**Vision and Scope Document**

**for**

**Web system for organization and management of a city library**

**Version 1.0**

**approved**

**Prepared by**

**Vladyslav Riabko**

**NURE**

**06.06.2025**

## Постановка мети

Метою роботи є розробка веб-системи, яка забезпечує автоматизацію процесів міської бібліотеки та оптимізує взаємодію між користувачами, книжковим фондом і подіями. Програмна система дозволяє реалізувати функціонал для реєстрації читачів, управління книжками, бронювання, проведення бібліотечних заходів, а також адміністрування системи з боку персоналу.

У процесі реалізації веб-застосунку були використані сучасні інструменти та технології, що забезпечують масштабованість, безпеку й зручність користування.

Клієнтська частина системи. Інтерфейс користувача реалізовано з використанням JavaScript і бібліотеки React.js. Для обміну даними із серверною частиною використовуються HTTP-запити через бібліотеку Axios. Для прискорення створення інтерфейсу та підвищення його доступності були використані TailwindCSS та Material UI – бібліотеки з широким набором адаптивних і готових компонентів.

Серверна частина системи. Для побудови серверної логіки було обрано мову Python і фреймворк Django REST Framework, який базується на Django і суттєво полегшує розробку RESTful API. Архітектура реалізована за принципами шаблону MVC, доповненого механізмами розмежування прав доступу та обробки запитів.

Для реалізації всієї системи обрано трирівневу архітектуру, яка складається з клієнтської частини (frontend), серверної логіки (backend) і бази даних (PostgreSQL).

## Загальний опис системи

Програмна система являє собою клієнт-серверний веб-застосунок для автоматизації роботи міської бібліотеки. Вона дозволяє здійснювати повний цикл керування бібліотечним процесом – від перегляду книжкового каталогу до адміністрування користувачів і подій. Система розроблена для зручної роботи як читачів, так і персоналу бібліотеки: бібліотекарів та адміністраторів.

Головна сторінка веб-додатку слугує для ознайомлення користувачів із бібліотечним каталогом. Тут представлено перелік доступних книг, пошук за автором або назвою, а також розділ із майбутніми заходами. Гості можуть переглядати загальнодоступну інформацію, а зареєстровані читачі — створювати бронювання, переглядати активні позичання, історію позичань і редагувати профіль.

Інтерфейс є інтуїтивно зрозумілим і доступним. Функціонал реалізовано за допомогою React.js із використанням бібліотеки TailwindCSS, що забезпечує сучасний та зручний вигляд сайту. Приклади інтерфейсу для читачів наведено на рисунках 1 і 2.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1 – Головна сторінка з відображенням каталогу книг

Изображение выглядит как текст, снимок экрана, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 – Сторінка перегляду книги

Бібліотекар має доступ до розширеного функціоналу: він може додавати нові книги, редагувати інформацію про наявні, оновлювати події, а також переглядати профілі користувачів і їхню історію бронювань.

Зокрема, реалізовано сторінку додавання книги, яка включає заповнення форми з назвою, автором, жанром, роком видання, описом і завантаженням обкладинки. Аналогічно реалізовано сторінку додавання подій. Усі операції виконуються без перезавантаження сторінки завдяки використанню Axios-запитів.

Крім того, бібліотекар може оформлювати офлайн-бронювання, що дозволяє користувачам не входити в систему щоразу, а просто звернутися до бібліотекаря для оформлення бронювання вручну. Приклади інтерфейсу для бібліотекаря наведено на рисунках 3 та 4.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3 – Сторінка додавання нової книги

Изображение выглядит как текст, снимок экрана, веб-страница, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 – Сторінка з переліком активних бронювань

Адміністратор має повний доступ до керування всією системою та володіє розширеними правами. Він може виконувати всі дії, доступні бібліотекарю, за винятком реєстрації самого себе та офлайн-бронювань, які адміністратору не потрібні за логікою його ролі. Додатково адміністратор має можливість змінювати глобальні параметри системи, такі як максимальна тривалість позичання, час початку роботи та тривалість робочого дня у будні й вихідні. Зміни набирають чинності миттєво після збереження. Інтерфейс для редагування цих налаштувань представлено на рисунку 5.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 5 – Сторінка редагування системних параметрів

Серверну частину системи реалізовано на основі Django та Django REST Framework. Вона обробляє всі запити з клієнта, забезпечує автентифікацію користувачів, керування ролями, обробку книжкового каталогу, бронювань і подій. Усі дані зберігаються в базі даних PostgreSQL, що гарантує цілісність і надійність обробки.

Особливістю системи є реалізація чіткої структури ендпоінтів, що дозволяє клієнтській частині отримувати всю необхідну інформацію через REST API. Наприклад, для отримання повного списку книг використовується ендпоінт /api/books/list, який повертає серіалізований перелік книг разом з усіма їхніми даними. Приклад праці єндпоінту наведено на рисунку 6.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 6 – Приклад відповіді ендпоінта /api/books/ з переліком усіх книг у системі

Система значно спрощує роботу бібліотекарів і адміністраторів, підвищує ефективність обслуговування відвідувачів і робить міську бібліотеку більш доступною через онлайн-інтерфейс. Завдяки гнучкій архітектурі вона може бути легко адаптована до інших освітніх або культурних закладів.

## Основний функціонал системи

У системі реалізовано чіткий розподіл ролей користувачів. Кожна роль має власний набір можливостей і обмежень, що забезпечує контроль доступу до різних функцій системи. Всього передбачено три основні ролі: читач, бібліотекар та адміністратор. Для кожної ролі було створено Use Case-діаграми, які демонструють відповідний функціонал системи. Діаграми наведено на рисунках 7, 8 та 9.

Изображение выглядит как диаграмма, текст, рисунок, зарисовка

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 7 – UseCase діаграма для ролі Читача

Изображение выглядит как текст, диаграмма, рисунок, зарисовка

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 8 – UseCase діаграма для ролі Бібліотекаря

Изображение выглядит как текст, диаграмма, рисунок, зарисовка

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9 – UseCase діаграма для ролі Адміністратора

Ролі мають наступний функціонал:

Читач:

* реєстрація та вхів в систему;
* перегляд каталогу книг;
* пошук книг за автором, назвою або жанром;
* перегляд заходів;
* бронювання доступних книг;
* перегляд власного профілю та його редагування;
* перегляд історії позичань та активних бронювань;
* відстеження термінів повернення книг;
* вихід із облікового запису.

Бібліотекар:

* реєстрація та вхів в систему;
* перегляд та пошук книг у каталозі;
* керування книжковим фондом (додавання, редагування, видалення книг);
* реєстрація повернень книг;
* обробка офлайн-бронювань користувачів;
* керування подіями (створення, редагування, видалення заходів);
* перегляд контактних даних користувачів;
* перегляд історії позичань читачів.

Адміністратор:

* видалення користувачів;
* перегляд та пошук книг у каталозі;
* керування книжковим фондом (додавання, редагування, видалення книг);
* реєстрація повернень книг;
* обробка офлайн-бронювань користувачів;
* керування подіями (створення, редагування, видалення заходів);
* перегляд контактних даних користувачів;
* перегляд історії позичань читачів.
* налаштування глобальних параметрів системи (термін позичання, графік роботи бібліотеки);

## Загальні обмеження системи

Програмна система має наступні обмеження:

* система працює лише при наявності з’єднання з мережею Інтернет;
* кожен із користувачів має власну роль (читач, бібліотекар, адміністратор), яка визначає доступ до функціоналу;
* якщо обліковий запис користувача не створено, доступ до персоналізованого функціоналу (бронювання, історія, профіль) відсутній;
* браузер користувача має бути сумісним із сучасними веб-технологіями;
* читачі не можуть створювати або змінювати бібліотечний контент (книги, події) — це право мають лише бібліотекарі та адміністратори;
* лише користувачі з роллю "адміністратор" мають доступ до видалення облікових записів, зміни графіку роботи бібліотеки та зміни терміну бронювання за умовчуванням;
* система не підтримує офлайн-режим і не гарантує збереження даних при втраті з’єднання під час активних дій (наприклад, бронювання);
* облікові записи мають бути створені за правилами, встановленими політикою конфіденційності бібліотеки;
* система не призначена для зберігання сторонніх даних, окрім інформації, що прямо стосується бібліотечної діяльності.

## Припущення та залежності системи

Розглядаються наступні припущення та залежності:

* існує актуальна потреба в системі, яка дозволяє автоматизувати облік книг, бронювання та керування бібліотечними подіями у зручному цифровому форматі;
* працівники бібліотеки та відвідувачі готові користуватися веб-системою для взаємодії з бібліотекою в онлайн-режимі;
* пристрій користувача повинен бути підключений до мережі Інтернет для повноцінної роботи з системою;
* для отримання доступу до персонального функціоналу (бронювання, історія, профіль) користувач має увійти до облікового запису;
* при втраті з’єднання з мережею користувач втрачає доступ до основного функціоналу системи;
* інтерфейс системи створено з урахуванням принципів доступності та простоти – з інтуїтивною навігацією, розділенням функціоналу за ролями та адаптивним дизайном.