Software Requirements Specification

1.0

14.06.2025

Онлайн-платформа для координації волонтерських ініціатив

Равлюк Кирило Станіславович

Супрун Дарина Андріївна

Шестаков Микола Сейранович

**ІСТОРІЯ ЗМІН**

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Опис | Автор | Коментарі |
| 11.06.2025 | Створено пункти 1.1 - 1.5, 2.1-2.4 | Супрун Дарина Андріївна |  |
| 13.06.2025 | Створено пункти 3.1-3.2 | Равлюк Кирило Станіславович |  |
| 13.06.2025 | Створено пункти 2.5, 3.3-3.5 | Шестаков Микола Сейранович |  |
| 14.06.2025 | Повністю оформлено документ | Равлюк Кирило Станіславович |  |

**ЗАТВЕРДЖЕННЯ ДОКУМЕНТУ**

Наступну специфікацію вимог до програмного забезпечення було прийнято та схвалено:

|  |  |  |  |
| --- | --- | --- | --- |
| Підпис | Друковане ім’я | Назва | Дата |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. ВСТУП

1.1 Огляд продукту

Метою цього продукту є створення єдиної онлайн-платформи для координації волонтерських ініціатив, яка дозволяє об’єднувати волонтерів, ініціаторів та адміністрацію в єдиному цифровому середовищі. Система спрощує пошук, участь і адміністрування волонтерських проєктів, надає інструменти для створення ініціатив, відбору учасників за критеріями, географічної візуалізації активностей, керування навичками та комунікації в межах проєктів.

Платформа дозволяє волонтерам знаходити проєкти за категоріями, локацією, датою та вимогами до навичок, а також переглядати власні участі та координувати доступність. Ініціатори можуть створювати проєкти, додавати категорії, керувати учасниками та призначати ролі в команді. Адміністратори мають доступ до повної модерації користувачів, ініціатив і навичок. Система включає підтримку автентифікації, авторизації, керування профілем, навичками та інтерфейсом завантаження аватарів.

Цільова аудиторія охоплює волонтерів, благодійні організації, громадські ініціативи та координаторів соціальних проєктів, які прагнуть підвищити ефективність комунікації, відбору кадрів та обліку волонтерської діяльності. Продукт стане особливо корисним у середовищі, де інформація про активності зазвичай розпорошена між неструктурованими каналами — соціальними мережами, чатами, особистими контактами. Платформа орієнтована на користувачів із різних регіонів України та за її межами, з англомовним інтерфейсом.

1.2 Мета

Платформа «Всеукраїнський волонтерський хаб» розробляється з метою впорядкування та автоматизації процесів взаємодії між волонтерами, ініціаторами соціальних ініціатив і адміністраторами. Сучасні волонтерські активності часто організовуються неструктуровано — через особисті повідомлення, соцмережі чи імпровізовані списки. Такий підхід ускладнює залучення учасників, контроль прогресу, облік навичок та ефективну комунікацію в командах. Основна мета продукту — створити централізовану цифрову екосистему, яка забезпечує прозору реєстрацію, зручний пошук ініціатив, відбір волонтерів за навичками чи географічним розташуванням, а також можливість взаємодії всередині проєктів.

Платформа дозволяє ініціаторам створювати волонтерські ініціативи із зазначенням опису, дедлайнів, місця проведення та необхідних категорій. Волонтери можуть переглядати активні ініціативи на інтерактивній мапі, фільтрувати їх за тематикою або регіоном, подаватися на участь та слідкувати за статусами. Також реалізовано механізм профілів, керування навичками й участю, можливість зміни ролі користувача, додавання та підтвердження навичок. Очікуваним результатом є зниження фрагментації волонтерського середовища, підвищення ефективності координації та залучення нових учасників до соціальних проєктів.

Переваги платформи включають централізацію всіх волонтерських ініціатив, прозорість даних, автоматизоване управління участю, підтримку різних ролей користувачів, розширюваність архітектури, а також використання сучасних технологій (React, .NET Core, MongoDB, SignalR), які забезпечують надійність, масштабованість і зручність у використанні.

1.3 Межі

Програмний продукт охоплює повноцінну реалізацію серверної частини на основі ASP.NET Core Web API з використанням C#, базу даних MongoDB Atlas та клієнтську частину, створену з використанням React, TypeScript, Zustand, React Query і SCSS. Система реалізує функціонал реєстрації користувачів із розмежуванням ролей (волонтер, ініціатор, адміністратор), створення та керування волонтерськими ініціативами, пошуку проєктів за фільтрами (категорії, локація, дата, ключові слова), приєднання до ініціатив, керування особистими навичками, верифікації ролей і завантаження аватарів. Передбачена підтримка інтерактивної мапи активностей на базі Leaflet, що дозволяє візуалізувати розташування волонтерських подій.

Платформа не включає багатомовність — інтерфейс представлений виключно англійською мовою. У системі відсутня інтеграція зі сторонніми сервісами автентифікації (Google, Facebook), а також механізми розсилки електронних листів або повідомлень, за винятком функції відновлення пароля. На даному етапі розробки не реалізовано внутрішній чат, рейтинг учасників, механізм «банку часу» та «фонду ідей» — ці модулі заплановані як майбутні розширення. Обмеження функціоналу також стосуються взаємодії з проєктами: ініціатори та адміністратори мають змогу керувати лише власними ініціативами, тоді як волонтери можуть приєднуватися, відписуватися та переглядати лише свої участі.

Система не надає підтримки користувачам з обмеженими можливостями, а також не гарантує повноцінну роботу у застарілих браузерах. У межах першого релізу передбачено лише базову перевірку доступності та валідацію на рівні користувацького інтерфейсу. Незважаючи на ці обмеження, архітектура системи дозволяє подальше масштабування функціоналу та інтеграцію додаткових сервісів без істотного рефакторингу основної логіки.

1.4 Посилання

У цій специфікації програмного забезпечення для платформи «Всеукраїнський волонтерський хаб» використовуються нормативні документи, технічні стандарти та офіційна документація щодо інструментів і технологій, що застосовуються у проєкті. Вони слугують основою для формулювання функціональних і нефункціональних вимог, побудови архітектури системи, реалізації безпечної автентифікації, організації REST API та побудови клієнтської частини на сучасному стеку.

1. ISO/IEC 25010:2011 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE): System and software quality models. Стандарт визначає модель якості програмного забезпечення. Доступно на сайті ISO (<https://www.iso.org/standard/35733.html>).
2. RFC 7519 – JSON Web Token (JWT). Опис формату передачі токенів для автентифікації користувачів. Доступно на IETF   
   (<https://datatracker.ietf.org/doc/html/rfc7519>).
3. RFC 6455 – The WebSocket Protocol. Специфікація протоколу, який використовується для реалізації двосторонньої комунікації через SignalR. Доступно на IETF (<https://datatracker.ietf.org/doc/html/rfc6455>).
4. OWASP Application Security Verification Standard (ASVS) – Рекомендації щодо захисту вебдодатків. Доступно на сайті OWASP (<https://owasp.org/www-project-application-security-verification-standard/>).
5. Документація ASP.NET Core – офіційна документація для розробки RESTful API та бекенду. Детальні матеріали доступні на сайті Microsoft Learn (<https://learn.microsoft.com/en-us/aspnet/core/>).
6. Документація MongoDB – офіційна база знань щодо використання нереляційної бази даних MongoDB. Докладну інформацію можна знайти на офіційному сайті (<https://www.mongodb.com/docs/>).
7. Документація React – офіційна документація бібліотеки React для побудови клієнтської. Ознайомитись із матеріалами можна на сайті React (<https://react.dev/>).
8. Документація React Query – бібліотека для кешування та синхронізації запитів. Актуальна документація доступна на сайті TanStack (<https://tanstack.com/query>).
9. Документація Zustand – бібліотека керування глобальним станом у React. З повною документацією можна ознайомитись на офіційному сайті (<https://docs.pmnd.rs/zustand>).
10. Документація Leaflet – бібліотека для побудови інтерактивних карт. Офіційна документація доступна на сайті проєкту (<https://leafletjs.com/>).
11. Методичні рекомендації щодо складання СРС (Специфікації вимог до програмного забезпечення) – відповідно до внутрішніх академічних стандартів навчального закладу, що регламентують структуру документа.

Зазначені джерела використовуються при розробці, тестуванні та документуванні системи для забезпечення її якості, безпеки, функціональності та відповідності сучасним інженерним практикам.

5. Означення та абревіатури

У цьому розділі наведено ключові терміни та скорочення, які використовуються в специфікації програмного забезпечення для платформи «Всеукраїнський волонтерський хаб». Вони покликані забезпечити однозначне розуміння концепцій, функціональних складових і технологічних рішень, що застосовуються під час розробки системи.

Означення:

* Волонтер – користувач системи, який приєднується до ініціатив і виконує завдання на добровільній основі.
* Ініціатор – користувач, який створює волонтерські проєкти, керує ними та запрошує волонтерів.
* Адміністратор – користувач із розширеними правами, який має доступ до модерації ініціатив, керування ролями, навичками, користувачами та категоріями.
* Ініціатива / проєкт – волонтерський захід або подія, створена ініціатором і доступна для участі волонтерів.
* Навичка – визначене в системі уміння чи компетенція, яку може мати волонтер (наприклад, водіння, медична допомога).
* Категорія – тематичний напрямок проєкту (екологія, гуманітарна допомога, логістика тощо).
* Участь – взаємозв’язок між користувачем та проєктом, що вказує на його задіяність у певній ініціативі.
* Інтерактивна мапа – візуальний компонент, що відображає місце розташування волонтерських подій.
* Банк часу – майбутній модуль, що дозволяє обліковувати години волонтерської діяльності та обмінювати їх на послуги або бонуси.
* Фонд ідей – запланований функціонал, де користувачі можуть подавати нові волонтерські ініціативи для обговорення та реалізації.

Абревіатури:

* SRS (Software Requirements Specification) – Специфікація вимог до програмного забезпечення.
* UI (User Interface) – Користувацький інтерфейс.
* API (Application Programming Interface) – Програмний інтерфейс взаємодії додатків.
* DB (Database) – База даних.
* JWT (JSON Web Token) – Формат токенів, що використовується для аутентифікації користувачів.
* HTTPS (HyperText Transfer Protocol Secure) – Протокол захищеної передачі даних в інтернеті.
* REST (Representational State Transfer) – Архітектурний стиль побудови веб-сервісів, орієнтований на ресурси.
* SignalR – Бібліотека ASP.NET для реалізації реального часу через WebSocket або інші протоколи.
* React – JavaScript-бібліотека для створення інтерфейсів користувача.
* React Query – Бібліотека для кешування та синхронізації асинхронних запитів у React-додатках.
* Zustand – Легка бібліотека для керування станом у React.
* Leaflet – Бібліотека JavaScript для побудови інтерактивних карт.
* EF Core (Entity Framework Core) – ORM-бібліотека для взаємодії з базами даних у .NET-додатках.
* OAuth – Протокол авторизації, що дозволяє надавати стороннім застосункам доступ до ресурсів користувача без розкриття пароля (не використовується у поточній реалізації, але згадується як потенційний).

2. ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Платформа «Всеукраїнський волонтерський хаб» створена для вирішення актуальної проблеми децентралізованості волонтерської діяльності в Україні та за її межами. У сучасних умовах волонтерські ініціативи часто виникають стихійно, поширюються через особисті контакти або соціальні мережі, що призводить до дублювання зусиль, втрати інформації та ускладненого доступу для нових учасників. Запропонований вебзастосунок покликаний змінити цю ситуацію шляхом формування єдиного цифрового простору, в якому усі сторони — волонтери, ініціатори, організації та адміністратори — можуть ефективно взаємодіяти, координувати дії та досягати спільної соціальної мети.

Передбачена архітектура системи забезпечує високий рівень масштабованості, що дозволяє використовувати її як на рівні локальних громад, так і у міжрегіональному або національному масштабі. Хаб може застосовуватись громадськими організаціями, благодійними фондами, навчальними закладами, муніципалітетами та незалежними ініціативними групами як базова цифрова інфраструктура для запуску або підтримки волонтерських кампаній. У перспективі система може бути інтегрована з іншими інформаційними платформами або державними системами, що працюють у сфері соціальної підтримки та координації добровольчих зусиль.

Функціональне ядро платформи зосереджене на трьох основних ролях користувачів: волонтер, ініціатор та адміністратор. Волонтери отримують змогу переглядати актуальні ініціативи, фільтрувати їх за категоріями, місцем або часом, відгукуватися на проєкти, відстежувати статуси та керувати власними навичками. Ініціатори, у свою чергу, мають доступ до створення проєктів, управління їхнім життєвим циклом, перегляду зацікавлених учасників і комунікації з ними. Адміністративний рівень забезпечує модерацію контенту, контроль якості, підтвердження навичок, зміну ролей користувачів і керування базовими довідниками системи.

Однією з ключових перспектив продукту є його гнучкість і готовність до розширення. Архітектура передбачає впровадження додаткових модулів — зокрема, «банку часу», який дозволяє обліковувати і винагороджувати волонтерську активність, «фонду ідей» — місця, де користувачі можуть подавати пропозиції нових ініціатив, а також системи повідомлень і рейтингу учасників. Такі функції дозволяють платформі перетворитися на не просто інструмент організації, а повноцінну спільноту з розвиненим соціальним і мотиваційним шаром.

Платформа є повністю веборієнтованою, з клієнтською частиною на React та серверною логікою на ASP.NET Core Web API. Використання MongoDB як гнучкої бази даних дозволяє легко масштабувати проєкт, а інтеграція з Leaflet забезпечує зручну геовізуалізацію активностей. Завдяки відмові від залежності від сторонніх сервісів авторизації та розсилок, платформа зберігає контрольовану екосистему, орієнтовану на безпеку, конфіденційність та стабільність.

Таким чином, «Всеукраїнський волонтерський хаб» — це не просто платформа для розміщення проєктів, а інструмент стратегічного рівня, що дозволяє розвивати соціальну активність, формувати довготривалі волонтерські зв’язки та сприяти формуванню сильної, відповідальної громадянської спільноти, незалежно від регіону, сфери допомоги чи досвіду користувача.

2.2 Функції продукту

Розроблювана платформа «Всеукраїнський волонтерський хаб» має на меті забезпечити ефективну цифрову взаємодію між волонтерами, ініціаторами та адміністраторами шляхом централізації волонтерських активностей у межах єдиного вебінтерфейсу. Основна функціональність системи охоплює повний життєвий цикл волонтерської ініціативи — від її створення, публікації та поширення, до набору команди, участі волонтерів і подальшого керування. Користувачі можуть створювати облікові записи, редагувати профілі, додавати особисту інформацію, включаючи біо, контактні дані, координати та аватар. Під час реєстрації або згодом кожен користувач обирає роль — волонтера, ініціатора чи адміністратора — що визначає обсяг доступних функцій.

Для волонтерів реалізовано функції пошуку та перегляду ініціатив за містом, датою, категорією або ключовими словами, а також можливість фільтрації проєктів за відстанню та навичками. Волонтери можуть приєднуватися до ініціатив, переглядати свої поточні та минулі участі, залишати ініціативу, а також керувати власним набором навичок і рівнем їх опанування. Система дозволяє їм бачити підтвердження навичок від ініціаторів або адміністраторів. Ініціатори, своєю чергою, мають змогу створювати нові проєкти, вказуючи їхню назву, опис, дедлайни, місце проведення та відповідні категорії. Вони можуть бачити список волонтерів, що долучилися, призначати ролі в межах команди, змінювати статуси участі або видаляти волонтерів з проєкту.

Функціонал адміністратора включає повну модерацію контенту: підтвердження або редагування ініціатив, керування користувачами, зміна ролей, редагування або видалення навичок і категорій. Усі користувачі мають доступ до інтерактивної мапи, яка відображає волонтерські події у їхньому регіоні, з можливістю фільтрації та перегляду деталей. Крім того, система підтримує механізми автентифікації та авторизації з використанням JWT-токенів, зберігання сесій через cookie, а також відновлення пароля за допомогою електронної пошти.

У платформу вбудована система участей, яка дозволяє фіксувати, переглядати та фільтрувати зв’язки між користувачами та проєктами. Це формує основу для майбутніх розширень — зокрема, системи «банку часу», рейтингу волонтерів, внутрішнього чату та модуля ідей. Таким чином, хаб забезпечує не лише базову функціональність для участі в соціально значущих активностях, але й виступає як цифрова основа для формування волонтерської екосистеми з високим рівнем координації, прозорості та залучення.

2.3 Характеристики користувачів

Платформа «Всеукраїнський волонтерський хаб» передбачає використання трьох основних ролей користувачів, кожна з яких має власний набір можливостей та рівень доступу до функціоналу системи. Всі користувачі проходять процес реєстрації та автентифікації, після чого отримують одну з ролей — волонтер, ініціатор або адміністратор. Таке чітке розмежування дозволяє оптимізувати роботу системи, розподілити відповідальність між учасниками та забезпечити прозоре управління волонтерськими ініціативами.

Волонтери є найчисленнішою категорією користувачів. Це люди різного віку, досвіду та соціального статусу, які бажають брати участь у суспільно корисних ініціативах. Від користувачів цього типу не очікується спеціальної технічної підготовки — платформа має інтуїтивно зрозумілий інтерфейс, що дозволяє легко знаходити відповідні проєкти, фільтрувати їх за місцем проведення, тематикою або термінами, а також подаватися на участь у декілька кліків. Волонтери можуть керувати власним профілем, оновлювати особисту інформацію, додавати навички та відмічати свою доступність по днях. За потреби вони можуть вийти з проєкту, залишити активну участь або змінити рівень своїх навичок. Деякі функції, як-от перегляд історії участей або підтвердження навичок, будуть доступні з боку адміністрації або ініціаторів.

Ініціатори — це зареєстровані користувачі, які мають змогу створювати власні волонтерські ініціативи. Як правило, це представники громадських організацій, благодійних фондів, місцевих об’єднань або просто активні громадяни, які мають ідею та потребують допомоги у її реалізації. Ініціатори повинні володіти базовими навичками цифрової грамотності, вміти працювати з формами, картами, фільтрами та навігацією в межах інтерфейсу платформи. Вони мають змогу керувати проєктами: додавати опис, місце, час проведення, тематичні категорії, переглядати список залучених волонтерів, призначати ролі в команді та підтверджувати волонтерські навички. Крім того, ініціатори можуть модифікувати створені ними проєкти або видаляти їх у разі потреби. У перспективі для цієї категорії передбачається додатковий функціонал аналітики, звітності та інструментів комунікації.

Адміністратори — це користувачі з повним доступом до внутрішньої інфраструктури платформи. Вони виконують роль модераторів, технічних керівників і контролюючих осіб. До їхніх обов’язків входить перевірка валідності та актуальності контенту, редагування або видалення ініціатив, зміна ролей користувачів, підтвердження або скасування навичок, керування категоріями, а також повне адміністрування користувацьких облікових записів. Адміністратор повинен мати впевнені навички користування цифровими системами, розуміти логіку REST API, знати базові принципи безпеки та верифікації, а також вміти працювати з системами обліку та управління даними.

Важливо зазначити, що платформа не передбачає підтримку користувачів із обмеженими можливостями, а також не оптимізована для мобільних пристроїв. Це означає, що всі категорії користувачів повинні мати доступ до персонального комп’ютера або ноутбука з сучасним браузером. З іншого боку, простота інтерфейсу, чітке структурування ролей і прозора логіка користування роблять платформу доступною навіть для новачків, які не мають досвіду участі в подібних цифрових системах. Це дозволяє залучати широку аудиторію — від студентів і пенсіонерів до професійних волонтерів і керівників проєктів. Завдяки цьому хаб створює гнучке, доступне та масштабоване середовище для розвитку волонтерського руху в цифрову епоху.

2.4 Загальні обмеження

Розробка вебплатформи «Всеукраїнський волонтерський хаб» супроводжується низкою загальних обмежень, які впливають на її функціональність, архітектурні рішення та користувацький досвід. Одним із ключових обмежень є відсутність мобільної версії застосунку: система орієнтована виключно на використання в десктопному середовищі через сучасні браузери. Це означає, що доступ до основних функцій можливий лише з комп’ютерів або ноутбуків, що мають достатній рівень продуктивності, стабільне інтернет-з’єднання та підтримку JavaScript. Платформа не підтримує адаптивний інтерфейс або окремі мобільні додатки, що унеможливлює повноцінну роботу на смартфонах і планшетах.

Ще одним обмеженням є одноосібна мовна підтримка — інтерфейс додатка повністю англомовний, а багатомовність не передбачена в поточному релізі. Це звужує аудиторію до користувачів, які володіють англійською мовою, і може створювати бар’єри для частини населення, особливо серед людей похилого віку або користувачів без знання іноземних мов. Також платформа не враховує потреби користувачів з обмеженими можливостями — не реалізовано механізмів доступності (наприклад, сумісність із екранними читачами, контрастний режим, навігація клавіатурою).

На рівні архітектури система має обмеження щодо інтеграції із зовнішніми сервісами. З міркувань безпеки, конфіденційності та контролю платформа не використовує сторонні механізми автентифікації (такі як Google OAuth або Facebook Login), а також не передбачає email-розсилок, окрім функції відновлення пароля. Це звільняє систему від зайвих залежностей, але обмежує її інтерактивні можливості, зокрема — відсутність нотифікацій, сповіщень про активність та запрошень у проєкти.

На рівні зберігання даних платформа використовує MongoDB Atlas на безплатному тарифному плані. Це накладає суттєві технічні обмеження: зокрема, ліміт у 512 МБ доступного обсягу зберігання, обмеження до 100 одночасних з’єднань, а також обмежена кількість запитів на секунду (не більше 100–200 OPS у пікових навантаженнях). Крім того, на безкоштовному рівні не доступні функції горизонтального масштабування, шардінгу, повноцінної аналітики продуктивності та резервного копіювання з гарантованим SLA. Це означає, що при збільшенні кількості користувачів або складності запитів можуть виникати затримки, збої доступу або перевищення ресурсних лімітів, що призведе до часткової або повної недоступності частин функціоналу. Подібні ризики особливо критичні при проведенні масштабних волонтерських кампаній або одночасному використанні платформи великою кількістю учасників.

Картографічна частина системи реалізована за допомогою бібліотеки Leaflet, яка є відкритою і не потребує ліцензування, проте має обмеження, пов’язані з використанням публічних картографічних тайлів. У більшості випадків для рендерингу карти застосовується безкоштовний провайдер OpenStreetMap, який має обмеження на кількість запитів з одного IP або домену — зазвичай до 1–2 тис. запитів на добу. Перевищення цього ліміту може призвести до блокування тайлів і недоступності карти. Leaflet також не підтримує офлайн-режим, кешування геоданих або захищені комерційні шари без додаткових плагінів і інтеграцій. Це означає, що при слабкому інтернет-з'єднанні або великій кількості користувачів мапа може завантажуватись повільно або частково.

Платформа не має мобільної версії та не оптимізована для пристроїв із невеликим екраном. Доступ до функціоналу забезпечується лише через десктопні браузери, які підтримують сучасні вебтехнології — зокрема JavaScript, CSS Grid, Flexbox, Web API. Користувачі мобільних пристроїв можуть зіткнутися з некоректним відображенням інтерфейсу або недоступністю частини функціоналу, особливо при взаємодії з мапою або формами.

Також варто враховувати обмеження, пов’язані з відсутністю реалізованих систем повідомлень, рейтингу волонтерів, «банку часу» чи «фонду ідей» — ці модулі передбачені в майбутніх релізах, проте на даному етапі не впроваджені. Це впливає на можливості зворотного зв’язку, підвищення мотивації учасників і формування довготривалих спільнот на базі платформи.

З погляду безпеки, система покладається на автентифікацію через JWT-токени, cookie-сесію та контроль доступу через ролі, проте не реалізовано двофакторну автентифікацію чи централізований механізм відстеження активності. Це створює потенційні виклики у майбутньому, коли кількість користувачів і обсяг чутливої інформації зростатимуть.

Таким чином, «Всеукраїнський волонтерський хаб» є гнучкою, але свідомо обмеженою у деяких аспектах системою, що фокусується на основному функціоналі та закладає фундамент для поступового розширення. Платформа розробляється із пріоритетом на простоту, доступність для широкого кола користувачів і надійність, однак з чітко визначеними рамками, які обумовлені як технічними, так і концептуальними рішеннями на поточному етапі розвитку.

2.5 Припущення й залежності

Розробка програмного забезпечення «Всеукраїнський волонтерський хаб» ґрунтується на низці технічних та організаційних припущень, які визначають обмеження і контекст використання системи. Вони охоплюють як властивості середовища виконання, так і стабільність сторонніх сервісів, до яких система має критичну залежність.

Передбачається, що доступ до функціоналу платформи здійснюється з використанням сучасних веббраузерів, які підтримують актуальні технології відображення інтерфейсу. Платформа реалізована з адаптивним дизайном, що забезпечує її коректну роботу як на десктопах, так і на мобільних пристроях. Систему не потрібно встановлювати на пристрій — вона є вебзастосунком. Вважається, що користувачі працюють в операційних системах, сумісних із сучасними браузерами, та мають постійне інтернет-з’єднання для стабільної взаємодії з інтерфейсом і динамічними компонентами платформи.

Для географічної візуалізації волонтерських ініціатив у системі використовується бібліотека Leaflet із використанням відкритих даних сервісу OpenStreetMap. Припускається, що безкоштовні тайл-сервери OpenStreetMap залишатимуться доступними для невеликого або середнього обсягу запитів. У разі перевищення цього ліміту можливе тимчасове обмеження доступу до картографічного шару, що може вплинути на зручність навігації. На етапі поточного релізу не передбачається використання платних або самостійно хостованих картографічних сервісів.

Уся інформація про користувачів, проєкти, участі та навички зберігається у хмарній базі даних MongoDB Atlas, яка використовується на безкоштовному тарифному плані. Це означає, що обсяг зберігання, кількість з’єднань та швидкість обробки запитів є обмеженими. Передбачається, що рівень навантаження не перевищуватиме можливостей тарифу, а резервне копіювання, SLA та горизонтальне масштабування не використовуються на поточному етапі розробки.

Механізм автентифікації базується на використанні JWT-токенів, що зберігаються у cookie-файлах браузера. Сторонні сервіси авторизації не залучаються. Реєстрація користувачів виконується через email і пароль, а захист облікових записів покладається на відповідальність самих користувачів. Підтримка двофакторної автентифікації не реалізована і не передбачена у межах поточної версії проекту.

Система має єдину реалізовану email-функцію — відновлення пароля. Надсилання інших типів повідомлень, сповіщення про участь у проєкті, зміна ролі, підтвердження навички тощо, не передбачено. Вважається, що користувачі регулярно перевіряють свою пошту, а стабільність доставки листів забезпечується обраним SMTP-сервером.

Координація між користувачами, зокрема призначення ролей, підтвердження навичок або керування участю у ініціативах, здійснюється вручну через інтерфейс платформи. У системі не передбачено реалізації внутрішніх чатів, системи повідомлень у реальному часі чи автоматичної модерації дій. Передбачається, що всі дії виконуються у відповідності до ролі користувача без необхідності додаткових засобів комунікації.

Архітектура платформи дозволяє розширення функціональності без суттєвої перебудови основного програмного коду. У майбутньому передбачається впровадження додаткових модулів, зокрема «банку часу», «фонду ідей», системи оцінювання волонтерів та функціоналу обміну повідомленнями. Їх реалізація не є критично необхідною для базового використання, однак розглядається як перспективний напрямок подальшого розвитку платформи.

3. КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

У цьому розділі наведено конкретні вимоги до зовнішніх інтерфейсів системи, зокрема інтерфейсу користувача, інтерфейсів взаємодії з апаратним забезпеченням, програмними компонентами та іншими зовнішніми системами, що забезпечують коректну інтеграцію та роботу програмної системи.

3.1.1 Інтерфейс користувача

Інтерфейс користувача вебдодатку розроблено з урахуванням принципів інтуїтивної зрозумілості та забезпечення якісного користувацького досвіду. Дизайн оформлено в світлій тематиці, реалізовано адаптивність для коректного відображення на різних типах пристроїв. Основними елементами інтерфейсу є навігаційне меню, яке забезпечує швидкий доступ до функціональних сторінок та адаптується відповідно до ролі користувача.

Для волонтерів передбачено доступ до функціоналу запису до ініціатив, пошуку та перегляду ініціатив (у тому числі через інтерактивну мапу), редагування профілю, додавання навичок та перегляду активної участі. Організатори мають змогу створювати, редагувати та видаляти ініціативи, керувати користувачами в межах проєктів, а також призначати ролі. Адміністратори отримують розширені права на керування категоріями, навичками, історією ініціатив та всіма користувачами системи. Для всіх ролей доступні сторінки авторизації, реєстрації, відновлення пароля через електронну пошту, перегляду профілю, ініціатив та інтерактивної мапи.

3.1.2 Апаратний інтерфейс

Вебдодаток розрахований на використання у сучасних браузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge та Safari, із забезпеченням повної підтримки HTML5, CSS3 та JavaScript ES6. Для стабільної роботи рекомендується використовувати пристрої з процесором не нижче Intel Core i3 або аналогічним, щонайменше 4 ГБ оперативної пам’яті та стабільне підключення до Інтернету зі швидкістю не менше 5 Мбіт/с. Додаток не вимагає спеціалізованого апаратного забезпечення та може функціонувати як на настільних комп’ютерах і ноутбуках, так і на мобільних пристроях із підтримкою адаптивного інтерфейсу.

3.1.3 Програмний інтерфейс

Програмна система побудована за клієнт-серверною архітектурою, де взаємодія між частинами здійснюється за допомогою REST API. Клієнтська частина, реалізована з використанням React, надсилає HTTPS-запити до серверної частини, розробленої на базі ASP.NET, яка обробляє ці запити, виконує бізнес-логіку та повертає структуровані дані у форматі JSON. Для збереження та отримання даних про користувачів, ініціативи та інші сутності система використовує нереляційну базу даних Atlas MongoDB. Передача та зберігання користувацьких аватарів здійснюється шляхом інтеграції з зовнішнім хмарним сервісом Azure Blob Storage, який взаємодіє безпосередньо з серверною частиною.

3.1.4 Комунікаційний протокол

Для забезпечення безпечного та надійного обміну даними між клієнтською та серверною частинами системи використовується протокол HTTPS, який забезпечує шифрування інформації на рівні транспортного шару. Використання HTTPS гарантує конфіденційність, цілісність та автентичність передаваних даних, що є особливо важливим при роботі з персональною інформацією користувачів.

3.1.5 Обмеження пам’яті

Клієнтська частина вебдодатку працює у веббраузері і використовує оперативну пам’ять пристрою користувача, рекомендований мінімум – 4 ГБ  для стабільної роботи. Серверна частина, що взаємодіє з базою даних Atlas MongoDB та хмарним сховищем Azure Blob Storage, у межах мінімального тарифного плану має обмеження: для Atlas MongoDB – обсяг сховища до 512 МБ і shared кластер з обмеженою оперативною пам’яттю (до 2 ГБ), а для Azure Blob Storage – базовий тариф передбачає до 5 ТБ сховища з обмеженою пропускною здатністю.

3.1.6 Операції

Користувачі системи можуть виконувати комплекс основних операцій над даними відповідно до своїх ролей. Волонтери мають можливість створювати записи про свою участь у ініціативах, здійснювати пошук та фільтрацію ініціатив, переглядати їх через стандартний список або інтерактивну мапу, оновлювати свій профіль та додавати навички. Організатори можуть створювати, редагувати та видаляти ініціативи, а також керувати користувачами проєкту, зокрема призначати або змінювати ролі. Адміністратори виконують всі CRUD-операції над категоріями навичок, мають повний доступ до керування користувачами, включно зі зміною ролей та видаленням, а також переглядом історії ініціатив. Для всіх користувачів доступні операції реєстрації, авторизації, відновлення пароля, перегляду та редагування профілю, а також перегляду ініціатив і інтерактивної мапи.

3.1.7 Функції продукту

Розроблювана платформа «Всеукраїнський волонтерський хаб» покликана забезпечити централізовану цифрову взаємодію між волонтерами, ініціаторами та адміністраторами через єдиний вебінтерфейс. Основні функції системи охоплюють повний життєвий цикл волонтерської ініціативи — від створення та розміщення проєктів до набору команди, участі волонтерів і подальшого управління.

* Реєстрація та керування обліковим записом (пріоритет високий): створення профілю, редагування особистих даних, кастомізація профілю з можливістю додавання особистої інформації, біографії, контактів, координат, завантаження аватара.
* Пошук і фільтрація ініціатив (пріоритет високий): перегляд проєктів за містом, датою, категорією, ключовими словами; фільтрація за відстанню.
* Управління участю волонтерів (пріоритет високий): приєднання та вихід з ініціатив, перегляд поточних участей, фільтрація ініціатив за категоріями, кінцевим строком та пошук за назвою ініціативи.
* Керування проєктами ініціаторами (пріоритет середній): створення, редагування, видалення проєктів, призначення ролей учасникам, зміна статусів волонтерів.
* Адміністрування платформи (пріоритет середній): управління користувачами, зміна ролей, редагування категорій та навичок.
* Інтерактивна мапа (пріоритет високий): візуалізація волонтерських подій із фільтрацією та переглядом детальної інформації.
* Автентифікація та авторизація (пріоритет високий): безпечний вхід через JWT-токени, збереження сесій, відновлення пароля через електронну пошту.

3.1.8 Припущення й залежності

Для коректного та стабільного функціонування системи необхідно враховувати низку ключових припущень та зовнішніх залежностей, які впливають на роботу інтерфейсів та взаємодію між компонентами програмного продукту.

* Наявність стабільного підключення користувача до інтернету з достатньою пропускною здатністю для коректної роботи вебдодатку та обміну даними з сервером.
* Використання сучасних веббраузерів із підтримкою стандартів HTML5, CSS3, JavaScript, а також забезпечення сумісності з основними браузерами (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).
* Коректна робота серверної частини на базі ASP.NET та MongoDB, забезпечення доступності хмарних сервісів, зокрема Azure Blob Storage для зберігання та передачі аватарів користувачів.
* Наявність належних прав доступу й авторизації для користувачів відповідно до їхніх ролей (волонтер, ініціатор, адміністратор).
* Відповідність тарифних планів хмарних сервісів вимогам щодо обсягів оперативної та постійної пам’яті, які потрібні для стабільної роботи системи.
* Сумісність між клієнтською частиною на React та серверними API через REST-протоколи з використанням HTTPS для безпечного обміну даними.

3.2 Властивості програмного продукту

3.2.1 Реєстрація

3.2.1.1 Загальна характеристика функції

Функція реєстрації забезпечує створення нового облікового запису користувача в системі. Вона є початковою точкою входу до платформи та передбачає збереження мінімального набору даних для подальшої авторизації та ідентифікації користувача. Реєстрація доступна для всіх охочих долучитися до волонтерської платформи та не вимагає попереднього підтвердження ролі.

3.2.1.2 Вхідні дані

Під час реєстрації користувач повинен надати електронну пошту, повне ім’я та пароль. Усі поля є обов’язковими до заповнення. Пароль повинен відповідати мінімальним вимогам щодо складності, а електронна пошта — бути у правильному форматі. Перед надсиланням даних до сервера перевіряється їх валідність на клієнтському рівні.

3.2.1.3 Вихідні дані

У результаті успішної обробки запиту користувач отримує повідомлення про успішну реєстрацію. На клієнтському рівні відображається відповідне підтвердження, після чого користувач має змогу перейти до форми входу в систему для авторизації. На сервері створюється новий обліковий запис, який у подальшому може бути доповнений іншими персональними або службовими даними, зокрема призначеною роллю. За замовчуванням новоствореному обліковому запису автоматично присвоюється роль волонтера, що надає базовий рівень доступу до функціоналу системи.

3.2.1.4 Обробка помилок

У випадку, якщо електронна адреса вже використовується, користувач отримує відповідне повідомлення про помилку. Якщо будь-яке з полів не заповнене або не відповідає вимогам, система повідомляє про необхідність коригування. У разі внутрішньої помилки сервера (наприклад, відсутність з’єднання з базою даних), користувачеві виводиться загальне повідомлення про збій із рекомендацією повторити спробу пізніше.

3.2.2 Вхід до системи

3.2.2.1 Загальна характеристика функції

Функція входу забезпечує авторизацію зареєстрованого користувача на платформі та відкриття доступу до персоналізованого функціоналу відповідно до його ролі. Вхід здійснюється через введення облікових даних, після чого відбувається перевірка на сервері та видача токена доступу. Цей токен зберігається у браузері у вигляді cookie для забезпечення подальшої аутентифікації під час користування сервісом, включаючи оновлення сторінки, навігацію та виконання захищених запитів.

3.2.2.2 Вхідні дані

Користувач вводить електронну адресу та пароль, які були вказані під час реєстрації облікового запису. Обидва поля є обов’язковими для заповнення з метою забезпечення безпеки доступу. На клієнтському рівні виконується базова перевірка на заповненість полів, а також правильність формату введеної електронної пошти. У разі виявлення помилок користувач отримує відповідне повідомлення з підказкою для коректного введення даних.

3.2.2.3 Вихідні дані

У разі успішної авторизації система повертає токен доступу, який зберігається на клієнті. Користувач отримує доступ до функціоналу, що відповідає його ролі.

3.2.2.4 Обробка помилок

Якщо облікові дані вказані неправильно, система повідомляє користувача про недійсну комбінацію логіна й пароля. Якщо акаунт не існує або видалений, виводиться відповідне попередження. У випадку проблем з підключенням до сервера або помилок бази даних користувач бачить повідомлення про технічну несправність і пропозицію спробувати знову пізніше.

3.2.3 Відновлення пароля

3.2.3.1 Загальна характеристика функції

Функція відновлення пароля дозволяє користувачам, які втратили доступ до свого облікового запису, відновити його шляхом отримання листа на електронну пошту. Процес складається з двох етапів: спочатку користувач ініціює запит, вказавши свою електронну адресу, після чого на неї надсилається лист із посиланням для зміни пароля. Перехід за цим посиланням відкриває спеціальну форму, де користувач вводить новий пароль і підтверджує його.

3.2.3.2 Вхідні дані

На першому етапі користувач вводить електронну адресу, прив’язану до відповідного облікового запису. На другому етапі, після переходу за посиланням із листа (яке містить токен підтвердження безпеки), користувач вводить новий пароль та його підтвердження. Усі поля є обов’язковими для заповнення, а паролі повинні відповідати встановленим вимогам до складності, мінімальної довжини та збігатися між собою для успішного завершення процедури скидання.

3.2.3.3 Вихідні дані

У результаті успішної зміни пароля користувач отримує підтвердження про оновлення, після чого може увійти до системи з новими обліковими даними. Токен з посилання автоматично деактивується після використання або через вичерпання терміну дії.

3.2.3.4 Обробка помилок

Якщо вказана електронна адреса не зареєстрована в системі, користувач отримає повідомлення про це без уточнення деталей для запобігання витоку інформації. Якщо токен недійсний, застарілий або вже використаний, система повідомляє про помилку й пропонує повторити процедуру відновлення. У разі технічних збоїв, таких як недоступність поштового сервера або бази даних, відображається загальне повідомлення з порадою спробувати пізніше.

3.2.4 Редагування профілю

3.2.4.1 Загальна характеристика функції

Функція редагування профілю дозволяє користувачеві оновлювати свої персональні дані в обліковому записі, що забезпечує актуальність та персоналізацію інформації на платформі. Користувач може змінити ім’я, дату народження, місцезнаходження, контактні дані, а також завантажити новий аватар. Ця можливість доступна лише після автентифікації, що гарантує захист особистої інформації та збереження цілісності даних. Зміни зберігаються у базі даних і миттєво оновлюють профіль користувача.

3.2.4.2 Вхідні дані

Користувач вводить або редагує такі дані: повне ім’я, дату народження, місце проживання (місто або координати), номер телефону чи інші контактні дані, а також має можливість завантажити новий аватар у форматі зображення (JPEG або PNG). Усі дані заповнюються вручну або завантажуються через інтерфейс, після чого проходять базову перевірку на валідність — наприклад, правильність формату дати чи електронної адреси.

3.2.4.3 Вихідні дані

Після успішного збереження змін користувач отримує підтвердження у вигляді повідомлення про оновлення профілю. На клієнтській стороні миттєво оновлюється відображення змінених даних, зокрема новий аватар, оновлене ім’я або інші контактні відомості. У базі даних відбувається оновлення відповідних полів облікового запису, а файл з аватаром зберігається у відповідному blob-сховищі із прив’язкою до профілю користувача.

3.2.4.4 Обробка помилок

У разі виникнення помилок, пов’язаних із валідацією (наприклад, некоректний формат телефону або порожнє поле імені), користувач отримує відповідне повідомлення з поясненням, що саме слід виправити. Якщо виникає помилка при завантаженні файлу (наприклад, файл перевищує дозволений розмір або має непідтримуваний формат), система повідомляє про неможливість збереження аватара. У випадку проблем із сервером або збереженням у базі даних, відображається загальне повідомлення про збій з рекомендацією спробувати пізніше.

3.2.5 Навички для волонтерів

3.2.5.1 Загальна характеристика функції

Ця функціональність дозволяє користувачам шукати навички за допомогою пошуку в існуючому списку, вибирати потрібну навичку та одразу вказувати рівень володіння нею при додаванні до свого профілю. Користувач також має можливість редагувати рівень вже доданих навичок або видаляти їх із профілю, якщо вони більше не актуальні або потребують оновлення згідно змін.

3.2.5.2 Вхідні дані

Під час додавання навички користувач вводить текст у пошукове поле для швидкого пошуку потрібної навички у списку, обирає відповідну навичку та вказує рівень володіння нею. Для редагування користувач може змінити рівень існуючої навички або видалити її. Всі введені дані проходять валідацію на коректність та обов’язковість.

3.2.5.3 Вихідні дані

Після успішного додавання, редагування або видалення навички користувач отримує підтвердження про виконану операцію. Відповідні зміни одразу відображаються у профілі користувача в розділі «Навички», а база даних оновлюється для подальшого використання системою.

3.2.5.4 Обробка помилок

Якщо користувач вводить некоректні дані, наприклад, залишає поле назви навички порожнім або вибирає неіснуючий рівень, система повідомляє про помилку та надає рекомендації щодо виправлення. При спробі додати навичку, яка вже є у профілі, відображається повідомлення про дублювання. У разі технічних проблем збереження інформації користувач отримає загальне повідомлення про збій із пропозицією повторити дію пізніше.

3.2.6 Функціональність ініціатив призначена для організатора

3.2.6.1 Загальна характеристика функції

Модуль ініціатив для організатора забезпечує створення, редагування, видалення волонтерських проєктів та керування командою волонтерів. Організатор може задавати ключові параметри ініціативи — назву, опис, категорії, місце проведення, дедлайни та статуси. Крім того, модуль дозволяє управляти складом команди: переглядати список волонтерів, призначати їм ролі в межах проєкту, а також видаляти учасників. Такий функціонал забезпечує ефективне управління ініціативою та підтримує прозорість у роботі з командою.

3.2.6.2 Вхідні дані

Для створення ініціативи організатор вводить назву проєкту, опис, обирає категорії, вказує місце проведення та дедлайн. При редагуванні — оновлює ці дані або змінює статус проєкту. Для керування командою організатор вибирає волонтерів зі списку, призначає ролі або видаляє їх із проєкту. Всі введені дані перевіряються на коректність і відповідність вимогам.

3.2.6.3 Вихідні дані

Після успішного створення або оновлення ініціативи система відображає підтвердження операції і оновлює інформацію в профілі проєкту. Зміни у складі команди миттєво відображаються в інтерфейсі організатора та стають доступними для інших учасників. Видалення проєкту або волонтерів супроводжується відповідними повідомленнями.

3.2.6.4 Обробка помилок

Якщо введені дані некоректні (наприклад, відсутня назва проєкту або вказаний недійсний дедлайн), система повідомляє про помилки та надає рекомендації для їх виправлення. При спробі призначити неіснуючого користувача відображається повідомлення про помилку. У разі технічних збоїв під час збереження інформації користувач отримує загальне повідомлення про помилку з порадою спробувати пізніше.

3.2.7 Перегляд ініціатив

3.2.7.1 Загальна характеристика функції

Функціональність перегляду ініціатив реалізована у двох форматах: традиційному списковому та інтерактивному картографічному. Обидва способи дозволяють користувачам ознайомлюватися з актуальними волонтерськими проєктами, застосовувати пошук і фільтрацію, а також отримувати детальну інформацію про кожну ініціативу. У списковому режимі ініціативи подаються у вигляді карток, що містять основні відомості. Водночас інтерактивна мапа дозволяє візуально переглядати розміщення ініціатив поблизу, обирати їх за місцем розташування та взаємодіяти з ними через геомітки.

3.2.7.2 Вхідні дані

Користувач може ввести або вибрати параметри для пошуку й фільтрації ініціатив. У списковому перегляді це можуть бути: ключові слова, категорії, дистанція до ініціативи, та кінцевий строк активності ініціативи. Для картографічного перегляду додатково запитуються координати користувача та максимальна відстань до ініціатив. Усі вхідні параметри перевіряються на коректність — наприклад, щоб уникнути негативних значень відстані або помилкового формату дати.

3.2.7.3 Вихідні дані

У списковому форматі користувач отримує перелік ініціатив, що відповідають заданим умовам. Кожна ініціатива представлена карткою з назвою, коротким описом, категорією, датами, і кнопкою перегляду деталей. Після натискання відкривається повна інформація про ініціативу з контактами та можливістю приєднатися. У режимі мапи система відображає маркери відповідних ініціатив на мапі. Натискання на маркер відкриває короткий опис і кнопку для переходу до повної інформації про проєкт. Застосування нових фільтрів або зміна місця розташування призводить до динамічного оновлення відображення без перезавантаження сторінки.

3.2.7.4 Обробка помилок

Якщо введено некоректні значення — наприклад, недопустимий формат дати, від’ємна відстань або порожні обов’язкові поля — система відображає підказки щодо виправлення. У разі технічних збоїв, таких як неможливість завантажити список ініціатив або маркери на мапі, користувач отримує загальне повідомлення про помилку з рекомендацією оновити сторінку або повторити спробу пізніше.

3.2.8 Функціональність адміністрування платформи

3.2.8.1 Загальна характеристика функції

Модуль адміністрування надає адміністраторам можливість повного контролю над платформою. Вони можуть виконувати стандартні CRUD-операції з довідниками навичок і категорій, створюючи, редагуючи, видаляючи та переглядаючи їх. Адміністратори мають доступ до повної історії проєктів, що дозволяє відстежувати зміни, статуси і активність у межах волонтерських ініціатив. Також модуль включає управління користувачами платформи — зміну ролей (волонтер, ініціатор, адміністратор) та видалення облікових записів, що забезпечує підтримку безпеки і актуальності користувацької бази.

3.2.8.2 Вхідні дані

Адміністратор вводить необхідну інформацію для управління навичками і категоріями, включаючи назви, описи та інші параметри. Для управління користувачами доступна інформація про облікові записи, ролі, а також статуси проєктів. Всі введені дані проходять валідацію, щоб уникнути помилок і зберегти цілісність системи.

3.2.8.3 Вихідні дані

Після успішного виконання дій адміністратор отримує підтвердження про внесені зміни. Оновлення миттєво відображаються у відповідних розділах платформи. Історія проєктів доступна для перегляду у вигляді логів змін, що дає змогу відстежувати дії ініціаторів і волонтерів. Зміни ролей користувачів одразу оновлюють їх права доступу.

3.2.8.4 Обробка помилок

Якщо адміністратор вводить некоректні дані, система повідомляє про помилку із поясненням. У разі технічних проблем або втрати зв’язку з сервером адміністратор отримає повідомлення з рекомендацією повторити дію пізніше. Всі помилки фіксуються у системному журналі для подальшого аналізу.

3.2.9 Функціональність перегляду активних ініціатив

3.2.9.1 Загальна характеристика функції

Модуль перегляду активних ініціатив надає користувачам можливість ознайомитися з поточними волонтерськими проєктами, доступними для участі. Користувачі можуть шукати ініціативи за назвою та застосовувати фільтри за категоріями та кінцевим строком, що допомагає швидко знаходити релевантні проєкти відповідно до своїх інтересів і можливостей. Додатково, користувачі мають можливість покинути ініціативу, в якій вони наразі беруть участь, що дозволяє гнучко керувати власною волонтерською активністю.

3.2.9.2 Вхідні дані

Для пошуку користувач вводить ключові слова або частину назви ініціативи у пошукове поле. Для фільтрації обираються відповідні категорії зі списку та встановлюється кінцевий строк за допомогою календарного інтерфейсу. Для виходу з ініціативи користувач підтверджує своє рішення через інтерфейс.

3.2.9.3 Вихідні дані

Система відображає список активних ініціатив, які відповідають критеріям пошуку та фільтрації. Після виходу з ініціативи користувач отримує підтвердження про успішне виконання операції, а його статус у цій ініціативі оновлюється у базі даних, що позначається у відповідних розділах платформи.

3.2.9.4 Обробка помилок

У разі відсутності ініціатив, які відповідають заданим параметрам пошуку чи фільтрації, користувач отримує відповідне повідомлення про відсутність результатів. Якщо виникають технічні проблеми під час виходу з ініціативи, система повідомляє про збій і пропонує повторити спробу пізніше. Всі помилки обробляються таким чином, щоб мінімізувати негативний досвід користувача.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Платформа повинна забезпечувати стабільну роботу під час типової експлуатації. В системі реалізовано базове логування на сервері, яке дозволяє виявляти критичні збої та помилки. Очікується, що при правильному використанні і дотриманні ресурсних обмежень система буде працювати без критичних збоїв не менше ніж 99% часу.

3.3.2 Доступність

Система орієнтована на постійний доступ до основного функціоналу без обмеження в часі, за винятком технічних перерв, які можуть виникати під час розгортання оновлень. Можливі короткочасні збої, оскільки використовується безкоштовна база даних, але очікувана доступність для користувачів становить не менше 98% часу. У разі критичних збоїв передбачається можливість ручного перезапуску сервісу

3.3.3 Безпека

Система реалізує аутентифікацію користувачів за допомогою JWT-токенів, що зберігаються у cookie з прапором HttpOnly. Всі паролі перед збереженням хешуються. Права доступу до функціоналу визначаються роллю користувача (волонтер, ініціатор, адміністратор). Передача даних між клієнтом і сервером здійснюється через HTTPS. Сторонні сервіси авторизації не використовуються. У поточній версії не реалізовані двофакторна автентифікація, обмеження кількості спроб входу.

3.3.4 Супроводжуваність

Архітектура застосунку модульна: серверна логіка розділена на сервіси, контролери та моделі. Це дозволяє локалізовано оновлювати або розширювати функціональність. У проекті використано добре структурований код та автоматичну документацію API через Swagger. Для забезпечення подальшого супроводу платформи передбачено логування помилок та рефакторинг, що полегшує технічне обслуговування та масштабування системи.

3.3.5 Переносимість

Програмне забезпечення розроблено як кросплатформене вебрішення. Клієнтська частина сумісна з усіма популярними браузерами та адаптована для мобільних пристроїв. Серверна частина працює на .NET Core, що забезпечує її сумісність з ОС Windows, Linux та можливість контейнеризації за допомогою Docker. Підтримується розгортання у хмарному середовищі, включаючи масштабовані хостингові платформи та сервіси.

3.3.6 Продуктивність

Основні запити (отримання списку ініціатив, категорій, навичок, участей) обробляються менш ніж за 2 секунди при звичайному навантаженні. Оптимізація продуктивності досягається за рахунок використання MongoDB-індексів та клієнтського кешування через React Query. Архітектура дозволяє масштабування в майбутньому.

3.4 Вимоги бази даних

База даних є критично важливою частиною функціонування системи «Всеукраїнський волонтерський хаб», оскільки забезпечує зберігання, структуровану організацію та ефективну обробку основних сутностей: користувачів, їхніх профілів, волонтерських ініціатив, навичок, категорій та участей у проєктах. Її структура повинна бути логічно побудованою, цілісною та масштабованою, щоб гарантувати стабільну роботу системи.

Основною вимогою до побудови схеми бази даних є підтримка зв’язків між об’єктами. Зокрема, кожен користувач пов’язаний зі своїм профілем, набором навичок, що можуть бути підтверджені іншими учасниками. Один користувач може брати участь у декількох ініціативах, а кожна ініціатива може охоплювати кілька категорій. Такі зв’язки реалізовані через допоміжні таблиці з підтримкою типу зв’язку «багато до багатьох», що дозволяє зберігати дані без дублювання та втрати цілісності.

База даних повинна забезпечувати зберігання всієї ключової інформації про проєкти, включаючи назву, опис, дати створення та завершення, координати для геовізуалізації, перелік пов’язаних категорій і список учасників. Також необхідно зберігати інформацію про ролі волонтерів у межах кожного проєкту, що дозволяє аналізувати рівень залучення та структуру команди.

З метою забезпечення високої продуктивності платформи база даних повинна підтримувати індексацію ключових полів, таких як назва проєкту, категорія, навичка, координати та дата завершення. Це дозволить реалізувати ефективні пошук і фільтрацію при великій кількості запитів. Для функціонування інтерактивної мапи важливою є наявність геоіндексу для підтримки запитів у радіусі.

У поточній реалізації використовується хмарна документно-орієнтована база даних MongoDB Atlas. Це передбачає обмеження за обсягом зберігання, кількістю одночасних з’єднань і швидкістю обробки запитів.

Структура бази даних повинна бути готова до розширення — зокрема до інтеграції додаткових модулів, таких як «банк часу», «фонд ідей» чи система оцінювання волонтерів. Це означає, що модель даних має бути побудована таким чином, щоб нові сутності можна було додавати без порушення логіки вже існуючих зв’язків і без необхідності складних міграцій.

3.5 Інші вимоги

Програмне забезпечення «Всеукраїнський волонтерський хаб» створене з освітньою метою і не передбачає комерційного використання на момент першого релізу. Платформа розробляється на основі відкритих технологій та бібліотек, дозволених для вільного використання в навчальних і некомерційних проєктах.

Інтерфейс системи англійською мовою, тому окрема локалізація не потрібна.

Розробка враховує етичні принципи: система підтримує розподіл ролей, персональні дані не передаються третім сторонам і доступні лише авторизованим користувачам відповідного рівня. Платформа не містить механізмів монетизації або дискримінаційної фільтрації та спрямована на забезпечення відкритої, безпечної й інклюзивної взаємодії між волонтерами й ініціативами.