**СПЕЦИФІКАЦІЯ ВИМОГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**A.1 Вступ**

**A.1.1 Огляд продукту**

QuestScout — це веб-платформа для бронювання квест-кімнат, яка дозволяє користувачам переглядати доступні квести, обирати дати та час, здійснювати оплату та керувати своїми бронюваннями. Адміністратори мають інструменти для управління кімнатами, бронюваннями, користувачами та відгуками.

**A.1.2 Мета**  
 Метою проекту є створення зручного, безпечного та ефективного сервісу для організації бронювання квест-кімнат. Платформа повинна спростити процес вибору та оплати для клієнтів, а також надати адміністраторам інструменти для аналітики та управління бізнес-процесами.

**A.1.3 Межі**

* Система не включає функціонал для фізичного контролю доступу до квест-кімнат (наприклад, електронні замки).
* Не передбачено інтеграції зі сторонніми системами бронювання (наприклад, Booking.com).
* Мобільний додаток не розробляється — лише веб-версія.
* Оплата підтримується лише через PayPal (інші платіжні системи, такі як Stripe або Приват24, не інтегровані).

**A.1.4 Означення та абревіатури**

* GraphQL — мова запитів для API, яка дозволяє клієнтам отримувати лише необхідні дані.
* JWT (JSON Web Token) — стандарт для безпечної передачі даних автентифікації.
* OAuth — протокол для авторизації через сторонні сервіси (наприклад, Google).
* Apollo Server — фреймворк для побудови GraphQL-серверів.
* Sequelize — ORM для роботи з базами даних у Node.js.

**A.2 Загальний опис**

**A.2.1 Перспективи продукту**

QuestScout розробляється як сучасна платформа для бронювання квест-кімнат, яка має замінити традиційні методи резервування (телефонні дзвінки, ручне ведення записів). Система автоматизує процес бронювання, спрощує управління для адміністраторів та надає зручний інтерфейс для користувачів.

У майбутньому можливі розширення:

* додаткові платіжні системи (Stripe, Приват24);
* система лояльності (бонусні бали, знижки);
* мобільний додаток для iOS та Android;
* інтеграція з календарями (Google Calendar, Outlook).

**A.2.2 Функції продукту**

Для користувачів:

* реєстрація та автентифікація (через email або Google);
* пошук квестів за фільтрами (жанр, складність, кількість гравців);
* перегляд доступних дат та часу;
* бронювання з оплатою через PayPal;
* перегляд історії бронювань та скасування (з поверненням коштів);
* залишення відгуків та оцінок.

Для адміністраторів:

* додавання, редагування та видалення квестів;
* управління бронюваннями (підтвердження, скасування);
* перегляд аналітики (дохід, популярні квести);
* модерація відгуків та відповіді на них;
* керування користувачами (блокування, надання прав).

**A.2.3 Характеристики користувачів**

Звичайні користувачі:

* клієнти, які бронюють квести для себе або компанії;
* не мають прав на редагування контенту;
* можуть залишати відгуки та керувати своїми бронюваннями.

Адміністратори:

* працівники квест-центру або власники бізнесу;
* мають доступ до панелі управління;
* відповідають за наповнення сайту, обробку платежів та підтримку користувачів.

**A.2.4 Загальні обмеження**

Технічні:

* система працює лише у веб-браузері (немає мобільного додатка);
* підтримка лише PayPal для оплати;
* не передбачено офлайн-режиму (потрібен інтернет).

Бізнес-обмеження:

* не можна бронювати квести менш ніж за 2 години до початку;
* максимальна кількість гравців обмежена характеристиками кімнати;
* повернення коштів лише за певних умов (наприклад, скасування за 24 години).

**A.2.5 Припущення і залежності**

Припущення:

* користувачі мають стабільний інтернет;
* адміністратори знають, як користуватися панеллю управління;
* квест-центр має власний графік роботи, який не змінюється дуже часто.

Залежності:

* робота платежної системи (PayPal);
* доступність Google OAuth для автентифікації;
* стабільність серверів баз даних (MySQL).

**A.3 Конкретні вимоги**

**A.3.1 Вимоги до зовнішніх інтерфейсів**

**A.3.1.1 Інтерфейс користувача**

Система надає сучасний веб-інтерфейс з адаптивним дизайном, що коректно відображається на комп'ютерах, планшетах та мобільних пристроях. Основні елементи:

* головна сторінка з каталогом квестів, фільтрами та пошуком;
* сторінка квесту з детальною інформацією, фотогалереєю, відгуками та календарем доступності;
* особистий кабінет для перегляду історії бронювань та управління обліковим записом;
* адмін-панель з інтуїтивним інтерфейсом для управління контентом;
* модальні вікна для бронювання, оплати та сповіщень.

Інтерфейс реалізований з використанням React та Tailwind CSS, що забезпечує плавну анімацію переходів та швидке реагування на дії користувача.

**A.3.1.2 Апаратний інтерфейс**

Система не вимагає спеціального апаратного забезпечення на стороні клієнта. Мінімальні вимоги:

* процесор з тактовою частотою від 1 ГГц;
* 2 ГБ оперативної пам'яті;
* веб-браузер останньої версії (Chrome, Firefox, Safari);
* роздільна здатність екрана не нижче 320×568 пікселів.

**A.3.1.3 Програмний інтерфейс**

Система використовує наступні API:

* GraphQL API для всіх операцій з даними;
* Google OAuth API для авторизації;
* Google Maps API для відображення локацій;

**A.3.1.4 Комунікаційний протокол**

* HTTPS для всіх зовнішніх з'єднань.
* WebSocket для реального оновлення даних про доступність квестів.
* REST для інтеграції з PayPal (у випадку необхідності).
* OAuth 2.0 для авторизації через Google.

**A.3.1.5 Обмеження пам'яті**

* Клієнтська частина: не більше 100 МБ постійного сховища (LocalStorage, IndexedDB).
* Серверна частина: до 1 ГБ оперативної пам'яті на екземпляр API.
* База даних: до 10 ГБ для початкового розгортання.

**A.3.1.6 Операції**

Бронювання квесту:

* вибір дати/часу;
* вказівка кількості гравців;
* оплата через PayPal;

Скасування бронювання:

* автоматичне повернення коштів;
* оновлення доступних слотів.

Адміністрування:

* додавання нового квесту;
* блокування користувачів;
* модерація відгуків.

**A.3.1.7 Функції продукту**

Система бронювань:

* + календар з відображенням доступності;
  + валідація мінімальної/максимальної кількості гравців;
  + захист від одночасного бронювання.

Платіжна система:

* + безпечне зберігання платежних даних;
  + механізми повернення коштів;
  + історія транзакцій.

Система відгуків:

* + модерація перед публікацією;
  + відповіді адміністрації;
  + рейтинг на основі середньої оцінки.

**A.3.1.8 Припущення і залежності**

Припущення:

* + користувачі мають активний PayPal акаунт для оплати;
  + адміністратори регулярно оновлюють інформацію про квести;
  + графік роботи квест-центру стабільний.

Залежності:

* + доступність Google OAuth сервісів;
  + стабільність хостинг-провайдера для серверної частини;
  + підтримка SSL сертифікатів для безпечного з'єднання.

**A.3.2 Властивості програмного продукту**

Модульність архітектури:

* чітке розділення на фронтенд (React) та бекенд (Node.js + GraphQL);
* незалежність компонентів (бронювання, оплата, відгуки);
* можливість оновлення окремих модулів без впливу на всю систему.

Масштабованість:

* горизонтальне масштабування API-серверів;
* оптимізовані запити до бази даних;
* можливість додавання нових квест-центрів без змін коду.

Інтегрованість:

* інтеграція з PayPal для платежів;
* підтримка соціальної авторизації (Google);
* API для потенційної інтеграції з CRM-системами.

Адаптивність:

* підтримка різних графіків роботи квестів;
* гнучкі налаштування правил бронювання;

**A.3.3 Атрибути програмного продукту**

**A.3.3.1 Надійність**

Система забезпечує стабільну роботу навіть у разі часткових збоїв за рахунок:

* транзакційної цілісності для критичних операцій (бронювання, оплата);
* механізму відновлення після збоїв (retry logic для платежів);
* резервного копіювання даних щогодини зі зберіганням протягом 30 днів;
* валідації вхідних даних на всіх рівнях (клієнт, сервер, БД);
* моніторингу помилок (Sentry для клієнта, Winston для сервера).

**A.3.3.2 Доступність**

Платформа забезпечує:

* крос-браузерну сумісність (Chrome, Firefox, Safari, Edge останніх версій);
* адаптивний дизайн для екранів від 320px до 4K;
* підтримку стандартів доступності WCAG 2.1 AA;
* альтернативні тексти для всіх графічних елементів;
* клавіатурну навігацію для користувачів з обмеженими можливостями;
* режим високої контрастності для людей з вадами зору.

Серверна частина забезпечує uptime 99.5% за рахунок:

* кластерного розгортання;
* автоматичного масштабування;
* географічно розподілених резервних копій.

**A.3.3.3 Безпека**

Захист даних:

* шифрування TLS 1.3 для всіх з'єднань;
* хешування паролів (bcrypt з salt rounds=12);
* ізоляція сесій через JWT з 15-хвилинним TTL;
* PCI DSS Compliance для обробки платежів.

Запобігання атакам:

* Rate limiting (100 запитів/хв з IP);
* захист від CSRF (SameSite cookies);
* санітація вхідних даних (XSS-фільтри);
* регулярні пентести (раз на квартал).

Контроль доступу:

* RBAC (ролі: гість, користувач, модератор, адмін);
* 2FA для адміністраторів;
* аудит логування всіх критичних дій.

**A.3.3.4 Супроводжуваність**

Архітектурні принципи:

* модульна структура (DDD підхід);
* документовані API (OpenAPI 3.0);
* CI/CD пайплайн з автоматичними перевірками.

Інструменти:

* логування в ELK-стеку;
* документація в GitBook;
* візуалізація залежностей (Lerna);
* контейнеризація (Docker + Kubernetes).

**A.3.3.5 Переносимість**

Підтримувані середовища:

* хмарні провайдери: AWS, Azure, GCP;
* БД: MySQL 8.0+, PostgreSQL 14+;
* ОС серверів: Linux (Ubuntu 22.04 LTS);
* Node.js 18+ / React 18+.

Процес міграції:

* скрипти для переносу даних (Sequelize migrations);
* конфігурація через env-змінні;
* Terraform для інфраструктури як коду.

**A.3.3.6 Продуктивність**

Ключові метрики:

* обробка до 100 одночасних бронювань;
* відгук API < 300 мс при 1000 RPS;
* завантаження сторінок < 1.5 сек при TTFB < 400 мс;
* база даних: обробка 10k+ записів квестів.

Оптимізації:

* пакетне завантаження React-компонентів;
* Connection pooling для БД;
* індексовані запити для пошуку;
* CDN для статичних ресурсов.

**A.3.4 Вимоги до бази даних**

Схема даних:

* нормалізація до 3NF для усунення надмірності;
* індексовані поля для пошуку (ID квестів, email користувачів);
* зв'язки між таблицями через foreign keys з каскадними оновленнями.

Типи даних:

* точні часові мітки для бронювань (TIMESTAMP WITH TIME ZONE);
* JSON для зберігання динамічних налаштувань квестів;
* DECIMAL(10,2) для фінансових операцій;
* ENUM для статусів бронювань ('pending', 'confirmed', 'cancelled', 'completed').

Реплікація:

* Master-slave конфігурація для читання/запису;
* щогодинна синхронізація між дата-центрами.

Швидкість запитів:

* <50 мс для CRUD-операцій;
* <200 мс для складних JOIN-запитів.

Обсяги даних:

* до 100,000 бронювань на рік;
* зберігання історії платежів протягом 5 років.

Резервування та відновлення:

* щоденні differential backups;
* Point-in-time recovery з точністю до 1 хвилини;
* тестування відновлення даних щокварталу.

Безпека:

* шифрування "at rest" для PII даних;
* ролевий доступ (admin/read-only);
* аудит змін (trigger-based logging).

**A.3.5 Інші вимоги**

Відповідність стандартам:

* GDPR для захисту персональних даних
* локалізовані умови конфіденційності

Локалізація:

* підтримка UTF-8 для мультимовного контенту
* динамічні часові зони для бронювань

Документація:

* ER-діаграми з описом зв'язків;
* Data dictionary з описом полів;
* приклади типових запитів для розробників.

**А.4 Додаткові матеріали**

**А.4.1 Список використаних джерел**

1. ISO/IEC/IEEE 29148:2018 – Systems and software engineering — Life cycle processes — Requirements engineering. Міжнародний стандарт, що визначає структуру та вимоги до документу SRS, який використовувався як основа для формату цієї специфікації.
2. Google Developers (2023) – Google Maps API Documentation. Офіційна документація щодо інтеграції Google Maps у вебдодатки, яка була використана для реалізації локаційного функціоналу QuestScout.
3. Apollo GraphQL (2023) – GraphQL Best Practices. Рекомендації щодо проектування API, оптимізації запитів та кешування даних, які використовувалися при розробці серверної частини.
4. OWASP (2023) – Application Security Verification Standard (ASVS). Стандарт безпеки вебдодатків, що вплинув на архітектурні рішення щодо захисту даних користувачів (JWT, HTTPS, SQL-ін’єкції).
5. React Documentation (2023) – Hooks, Context API, and Performance Optimization. Офіційні матеріали React, які стали основою для побудови інтерактивного інтерфейсу.
6. Sequelize ORM (2023) – Database Migrations and Model Associations. Документація з роботи з реляційними базами даних, що застосовувалася для проектування схеми БД.
7. GDPR (2018) – General Data Protection Regulation. Європейський регламент, який визначив вимоги до обробки персональних даних (згоди, шифрування, право на видалення).