів, таких як Docker, забезпечує спільне середовище виконання на будь-якій платформі. Код не містить значної кількості хост-залежних фрагментів і протестований на Windows. Вимірювання портативності відбуватиметься шляхом запуску застосунку в різних середовищах та оцінки змін, потрібних для успішної роботи. Таким чином, переносимість оцінюється як висока.

**3.7 Organizing the Specific Requirements**

**3.7.1 System Mode**

Веб-застосунок «MonopolyUA» має різні режими роботи, зокрема режим реєстрації та авторизації, режим роботи з профілем користувача, ігровий режим, режим роботи магазину та лідерборду. Кожен режим має свої особливості інтерфейсу та логіки, тому вимоги до системи формулюються окремо для кожного режиму, враховуючи специфіку відображення даних і поведінки системи, що дозволяє підвищити зручність та продуктивність.

**3.7.2 User Class**

Система розподіляє функціонал залежно від класу користувача. Зареєстровані гравці мають доступ до повного набору можливостей — участь в іграх, покупка скінів, додавання друзів. Адміністратори отримують розширені права для керування контентом і користувачами, а незареєстровані користувачі можуть лише ознайомитися з правилами та зареєструватися.

**3.7.3 Objects**

Основними об’єктами системи є користувач, ігрова сесія, ігрове поле, інвентар скінів, товари магазину, чат, рейтингові таблиці та списки друзів. Кожен об’єкт має свої атрибути (наприклад, профіль користувача містить ім’я, аватар, статистику), а також функції (додавання друга, створення лоббі, відправка повідомлення), що забезпечують реалізацію бізнес-логіки.

**3.7.4 Feature**

Функції системи визначаються як конкретні послуги, які вона надає користувачам, наприклад, реєстрація, авторизація, створення та пошук ігрових сесій, купівля скінів, обмін повідомленнями в чаті, оновлення профілю. Для кожної функції описується послідовність дій, необхідних для її виконання, включно з вхідними даними та очікуваною відповіддю системи.

**3.8 Additional Comments**

Для веб-застосунку «MonopolyUA» доцільно застосовувати одночасно кілька способів організації специфічних вимог, описаних у пункті 3.7, оскільки це допоможе краще структурувати вимоги та врахувати різні аспекти роботи системи.

Зокрема, організація вимог за режимами роботи системи (реєстрація, ігровий режим, магазин тощо) допоможе чітко виділити логіку для кожного сценарію використання. Водночас розподіл за класами користувачів (гравці, адміністратори, гості) дозволить чітко визначити права доступу та функціональні можливості.

Використання об’єктного підходу сприяє кращому опису структури системи та взаємодії між основними компонентами, такими як ігрова сесія, користувачі та інші. Організація вимог за функціями дозволяє деталізувати послідовність дій користувача та реакцію системи, що важливо для реалізації бізнес-логіки.

**4. Change Management Process**

Зміни в вимогах до веб-застосунку «MonopolyUA» будуть контролюватися через чітко встановлений процес управління змінами. Клієнт не може просто зателефонувати або висловити нову ідею усно — всі пропозиції щодо змін мають подаватися офіційно у письмовій формі електронною поштою. Після отримання запиту команда проекту проводить його оцінку з урахуванням технічних можливостей, впливу на поточний план робіт і бюджету. Всі рішення щодо впровадження змін ухвалюються колективно командою.

**5. Document Approvals**

**TBD.**

**6. Supporting Information**

Цей розділ містить допоміжні матеріали, що полегшують читання, розуміння та використання документа Специфікації вимог до програмного забезпечення (SRS). Хоча вони не є безпосередньою частиною формальних вимог, ці матеріали служать важливим доповненням для розробників, тестувальників, замовників та інших зацікавлених осіб.

Зміст документа представлено на початку файлу й охоплює всі розділи: від вступу до функціональних і нефункціональних вимог, включно з додатками.

Інтерактивний або текстовий покажчик термінів, абревіатур і ключових понять, які використовуються у документі. У разі публікації в цифровому вигляді, передбачена можливість пошуку за ключовими словами (наприклад: "WebSocket", "JWT", "Redis").