( Вебзастосунок для підтримки гри Brick Prince)

(Балицький Ігор Олександрович)

**Software Requirements Specification**

**Document**

**Version: 1** **Date: (05/20/2025)**

1. Вступ

1.1 Призначення

Цей документ є специфікацією програмних вимог (SRS) до вебзастосунку Brick Prince. Він призначений для розробників, тестувальників, замовників, технічних експертів і всіх зацікавлених сторін, які беруть участь у процесі створення, тестування та підтримки даного програмного забезпечення. Документ визначає функціональні та нефункціональні вимоги до системи, а також встановлює межі та основні припущення про її реалізацію.

Метою документа є надання чітких і зрозумілих вимог до програмного продукту, що слугуватимуть основою для його проєктування, розробки та перевірки.

1.2 Область застосування

Brick Prince — це сучасний вебзастосунок, розроблений як інформаційна платформа для підтримки мобільної гри однойменного ігрового проєкту. Застосунок дозволяє користувачам отримувати гру, стежити за новинами, залишати відгуки, оцінювати інші коментарі, а адміністраторам — модерувати контент і керувати користувачами.

Система охоплює такі функції:

* реєстрація та вхід користувачів;
* перегляд новин про гру;
* завантаження гри за email;
* створення та редагування відгуків;
* оцінка відгуків через лайки/дизлайки;
* панель адміністратора з доступом до користувачів, новин, відгуків;
* підтримка багатомовного інтерфейсу.

Програмне забезпечення призначене для використання через сучасні веб браузери на настільних комп’ютерах і мобільних пристроях. Система не включає безпосередньо геймплей гри, її серверну частину або функції внутрішньоігрового процесу — лише зовнішню підтримку гри з боку спільноти та адміністраторів.

Основною метою є створення зручного й масштабованого інструменту для взаємодії між гравцями та розробниками проєкту Brick Prince.

1.3 Терміни, скорочення та абревіатури

У цьому документі використовуються такі терміни та скорочення:

* SRS — Software Requirements Specification, специфікація програмних вимог
* UI — User Interface, користувацький інтерфейс
* UX — User Experience, досвід користувача
* CRUD — Create, Read, Update, Delete — базові операції з даними
* SQLite — легка реляційна база даних
* REST API — архітектурний стиль прикладного інтерфейсу для взаємодії клієнт-сервер
* HTML/CSS/JS — мови для створення інтерфейсів вебзастосунків
* FastAPI — фреймворк для побудови веб-API на Python
* JWT — JSON Web Token — формат для авторизації
* i18n — Internationalization — підтримка багатомовності
* SMTP — Simple Mail Transfer Protocol — протокол надсилання електронних листів

1.4 Перелік використаних джерел

У документі було використано наступні джерела:

1. ISO/IEC/IEEE 29148:2018. Systems and software engineering — Life cycle processes — Requirements engineering.
2. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання.
3. Офіційна документація FastAPI. URL: https://fastapi.tiangolo.com
4. Офіційна документація SQLite. URL:<https://sqlite.org>
5. Mozilla Developer Network (MDN): HTML, CSS, JavaScript Docs. URL:<https://developer.mozilla.org>
6. Python Software Foundation. Official Python Docs. URL:<https://docs.python.org>
7. Bootstrap Documentation. URL:<https://getbootstrap.com>
8. Git SCM Documentation. URL:<https://git-scm.com/doc>
9. RFC 5321 – Simple Mail Transfer Protocol (SMTP). URL: https://datatracker.ietf.org/doc/html/rfc5321

1.5 Огляд

Даний документ специфікації програмних вимог складається з трьох основних розділів:

* Розділ 1 надає загальну інформацію про призначення, обсяг, скорочення та джерела, що використовуються.
* Розділ 2 описує загальну структуру програмного продукту, включаючи його функціональні можливості, характеристики користувача, обмеження та припущення.
* Розділ 3 детально описує конкретні вимоги до функціоналу, інтерфейсів, продуктивності, безпеки, атрибутів системи тощо. Саме на цей розділ орієнтуються розробники під час реалізації програмного продукту.

Після ознайомлення з цим документом користувачі, розробники та інші зацікавлені особи отримають чітке уявлення про вимоги до вебзастосунку Brick Prince, що дозволить успішно реалізувати його функціональні та нефункціональні компоненти.

2. Загальний опис

Цей розділ описує загальні фактори, що впливають на розробку програмного продукту Brick Prince WebApp, а також пояснює передумови та контекст для кращого розуміння вимог, сформульованих у розділі 3. Інформація, наведена тут, орієнтована на кінцевого замовника або користувача й подається у зрозумілій формі без складної технічної специфікації.

2.1 Перспектива програмного продукту

Вебзастосунок Brick Prince є незалежною інформаційною системою, створеною спеціально для підтримки гравців, розробників і адміністрації однойменної гри у жанрі roguelike. Він забезпечує низку функцій: надання доступу до гри для незареєстрованих користувачів, зберігання відгуків, перегляд новин, взаємодію з користувачами, а також адміністрування контенту. Продукт є повністю самостійним і не потребує зовнішніх комерційних сервісів для основної функціональності.

Програмне забезпечення побудовано за клієнт-серверною архітектурою з REST API, де фронтенд реалізовано на HTML/CSS/JS, а серверна частина — на Python з використанням FastAPI. База даних — SQLite. Застосунок розгортається локально або на сервері, не вимагає складного середовища або ліцензованих систем.

Хоча подібні системи управління спільнотою гравців вже існують (наприклад, Steam Community, itch.io profiles), Brick Prince WebApp має більш вузьку спеціалізацію — він націлений на обслуговування однієї конкретної гри з урахуванням її унікальної механіки та специфічного аудиторного фокусу. Це дозволило реалізувати набір функцій, які відповідають саме потребам розробників гри та її користувачів, без перевантаження сторонніми можливостями.

У разі інтеграції з іншими сервісами (наприклад, надсиланням листів через SMTP або оновленням клієнта гри) інтерфейси взаємодії реалізуються відповідно до стандартних протоколів, забезпечуючи масштабованість і майбутню сумісність.

2.1.1 Інтерфейси системи

Система Brick Prince WebApp взаємодіє з зовнішніми службами лише частково. Основною інтеграцією є модуль надсилання електронних листів. Цей функціонал реалізовано через SMTP-протокол, що дозволяє надсилати листи з ключами доступу до гри користувачам, які ввели свій email на головній сторінці. Жодна стороння система типу платіжних шлюзів або соціальних мереж не інтегрується з нашим застосунком. У разі розширення функціональності можливе додавання API взаємодії з ігровим клієнтом або сервісами аналітики, однак наразі система залишається ізольованою та самодостатньою.

2.1.2 Користувацькі інтерфейси

Інтерфейс програми побудований на основі сучасного графічного веб-інтерфейсу (GUI), що дозволяє взаємодіяти із системою за допомогою миші та клавіатури. Особлива увага приділена зручності користування: меню, кнопки та форми розташовані логічно, з дотриманням принципів UX. Інтерфейс доступний українською та англійською мовами, перемикач знаходиться у правому верхньому куті.

Дизайн сайту адаптований до різних розмірів екранів і не містить надлишкових візуальних ефектів, що забезпечує швидке завантаження і комфортну роботу навіть на слабких пристроях. Враховано потреби користувачів з мінімальним досвідом — усі функції інтуїтивно зрозумілі, а помилки вводу супроводжуються поясненнями.

2.1.3 Апаратні інтерфейси

Програмне забезпечення Brick Prince WebApp не вимагає безпосередньої взаємодії з апаратним забезпеченням. Жодних спеціалізованих пристроїв, драйверів або додаткового обладнання не використовується. Система функціонує у веббраузері, що дозволяє уникнути апаратної залежності. Таким чином, вимоги до апаратних інтерфейсів відсутні.

2.1.4 Програмні інтерфейси

Застосунок взаємодіє з низкою програмних компонентів, які є необхідними для його роботи. Основним компонентом є СКБД SQLite, з якою серверна частина на Python/FastAPI працює напряму через відповідні бібліотеки. У частині роботи з поштовим сервером використовується бібліотека smtplib для встановлення SMTP-з'єднання.

Інтерфейс клієнт-сервер реалізовано у форматі REST API. Фронтенд надсилає HTTP-запити на бекенд, отримуючи у відповідь JSON-дані. Для забезпечення кросбраузерності використовуються стандартні технології HTML, CSS і JavaScript без використання специфічних фреймворків.

2.1.5 Комунікаційні інтерфейси

Основною комунікаційною взаємодією є передача даних між клієнтом і сервером через протокол HTTP(S). Всі запити до API реалізовано як REST-запити, відповіді формуються у форматі JSON. Надсилання листів відбувається через SMTP-з'єднання з налаштованим поштовим сервером. Встановлення з'єднання та автентифікація відбуваються відповідно до RFC 5321.

2.1.6 Обмеження пам’яті

Система Brick Prince WebApp не накладає суворих обмежень на використання оперативної чи постійної пам’яті. Оскільки база даних працює на SQLite та всі дані локально зберігаються у вигляді простих таблиць, навіть слабкі сервери або локальні машини можуть успішно запускати застосунок. У мінімальній конфігурації достатньо 512 МБ оперативної пам’яті, проте реальні вимоги можуть бути ще нижчими.

2.1.7 Операції

Застосунок підтримує як інтерактивний режим роботи користувача (авторизація, додавання відгуку, редагування профілю), так і пасивний перегляд публічної інформації (новини, відгуки, отримання гри). Резервне копіювання даних здійснюється вручну — копією файлу бази даних SQLite. Відновлення відбувається шляхом заміни файлу на резервний. Система не потребує обслуговування вночі чи спеціального режиму для виконання фонових задач.

2.1.8 Вимоги до адаптації під конкретну локацію

Для встановлення та запуску застосунку не потрібно жодних спеціальних змін у середовищі користувача. Система не вимагає фізичної адаптації серверного приміщення або закупівлі додаткового обладнання. Єдине обмеження — наявність браузера та Python-оточення на сервері. Перед встановленням необхідно створити та ініціалізувати базу даних, а також налаштувати SMTP, що виконується один раз згідно з інструкцією в документації.

2.2 Функції програмного продукту

Програмне забезпечення Brick Prince WebApp виконує роль вебзастосунку, що забезпечує зручну взаємодію користувача з інформацією про гру «Brick Prince», а також дозволяє залишати відгуки, переглядати новини про гру та отримувати копію гри на електронну пошту.

Основні функції системи охоплюють такі ключові напрями:

1. Надсилання гри незареєстрованим користувачам.  
 На головній сторінці сайту розміщено форму, в яку користувач може ввести адресу електронної пошти. Після надсилання адреси система автоматично генерує лист з посиланням на завантаження гри та надсилає його за допомогою SMTP-сервера. Це дозволяє поширювати гру серед нових гравців без необхідності проходження реєстрації.

2. Реєстрація та авторизація користувачів.  
 Система надає можливість створення облікового запису з унікальною адресою електронної пошти та паролем. Після успішної реєстрації користувач отримує доступ до особистого кабінету, де може переглядати і змінювати свої дані. Авторизовані користувачі отримують додаткові можливості, зокрема — залишати відгуки про гру.

3. Перегляд та керування відгуками.  
 Кожен зареєстрований користувач має право створити один відгук, який можна згодом редагувати або видалити. Всі відгуки виводяться на сторінці з пагінацією. Відгуки містять ім’я автора, текст, дату, оцінку гри (від 1 до 10), а також можливість поставити «лайк» або «дизлайк». Лише зареєстровані користувачі можуть взаємодіяти з реакціями.

4. Реакції на відгуки (лайки/дизлайки).  
 Користувач може поставити одну реакцію на відгук: лайк або дизлайк. Повторне натискання прибирає реакцію, а зміна типу реакції автоматично оновлює стан у базі. Це стимулює прозору оцінку вмісту без спаму.

5. Перегляд новин про гру.  
 На окремій сторінці виводиться стрічка новин, створена адміністраторами. Вона включає заголовок, дату публікації та повний текст кожної новини. Таким чином, користувачі можуть бути поінформовані про останні оновлення, зміни у грі або майбутні події.

6. Панель адміністратора.  
 Користувачі з роллю «admin» мають доступ до адміністративної панелі, де можуть:

* переглядати всіх зареєстрованих користувачів;
* змінювати роль або видаляти користувачів;
* створювати, редагувати або видаляти новини;
* переглядати список усіх відгуків та при потребі модерувати їх.

7. Локалізація інтерфейсу.  
 Інтерфейс сайту підтримує перемикання між українською та англійською мовами. Це забезпечує зручність використання для ширшої аудиторії. Мовне перемикання зберігається під час навігації між сторінками.

8. Адаптивний дизайн.  
 Сайт оптимізовано для перегляду на різних пристроях, включно з мобільними. Навігація та відображення вмісту адаптуються до розміру екрану, що забезпечує зручний доступ до всіх функцій незалежно від типу пристрою.

2.3 Характеристики користувачів

Цільова аудиторія вебзастосунку Brick Prince WebApp охоплює широкий спектр користувачів, від звичайних гравців мобільної гри до адміністраторів, відповідальних за її підтримку та розвиток. Загальні характеристики користувачів мають прямий вплив на побудову інтерфейсу, логіку взаємодії та рівень доступних функцій.

Кінцеві користувачі гри (гравці):  
Це переважно молоді люди віком від 13 до 35 років, з базовим або середнім рівнем цифрової грамотності. Вони мають досвід користування мобільними додатками, знайомі з базовими вебінтерфейсами та очікують просту й інтуїтивну взаємодію. Через це інтерфейс було реалізовано у зрозумілій формі з адаптивним дизайном і логічною структурою навігації, а вся взаємодія з системою відбувається в кілька кроків — наприклад, надсилання гри на пошту або залишення відгуку.

Зареєстровані користувачі:  
Ці користувачі, окрім доступу до публічного вмісту, мають змогу створювати та редагувати відгуки. Для них було реалізовано особистий кабінет та базові можливості керування профілем. Ця категорія не потребує спеціальної підготовки, тому функціонал побудовано максимально прозоро, з використанням знайомих елементів інтерфейсу (кнопки, форми, підтвердження дій).

Адміністратори системи:  
Це обмежене коло осіб, які мають розширені повноваження у вебзастосунку. Вони відповідають за створення новин, управління користувачами та модерування відгуків. Їх рівень технічної підготовки вищий за середній, тому інтерфейс адміністративної панелі передбачає доступ до функцій керування базою даних через прості форми, без потреби писати SQL-запити або взаємодіяти зі складною системою адміністрування.

Таким чином, при проєктуванні інтерфейсу особливу увагу було приділено простоті, зручності та наочності, щоб забезпечити доступність для гравців без технічної підготовки, при цьому не обмежуючи можливості досвідчених користувачів і адміністраторів.

2.4 Обмеження

Під час розробки вебзастосунку Brick Prince WebApp було враховано низку обмежень, які впливають на технічні рішення, архітектуру та реалізацію окремих функціональних можливостей. Ці обмеження не пов'язані безпосередньо з функціоналом, але визначають рамки, в межах яких має працювати програмне забезпечення.

Регуляторні політики та безпека даних.  
Усі дії, пов’язані зі відгуками і збереженням персональних даних користувачів (наприклад, ім’я, email), мають відповідати основним вимогам захисту конфіденційної інформації. Застосунок не зберігає платіжну інформацію, а донати є симулятивними — задля уникнення юридичних наслідків та підвищення безпеки.

Обмеження апаратного середовища.  
Вебзастосунок орієнтований на запуск у звичайному браузері без використання спеціалізованого обладнання. Він має коректно працювати на більшості сучасних пристроїв — як на стаціонарних ПК, так і на мобільних пристроях. Система не має доступу до специфічного апаратного забезпечення користувача, і тому не взаємодіє з зовнішніми пристроями.

Інтерфейси з іншими застосунками.  
Система використовує лише внутрішні API, які були спеціально реалізовані для цього проєкту. Взаємодії з сторонніми сервісами або застосунками не передбачено, що спрощує архітектуру та підвищує автономність рішення.

Паралельна робота та масштабування.  
Застосунок не підтримує горизонтальне масштабування на рівні користувача — кожен клієнт працює із системою незалежно. Паралельна обробка запитів на бекенді реалізована через стандартні можливості FastAPI.

Контроль та аудит.  
Повноцінна система аудиту не реалізована, однак облік донатів та створення новин фіксується із прив’язкою до користувача, що забезпечує базовий рівень простежуваності дій.

Вимоги до мови розробки.  
Весь серверний код реалізовано на мові Python (версія 3.10+) з використанням фреймворку FastAPI. Клієнтська частина реалізована з використанням HTML, CSS, JavaScript та бібліотеки Fetch API для взаємодії з сервером.

Надійність і критичність.  
Оскільки система не є життєво критичною, жорсткі вимоги до безвідмовності або високої доступності не встановлювалися. Проте передбачено захист від некоректного введення даних, перевірку прав доступу та інші базові механізми забезпечення стабільності.

Таким чином, визначені обмеження не лише задають межі технічної реалізації, а й допомагають зосередити розробку на ключових аспектах взаємодії користувача з системою без перевантаження зайвими складностями.

2.5 Припущення та залежності

У процесі створення вебзастосунку Brick Prince WebApp було зроблено низку припущень, які безпосередньо вплинули на формулювання функціональних і нефункціональних вимог. Зміна будь-якого з цих чинників може призвести до перегляду окремих частин вимог, викладених у цьому документі.

По-перше, передбачається, що користувачі мають стабільне підключення до Інтернету і можуть відкривати вебзастосунок у сучасному браузері, що підтримує JavaScript та CSS3. Якщо підключення до мережі відсутнє або браузер застарілий, робота з системою стає неможливою або обмеженою.

По-друге, застосунок передбачає локальний запуск серверної частини або її розгортання на хостингу, що підтримує Python 3.10+, FastAPI та SQLite. Зміна середовища виконання, наприклад, перехід на іншу СУБД (MySQL або PostgreSQL), потребуватиме модифікацій у частині обробки запитів та ініціалізації бази даних.

По-третє, вихідною передумовою є використання тільки власного API без сторонніх сервісів або інтеграцій. Наприклад, поштові повідомлення формуються та надсилаються із сервера за допомогою стандартної бібліотеки, а не сторонніх поштових сервісів (Mailgun, SendGrid тощо). У разі потреби інтеграції з зовнішніми сервісами вимоги до системи значно зміняться.

Також, очікується, що кількість одночасних користувачів буде відносно невеликою, що дозволяє уникнути реалізації повноцінного механізму масштабування чи балансування навантаження. У разі зростання популярності проєкту й потреби обробляти запити великої кількості користувачів, архітектура серверної частини потребуватиме перегляду.

Крім того, передбачено, що адміністрування застосунку здійснюється технічно підготовленими користувачами, що мають прямий доступ до серверної частини або до панелі адміністратора. Автоматизоване резервне копіювання або розподілена система керування ролями наразі не впроваджені.

Таким чином, зазначені припущення і залежності окреслюють межі, у яких проєкт функціонує стабільно, і визначають потенційні точки адаптації системи у разі зміни зовнішніх умов або очікувань користувачів.

2.6 Розподіл вимог

У межах першої реалізації вебзастосунку Brick Prince WebApp було зосереджено увагу на реалізації основного функціоналу, необхідного для повноцінної роботи користувача із системою. Проте, враховуючи обмеження у часі, ресурсах і складності окремих компонентів, частина вимог була перенесена на наступні ітерації або розглядається як потенційне розширення у майбутніх версіях.

Функціонал, реалізований у поточній версії:

* Реєстрація та авторизація користувачів;
* Перегляд новин про гру;
* Перегляд та створення відгуків;
* Оцінювання відгуків (лайки/дизлайки);
* Надсилання гри на email користувачам;
* Панель адміністратора з можливістю керування новинами, відгуками та користувачами.

Функціонал, перенесений до наступних версій:

* Підтримка багатомовності в усьому інтерфейсі. На даному етапі реалізовано часткову локалізацію, однак повне охоплення інтерфейсу ще потребує часу. Планується включення розширеної підтримки української та англійської мов у майбутніх релізах.
* Система повідомлень або сповіщень для користувачів. Вона дозволила б інформувати про зміну статусу відгуку або новини. Цей функціонал наразі відкладений через складність реалізації асинхронної логіки на стороні сервера.
* Розширена система фільтрів для відгуків. На сьогодні реалізовано лише пагінацію та базове сортування. Фільтрація за рейтингом, датою або наявністю реакцій буде додана в наступному оновленні.
* Візуальна система аналітики або статистики в адмінпанелі. Можливість бачити кількість зареєстрованих користувачів, середній рейтинг, активність користувачів тощо запланована для майбутніх версій.

Цей розподіл дає змогу ефективно планувати подальшу розробку, а також адаптувати застосунок до зростаючих потреб цільової аудиторії. Кожен з відкладених компонентів може бути розглянутий окремо під час наступного життєвого циклу проєкту, з урахуванням технічної та часової доцільності.

3. Конкретні вимоги (Specific Requirements)

Цей розділ містить деталізовані функціональні та нефункціональні вимоги до вебзастосунку Brick Prince. Усі вимоги формулюються у формі, придатній для тестування, відстеження, реалізації та перевірки.

3.1 Зовнішні інтерфейси (External Interfaces)

3.1.1 Інтерфейс користувача:

* Сторінки входу, реєстрації, профілю користувача, новин, відгуків, адмінпанелі.
* Формати вводу: текстові поля, паролі, числові значення.
* Формати виводу: HTML-сторінки, таблиці, картки, модальні вікна.

3.1.2 Інтерфейси бази даних:

* SQLite через SQLAlchemy (або аналогічний ORM).
* Формати: JSON-відповіді API, SQL-запити.

3.1.3 Інтерфейси адміністратора:

* CRUD-функціональність для користувачів, новин, відгуків.

3.1.4 Комунікаційні інтерфейси:

* HTTP(S), RESTful API.

3.2 Функціональні вимоги (Functions)

3.2.1 Реєстрація та автентифікація:

* Система повинна забезпечувати реєстрацію з валідацією email та паролю.
* Користувач може увійти, вийти та змінити свої дані профілю.

3.2.2 Перегляд та створення відгуків:

* Користувач може створити лише один відгук, редагувати чи видалити його.
* Відгуки відображаються з датою, рейтингом, текстом, кількістю лайків/дизлайків.

3.2.3 Реакції на відгуки:

* Користувач може поставити лайк або дизлайк; повторне натискання скасовує дію.
* Можливе лише одне активне оцінювання на один відгук від користувача.

3.2.4 Адмінпанель:

* Можливість редагування/видалення новин, користувачів та відгуків.
* Пошук за email та назвами.

3.2.5 Надсилання гри:

* Незареєстрований користувач може ввести email для отримання гри.

3.3 Вимоги до продуктивності (Performance Requirements)

* 90% запитів до API повинні оброблятись менше ніж за 500 мс.
* Система має підтримувати одночасну роботу щонайменше 50 користувачів.

3.4 Логічні вимоги до бази даних (Logical Database Requirements)

* Таблиці: users, feedback, news, likes, reviews, settings.
* Всі зовнішні ключі мають бути збережені, забезпечено цілісність даних.
* Відгуки та реакції мають бути зв'язані з користувачем.

3.5 Обмеження дизайну (Design Constraints)

* Використання SQLite як БД.
* Реалізація на Python (FastAPI) для бекенду, HTML/CSS/JS для фронтенду.

3.5.1 Дотримання стандартів:

* Формати CSV для експорту/імпорту.
* RESTful структурування API.

3.6 Атрибути програмної системи (Software System Attributes)

3.6.1 Надійність:

* Система повинна функціонувати без збоїв при стандартному використанні.
* При втраті зв'язку відображається повідомлення, без втрати введених даних.

3.6.2 Доступність:

* Доступ до клієнтської частини 24/7, за винятком технічного обслуговування.

3.6.3 Безпека:

* Захист від SQL-ін’єкцій, XSS, CSRF.
* Паролі зберігаються в хешованому вигляді.

3.6.4 Підтримуваність:

* Структура проєкту повинна дозволяти легке доповнення нових модулів.

3.6.5 Портативність:

* Система має бути сумісною з основними браузерами (Chrome, Firefox, Edge).
* Можливість розгортання на будь-якому сервері з підтримкою Python 3.10+.

3.7 Організація специфікації вимог

Вимоги системи організовано за функціональною ієрархією, згруповано за основними підсистемами: користувацька частина, система відгуків, адмінпанель, механізм надсилання гри. Кожна група має чіткі, уніфіковані вимоги у вигляді специфікацій до вводу, обробки та виводу.