**Software Requirements Specification for**

**<** **Програмна система для керування сервісами та взаємодії користувачів платформи замовлення послуг «Doerly»>**

**Version 1.0 approved**

**Prepared by <Lukianchuk Danylo, Pyroh Andrii, Dzabakhidze Davyd, Merkulov Arsenii, Bashenko Ivan>**

**<PZPI-21-3>**

**<15.06.2025>**

**ЗМІСТ**

[1 ВСТУП](#_heading=h.voeqai7yny4b) 3

[1.1 Огляд продукту](#_heading=h.mtd41jr5ya33) 3

[1.2 Мета 4](#_heading=h.9v34g7s1w0ni)

[1.3 Межі](#_heading=h.38h229mno2k2) 5

[1.4 Посилання](#_heading=h.7qds28ac9dot) 6

[1.5 Означення та абревіатури](#_heading=h.hx7gt7aadyk6) 6

[2 ЗАГАЛЬНИЙ ОПИС 6](#_heading=h.p8sbutptmvo)

[2.1 Перспективи продукту 6](#_heading=h.ovj12wiwylow)

[2.2 Функції продукту 6](#_heading=h.fzh6csbgpsiz)

[2.3 Характеристики користувачів 8](#_heading=h.pg5finozbvlg)

[2.4 Загальні обмеження 9](#_heading=h.sr4tkey6uggv)

[2.5 Припущення й залежності 9](#_heading=h.lc32rcceaimz)

[3 КОНКРЕТНІ ВИМОГИ](#_heading=h.rqwml3g566ui) 14

[3.1 Вимоги до зовнішніх інтерфейсів](#_heading=h.3tk64to5zr0s) 15

[3.1.1 Інтерфейс користувача](#_heading=h.f481mnyv7k5h) 15

[3.1.2 Апаратний інтерфейс](#_heading=h.7q8xdaejun85) 15

[3.1.3 Програмний інтерфейс](#_heading=h.qoxjcse4q2c7) 15

[3.1.4 Комунікаційний протокол](#_heading=h.rms3os8mm9f) 16

[3.1.5 Обмеження пам’яті](#_heading=h.qk4lkajv4b4u) 16

[3.1.6 Операції](#_heading=h.1ra1duf3kbh) 17

[3.2 Властивості програмного продукту](#_heading=h.m7ydxnrh7wmv) 17

[3.3 Атрибути програмного продукту](#_heading=h.k3kt08af1nm3) 17

[3.3.1 Надійність](#_heading=h.aecduj1cu68g) 17

[3.3.2 Масштабованість](#_heading=h.oyrcy3f99a1k) 17

[3.3.3 Відкрита до відлагодження](#_heading=h.qtytyk9gxjmi) 18

[3.3.4 Адаптивність](#_heading=h.jplsnilwr4z9) 18

[3.3.5 Швидкодія](#_heading=h.85wx2ttr7au) 18

[3.3.6 Підтримка багатьох браузерів](#_heading=h.ikv5l52kci8z) 19

[3.3.7 Безпека даних](#_heading=h.86wwra8cttu) 19

[3.3.8 Робота з одночасним доступом](#_heading=h.jsorapkgzthr) 19

[3.3.9 Захист від помилок](#_heading=h.9b001yn2390e) 19

[3.4 Вимоги бази даних](#_heading=h.sjbuzoeg0qf2) 19

**1 ВСТУП**

* 1. Огляд продукту

У сучасному цифровому середовищі все більшої популярності набувають онлайн-платформи, які об'єднують замовників і виконавців різноманітних послуг. Такі сервіси створюють нові можливості для самозайнятих фахівців, малого бізнесу та звичайних користувачів, які прагнуть швидко та зручно знаходити виконавців для виконання побутових, технічних, креативних або спеціалізованих завдань. Водночас для ефективного функціонування таких платформ необхідна надійна програмна система, яка забезпечуватиме взаємодію між сторонами, прозору логіку замовлення послуг, підтримку фінансових операцій, облік користувацької активності, рейтингів, повідомлень та іншого супутнього функціоналу.

Описуваний продукт — це програмна система для керування сервісами та взаємодії користувачів у рамках платформи замовлення послуг. Вона орієнтована на побудову повноцінної цифрової екосистеми, у якій кожен користувач може виступати як замовником, так і виконавцем. Система охоплює широкий спектр функціональних модулів, зокрема модуль створення і обробки замовлень, модуль пропозицій від виконавців, систему повідомлень, систему оплат і внутрішніх балансів, модуль керування профілями користувачів, а також адміністративну панель для моніторингу та модерації активностей на платформі.

На відміну від існуючих рішень, система проєктувалась як модульний моноліт із розділенням відповідальностей між сервісами, що дозволяє підтримувати узгодженість логіки при збереженні високої продуктивності. Основна увага приділяється гнучкості, масштабованості та зручності інтерфейсів для кінцевого користувача. У рамках розробки враховуються як технічні, так і бізнес-вимоги, що дозволяє адаптувати систему під реальні потреби сервісного ринку. Програмна система покликана спростити процес організації послуг, підвищити ефективність взаємодії між користувачами, а також забезпечити високий рівень надійності та безпеки даних.

Передбачається, що дана система знайде застосування у багатьох сферах — від локальних сервісів (ремонт, доставка, прибирання) до фриланс-платформ, маркетплейсів послуг, внутрішніх корпоративних систем або мобільних додатків для керування запитами на виконання завдань. Таким чином, продукт має потенціал як для комерційного використання, так і для подальшої адаптації в рамках спеціалізованих проєктів.

* 1. Мета

Метою цього проєкту є розробка програмної системи для керування сервісами та взаємодії користувачів у межах онлайн-платформи замовлення послуг. Основне призначення системи полягає у створенні інструменту, який дозволяє зручно та ефективно організовувати процеси публікації, пошуку, виконання та оплати замовлень, забезпечуючи при цьому прозору взаємодію між замовниками та виконавцями.

Система має забезпечити зручний механізм керування замовленнями, підтримку двосторонньої комунікації, систему рейтингів та відгуків, а також надійні засоби проведення оплат і гарантування зобов’язань сторін. Архітектура проєкту передбачає поділ функціональності на окремі модулі, що дозволяє забезпечити гнучкість, підтримку подальшої масштабованості та адаптації під специфічні потреби платформи.

Проєкт орієнтований як на приватних замовників і виконавців, так і на бізнес-користувачів, які бажають оптимізувати взаємодію в межах надання чи отримання послуг. Очікується, що створене рішення не лише задовольнить функціональні потреби типового сервісу замовлень, але й стане основою для розширення платформи в напрямку інших супутніх сервісів, таких як аналітика, підтримка, модерація тощо.

* 1. Межі

У рамках даної роботи реалізовано клієнт-серверну архітектуру програмної системи, яка складається з кількох функціональних частин, зокрема основного застосунку для замовників і виконавців, а також адміністративної панелі.

Основна серверна частина (бекенд) побудована на базі ASP.NET Core із використанням мови програмування C# та платформи .NET 9. Вона реалізує бізнес-логіку платформи, обробку HTTP-запитів, керування користувачами, створення та обробку замовлень, а також забезпечує механізми авторизації та взаємодії між учасниками системи. Для роботи з базою даних застосовується ORM Entity Framework Core, а основною системою керування базами даних обрано PostgreSQL.

Клієнтська частина основного застосунку створена на Angular 18 із використанням HTML, SCSS та TypeScript. Вона забезпечує інтерфейс для замовників і виконавців, даючи можливість створювати, переглядати, приймати або відхиляти замовлення, а також вести комунікацію між сторонами.

Адміністративна панель реалізована окремим фронтенд-застосунком на основі React, що дозволяє адміністраторам керувати контентом платформи, модерацією користувачів і замовлень, а також отримувати доступ до аналітичної інформації.

Для зберігання медіафайлів, що супроводжують замовлення або профілі користувачів, використовується хмарний сервіс Azure Blob Storage, який дозволяє надійно зберігати дані поза межами основного серверного середовища з високим рівнем доступності.

Таким чином, у проєкті чітко розмежовані відповідальності між різними компонентами системи, що забезпечує її модульність, масштабованість і зручність подальшого супроводу.

* 1. Посилання

Документ посилається на стандарт ДСТУ 3008:2015 «Документація. Звіти у сфері наукових досліджень та розробок. Структура і правила оформлення». Текст має офіційний стиль, IEEE Std 830-1993 «Рекомендована практика IEEE для специфікацій вимог до програмного забезпечення».

* 1. Означення та абревіатури

DI – Dependency Injection

ПЗ – програмне забезпечення

DTO – Data Transfer Object

БД – база даних

СУБД – система управління базами даних

ORM – Object Relational Mapping

CI/CD – Continuous Integration/Continuous Deployment

API – Application Programming Interface

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

Модуль управління нашої платформи для замовлення послуг створений для спрощення адміністративних завдань і покращення управління профілями користувачів. У контексті зростання гіг-економіки наш продукт відповідає попиту на ефективні інструменти управління послугами. Збудований на сучасному технологічному стеку, що включає .NET Core і Angular, він забезпечує масштабованість і гнучкість для впровадження нових функцій, таких як рекомендації на базі штучного інтелекту та верифікація транзакцій через блокчейн.

У майбутньому ми плануємо розширити географію присутності платформи та перелік доступних послуг. Для збільшення активності користувачів і обсягу транзакцій ми впроваджуватимемо стратегії залучення, такі як програми лояльності та реферальні бонуси. Також ми розглядаємо можливості інтеграції з популярними сторонніми сервісами та партнерства з іншими постачальниками послуг для розширення функціональності платформи.

Хоча ми очікуємо викликів, таких як конкуренція з усталеними платформами (наприклад, Upwork чи Fiverr) та необхідність забезпечення безпеки даних, ми зосередимося на створенні кращого користувацького досвіду та дотриманні міжнародних стандартів захисту даних. Наша довгострокова мета — стати провідною платформою для постачальників і споживачів послуг, пропонуючи комплексні інструменти, що адаптуються до потреб галузі, та створивши самодостатню екосистему, яка сприяє постійному зростанню.

* 1. Функції продукту

Продукт розроблено для забезпечення потреб користувачів платформи замовлення послуг. Основний функціонал поділено на дві ключові складові: профіль користувача, доступний усім зареєстрованим користувачам (замовникам і виконавцям), та панель адміністратора, призначену для управління платформою.

Додатково реалізовано підсистеми комунікацій, каталогу та управління замовленнями, а також визначено функціонал для основних ролей — замовника та виконавця.

Профіль користувача дозволяє презентувати себе, управляти особистою та професійною інформацією, а також відстежувати активність на платформі.

Функціональні вимоги до профілю користувача:

* Перегляд і редагування основних даних: ім’я, прізвище, дата народження, стать;
* Додавання додаткової особистої інформації (наприклад, коротка біографія);
* Вказівка орієнтовного міста та країни проживання;
* Завантаження та оновлення фотографії профілю;
* Завантаження і редагування резюме з детальною інформацією про навички, досвід та кваліфікацію, важливі для замовників;
* Заповнення та редагування переліку категорій послуг, у яких користувач має компетенції для прийому замовлень;
* Вибір мов і зазначення рівнів їх володіння.

Панель адміністратора забезпечує контроль над платформою, модерацію активності користувачів і підтримку стабільної роботи системи.

Функціональні вимоги до панелі адміністратора:

* Перегляд списку всіх зареєстрованих користувачів із можливістю пошуку за ім’ям;
* Блокування акаунтів;
* Створення та редагування категорій і підкатегорій послуг у системі;
* Створення та редагування фільтрів категорій послуг;
* Перегляд усіх замовлень та їх детальної інформації з фільтрацією за виконавцем або замовником;
* Відображення аналітики про активність користувачів у системі;
* Відображення аналітики про фінансові оплати та відкриті рахунки.

Програмний застосунок передбачає дві основні ролі: замовник і виконавець.

Функції для всіх користувачів:

* Реєстрація в системі;
* Підтвердження електронної поштової адреси;
* Вхід у систему;
* Відновлення паролю;

Функціонал замовника:

* Оплата після наданих послуг;
* Поетапна оплата за надані послуги;
* Встановлення рейтингу виконавцю послуги;
* Надання текстового відгуку про роботу виконавця;
* Перегляд історії транзакцій;

Функціонал виконавця:

* Отримання оплати за надані послуги;
* Перегляд історії транзакцій;
* Перегляд відгуків на надані послуги;
* Надання відгуків про надані послуги;

Підсистема комунікацій забезпечує обмін повідомленнями між користувачами з інтеграцією до модулів профілю та авторизації.

Функціональні вимоги до комунікацій:

* Миттєве спілкування: користувачі бачать нові повідомлення одразу після надсилання без оновлення сторінки;
* Інтеграція з платформою: чітко визначені правила обміну даними з модулями авторизації та профілів, без тісного зв’язку між ними;
* Безпека: лише зареєстровані та авторизовані користувачі можуть переглядати, надсилати або отримувати повідомлення;
* Готовність до великої кількості користувачів: стабільна робота при великій кількості активних користувачів і повідомлень;
* Надсилання текстових повідомлень із збереженням інформації про відправника, вміст, час надсилання та статус ("Відправлено", "Доставлено", "Прочитано").

Підсистема каталогу забезпечує ефективний пошук, перегляд та взаємодію між замовниками і виконавцями послуг, надаючи зручну та інформативну навігацію.

Функціональні можливості каталогу:

* Пошук замовлень та послуг за ключовими словами;
* Застосування фільтрів до результатів пошуку (категорія, ціна, рейтинг, час виконання, мова тощо);
* Сортування замовлень та послуг за ціною та ім’ям;
* Перегляд детальної сторінки послуги або замовлення з описом, ціною, термінами виконання, прикладами робіт;
* Можливість переходу на сторінку публічного профілю виконавця з відгуками, рейтингом, статистикою виконаних замовлень та портфоліо;
* Оформлення замовлення безпосередньо зі сторінки послуги в кілька кліків;
* Створення нової послуги із заповненням назви, опису, вартості, тривалості виконання, категорії, тегів та умов;
* Редагування або деактивація власних послуг чи замовлень виконавцем;
* Адаптивний інтерфейс для комфортної взаємодії на мобільних пристроях;
* Автоматична рекомендація схожих послуг на основі категорії та рейтингу.

Підсистема замовлень забезпечує повний цикл управління замовленнями — від створення до завершення, з гнучкою взаємодією між виконавцями та замовниками. Інтерфейс адаптивний, а логіка стійка до помилок і маніпуляцій.

Функціональні вимоги до управління замовленнями:

* Створення нового замовлення з усією необхідною інформацією;
* Отримання переліку всіх замовлень користувача;
* Перегляд детальної інформації про окреме замовлення;
* Редагування існуючого замовлення;
* Скасування або видалення замовлення;
* Автоматичне відображення рекомендованих виконавців на основі категорії та локації;
* Підтвердження виконання замовлення з боку замовника та виконавця;
* Перехід на сторінку оплати замовлення через інтегровану платіжну систему;
* Формування запиту на виконання замовлення з боку замовника або виконавця;
* Перегляд списку вхідних та вихідних запитів на виконання;
* Детальний перегляд запиту на виконання замовлення з інформацією про користувача та статус;
* Прийняття, відхилення або відкликання запиту на виконання замовлення.

Цей розділ чітко описує функціональні можливості продукту, забезпечуючи легке розуміння для всіх зацікавлених сторін.

* 1. Характеристики користувачів

Програмна платформа для замовлення послуг орієнтована на три основні категорії користувачів: замовники, виконавці та адміністратори. Кожна категорія має унікальні ролі, цілі та технічні вимоги, що визначають їхню взаємодію з системою.

Замовники – це фізичні або юридичні особи, які шукають послуги через платформу. Це можуть бути приватні клієнти, представники малого бізнесу або компанії, що потребують виконання разових чи регулярних завдань, наприклад, розробки вебсайту, прибирання офісу або юридичної консультації. Вони зазвичай мають вік від 18 до 60 років, різний соціальний та професійний статус, базові навички роботи з вебзастосунками та мобільними пристроями, а також знання основ онлайн-платформ.

Замовники можуть використовувати смартфони, планшети, ноутбуки чи настільні комп’ютери з сучасними браузерами, такими як Chrome, Safari або Firefox. Їхні основні цілі – швидкий пошук і замовлення послуг, перегляд профілів виконавців, їхніх рейтингів і відгуків, безпечна оплата та прозора комунікація, а також отримання якісних результатів у встановлені терміни. Для цього їм потрібен інтуїтивно зрозумілий інтерфейс для пошуку та фільтрації послуг, можливість оцінювати виконавців і залишати відгуки, доступ до історії транзакцій і статусів замовлень, а також захист персональних даних і безпечні платіжні операції.

Виконавці – це фрілансери, спеціалісти або компанії, які пропонують свої послуги через платформу. Вони можуть надавати цифрові послуги, такі як графічний дизайн або програмування, локальні послуги, як-от ремонт або прибирання, або професійні послуги, наприклад, юридичні консультації чи бухгалтерія. Виконавці зазвичай віком від 18 до 55 років, часто самозайняті особи або мікропідприємці з середнім або високим рівнем технічної грамотності. Вони здатні створювати та редагувати профіль, завантажувати портфоліо та працювати з чатом.

Виконавці можуть використовувати мобільні пристрої для оперативного реагування та комп’ютери для детального налаштування профілю. Їхні основні цілі – презентація своїх навичок і залучення клієнтів, отримання замовлень і своєчасної оплати, а також формування позитивної репутації через рейтинги та відгуки. Для цього їм потрібен зручний профіль для демонстрації досвіду, портфоліо та навичок, прозора система отримання оплати та перегляду транзакцій, можливість відповідати на відгуки та оперативно спілкуватися із замовниками, а також фільтри для пошуку релевантних замовлень за категоріями та локацією.

Адміністратори – це внутрішні користувачі платформи, відповідальні за її управління, модерацію контенту та підтримку стабільної роботи. Це можуть бути співробітники компанії-розробника або спеціально призначені модератори віком від 25 до 50 років, зазвичай із досвідом у технічній підтримці чи адмініструванні.

Адміністратори мають високий рівень технічної грамотності, здатність працювати з аналітичними інструментами, базами даних і панелями управління. Вони переважно використовують настільні комп’ютери або ноутбуки для виконання складних адміністративних завдань. Їхні основні цілі – забезпечення безпеки та якості контенту на платформі, моніторинг активності користувачів і вирішення конфліктних ситуацій, а також аналіз фінансових і користувацьких даних для оптимізації роботи системи. Для цього їм потрібен доступ до панелі адміністратора з функціями модерації, аналітики та управління користувачами, можливість блокування акаунтів і редагування категорій послуг, звіти про активність користувачів і фінансові операції, а також інструменти для швидкого реагування на скарги чи порушення.

* 1. Загальні обмеження

Програмна система має наступні обмеження, які визначають рамки її функціонування та використання:

1. необхідність Інтернет-підключення так як система працює лише за наявності стабільного підключення до Інтернету, оскільки всі дані зберігаються та обробляються на сервері;
2. обмеження для неавторизованих користувачів:
   1. неавторизовані користувачі не можуть створювати та відгукуватись на замовлення, переглядати профілі інших користувачів або брати участь у комунікації через чат;
   2. доступ до функціоналу профілю та панелі адміністратора можливий лише після успішної авторизації;
3. обмеження для користувачів:
   1. користувач може завантажити лише одне резюме у профіль, і його розмір не повинен перевищувати встановлений ліміт в 2 МБ, його формат має бути допустимим системою (PDF);
   2. фотографія профілю має відповідати вимогам до розміру та формату – не більше 2 МБ, формати – Jpg, Png та Jpeg;
   3. користувач не може приймати замовлення, якщо не заповнив професійну інформацію у профілі (перелік категорій послуг і резюме);
   4. користувач може залишити лише один відгук про виконавця або замовника для кожного завершеного замовлення;
   5. виконавець не може отримати оплату, доки замовлення не буде підтверджено як завершене замовником;
4. обмеження для адміністраторів:
   1. адміністратор не має доступу до приватних повідомлень у чаті між користувачами;
   2. адміністратор не може змінювати фінансові дані користувачів (історію транзакцій, баланс.
   3. Припущення й залежності

Система базується на таких припущеннях щодо користувачів та середовища:

* користувачі мають комп'ютер або мобільний пристрій з сучасним веб-браузером, який підтримує HTML5 та CSS3 для коректної роботи веб-застосунку;
* система призначена для роботи на популярних операційних системах, таких як Windows, Linux та macOS;
* для доступу до функціоналу системи користувачі повинні мати стабільне підключення до Інтернету;
* користувачі мають власну електронну скриньку для реєстрації та комунікації в системі;
* користувачі володіють хоча б однією з мов, доступних в інтерфейсі системи.

Функціонування системи залежить від наступних умов:

* функціонал системи доступний лише після успішної реєстрації та авторизації користувача;
* замовники повинні мати достатню суму на своїй банківській картці для оплати замовлення;
* можливість залишати відгуки та оцінки залежить від завершення замовлення;
* редагування профілю та інші налаштування доступні лише для авторизованих користувачів;
* панель адміністратора доступна лише користувачам з відповідною роллю адміністратора.

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

3.1.1 Інтерфейс користувача

Інтерфейс програмної системи реалізовано як односторінковий вебзастосунок (SPA) з використанням Angular 19. Архітектура фронтенду побудована за принципами модульності та компонентного підходу:

* Основний код розміщений у директорії src/app з чітким розділенням на:
* @core - базові сервіси та інтерфейси
* @shared - перевикористовувані компонентиmodules - функціональні модулі системи
* shared-pages - спільні сторінки

Для реалізації візуальних елементів використовується бібліотека PrimeNG, що надає готові компоненти для побудови сучасного інтерфейсу. Система стилів побудована на SCSS з використанням змінних для забезпечення консистентності дизайну.

3.1.2 Апаратний інтерфейс

Мінімальні вимоги до апаратного забезпечення:

* Доступ до мережі Інтернет (WiFi або мобільний інтернет)
* Операційна система: Windows 10/11macOS 10.15+Linux (Ubuntu 20.04+, Debian 11+)
* Мобільні пристрої: iOS 14+, Android 8.0+
* Браузери: Google Chrome 90+Firefox 90+Safari 14+Microsoft Edge 90+

3.1.3 Програмний інтерфейс

Бекенд реалізований як мікросервісна архітектура на .NET 8 з використанням REST API. Основні компоненти:

* Doerly.Host - головний сервіс, що об'єднує всі модулі

Модульна структура:

* Doerly.Module.Authorization - автентифікація та авторизація
* Doerly.Module.Profile - управління профілями
* Doerly.Module.Order - обробка замовлень
* Doerly.Module.Catalog - управління каталогом
* Doerly.Module.Statistics - збір та аналіз статистики
* Doerly.Module.Notification - система сповіщень

API захищене за допомогою JWT-автентифікації з підтримкою ролей та політик доступу. Для взаємодії з базою даних використовується Entity Framework Core.

3.1.4 Комунікаційний протокол

Система використовує HTTP/HTTPS для комунікації між компонентами. Реалізовано: REST API для основних операцій, WebSocket для real-time сповіщень

3.1.5 Обмеження пам’яті

Система реалізована з урахуванням оптимального використання ресурсів на обох рівнях - клієнтському та серверному. На серверній стороні архітектура побудована таким чином, що дозволяє ефективно розподіляти навантаження між різними модулями. Основний сервіс (Doerly.Host) виконує роль координатора, що дозволяє зменшити навантаження на окремі модулі.

Для серверної частини встановлено наступні вимоги:

* Мінімальні вимоги: 4 ГБ оперативної пам'яті, що достатньо для базової роботи системи та обробки одночасних запитів від 5-10 користувачів
* Рекомендовані вимоги: 8 ГБ оперативної пам'яті, що забезпечує комфортну роботу з системою для 20-30 одночасних користувачів
* Продукційне середовище: 16+ ГБ оперативної пам'яті для забезпечення стабільної роботи при високому навантаженні

Клієнтська частина, будучи веб-додатком на Angular, оптимізована для роботи в сучасних браузерах. Система використовує ряд механізмів для оптимізації використання пам'яті:

* Ліниве завантаження модулів (lazy loading), що дозволяє завантажувати компоненти тільки при необхідності
* Кешування даних на клієнтській стороні для зменшення кількості запитів до сервера
* Оптимізація статичних ресурсів через компресію та мініфікаціюВикористання віртуального скролінгу для роботи з великими списками даних

3.1.6 Операції

Система реалізує повний набір CRUD-операцій через REST API, що забезпечує гнучку взаємодію з даними. Кожна операція реалізована з урахуванням безпеки та валідації даних.

* GET-операції дозволяють отримувати списки сутностей з підтримкою пагінації та фільтрації, детальну інформацію про конкретну сутність, агреговані дані та статистику. Всі GET-запити підтримують параметри для налаштування формату відповіді.
* POST-операції використовуються для створення нових сутностей з валідацією вхідних даних, масового створення сутностей через batch-операції, завантаження файлів з перевіркою типів та розмірів, а також виконання складних операцій, що потребують обробки на сервері.
* PUT та PATCH операції забезпечують повне та часткове оновлення сутностей відповідно. Вони включають валідацію змін перед застосуванням та використовують оптимістичне блокування для запобігання конфліктів.
* DELETE-операції реалізують м'яке видалення з можливістю відновлення, каскадне видалення пов'язаних сутностей, перевірку прав доступу та логування операцій видалення.

Додатково система підтримує асинхронні операції для довгих процесів, систему черг для обробки масових операцій, механізм відстеження статусу виконання операцій та підтримку транзакцій для складних операцій.

3.2 Властивості програмного продукту

Програмний продукт побудований із використанням сучасного технологічного стеку, що забезпечує швидкість розробки, масштабованість і зручність супроводу. Основний акцент зроблено на модульність і гнучкість: система підтримує генерацію програмних проєктів на основі шаблонів або розмітки, динамічне створення компонентів (моделі, сервіси, контролери), а також можливість розширення за рахунок підключення нових фреймворків. Архітектура побудована за принципами чистого коду та з використанням REST API, що гарантує простоту інтеграції з іншими системами. Продукт орієнтований як на індивідуальне використання, так і на командну розробку, з перспективою подальшого розширення функціональності.

[3.3 Атрибути програмного продукту](#_heading=h.k3kt08af1nm3)

3.3.1 Надійність

Архітектура побудована за принципами чистого коду та з використанням REST API, що гарантує простоту інтеграції з іншими системами. Продукт орієнтований як на індивідуальне використання, так і на командну розробку, з перспективою подальшого розширення функціональності.

3.3.2 Масштабованість

Архітектура системи дозволяє легко масштабувати її як горизонтально, так і вертикально. Модульна структура забезпечує незалежність модулів з чітко визначеними інтерфейсами, можливість додавання нових модулів без впливу на існуючі та мікросервісну архітектуру для незалежного масштабування компонентів.

Горизонтальне масштабування досягається через підтримку балансування навантаження, реплікацію бази даних, кешування на різних рівнях та контейнеризацію для легкого розгортання. Вертикальне масштабування забезпечується оптимізацією використання ресурсів, можливістю збільшення потужності окремих компонентів та ефективним використанням багатопроцесорних систем.

Програмна система повинна мати можливість відновлення з негативного стану та бути доступною для користувачів 99% часу.

3.3.3 Відкрита до відлагодження

Система надає розробникам та адміністраторам потужні інструменти для відлагодження та моніторингу. Логування забезпечує структуровані логи з контекстом виконання, різні рівні деталізації для різних середовищ, централізований збір логів та аналітичні інструменти для пошуку в логах.

Трейсинг дозволяє відстежувати запити через систему, вимірювати час виконання операцій, аналізувати залежності між компонентами та візуалізувати шляхи виконання. Система збирає метрики використання ресурсів, моніторить продуктивність, відправляє алерти при досягненні критичних значень та надає графіки та діаграми для аналізу.

Документація включає Swagger для API, технічну документацію, приклади використання та опис архітектури.

3.3.4 Адаптивність

Інтерфейс системи розроблений з урахуванням різних пристроїв та розмірів екрану. Responsive дизайн реалізований через використання CSS Grid та Flexbox, адаптивні компоненти PrimeNG, медіа-запити для різних розмірів екрану та оптимізацію для різних пристроїв.

Mobile-first підхід передбачає пріоритет мобільної версії при розробці, оптимізовані компоненти для сенсорних екранів, адаптивні форми та елементи керування, а також оптимізацію продуктивності для мобільних пристроїв.

3.3.5 Швидкодія

Система оптимізована для забезпечення високої продуктивності. Кешування реалізовано на багатьох рівнях: база даних, сервер, клієнт. Система кешує запити, статичні ресурси та забезпечує інтелектуальне оновлення кешу

.Оптимізація запитів включає ефективні SQL-запити, індексацію бази даних, оптимізацію ORM та асинхронну обробку. Компресія застосовується для Gzip/Brotli стиснення відповідей, мініфікації статичних ресурсів, оптимізації зображень та кешування на рівні CDN.

3.3.6 Підтримка багатьох браузерів

Програмна система має бути доступною та працювати однаковим чином на всіх сучасних браузерах.

3.3.7 Безпека даних

Програмна система повинна зберігати лише необхідну для ідентифікації користувача приватну інформацію, мінімізуючи обсяг персональних даних. Уся критична інформація зберігається в зашифрованому або захешованому вигляді. Для автентифікації та авторизації користувачів застосовується JWT (JSON Web Token), що дозволяє реалізувати безпечну безстейтову взаємодію між клієнтом і сервером, а також ефективно контролювати доступ до ресурсів системи.

3.3.8 Робота з одночасним доступом

Програмна система повинна мати можливість обробки паралельних запитів без впливу запитів на результат один одного, якщо це не пряма дія спрямована на зміну стану системи.

3.3.9 Захист від помилок

Частини інтерфейсу програмної системи повинні бути доступними лише при достатній кількості введених даних для запобігання подальших помилок при обробці цих даних.

3.4 Вимоги бази даних

Система використовує Microsoft SQL Server з розширеними можливостями. ORM Entity Framework Core забезпечує code-first підхід до розробки, автоматичні міграції, ліниве завантаження та оптимізовані запити.

Оптимізація бази даних включає ефективні індекси, партиціонування даних, кешування запитів та оптимізацію схеми. Висока доступність забезпечується через реплікацію даних, автоматичне резервне копіювання, відновлення після збоїв та моніторинг стану.

Безпека бази даних реалізована через шифрування даних, аудит доступу, контроль прав та захист від SQL-ін'єкцій.