ДОДАТОК В

Специфікація програмного забезпечення

1 ВСТУП

1.1 Огляд

Даний документ є специфікацією вимог до програмної системи для управління платформою нерухомості, яка забезпечує публікацію, пошук і адміністрування оголошень про оренду та продаж нерухомих об’єктів. Система складається з серверної частини (NestJS), клієнтської веб-частини (React + TypeScript), бази даних (MySQL) та механізмів безпечного доступу до API. Специфікація містить загальний опис, функціональні можливості, характеристики користувачів, зовнішні інтерфейси, обмеження та конкретні вимоги.

1.2 Мета

Метою цієї специфікації є чітке визначення функціональності, структури та вимог до платформи для реалізації повного циклу управління оголошеннями нерухомості, від створення до перегляду й модерації. Це забезпечить зручний, безпечний і масштабований сервіс для власників об’єктів, агентів, адміністраторів і кінцевих користувачів.

1.3 Межі

Система охоплює бекенд-сервіс (NestJS + TypeORM + MySQL), фронтенд-додаток (React + Axios + JWT), зовнішній API-документатор (Swagger) та адміністрування у браузері або мобільному середовищі. Система підтримує багаторольову модель доступу, обробку мультимедійних файлів для оголошень, пошук за фільтрами та інтеграцію зі сторонніми сервісами розсилки пошти.

1.4 Посилання

Розробка здійснена відповідно до стандартів IEEE Std 830-1998 та ДСТУ ISO/IEC 15910:2002. Структура документа відповідає рекомендаціям IEEE для розробки специфікацій вимог до ПЗ.

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

У сучасних умовах ринок нерухомості потребує цифрових рішень для швидкої й безпечної взаємодії користувачів із продавцями, агентами та адміністраторами. Застарілі або розрізнені системи ускладнюють пошук житла, знижують ефективність і рівень обслуговування клієнтів.

Розроблювана платформа об’єднує всі ці процеси у єдиний цифровий сервіс із прозорим управлінням оголошеннями, заявками на перегляд і безпечною авторизацією користувачів. У системі реалізовано розподіл доступу на основі ролей, можливість адміністрування, завантаження фотографій, фільтрації об’єктів і моніторингу їхнього статусу. Архітектура побудована на сучасному стеку технологій (NestJS, React, MySQL), що дозволяє масштабувати систему й розширювати функціональність у майбутньому.

2.2 Функції продукту

Система підтримує наступні функції:

FR-1: реєстрація й вхід користувачів (JWT);

FR-2: створення й редагування профілю користувача;

FR-3: управління нерухомістю (створення, редагування, видалення оголошень);

FR-4: завантаження фотографій об’єктів нерухомості (з валідацією розміру й формату);

FR-5: перегляд, пошук і фільтрація оголошень (тип, локація, ціна тощо);

FR-6: подача запитів на перегляд або консультацію щодо нерухомості;

FR-7: управління користувачами та ролями (адміністратор);

FR-8: модерація оголошень, управління статусами «активне/в архіві»;

FR-9: сповіщення користувачів про статуси заявок (SMTP);

FR-10: генерація Swagger-документації для розробників і тестування API.

2.3 Характеристики користувачів

Користувачами платформи є:

* **Гість** — переглядає публічні оголошення без реєстрації.
* **Зареєстрований користувач** — може подавати запити й зберігати оголошення у вибране.
* **Агент/Власник** — додає й управляє своїми оголошеннями.
* **Адміністратор** — модерує контент, управляє користувачами.  
  Усі користувачі мають базові навички користування веб-сайтами або мобільними застосунками.

2.4 Загальні обмеження

Інтерфейс платформи доступний українською та англійською мовами. Для роботи системи необхідне постійне підключення до Інтернету. Надсилання системних повідомлень здійснюється через зовнішній SMTP-сервіс. Завантажувані зображення об’єктів нерухомості обмежуються розміром до 5 МБ і підтримують лише формати JPG та PNG. Також передбачено дотримання норм законодавства щодо захисту персональних даних користувачів і забезпечення безпечного зберігання платіжної інформації.

2.5 Припущення і залежності

Передбачається, що користувачі матимуть доступ до сучасного браузера або мобільного пристрою для перегляду та управління оголошеннями. Серверна частина розгорнута на хмарному хостингу й забезпечує стабільну цілодобову доступність платформи. Система інтегрується зі стороннім сервісом розсилок для відправлення повідомлень. Підтримується захищений протокол HTTPS, а для управління сесіями й авторизацією використовується JWT. Усі паролі користувачів зберігаються в захищеному вигляді у базі даних із застосуванням сучасних криптографічних алгоритмів.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувачів

Інтерфейс платформи має бути сучасним, зрозумілим і простим у використанні. Користувач повинен легко знаходити необхідний функціонал завдяки логічній навігації й адаптивному дизайну. Веб-інтерфейс розробляється на базі React із підтримкою багатомовності (українська й англійська мови). Для зручного доступу з мобільних пристроїв застосунок є адаптивним, тому окремий мобільний застосунок не потрібен — усі сторінки коректно відображаються на будь-якому розмірі екрану.

3.1.2 Програмний інтерфейс

Серверна частина платформи реалізована на фреймворку NestJS (Node.js), що гарантує масштабованість і стабільність обробки запитів. Комунікація між клієнтським і серверним компонентами здійснюється за допомогою REST API у форматі JSON через захищений протокол HTTPS.

Для управління доступом використовується автентифікація й авторизація на базі JWT (JSON Web Token). Після успішного логіну користувачу видається токен, який додається до заголовків кожного подальшого запиту для підтвердження прав доступу.

Веб-клієнт побудовано на React і здійснює запити до API через Axios. Дані сесій і токени на стороні браузера зберігаються у Local Storage або Session Storage.

Система також підтримує інтеграцію зі стороннім SMTP-сервісом для надсилання повідомлень користувачам і платіжним сервісом PayPal для управління підписками й оплатами.

Функціонал API охоплює всі типові операції платформи нерухомості — управління користувачами й їхніми ролями, роботу з оголошеннями, завантаження й обробку зображень об’єктів нерухомості, перегляд і фільтрацію списку об’єктів. Кожний запит обробляється з урахуванням можливих помилок, при цьому сервер повертає стандартизовані коди статусів HTTP (200, 400, 401, 403, 404, 500) і JSON-відповіді з інформацією про помилки.

3.2 Властивості програмного продукту

Програмна система повинна забезпечувати реалізацію наступного функціоналу:

* реєстрація й авторизація користувачів із розмежуванням ролей (незареєстрований користувач, зареєстрований користувач, агент, адміністратор);
* управління профілем користувача, зокрема редагування персональних даних;
* створення, редагування й видалення оголошень про продаж або оренду нерухомості;
* можливість завантажувати й переглядати фото для кожного об’єкта нерухомості;
* пошук і фільтрація оголошень за різними критеріями (тип об’єкта, ціна, розташування);
* інтеграція з платіжною системою PayPal для оформлення платних послуг або розширених тарифів розміщення оголошень;
* перегляд повного опису кожного об’єкта нерухомості (характеристики, розташування, ціна, фото, контактна інформація агента);
* управління доступами й дозволами користувачів відповідно до їх ролі у системі;
* синхронізація даних між фронтендом, бекендом, базою даних MySQL та зовнішніми сервісами для забезпечення цілісності й актуальності інформації.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Система стабільно працює при типових навантаженнях. Всі винятки обробляються централізовано, а користувач отримує чітке повідомлення про причину помилки.

3.3.2 Доступність

Доступ до платформи забезпечений цілодобово з будь-якого місця. Мобільний застосунок підтримує офлайн-режим із подальшою синхронізацією даних при появі підключення до мережі.

3.3.3 Безпека

Усі паролі зберігаються лише у захищеному (хешованому) вигляді. Доступ до функціоналу розмежовано залежно від ролі користувача. Авторизація реалізована на основі верифікації токенів JWT, що забезпечує безпечний обмін даними.

3.3.4 Супроводжуваність

Архітектура розроблена відповідно до принципів чистої архітектури та розділення відповідальностей між модулями. Логіка, UI, API та робота з базою даних розділені на окремі шари, тому будь-який компонент можна змінювати або тестувати ізольовано від решти.

Веб-застосунок побудовано на React із компонентним підходом для легкого повторного використання UI-елементів.  
Мобільний застосунок побудовано за патерном MVC для чіткого розділення логіки, представлення й управління даними.

У системі реалізовано логування подій і помилок для оперативного моніторингу роботи компонентів.

Використання TypeScript у бекенді дозволяє уникати багатьох типових помилок ще на етапі розробки.

Завдяки модульній побудові API та гнучкій схемі бази даних розширення функціональності не потребує змін у загальній архітектурі системи.