**Software Requirements Specification for**

**<** **Програмна система для ведення**

**деталізованої волонтерської звітності**

**як засобу підвищення залученості донорів >**

**Version 1.1 approved**

**Prepared by <Поліна Берест>**

**<ПЗПІ-21-3>**

**<02.06.2025>**

**ЗМІСТ**

[1 ВСТУП](#__RefHeading___Toc3267_16749813) 3

[1.1 Огляд продукту](#__RefHeading___Toc3269_16749813) 3

[1.2 Мета 4](#__RefHeading___Toc3271_16749813)

[1.3 Межі 4](#__RefHeading___Toc3273_16749813)

[1.4 Посилання 5](#__RefHeading___Toc3275_16749813)

[1.5 Означення та абревіатури 5](#__RefHeading___Toc3277_16749813)

[2 ЗАГАЛЬНИЙ ОПИС 6](#__RefHeading___Toc3279_16749813)

[2.1 Перспективи продукту 6](#__RefHeading___Toc3281_16749813)

[2.2 Функції продукту 6](#__RefHeading___Toc3283_16749813)

[2.3 Характеристики користувачів 7](#__RefHeading___Toc3285_16749813)

[2.4 Загальні обмеження 8](#__RefHeading___Toc3287_16749813)

[2.5 Припущення й залежності 8](#__RefHeading___Toc3289_16749813)

[3 КОНКРЕТНІ ВИМОГИ 9](#__RefHeading___Toc3291_16749813)

[3.1 Вимоги до зовнішніх інтерфейсів 9](#__RefHeading___Toc3293_16749813)

[3.1.1 Інтерфейс користувача 9](#__RefHeading___Toc3295_16749813)

[3.1.2 Апаратний інтерфейс 9](#__RefHeading___Toc3297_16749813)

[3.1.3 Програмний інтерфейс 10](#__RefHeading___Toc3299_16749813)

[3.1.4 Комунікаційний протокол 10](#__RefHeading___Toc3301_16749813)

[3.1.5 Обмеження пам’яті 10](#__RefHeading___Toc3516_16749813)

[3.1.6 Операції 10](#__RefHeading___Toc3518_16749813)

[3.2 Властивості програмного продукту 11](#__RefHeading___Toc3520_16749813)

[3.3 Атрибути програмного продукту 12](#__RefHeading___Toc3522_16749813)

[3.3.1 Надійність 12](#__RefHeading___Toc3524_16749813)

[3.3.2 Масштабованість 12](#__RefHeading___Toc3526_16749813)

[3.3.3 Адаптивність 12](#__RefHeading___Toc3530_16749813)

[3.3.4 Підтримка платформ](#__RefHeading___Toc3534_16749813) 13

[3.3.5 Безпека даних](#__RefHeading___Toc3536_16749813) 13

[3.3.6 Захист від помилок](#__RefHeading___Toc3540_16749813) 13

[3.4 Вимоги бази даних 13](#__RefHeading___Toc3542_16749813)

**1 ВСТУП**

* 1. Огляд продукту

В умовах повномасштабної війни волонтерський рух набув особливого значення та став важливим суспільним явищем в Україні. Зусилля волонтерів спрямовані на підтримку як Збройних Сил України, так і цивільного населення. Швидкість та адресність волонтерської допомоги часто дозволяють вирішити нагальні проблеми там, де державна система зволікає або не має достатнього ресурсу.

Незважаючи на стрімкий розвиток волонтерського руху, індивідуальні волонтери досі долають цілу низку викликів. Оскільки такі волонтери не входять до складу організацій, їхня можливість розширювати свою активну аудиторію та залучати фінансування на проєкти обмежується колом родичів, колег та знайомих або підписників у соціальних мережах а також фінансовими можливостями, політичними поглядами та колом інтересів таких потенційних донорів.

Державні платформи не підтримують локальні ініціативи та не пристосовані, щоб швидко реагувати на термінові запити. Спеціалізованих сервісів для фандрейзингу та звітування перед донорами для створення атмосфери довіри та залучення донорів на постійну основу, чого потребують українські волонтери, наразі не було створено.

Нагальною потребою індивідуальних волонтерів є пожвавлення зборів та залучення нових донорів, зацікавлених в їхній підтримці у подальшому.

Задовільнити таку потребу можна, надавши ринку рішення з усунення обмеження по розміру аудиторії волонтерів: платформу для прозорого представлення та легкого відстеження волонтерської діяльності та витрат, на якій зацікавлені донори зможуть обрати волонтера за його попередньою діяльністю та зробити внесок.

Для волонтерів система має стати платформою для відкритої звітності, завоювання довіри та збору коштів, для простих користувачів ­ ресурсом для дослідження діяльності волонтерів та ухвалення рішення про благодійний внесок.

* 1. Мета

Метою роботи є проєктування програмної системи для пожвавлення зборів активних індивідуальних волонтерів шляхом залучення нових донорів, зацікавлених в підтримці волонтерів, що надають певні категорії допомоги, та вибудови системи довіри між волонтером, що прозоро веде свою діяльність, та донорами.

Функціонал системи має бути розділений відповідно до трьох ролей: волонтера, простого користувача (для користувачів, які ознайомлюються з волонтерськими профілями та донатять) та адміністратора (який водночас виступатиме модератором контенту).

Передбачається створення серверного застосунку для проведення основних операцій за бізнес-логікою, мобільного застосунку для волонтера для швидкого звітування та веб-сайту, що виступатиме «вітриною» волонтерської діяльності та на якому простим користувачам буде зручно з будь-якого пристрою ознайомлюватися з профілями та звітами волонтерів.

* 1. Межі

У рамках роботи буде реалізовано програмну систему, складовими якої будуть серверний застосунок, в сервісах якого обробляються операції за бізнес-логікою, маніпуляції з даними системи; мобільний клієнт – застосунок для волонтера, в якому можна створити та оновлювати інформацію про профіль волонтера, створювати звіти про надану допомогу; веб-клієнт, на якому потенційні донори можуть ознайомитися з діяльністю волонтерів, переглянути їхні звіти та статистику надання допомоги, обрати волонтера, який вартий довіри, та зробити благодійний внесок. Через веб-клієнт також буде здійснюватися адміністративний доступ.

Основними технологіями, використаними для розробки програмної системи, будуть мови програмування Java (для мобільного застосунку), C# (платформа .NET 7, ASP.NET Core для серверного застосунку), фреймворк Angular Framework 17 із використанням мови програмування TypeScript (для веб-застосунку), протоколи REST API, OData. MS SQL Server буде використано як базу даних, для авторизації будемо послуговуватися JWT, Google OAuth.

* 1. Посилання

Документ посилається на стандарт ДСТУ 3008:2015 «Документація. Звіти у сфері наукових досліджень та розробок. Структура і правила оформлення» та на документ, що встановлює кращі практики з написання специфікацій до програмного забезпечення, IEEE 830-1993 «IEEE Recommended Practice for Software Requirements Specifications».

* 1. Означення та абревіатури

DI – Dependency Injection

JSON ­ JavaScript Object Notation

JWT ­ JSON Web Token

MVP ­ Minimum Viable Product

ПЗ – програмне забезпечення

БД – база даних

СУБД – система управління базами даних

1. **ЗАГАЛЬНИЙ ОПИС**
   1. Перспективи продукту

Програмна система VolunteerReport має значний потенціал для подальшого розвитку як соціального, так і технологічного продукту. У короткостроковій перспективі система може бути використана як MVP для тестування ідеї цифрової платформи прозорої звітності волонтерів та взаємодії з донорами в Україні. Уже на етапі запуску вона вирішує актуальні проблеми, зокрема обмежений доступ індивідуальних волонтерів до аудиторії потенційних донорів та відсутність централізованого інструменту для ведення звітності.

У середньостроковій перспективі платформа може масштабуватись:

– на регіональному рівні — для підтримки локальних волонтерських ініціатив;

– на національному — як альтернатива або доповнення до державних сервісів, які не підтримують швидкого реагування на запити;

– на міжнародному — за умови додавання підтримки більшої кількості платіжних сервісів, що дозволить українським волонтерам залучати кошти від діаспори або іноземних благодійників.

У довгостроковій перспективі продукт має комерційний потенціал. Його можна використовувати як як white-label рішення для громадських організацій.

* 1. Функції продукту

Система підтримує три основні ролі користувачів: волонтер, звичайний користувач (донор) та адміністратор.

Функціонал волонтера:

– реєстрація та авторизація через Google-акаунт;

– створення та редагування профілю волонтера;

– створення та видалення звітів про надану допомогу;

– прикріплення до звіту унікального фото.

Функціонал користувача (донора):

– реєстрація та авторизація через Google-акаунт;

– перегляд волонтерських профілів з фільтрацією за категорією допомоги, датою створення;

– перегляд звітів з можливістю фільтрації за категорією або конкретним волонтером;

– перегляд статистики волонтерської активності;

– отримання реквізитів для донату;

– подання скарги на волонтерський профіль (не більше двох на добу).

Функціонал адміністратора:

– управління волонтерськими профілями (блокування, розблокування, приховування);

– управління звітами волонтерів;

– модерація скарг від користувачів;

– управління категоріями допомоги (додавання, редагування, архівування).

* 1. Характеристики користувачів

Таблиця 2.1 – Користувачі системи

|  |  |  |  |
| --- | --- | --- | --- |
| **Користувач** | **Очікуваний рівень навичок** | **Пристрої** | **Основні цілі** |
| Волонтер | Середній (використання мобільних застосунків, Google-акаунт). | Смартфон (Android) | Вести звітність, отримати підтримку. |
| Донор | Низький–середній (використання браузера, фільтрація). | ПК, смартфон, планшет | Знайти надійного волонтера, задонатити ­ наблизити перемогу. |
| Адміністратор | Середній або високий. | ПК, смартфон, планшет | Модерувати контент, керувати системою. |

* 1. Загальні обмеження

У розробці буде прийнято такі обмеження:

– мобільний застосунок працює лише на Android (SDK ≥ 24);

– автентифікація користувачів виключно через Google-акаунт (OAuth 2.0);

– один волонтерський звіт містить лише одне фото;

– перевірка плагіату фото здійснюється лише в межах системи;

– редагувати волонтерський звіт після публікації неможливо, лише видалити;

– видалення волонтерського акаунту неможливе – тільки блокування.

* 1. Припущення й залежності

У межах розробки буде прийнято такі припущення та залежності:

* користувачі мають стабільне інтернет-з’єднання;
* волонтери мають смартфон з камерою та ОС Android;
* усі користувачі мають Google-акаунт;
* користувачі-донори та адміністратори користуються сучасними браузерами;
* система використовує MS SQL Server, Angular, ASP.NET Core, Java SDK.
* .

1. **КОНКРЕТНІ ВИМОГИ**
   1. Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Інтерфейси веб- та мобільного застосунків мають мати схожі елементи та візуальний стиль (формат акцентування уваги, кольорову палітру, шрифти).

Вимоги до інтерфейсу мобільного застосунка (для волонтера):

– інтерфейс оптимізований для смартфонів із сенсорним керуванням;

– мінімалістичний дизайн з акцентом на простоту створення звіту;

– форма створення звіту передбачає: вибір категорії допомоги, введення одиниці виміру, кількості, вартості, опису та додавання фото;

– підтвердження можливості публікації звіту здійснюється лише після проходження перевірки на унікальність фото;

– усі елементи мають бути доступні однією рукою (UX для черги на пошті).

Вимоги до інтерфейсу веб-застосунка (для донора та адміністратора):

– інтерфейс адаптивний (працює з ПК, планшетів, мобільних);

– розділення на головну сторінку, профілі волонтерів, перегляд звітів, авторизацію;

– для адміністраторів — доступ до панелі керування з розділами «Користувачі», «Звіти», «Категорії», «Скарги»;

– реалізовано фільтрацію (за категорією допомоги, активністю, датою) та сортування.

3.1.2 Апаратний інтерфейс

Мінімальні вимоги до апаратного інтерфейсу для мобільної частини:

­ ОС Android 7.0 (SDK 24) або вище;

­ камера (для фото);

­ оперативна пам’ять: не менше 2 ГБ.

Мінімальні вимоги до апаратного інтерфейсу для запуску веб-застосунку:

­ ОС: Windows 10+, macOS, Linux;

­ браузер: Chrome, Firefox, Safari, Edge.

3.1.3 Програмний інтерфейс

Програмний інтерфейс системи має бути реалізований у вигляді REST API, що забезпечує обмін даними між клієнтськими та серверною частинами. Розширенням для REST API виступить OData (для зручних фільтрації та сортування). JSON має бути обрано як стандартний формат обміну даними в системі. Авторизація запитів має здійснюватися через Bearer Token (JWT).

Серверна частина з АРІ має бути побудована на ASP.NET Core, використовувати Entity Framework Core для доступу до БД.

3.1.4 Комунікаційний протокол

Комунікація між різними компонентами системи здійснюватиметься через протоколи HTTP/HTTPS та OData. Архітектура має бути побудованою за клієнт-серверним принципом, де сервер виконує основні усі функції з бізнес логіки та управління даними в системі.

3.1.5 Обмеження пам’яті

Система для стабільної роботи з одночасними запитами потребує мінімум 4 ГБ оперативної пам’яті на сервері. У випадку роботи з завантаженим фото потреба буде дещо збільшуватися.

3.1.6 Операції

Взаємодія між клієнтами і серверною частиною системи буде здійснюватися через HTTPS-протокол. Один з клієнтів надсилає запит до API, після обробки отримує відповідь у форматі JSON. Для доступу до ресурсів використовуються URL-адреси, які ведуть до методів контролерів, які відкривають функціонал системи.

URL-ресурс супроводжується HTTP-методом, який і визначає тип операції, що виконується. Операціями, що буде використано в системі, є:

* GET –для отримання даних з сервера;
* POST – для створення нових записів;
* PUT – для оновлення даних;
* DELETE – для видалення даних з системи.

3.2 Властивості програмного продукту

Програмна система VolunteerReport має бути розроблена як багатокомпонентний клієнт-серверний застосунок, що включає серверну частину (бекенд), веб-інтерфейс для донорів та адміністраторів і мобільний клієнт для волонтерів. Однією з ключових властивостей такої системи є її модульність та масштабованість: кожен з компонентів може бути розгорнутий незалежно, що дозволяє оптимізувати ресурси, пришвидшити оновлення та спростити супровід.

Система має підтримувати адаптивний дизайн для веб-застосунку, що забезпечить комфортне використання з різних пристроїв, включаючи персональні комп’ютери, планшети та мобільні телефони. Застосунок автоматично підлаштовується під розмір екрану користувача, зберігаючи зручність навігації та доступність основних функцій.

Особливу увагу має бути приділено безпеці. Для автентифікації користувачів буде використано Google OAuth 2.0, що дозволить уникнути обробки чутливої інформації всередині системи. Доступ до функціоналу системи здійснюється згідно з роллю користувача, що мінімізує ризик несанкціонованих дій. Також у системі буде реалізовано механізми серверної й клієнтської валідації, що дозволить запобігати помилкам введення даних та забезпечити стабільну роботу навіть у випадку некоректних дій з боку користувача.

Система має підтримувати локалізацію інтерфейсу українською та англійською мовами, що дозволить охопити як внутрішню, так і міжнародну аудиторію донорів. Завдяки впровадженню протоколу OData, клієнтські частини зможуть ефективно взаємодіяти з бекендом, реалізуючи складні фільтрації та сортування без перевантаження серверу або мережі.

Таким чином, система матиме набір властивостей, які необхідні для успішного впровадження у волонтерській сфері ­ це прозорість, безпека, стабільність, гнучкість масштабування, ефективність обробки даних і підтримка різних платформ.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Система має бути доступною та працездатною впродовж тривалого часу. Для цього має бути реалізовано обробку винятків на сервері, логування помилок та стабільну авторизацію через Google OAuth. Особлива увага має приділитися валідації вхідних даних як на стороні клієнта, так і на сервері. Це дозволить мінімізувати ризик помилкових запитів та забезпечить безперервну роботу основного функціоналу навіть при часткових збоях мережі або системних служб.

3.3.2 Масштабованість

Архітектура системи має дозволяти масштабувати її компоненти. Серверна частина має бути побудована з урахуванням розділення логіки на модулі, що забезпечить легке розширення функціоналу. Застосування REST API та OData дозволить інтегрувати додаткові клієнтські додатки. Окремі підсистеми — мобільний застосунок, вебінтерфейс і серверна частина — можуть розгортатися на окремих серверах, що дозволяє збільшувати ресурси залежно від навантаження.

3.3.3 Адаптивність

Інтерфейс веб-застосунку має бути розроблений з урахуванням адаптивності — він має коректно відображатися на екранах різних розмірів та орієнтацій. Це дозволить користувачам (донорам або адміністраторам) працювати із системою як з комп’ютера, так і з мобільного пристрою без потреби встановлювати додаткове програмне забезпечення. Мобільний застосунок, у свою чергу, буде оптимізований для зручного користування однією рукою, що особливо важливо для волонтерів, які часто працюють у динамічних умовах.

3.3.4 Підтримка платформ

Вебзастосунок має працювати у всіх сучасних браузерах (Chrome, Firefox, Edge, Safari), він не потребуватиме встановлення.

Мобільний клієнт буде створено для ОС Android з використанням Java SDK (рівень 24 і вище). Серверна частина може бути розгорнута на будь-якому сервері або хмарному середовищі, яке підтримує ASP.NET Core та MS SQL Server. Це дозволяє обирати гнучке середовище для розгортання відповідно до доступної інфраструктури.

3.3.5 Безпека даних

Система має бути побудована з урахуванням принципів безпечного зберігання та обробки даних. Авторизація буде здійснюватися через Google OAuth, що виключить зберігання паролів. Доступ до функціоналу має бути обмежено згідно з роллю користувача (волонтер, донор, адміністратор). Передача даних має здійснюватися виключно через захищене з’єднання HTTPS.

3.3.6 Захист від помилок

Система має передбачати обробку всіх очікуваних типів помилок: від некоректного введення до мережевих збоїв. У разі помилки користувач має отримати зрозуміле повідомлення, яке дозволить скоригувати дію. Критичні помилки мають логуватися для подальшого аналізу адміністратором або розробниками. Таким чином, система буде підтримувати як превентивний захист (валидація, обмеження ролей), так і реактивний (відновлення роботи після виняткових ситуацій).

3.4 Вимоги бази даних

Для зберігання та обробки інформації в системі буде використовуватися реляційна база даних, реалізована на платформі Microsoft SQL Server 2019. Взаємодія з БД буде здійснюватися за допомогою ORM Entity Framework Core, що дозволить зручно працювати з даними у вигляді об'єктів у коді та забезпечить ефективну реалізацію CRUD-операцій.

Структура бази даних має відповідати відповідає предметній області системи. Основними сутностями є: користувачі, волонтери, волонтерські профілі, звіти, категорії допомоги, скарги та допоміжні таблиці для відстеження змін: наприклад, блокувань профілів. Таблиці пов’язані між собою зв’язками типу «один-до-багатьох» або «багато-до-одного». Наприклад, один волонтер може мати багато звітів, один волонтер може мати лише один відповідний запис в таблиці заблокованих профілів.

База даних має проектуватися з урахуванням подальшої оптимізації запитів: має бути використано індекси для полів, що часто використовуються у фільтрації (наприклад, категорія допомоги).