**Software Requirements Specification**

**for**

**Planned Social Network**

**Version 1.0 approved**

**Prepared by Biletskyi Danylo**

**SocialB**

**14.06.2025**

**Table of Contents**

1. Introduction 4

1.1 Purpose 4

1.2 Document Conventions 4

1.3 Intended Audience and Reading Suggestions 4

1.4 Project Scope 5

1.5 References 6

2. Overall Description 7

2.1 Product Perspective 7

2.2 Product Features 7

2.3 User Classes and Characteristics 8

2.4 Operating Environment 8

2.5 Design and Implementation Constraints 9

2.6 User Documentation 9

2.7 Assumptions and Dependencies 10

3. System Features 10

3.1 Авторизація та автентифікація 10

3.2 Створення, редагування, видалення та перегляд публікацій 12

3.3 Створення запланованих публікацій 14

3.4 Можливість підписуватися та мати підписників 15

3.5 Можливість коментувати публікації 16

3.6 Можливість ставити вподобайки до публікацій 18

3.7 Адміністративні функції 19

4. External Interface Requirements 21

4.1 User Interfaces 21

4.2 Hardware Interfaces 22

4.3 Software Interfaces 22

4.4 Communications Interfaces 22

5. Other Nonfunctional Requirements 23

5.1 Performance Requirements 23

5.2 Safety Requirements 23

5.3 Security Requirements 23

5.4 Software Quality Attributes 24

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**1 Introduction**

**1.1 Purpose**

Основна ціль соціальної мережі з системою запланованих публікацій це надати зручну для використання платформу, для авторів контенту та звичайних глядачів, котра дозволить їм із задоволенням проводити час використовуючи дану програмну систему, читаючи публікації та спілкуючись у коментарях. Зазвичай, система повідомлень користувачів про нову публікацію є виснажливою, через що, дуже часто функціонал повідомлень вимикається користувачами. Цей проєкт створений для того, аби вирішити цю проблему.

**1.2 Document Conventions**

Для написання даного SRS документу були використані правила оформлення зазначені у ДСТУ 3008-2015.

**1.3 Intended Audience and Reading Suggestions**

Цей документ призначений в першу чергу для розробників. Завдяки цьому документу розробники зможуть розуміти, який саме проект необхідно розробляти і які вимоги необхідно виконувати. Цей документ також призначений для тестувальників. Він може використовуватися ними для побудови тестів різних типів та для перевірки відповідності поставленим вимогам. Наступна група людей, для якої призначений даний документ, це бізнес партнери. Цей документ допоможе їм сформувати певні очікування від проєкту, а також використовувати його для введення нових вимог у відповідності до змін ринку. Специфікація програмного забезпечення буде корисна менеджерам проєкту, а також керівникам окремих команд розробників. Цей документ допоможе їм доносити необхідну інформацію розробникам, а також взаємодіяти з бізнес партнерами щодо уточнення вимог, їх зміни тощо.

Ну і на кінець, цей документ буде корисний користувачам. Завдяки цьому документу, користувачі зможуть зрозуміти, на що саме вони можуть очікувати від проєкту і як він буде реалізований.

Надалі у документі будуть зазначатися наступні дані:

* розділ 2 – загальний опис проєкту, котрий призначений для бізнес партнерів та користувачів, що дозволить їм зрозуміти суть, призначення та скласти враження про функціонал проєкту;
* розділи 3-4 – загальний перелік функціональних вимог, призначений для тестувальників та розробників, для розуміння проєкту та його необхідної функціональності;
* розділ 5 – перелік нефункціональних вимог необхідних тестувальникам для перевірки їх відповідності проєктом.

**1.4 Project Scope**

Соціальна мережа з системою запланованих публікацій націлена на покращення відчуттів користувачів від проводження часу у соціальній мережі. Проєкт створено у вигляді веб-застосунку для надання можливості використання на якомога більшій кількості платформ. Користувачам буде наданий широкий вибір функціональності, включаючи створення публікацій із зображеннями і без, запланованих і звичайних. Зокрема, користувачі матимуть можливість коментувати публікації та ставити «вподобайку» на публікацію, котра їм до вподоби. Для зручного використання соціальної мережі буде впроваджена система підписок, що дозволить людям слідкувати за обліковими записами, котрі їм подобаються.

В майбутньому буде введена можливість створювати «дерева коментарів», відповідаючи людям на їх коментарі. Також, буде введена функціональність приватних переписок, що дозволить користувачам спілкуватись між собою не виходячи із соціальної мережі. Планується зокрема широкий вибір налаштування зовнішнього вигляду системи користувачем, такої як колір фону, кнопок, тощо.

**1.5 References**

ДСТУ 3008-2015:<https://dl.nure.ua/mod/page/view.php?id=661258>

Документація ASP.NET Core: <https://learn.microsoft.com/uk-ua/aspnet/core/fundamentals/minimal-apis/overview?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website>

Документація Angular: <https://angular.dev/overview>

Сервіс збереження медіа-контенту Cloudinary: <https://cloudinary.com>

**2 Overall Description**

**2.1 Product Perspective**

Розроблюване програмне забезпечення є новим, самодостатнім веб-застосунком, який не є частиною існуючого програмного комплексу, але може бути інтегрованим з іншими системами через відкриті API. Цей застосунок не є продовженням чи оновленням вже існуючих продуктів, а створюється з нуля відповідно до сучасних потреб користувачів соціальних мереж.

Застосунок має модульну архітектуру, що дозволяє легко розширювати функціональність у майбутньому. Уся система базується на веб-технологіях та може бути використана через браузер на різних пристроях (десктоп, мобільні телефони, планшети).

Основними складовими соціальної мережі є:

* сервер – відповідає за обробку бізнес-логіки, управління базами даних, автентифікацію, запити до API;
* інтерфейс користувача – реалізований за допомогою фреймворку Angular, він надає доступ до всіх основних функцій соціальної мережі: реєстрація / вхід, перегляд публікацій, створення публікацій, заплановані публікації, коментування, лайки тощо;
* база даних PostgreSQL – зберігає основні дані користувачів, публікацій, коментарів, посилання на зображення тощо;
* база даних Neo4j – зберігає дані про підписки користувачів одне на одного за допомогою графової структури.

**2.2 Product Features**

Соціальна мережа із системою запланованих публікацій має наступний функціонал:

* вхід та реєстрація нового облікового запису з можливістю встановлення власного зображення профілю;
* створення, редагування, видалення та перегляд публікацій;
* створення запланованих публікацій, котрі стануть доступними іншим користувачам у певну дату та про які будуть сповіщені підписники за допомогою електронної розсилки;
* можливість підписуватися та мати підписників;
* можливість коментувати різні публікації;
* можливість ставити вподобайки на різні публікації що сподобалися;
* можливість блокувати користувачів адміністрацією, видаляти їх публікації або коментарі;
* можливість використовувати фотографії у публікаціях (до 4 фотографій).

**2.3 User Classes and Characteristics**

В даній програмній системі користувачі поділяються в основному на три основні види.

Неавторизований користувач – користувач котрий не має облікового запису, або ще не увійшов до нього. Такий користувач має дуже обмежений доступний функціонал і може лише увійти до облікового запису або створити новий.

Авторизований користувач, або просто користувач – має доступ до всього основного функціоналу системи. Такий користувач може створювати звичайні або заплановані публікації, писати коментарі, ставити вподобайки на публікації тощо. Такими користувачами є більша частина всієї клієнтської бази.

Адміністратори – користувачі котрі мають розширені повноваження. Таким користувачам доступний весь основний функціонал, а також додатковий, як от наприклад видалення чужих публікацій, коментарів, або блокування чи розблокування користувацьких записів загалом.

**2.4 Operating Environment**

Так як соціальна мережа це веб застосунок, вона може працювати на будь-якому пристрої з браузером, наприклад комп’ютери, ноутбуки, смартфони, планшети, тощо. Операційна система також не впливає на працездатність застосунку, тому є підтримка Windows, Linux, MacOS та Android. Клієнт соціальної мережі підтримується всіма сучасними браузерами, такими як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari тощо. Варто помітити, що має бути забезпечене стабільне інтернет з’єднання для роботи соціальної мережі.

**2.5 Design and Implementation Constraints**

У процесі розробки даної програмної системи існує ряд обмежень, які впливають на вибір технологій, архітектури та інструментів розробки. До основних обмежень належать:

* розробка клієнтської частини з використанням Angular 17;
* розробка серверної частини з використанням ASP.NET Core та .NET 8;
* використання реляційної бази даних PostgreSQL для зберігання основних даних про користувачів, публікації, коментарі тощо;
* використання графової бази даних Neo4j для зберігання підписок користувачів один на одного.

Зокрема, програмна система повинна бути спроектована з підтримкою усіма компонентами системи запуску на серверних середовищах Linux та Windows.

Клієнтська частина має взаємодіяти з серверною за допомогою REST API, що накладає певну структуру на формат запитів і відповідей.

Весь код має відповідати загальноприйнятим стандартам і стилістичним вимогам для відповідних мов програмування. Також рекомендоване застосування практик чистого коду та принципів SOLID для забезпечення можливостей простої підтримки системи в майбутньому.

**2.6 User Documentation**

Відповідні користувацькі інструкції будуть сформовані та надаватимуться користувачеві на основних етапах взаємодії. Система зв’язку з розробниками дозволить користувачам звертатися з приводу проблем, що виникають або питань щодо роботи певного функціоналу.

**2.7 Assumptions and Dependencies**

AS-1: Припускається, що користувачі системи матимуть постійний доступ до стабільного інтернет-з’єднання.

AS-2: Припускається, що всі сучасні браузери підтримуватимуть необхідні можливості HTML5, CSS3 та JavaScript + TypeScript.

AS-3: Припускається, що користувачі мають базовий досвід з роботи у соціальних мережах, з комп’ютером та із завантаженням фото.

AS-4: Інтерфейс застосунку розроблений англійською, тому припускається що користувачі мають достатньо високий рівень знань англійської.

DE-1: Система залежить від бази даних PostgreSQL для збереження основних структурованих даних. Будь-які зміни у підтримці PostgreSQL можуть вплинути на роботу проєкту.

DE-2: Система залежить від графової бази даних Neo4j для реалізації підписок між користувачами. Зміни у функціональності Neo4j можуть вимагати відповідних адаптацій у будові соціальної мережі.

DE-3: Серверна частина системи залежить від .NET 8 і всіх його компонентів.

DE-4: Клієнтська частина системи залежить від Angular 17 та бібліотек що використовуються в ній.

DE-5: Система збереження фото залежить від хмарного сервісу Cloudinary. Будь-які зміни в хмарному сервісі можуть вплинути на працездатність збереження фото-контенту.

**3 System Features**

**3.1 Авторизація та автентифікація**

3.1.1 Description and Priority

Функціональність авторизації, а саме створення нового облікового запису та вхід до вже існуючого мають високий пріоритет у соціальній мережі. Ця функціональність має бути виконана за допомогою підходу з JSON Web Tokens, що забезпечить працездатність, ефективність та безпеку системи авторизації та автентифікації. Для використання цієї функціональності, при реєстрації користувач має надати своє користувацьке ім’я, адресу електронної пошти та пароль. За бажанням, користувач може додати також зображення профілю. Для входу користувач має надати лише адресу електронної пошти та пароль.

3.1.2 Stimulus/Response Sequences

Якщо реєстрація була не вдала, користувач побачить повідомлення про помилку, а також як її виправити у правому верхньому кутку. У випадку вдалої реєстрації, користувач буде перенаправлений на свою сторінку профіля. Під час входу до акаунту, як і під час реєстрації, у разі помилки користувач побачить причину помилки та як її виправити у правому верхньому кутку. Якщо вхід був успішним, користувач буде перенаправлений на сторінку профіля.

3.1.3 Functional Requirements

REQ-1: Для автентифікації використовується підтримка ASP.NET Core для JWT автентифікації.

REQ-2: Користувач повинен мати змогу зареєструвати новий обліковий запис, надаючи унікальну адресу електронної пошти, користувацьке ім'я та пароль.

REQ-3: Після успішної реєстрації, користувач повинен отримати JWT токен, котрий зберігатиметься у браузері за допомогою localStorage.

REQ-4: Під час входу користувач вводить електронну пошту та пароль. У разі успішного входу система повинна видавати новий JWT-токен.

REQ-5: Якщо користувач вводить неправильну електронну пошту або пароль, система повинна повертати відповідне повідомлення про помилку з поясненням.

REQ-6: Користувач має мати змогу добровільно вийти з облікового запису, що видаляє JWT токен з клієнта.

REQ-7: Користувачі без токена або з недійсним токеном не мають доступу до захищених маршрутів або API.

**3.2 Створення, редагування, видалення та перегляд публікацій**

3.2.1 Description and Priority

Функціональність щодо взаємодії з публікаціями має високу пріоритетність. Зареєстрований користувач має мати змогу без проблем створити публікацію за допомогою введення тексту до спеціального поля, а також за бажанням, додавання до 4-х зображень на вибір. Після створення публікації, користувач має мати змогу до редагування створеної публікації або її видалення. До редагування не входить взаємодія з фото-контентом, тобто після додавання фото до публікації, користувач не має права видалити її чи редагувати. Єдиним шляхом є видалення публікації та заново створити вже з редагованим фото. Користувач має мати можливість переглядати чужі публікації, однак не має мати доступ до видалення чи редагування таких публікацій.

3.2.2 Stimulus/Response Sequences

Якщо створення публікації пройшло успішно, користувач перенаправляється до свого профілю, де він може побачити новостворений пост. Якщо ж під час створення публікації виникла якась проблема, користувач може побачити повідомлення про помилку справа зверху. Помилки під час додавання зображень до публікації також відображаються у правому верхньому куті. Видалення публікації має відображатися миттєво, після підтвердження видалення за допомогою діалогового вікна. Редагування публікації має перенаправляти користувача до сторінки профілю після вдалої зміни.

3.2.3 Functional Requirements

REQ-1: Система повинна дозволяти авторизованому користувачу створити нову публікацію, ввівши текст у відповідне поле.

REQ-2: При створенні публікації користувач може додати до 4-х зображень у форматах JPEG або PNG.

REQ-3: Система повинна відображати повідомлення про помилку, якщо користувач намагається додати більше 4 зображень або файл у недопустимому форматі.

REQ-4: Після успішного створення публікації користувач має бути перенаправлений на сторінку свого профілю, де публікація буде відображатися.

REQ-5: Авторизований користувач повинен мати можливість редагувати текст своєї публікації.

REQ-6: Користувач не має можливості редагувати чи видаляти зображення, прикріплені до публікації. Для цього потрібно видалити всю публікацію.

REQ-7: Авторизований користувач повинен мати можливість видалити власну публікацію через підтверджувальне діалогове вікно.

REQ-8: Видалення публікації повинно миттєво оновлювати інтерфейс користувача (UI), прибираючи публікацію зі списку.

REQ-9: Користувачі повинні мати змогу переглядати публікації інших користувачів.

REQ-10: Користувач не має доступу до редагування або видалення публікацій інших користувачів.

REQ-11: Система повинна показувати повідомлення про помилку у правому верхньому куті у випадку невдалої спроби створення, редагування або видалення публікації.

**3.3 Створення запланованих публікацій**

3.3.1 Description and Priority

Функціональність щодо створення запланованих публікацій має високу пріоритетність. Заплановані публікації є підтипом публікацій, котрі мають певну заплановану дату та невидимі для всіх окрім автора, до настання запланованої дати. Після настання такої дати, всі користувачі, підписані на автора, отримують повідомлення про появу такої публікації.

3.3.2 Stimulus/Response Sequences

Якщо створення запланованої публікації пройшло успішно, користувач повертається до сторінки профілю де може побачити заплановану публікацію та дату її появи. Якщо створення пройшло не успішно, користувач бачить повідомлення про помилку у правому верхньому кутку.

3.3.3 Functional Requirements

REQ-1: Авторизований користувач повинен мати можливість створити заплановану публікацію, ввівши текст у відповідне поле.

REQ-2: Під час створення запланованої публікації користувач зобов’язаний вказати дату та час публікації в майбутньому. Дата у минулому або теперішньому часі не допускається.

REQ-3: До запланованої публікації можна додати до 4 зображень у форматі JPEG або PNG, так само як і до звичайної публікації.

REQ-4: Після настання запланованої дати, публікація автоматично стає видимою для всіх користувачів.

REQ-5: Після активації публікації, система повинна автоматично відправити email-сповіщення всім підписникам автора публікації.

REQ-6: Якщо під час створення запланованої публікації виникла помилка (наприклад, неправильний формат дати або відсутній текст), користувач повинен побачити повідомлення про помилку у правому верхньому кутку.

REQ-7: Після успішного створення запланованої публікації користувач має бути перенаправлений на сторінку профілю, де буде відображено нову публікацію з вказаною запланованою датою.

**3.4 Можливість підписуватися та мати підписників**

3.4.1 Description and Priority

Функціональність щодо реалізації системи підписок має менший пріоритет порівняно з попередніми. Користувачі мають мати можливість натиснути на кнопку біля профілю іншого користувача та підписатися на них. Список підписок та підписників має бути доступним користувачам при переході на сторінку профіля та після натискання на відповідні написи.

3.4.2 Stimulus/Response Sequences

Якщо процес слідування за іншим користувачем пройшов успішно, користувач побачить як кнопка «Follow» змінилася на «Unfollow». Якщо ж в ході процесу виникла помилка, повідомлення про неї з’явиться у правому верхньому кутку. Відкриття списку підписників та каналів на які підписаний користувач має перенаправити на сторінки відповідних списків та відобразити відповідних користувачів.

3.4.3 Functional Requirements

REQ-1: Авторизований користувач повинен мати можливість підписатися на іншого користувача, натиснувши кнопку «Follow» на його сторінці профілю.

REQ-2: Після успішної підписки кнопка «Follow» має змінюватися на «Unfollow».

REQ-3: Повторне натискання кнопки «Unfollow» повинно видалити підписку та повернути кнопку до стану «Follow».

REQ-4: Користувач повинен не мати можливості ні при яких умовах підписатися на самого себе.

REQ-5: Користувач має мати можливість переглядати список усіх користувачів, на яких він підписаний, натиснувши на відповідний розділ у своєму профілі.

REQ-6: Користувач має мати можливість переглядати список усіх своїх підписників через окрему вкладку у власному профілі.

REQ-7: Якщо під час підписки або відписки виникла помилка (наприклад, відсутнє з’єднання з сервером), система повинна повідомити користувача про помилку у правому верхньому кутку.

REQ-8: Усі дані про підписки повинні зберігатися у графовій базі даних Neo4j, що дозволяє ефективно здійснювати пошук та аналіз зв'язків між користувачами.

**3.5 Можливість коментувати публікації**

3.5.1 Description and Priority

Функціональність щодо коментування публікацій має менший пріоритет порівняно з попередніми. Користувачі мають мати можливість написати коментар до обраної публікації. Для цього має бути заповнене поле коментування. Власник коментаря має мати можливість видалити свій коментар. При виникненні помилки при створенні чи видаленні коментаря, повідомлення про помилку має з’явитися у правому верхньому кутку.

3.5.2 Stimulus/Response Sequences

Якщо процес створення коментаря закінчився успішно, він має з’явитися під публікацією. Видалення коментаря має відбуватися за допомогою діалогового вікна і показувати результат видалення прибираючи коментар.

3.5.3 Functional Requirements

REQ-1: Авторизований користувач повинен мати можливість залишити коментар під будь-якою публікацією, заповнивши текстове поле коментування.

REQ-2: Коментар має містити щонайменше один символ і не повинен перевищувати встановлений ліміт.

REQ-3: Після успішного додавання коментаря, він повинен з’явитися миттєво під відповідною публікацією без необхідності перезавантаження сторінки.

REQ-4: Користувач повинен мати можливість видалити лише власні коментарі, натиснувши на кнопку видалення, що відкриває діалогове вікно підтвердження.

REQ-5: Після підтвердження видалення, коментар має бути видалений з інтерфейсу користувача без потреби оновлення сторінки.

REQ-6: У випадку помилки при створенні або видаленні коментаря (наприклад, через проблеми з підключенням або сервером), система має показати повідомлення про помилку у правому верхньому кутку.

REQ-7: Коментарі повинні відображатися в хронологічному порядку — від найновіших до найстаріших.

**3.6 Можливість ставити вподобайки до публікацій**

3.6.1 Description and Priority

Функціональність щодо можливості лайкати публікації має невеликий пріоритет. Користувачі мають мати можливість натиснути на іконку сердечка під публікацією та при успішному виконанні, сердечко загориться червоним. Для прибирання позначки, користувач має повторно натиснути на сердечко.

3.6.2 Stimulus/Response Sequences

Якщо вподобайка була поставлена – сердечко загориться червоним, якщо прибрана – сердечко стане знову прозорим. Якщо під час цього процесу виникла якась помилка – у правій верхній частині сторінки з’явиться повідомлення про помилку.

3.6.3 Functional Requirements

REQ-1: Авторизований користувач має мати можливість поставити вподобайку до будь-якої публікації, натиснувши на іконку сердечка.

REQ-2: Якщо публікацію вже вподобано, натискання на іконку сердечка має скасувати вподобайку.

REQ-3: Стан іконки сердечка має змінюватися в реальному часі: червоний — вподобано, прозорий — не вподобано.

REQ-4: Кількість вподобайок має оновлюватися автоматично після кожної взаємодії.

REQ-5: У разі виникнення помилки (наприклад, відсутність з’єднання з сервером), система повинна показати повідомлення про помилку у правому верхньому куті.

REQ-6: Вподобайки мають зберігатися у базі даних з прив’язкою до ідентифікаторів користувача та публікації.

REQ-7: Один користувач може поставити лише одну вподобайку до кожної публікації.

**3.7 Адміністративні функції**

3.7.1 Description and Priority

Адміністративні функції мають високий пріоритет. Адміністратори мають мати можливість видаляти коментарі чи публікації. Зокрема адміністратори повинні мати доступ до блокування чи розблокування користувацьких облікових записів. При видаленні публікації чи коментаря, має накоплюватись критичне значення, при наборі якого в результаті користувач буде заблокований системою автоматично.

3.7.2 Stimulus/Response Sequences

Якщо адміністратор видалив публікацію чи коментар то видалення має відбутися моментально. Якщо виникла помилка при видаленні, повинно з’явитися повідомлення у правому верхньому кутку. При блокуванні облікового запису користувача, мають пройти відповідні візуальні зміни у його профілі. При розблокуванні акаунту, ці зміни мають скасуватися.

3.7.3 Functional Requirements

REQ-1: Адміністратори повинні мати можливість видаляти будь-яку публікацію незалежно від її автора.

REQ-2: Адміністратори повинні мати можливість видаляти будь-який коментар незалежно від автора.

REQ-3: Після кожного видалення публікації або коментаря адміністратором має оновлюватися лічильник порушень, прив'язаний до користувача.

REQ-4: При досягненні визначеного порогового значення лічильника порушень, обліковий запис користувача має автоматично блокуватися системою.

REQ-5: Адміністратори повинні мати можливість вручну блокувати обліковий запис користувача, незалежно від накопичених порушень.

REQ-6: Адміністратори повинні мати можливість розблокувати раніше заблокований обліковий запис.

REQ-7: У разі блокування облікового запису профіль користувача має змінювати свій вигляд.

REQ-8: Після розблокування облікового запису мають бути відновлені стандартні функції та вигляд профілю користувача.

REQ-9: У випадку помилки при виконанні адміністративної дії (видалення, блокування, розблокування), повинно з’явитися повідомлення у правому верхньому кутку.

REQ-10: Усі адміністративні дії повинні бути зафіксовані у системному журналі.

**4 External Interface Requirements**

**4.1 User Interfaces**

UI-1:

Повинна бути головна сторінка соціальної мережі із вказаною основною інформацією та посиланнями.

UI-2:

Повинна бути навігаційна панель, котра дозволить переміщати користувача до сторінки входу, реєстрації, чи інших сторінок системи.

UI-3:

Повинна бути сторінка входу та реєстрації користувача з відповідними формами для заповнення.

UI-4:

Повинна бути сторінка профілю користувача, де вказані такі дані як: ім’я, дата реєстрації, кількість підписників та користувачів, на які підписаний власник. Також знизу від головного профіля, має бути список усіх публікацій користувача.

UI-5:

Має бути форма створення публікацій та вибору типу, а також кнопки для завантаження зображень.

UI-6:

Має бути сторінка публікації, де користувач може глянути на публікацію детальніше та на коментарі під нею.

UI-7:

Має бути форма для створення коментарів під публікацією.

UI-8:

Має бути відповідний інтерфейс для зміни даних користувача таких як ім’я та фото.

UI-9:

Повинен існувати інтерфейс для адміністративної панелі.

**4.2 Hardware Interfaces**

Цей програмний продукт є соціальною мережею, яка працює як веб-застосунок, тому не має прямої взаємодії з апаратними компонентами системи за межами стандартних пристроїв вводу/виводу.

**4.3 Software Interfaces**

Цей програмний продукт має наступні програмні інтерфейси:

* браузери - Mozilla Firefox, Google Chrome, Opera, Microsoft Edge, Safari;
* операційні системи – windows 7+, Mac OS, Linux;
* реляційна база даних - PostgreSQL 13+;
* графова база даних - Neo4j 5.0+;
* бібліотека сервісу збереження фото-контенту – Cloudinary 1.27+.

**4.4 Communications Interfaces**

Соціальна мережа використовуватиме протокол HTTP для взаємодії клієнту та серверу. Для взаємодії з базою даних PostgreSQL використовуватимуться запити SQL. Для взаємодії з базою даних Neo4j використовуватиметься тип зв’язку Bolt.

**5 Other Nonfunctional Requirements**

**5.1 Performance Requirements**

Дана програмна система має суворі вимоги щодо продуктивності, адже затримки роботи соціальної мережі призведуть до швидкої втрати користувацької бази. Цими вимогами є:

* час відповіді на основні запити до серверу не має перевищувати 2 секунди при стабільному інтернет-з’єднанні;
* сторінка профілю користувача повинна завантажуватися не більше ніж за 1.5 секунди у 90% випадків;
* система має бути здатною до масштабування та має виконувати свою роботу без переривань від користувачів;
* операції з високою частотою використання повинні оброблятися та відображатися в інтерфейсі протягом 500 мілісекунд.

**5.2 Safety Requirements**

Вимоги до цілісності цього програмного продукту виглядають наступним чином:

* дії, що можуть привести до втрати даних, мають бути підтверджені користувачем за допомогою діалогового вікна;
* дані користувачів не мають бути змінені або видалені без їх згоди або запиту, однак це можливо лише при порушенні користувачами правил соціальної мережі;
* у разі виникнення системної помилки, користувач має побачити повідомлення про рекомендовані дії;
* продукт має відповідати політикам безпеки даних, які прийняті в ЄС.

**5.3 Security Requirements**

Вимоги до безпеки цього програмного продукту виглядають наступним чином:

* дані, що передаються між клієнтом і сервером мають бути зашифровані через HTTPS запити;
* автентифікація користувачів має виконуватись за допомогою JWT токенів;
* паролі користувачів повинні зберігатися у зашифрованому вигляді із застосуванням надійних алгоритмів шифрування;
* має бути реалізована можливість видалення облікового запису та всіх пов’язаних з ним даних на вимогу користувача.

**5.4 Software Quality Attributes**

Вимоги до якості атрибутів програмного забезпечення цього програмного продукту виглядають наступним чином:

* інтерфейс користувача має бути інтуїтивно зрозумілим, з логічною структурою та без наявних помилок;
* система повинна бути розроблена з дотриманням принципів розділення обов’язків з використанням шаблонів проектування для спрощення підтримки;
* сервіс має бути доступний не менше ніж 99% часу на місяць, з урахуванням технічних пауз і оновлень;
* клієнтська частина має працювати на всіх сучасних браузерах;
* система повинна підтримувати можливість масштабування для додавання нових функцій без переписування основного ядра