

DevLib

Software Requirements Specification

Version: 1.0

10.02.2024

Bystrytska Anastasiia
Lead Software Engineer

Історія Змін (Revision History)

Date	Description	Author	Comments
10.03.2024	Version 1	Anastasiia Bystrytska	First Revision

Затвердження Документу (Document Approval)

Специфікація вимог до програмного забезпечення була прийнята та схвалена наступними особами:

Signature	Printed Name	Title	Date
	A.Bystrytska	Lead Software Eng.	
	O.Ihnatiev	Back-End Developer	
	O. Kolesnyk	Back-End Developer	
	V. Kravchenko	Front-End Developer	
	O. Lavrovych	Front-End Developer	
	O. Chalyi	Back-End Developer	
	P. Berezin	Back-End Developer	
	Y. Kravtsov	Front-End Developer	

Таблиця контенту (Table of Contents)

ІСТОРІЯ ЗМІН	II
ЗАТВЕРДЖЕННЯ ДОКУМЕНТУ	II
1. Вступ	5
1.1 Мета	5
1.2 Обсяг	5
1.3 Визначення, скорочення та аббревіатури	5
1.4 Посилання	6
1.5 Огляд	6
2. ЗАГАЛЬНА ХАРАКТЕРИСТИКА	7
2.1 Перспектива продукту	7
2.2 Функції продукту	8
2.3 Характеристики користувача	9
2.4 Загальні обмеження	9
2.5 Припущення та залежності	10
3. ОСОБЛИВІ ВИМОГИ	2
3.1 Вимоги до зовнішнього інтерфейсу	10
3.1.1 Користувацькі інтерфейси	10
3.1.2 Інтерфейси апаратного забезпечення	10
3.1.3 Інтерфейси програмного забезпечення	10
3.1.4 Інтерфейси зв'язку	11
3.2 Функціональні вимоги	12
3.2.1 Реєстрація та авторизація	14
3.2.2 Каталог матеріалів	15
3.2.3 Завантаження книг у форматі PDF	16
3.2.4 Оцінки та відгуки	17
3.2.5 Можливість зберігати книги у “закладки”, переглядати історію завантажень і читання	17
3.2.6 Читання книг на сайті	18
3.2.7 Пошук книг, сортування та фільтрація результатів	19
3.2.8 Форум для будь-яких користувачів	21
3.2.9 Форум для зареєстрованих користувачів	22
3.3 Use Case Діаграми	22
3.3.1 Незареєстрований Користувач Use Case	22
3.3.2 Зареєстрований Користувач Use Case	23
3.3.3 Адміністратор Use Case	24
3.4 Класи та Об'єкти	24
3.4.1 <Клас User>	24
3.4.2 <Клас Catalog>	25
3.4.3 <Клас Book>	25
3.4.4 <Клас Forum>	25
3.4.5 <Клас Bookmark>	26
3.5 Нефункціональні Вимоги	27
3.5.1 Продуктивність	27
3.5.2 Надійність	27
3.5.3 Доступність	27
3.5.4 Безпека	27
3.5.5 Обслуговування	27
3.5.6 Портативність	27
3.6 Інверсні Вимоги	28
3.7 Обмеження Дизайну	28
3.8 Логічні вимоги до бази даних	28
3.9 Інші Вимоги	29
4. МОДЕЛІ АНАЛІЗУ	30

4.1 ДІАГРАМИ ПОСЛІДОВНОСТЕЙ	30
4.2 ДІАГРАМИ ПОТОКІВ ДАНИХ	34
4.3 ДІАГРАМИ СТАНУ-ПЕРЕХОДУ	36
5. CHANGE MANAGEMENT PROCESS	39
A. APPENDICES	40
A.1 APPENDIX 1	40
A.2 APPENDIX 2	40

1. Вступ (Introduction)

Документ Специфікації вимог до програмного забезпечення (SRS) для проекту **DevLib** надає детальний огляд функціональності та вимог до створення цього програмного продукту. Він призначений для забезпечення розробників усією необхідною інформацією для проектування та реалізації бібліотеки, що відповідає вимогам цього документа. Вимоги базуються на рекомендаціях стандарту IEEE для складання SRS.

1.1 Призначення (Purpose)

DevLib. Документ розрахований на команду розробників, які будуть працювати над проектом, включаючи інженерів-програмістів, архітекторів системи та тестувальників. Цей документ також може бути корисним для зацікавлених сторін, які хочуть зрозуміти функціональні можливості майбутнього продукту.

1.2 Сфера застосування (Scope)

DevLib – це веб-додаток, який функціонує як онлайн-бібліотека для програмістів, надаючи доступ до книг, документації та інших ресурсів для навчання й професійного розвитку в галузі програмування. Користувачі матимуть можливість переглядати та шукати необхідні матеріали, залишати оцінки та коментарі в стилі Reddit. Крім того, додаток включає довідник із категоріями для швидкої навігації, наприклад, для операторів або функцій, що значно полегшує доступ до специфічних розділів коду та документації.

Оцінка матеріалів здійснюватиметься за допомогою п'ятизіркової системи, де кожен користувач зможе залишити власну оцінку, сприяючи формуванню надійної бази ресурсів. DevLib не включає функції кодування чи середовища розробки, зосереджуючись на перегляді, оцінці та обговоренні документації. Продукт розроблено для максимальної зручності та ефективності користувачів, забезпечуючи швидкий доступ до найважливіших матеріалів, необхідних для професійного росту в програмуванні.

1.3 Визначення, скорочення та аббревіатури (Definitions, Acronyms, and Abbreviations)

У цьому підрозділі наведені визначення термінів, акронімів та аббревіатур, що використовуються в документі та необхідні для правильного тлумачення вимог до програмного забезпечення **DevLib**.

- **DevLib:** Онлайн-бібліотека для програмістів, яка надає доступ до книг та документації, а також можливості для обговорення та оцінки матеріалів.
- **SRS (Software Requirements Specification):** Специфікація вимог до програмного забезпечення, документ, що описує функціональні та нефункціональні вимоги до системи.
- **UI (User Interface):** Користувальницький інтерфейс, частина програми, з якою взаємодіє користувач.

- **API** (Application Programming Interface): Інтерфейс програмування додатків, набір функцій, що дозволяють програмістам взаємодіяти з програмним забезпеченням.
- **DBMS** (Database Management System): Система управління базами даних, програмне забезпечення, що дозволяє створювати та управляти базами даних.
- **React**: Бібліотека JavaScript для створення користувацьких інтерфейсів.
- **.NET**: Платформа для розробки програмного забезпечення від Microsoft, що використовується для створення різноманітних додатків.
- **AS** (Assumption) Припущення.
- **DE** (Dependencies) Залежності.
- **STD** (State-Transition Diagrams): Діаграма, що визначає зміну станів об'єкту у часі, одна з діаграм моделювання поведінки в UML. Подає об'єкт як автомат з теорії автоматів зі стандартизованими умовними позначеннями.
- **DFD** (Data Flow Diagram): Модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма потоків даних також може використовуватись для візуалізації процесів обробки даних (структурне проектування).
- **Use Case**: Графічне зображення можливої взаємодії користувача з системою. Діаграма визначає послідовність дій, які виконуються для виконання цієї цілі, і включає можливі варіанти розвитку подій.

1.4 Посилання (References)

У цьому підрозділі наведений перелік документів, на які посилаються в інших місцях специфікації вимог до програмного забезпечення (SRS) для проекту **DevLib**.

- IEEE Standard for Software Requirements Specifications. IEEE Std 830-1998. IEEE, 1998.
[Джерело: IEEE Xplore Digital Library](#)
- Sommerville, I. Software Engineering. 10th ed. Pearson, 2015.
Джерело: Pearson Education
- Pressman, R. S. Software Engineering: A Practitioner's Approach. 9th ed. McGraw-Hill Education, 2014.
Джерело: McGraw-Hill Education
- Mozilla Developer Network (MDN) Web Docs. JavaScript Guide.
[Джерело: MDN Web Docs](#)

1.5 Огляд (Overview)

Цей підрозділ надає огляд структури та змісту документу специфікації вимог до програмного забезпечення (SRS) для проекту DevLib.

1. Зміст решти SRS:

Документ SRS містить детальний опис функціональних і нефункціональних вимог до програмного забезпечення DevLib. У ньому наведені такі основні розділи:

- **Загальний опис** (Розділ 2): охоплює загальні характеристики продукту, включаючи його перспективу, функції, характеристики користувачів, загальні обмеження та припущення.
- **Специфічні вимоги** (Розділ 3): найбільш об'ємний розділ, що містить детальний опис усіх специфічних вимог до системи, включаючи вимоги до зовнішніх інтерфейсів, функціональні та нефункціональні вимоги, а також моделі використання.
- **Аналіз моделей** (Розділ 4): представляє моделі, які використовуються для розробки специфічних вимог, такі як діаграми послідовностей, діаграми потоків даних та діаграми станів.
- **Процес управління змінами** (Розділ 5): описує процедури оновлення SRS у разі зміни обсягу проекту або вимог.

2. Організація SRS:

Документ структурований у логічному порядку, щоб полегшити розуміння та доступ до інформації. Кожен розділ має свою тему та мету, що забезпечує чіткість викладу. Основні розділи і підрозділи розташовані в порядку, який починається з загальної інформації про проект і поступово переходить до конкретних вимог та аналізу. Кожен розділ пронумерований, що дозволяє легко посилатися на конкретні вимоги або аспекти документа. Додатки, якщо такі є, надають додаткову інформацію, яка може бути корисною, але не є частиною основного набору вимог.

2. Загальна характеристика (General Description)

2.1 Перспектива продукту (Product Perspective)

DevLib — це незалежний онлайн-продукт, створений для програмістів, який об'єднує функціональність великої бібліотеки з навчальними матеріалами та інтерактивного форуму. Основною метою DevLib є полегшення доступу до якісних ресурсів для програмування та сприяння обміну знаннями в спільноті розробників.

Продукт складається з двох основних компонентів:

- **Клієнтська сторона**, яка представлена веб-додатком, через який користувачі можуть взаємодіяти з системою, переглядати матеріали, обговорювати теми на форумах, і завантажувати ресурси.

- **Серверна сторона**, яка відповідає за управління базою даних книг, обробку запитів на завантаження матеріалів, взаємодію з форумами та чатами, а також забезпечує авторизацію та реєстрацію користувачів.

У порівнянні з іншими подібними платформами, DevLib надає розширені можливості взаємодії, таких як інтегрована система оцінювання контенту, можливість читання книг безпосередньо на сайті, а також інтерактивний форум для обговорення технічних питань.

2.2 Функції продукту(Product Functions)

Реєстрація та авторизація:

- Користувачі можуть створювати акаунти через email або соціальні мережі (Google, GitHub).
- Авторизація дозволяє отримати доступ до персональних налаштувань, збережених матеріалів і участі в форумах.

Каталог матеріалів:

- Веб-платформа надає доступ до великої бібліотеки матеріалів з програмування (книги, статті, відео, tutorіали).
- Користувачі можуть шукати ресурси за назвами, авторами, категоріями або ключовими словами.

Завантаження книг:

- Книги з програмування доступні для завантаження у форматі PDF, що дозволяє користувачам читати їх офлайн на власних пристроях.

Оцінки та відгуки:

- Користувачі можуть залишати оцінки та відгуки про книги, допомагаючи іншим учасникам вибрати кращі ресурси.

Збереження книг у закладки:

- Можливість зберігати книги у закладки для швидкого доступу та переглядати історію завантажень.

Читання книг на сайті:

- DevLib пропонує інтерактивний інтерфейс для читання книг прямо на сайті з можливістю робити замітки та виділення.

Пошук і фільтрація матеріалів:

- Пошук книг з можливістю сортувати результати за датою, рейтингом, популярністю, авторами та іншими параметрами.

Форум та чати:

- Інтерактивний форум для обговорення технічних питань, де користувачі можуть ставити запитання, обмінюватися досвідом і відповідати на запити інших програмістів.
- Текстові чати для обговорення в реальному часі.

2.3 Характеристики користувача(User Characteristics)

Платформа DevLib має кілька важливих обмежень, які можуть впливати на її функціонування. По-перше, для використання більшості функцій, таких як доступ до повного каталогу матеріалів, завантаження книг та участь у форумі, користувачам необхідно пройти реєстрацію та авторизацію. Це створює бар'єр для тих, хто бажає отримати ресурси без створення акаунта. По-друге, система залежить від стабільного інтернет-з'єднання, оскільки більшість операцій, як-от перегляд книг або участь у дискусіях, відбуваються онлайн. Крім того, функціональність платформи обмежується підтримкою тільки певних форматів файлів для завантаження книг і матеріалів. Також важливо зазначити, що платформа може мати обмеження в масштабуванні при значному зростанні кількості користувачів, що може потребувати додаткових ресурсів для забезпечення стабільної роботи.

2.4 Загальні обмеження(General Constraints)

Платформа DevLib має кілька важливих обмежень, які можуть впливати на її функціонування. По-перше, для використання більшості функцій, таких як доступ до повного каталогу матеріалів, завантаження книг та участь у форумі, користувачам необхідно пройти реєстрацію та авторизацію. Це створює бар'єр для тих, хто бажає отримати ресурси без створення акаунта. По-друге, система залежить від стабільного інтернет-з'єднання, оскільки більшість операцій, як-от перегляд книг або участь у дискусіях, відбуваються онлайн. Крім того, функціональність платформи обмежується підтримкою тільки певних форматів файлів для завантаження книг і матеріалів. Також важливо зазначити, що платформа може мати обмеження в масштабуванні при значному зростанні кількості користувачів, що може потребувати додаткових ресурсів для забезпечення стабільної роботи. Крім того, система DevLib буде підтримувати лише завантаження книг у форматі PDF, що може обмежити користувачів, які шукають матеріали в інших форматах, таких як ePub чи fb2. Це обмеження пов'язане з тим, що формат PDF є найпоширенішим для технічних і навчальних матеріалів, однак для деяких користувачів це може стати незручністю.

2.5 Припущення та залежності(Assumptions and Dependencies)

AS-1: Популярність серед програмістів та студентів: припускається, що платформа DevLib здобуде популярність серед програмістів різних рівнів, зокрема студентів, які активно шукають навчальні ресурси. Вона стане привабливою для тих, хто прагне розширювати свої знання з програмування за допомогою книг і матеріалів.

AS-2: Професіонали та викладачі: очікується, що платформа знайде користувачів серед професіоналів, які працюють у галузі ІТ, а також серед викладачів, які шукають додаткові ресурси для своїх студентів. Це розширить аудиторію та сприятиме поширенню платформи в освітніх колах.

AS-3: Інтернет-з'єднання: передбачається, що користувачі матимуть стабільний доступ до Інтернету, оскільки більшість функцій платформи, таких як перегляд книг та участь у форумі, вимагають постійного підключення.

DE-1: Взаємодія користувачів: система DevLib залежить від активної участі користувачів, зокрема їхніх відгуків, оцінок та рекомендацій матеріалів. Успішне функціонування платформи передбачає активну спільноту, яка ділиться знаннями та допомагає підтримувати якісний контент.

DE-2: Підтримка PDF: система залежатиме від того, що користувачі завантажуватимуть матеріали виключно у форматі PDF. Це може обмежити можливості для користувачів, які воліли б використовувати інші формати для читання.

3. Особливі Вимоги (Specific Requirements)

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project's software design, implementation, and testing.

Each requirement in this section should be:

- *Correct*
- *Traceable (both forward and backward to prior/future artifacts)*
- *Unambiguous*
- *Verifiable (i.e., testable)*
- *Prioritized (with respect to importance and/or stability)*
- *Complete*
- *Consistent*
- *Uniquely identifiable (usually via numbering like 3.4.5.6)*

Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

3.1 Вимоги до зовнішнього інтерфейсу (External Interface Requirements)

3.1.1 Користувацькі інтерфейси (User Interfaces)

Інтерфейс користувача для системи «DevLib» створений для надання користувачам доступу до бібліотеки книг і матеріалів з програмування, а також форум для обговорення технічних тем і спілкування в чатах. Платформа спрямована на забезпечення доступу до навчальних ресурсів і сприяння обміну знаннями серед розробників. Користувачі матимуть можливість:

- здійснювати реєстрацію та авторизацію за допомогою email або соціальних мереж для доступу до розширених функцій платформи;
- переглядати каталог матеріалів з програмування, який включає книги, статті, відеоуроки та інші ресурси;
- завантажувати книги у форматі PDF на свій пристрій для подальшого використання офлайн;
- оцінювати та залишати відгуки щодо матеріалів, щоб допомогти іншим користувачам обирати корисні ресурси;
- зберігати матеріали у закладки та переглядати історію завантажень і читання;
- читати книги безпосередньо на сайті, без необхідності завантаження;
- шукати книги та інші матеріали за ключовими словами та використовувати фільтри для сортування результатів (наприклад, за рейтингом, датою публікації тощо).

Інтерфейс користувача буде розроблений для зручного та інтуїтивного використання, щоб забезпечити комфортне взаємодію з платформою для користувачів різного рівня.

3.1.2 Інтерфейси апаратного забезпечення (Hardware Interfaces)

Система «DevLib» не потребує специфічних апаратних засобів для доступу. Основні вимоги до обладнання включають:

- персональний комп'ютер або мобільний пристрій з веб-браузером (рекомендовані браузери: Google Chrome, Mozilla Firefox, Safari, Edge);
- інтернет-з'єднання, необхідне для доступу до ресурсів і форуму платформи;
- дисплей з достатньою роздільною здатністю, який дозволить комфортно читати матеріали і взаємодіяти з інтерфейсом.

Користувачам не потрібно спеціального обладнання, лише доступ до стандартного Інтернет-пристрою.

3.1.3 Інтерфейси програмного забезпечення (Software Interfaces)

Система «DevLib» буде інтегруватися з кількома програмними компонентами для забезпечення основної функціональності. Основні інтерфейси програмного забезпечення включають:

1. База даних: платформа використовуватиме реляційну базу даних (наприклад, MySQL або PostgreSQL) для зберігання інформації про користувачів, книги, відгуки, рейтинги та інші матеріали. Взаємодія з базою даних

- здійснюватиметься через API з використанням SQL-запитів, забезпечуючи швидкий доступ до даних і їхню безпеку.
2. Модуль аутентифікації та авторизації: система підтримуватиме різні методи авторизації, включаючи стандартний вхід за допомогою email і пароля, а також через сторонні сервіси, такі як Google або Facebook (OAuth 2.0). Цей інтерфейс забезпечуватиме безпечну аутентифікацію користувачів і контроль доступу до розширених функцій платформи.
 3. API для сторонніх інтеграцій: «DevLib» передбачає можливість взаємодії з іншими сервісами через REST API. Це може включати інтеграції з зовнішніми навчальними платформами, платіжними системами або системами відстеження прогресу користувачів у навчанні.
 4. PDF-переглядач: платформа інтегруватиме спеціальний інструмент для перегляду PDF-файлів онлайн, що дозволить користувачам читати книги безпосередньо на сайті. Цей компонент буде сумісний з браузерами і не вимагатиме додаткового програмного забезпечення.
 5. Модуль пошуку та фільтрації: для зручності користувачів, система забезпечуватиме швидкий пошук книг та інших матеріалів за ключовими словами. Також будуть доступні функції фільтрації за рейтингом, датою публікації, жанром тощо. Цей інтерфейс взаємодіятиме з базою даних для динамічного отримання результатів.
 6. Система оцінок та відгуків: користувачі зможуть оцінювати матеріали та залишати відгуки, що допоможе формувати рейтинг ресурсів. Інтерфейс програмного забезпечення для оцінок і відгуків буде взаємодіяти з базою даних для зберігання цієї інформації та надаватиме відповідні функції для перегляду рейтингу.

Таким чином, ці програмні інтерфейси забезпечать повну функціональність платформи «DevLib» та її інтеграцію з іншими сервісами для ефективного використання користувачами.

3.1.4 Інтерфейси Зв'язку (Communications Interfaces)

Для забезпечення роботи платформи «DevLib» користувачам необхідний постійний доступ до мережі Інтернет. Основні вимоги до інтерфейсів зв'язку включають:

- підключення до Інтернету через будь-який пристрій із веб-браузером, що дозволяє користувачам взаємодіяти з бібліотекою ресурсів, форумом та іншими функціями платформи;
- протоколи безпечної передачі даних (HTTP/HTTPS) для забезпечення захисту конфіденційної інформації користувачів при реєстрації, авторизації та обміні даними;
- підтримка REST API для інтеграції з іншими сервісами, такими як системи авторизації через соціальні мережі або сторонні сервіси для спрощення доступу до платформи;
- відправка сповіщень користувачам (email або push-сповіщення) про нові книги, повідомлення на форумі або особисті повідомлення в чатах.

Цей інтерфейс забезпечує надійне і безпечне з'єднання для взаємодії користувачів з системою.

3.2 Функціональні вимоги (Functional Requirements)

3.2.1 Реєстрація та авторизація

3.2.1.1 Введення (Introduction)

Функціональність реєстрації та авторизації дозволяє користувачам створювати акаунти для доступу до платформи «DevLib» та входити до системи для використання розширених функцій, таких як оцінки, відгуки, завантаження та закладки. Реєстрація може здійснюватись за допомогою email або через сторонні сервіси, як-от Google або Facebook. Це забезпечує зручність для користувачів, які можуть вибрати найбільш зручний спосіб входу.

3.2.1.2 Вхідні дані (Inputs)

- логін, email і пароль: користувач вводить свою електронну пошту та пароль для стандартної реєстрації. Логін (4-32 символи), електронна пошта - email, пароль (8-32 символи), повторний пароль (8-32 символи).
- дані профілю з соціальних мереж: при авторизації через соціальні сервіси, система отримує дані профілю (ім'я, адреса електронної пошти) від обраного сервісу (Google, Facebook тощо).

3.2.1.3 Обробка (Processing)

- валідація даних: система при реєстрації перевіряє коректність введених даних (правильність формату email, логін, довжина пароля, наявність спеціальних символів). Якщо ці дані введено правильно, сервер перевіряє, чи існують користувачі з вказаним логіном або електронною поштою. У випадку, якщо користувач з такими даними вже існує, процес реєстрації завершується невдало, і користувач отримує повідомлення про помилку. При авторизації йде перевірка існування користувача: чи існує користувач з введеним логіном та паролем. Якщо все введено правильно, то користувач отримує доступ до облікового запису. У разі неправильного паролю або відсутності користувача з введеним логіном, сервер повідомляє про помилку авторизації, і користувачу потрібно повторити процес або відновити пароль у разі забуття.
- активація акаунта: при реєстрації через email, система надсилає лист з підтвердженням для активації акаунта. Лист містить унікальне посилання для підтвердження адреси електронної пошти.
- авторизація через OAuth: система використовує протокол OAuth для забезпечення безпеки передачі даних при авторизації через соціальні мережі. Після авторизації користувач перенаправляється на платформу «DevLib».
- збереження сесії: після успішної авторизації, система створює сесію для користувача, яка зберігає його дані для подальшого використання.

3.2.1.4 Вихідні Дані (Outputs)

Користувач отримує підтвердження про успішну реєстрацію або авторизацію через повідомлення на екрані. Система створює новий акаунт у базі даних, зберігаючи всі необхідні дані. Після авторизації, користувач має доступ до свого профілю, де може користуватися розширеними функціями платформи. У випадку, якщо дані введені некоректно або відсутні обов'язкові поля, система відображає повідомлення з відповідним текстом помилки перед реєстраційною формою.

3.2.1.5 Обробка Помилки (Error Handling)

1. Некоректний email, login, пароль, обов'язкові поля не були заповнені: система відображає повідомлення про помилку, яке пояснює, чому дані не були прийняті.

2. Помилка авторизації через соціальні мережі: якщо виникає проблема з авторизацією через соціальні сервіси, система повідомляє користувача про неможливість завершити процес.

3. Акаунт вже існує: якщо введений email вже зареєстрований, система пропонує користувачу відновити пароль або вказує, що акаунт вже активний.

4. Виникли технічні проблеми під час обробки запиту, система відображає повідомлення про помилку і рекомендує спробувати зареєструватися або авторизуватися пізніше.

3.2.2 Каталог матеріалів

3.2.2.1 Введення (Introduction)

Ця функціональність забезпечує користувачам доступ до широкого каталогу матеріалів з програмування, таких як книги, статті, відео та tutorіали. Користувачі можуть переглядати, сортувати і фільтрувати доступні ресурси за різними параметрами, що робить процес пошуку більш ефективним.

3.2.2.2 Вхідні дані (Inputs)

- параметри пошуку: користувач вводить ключові слова, які відображають тематику, що його цікавить;
- категорії, фільтри: користувач вибирає категорії матеріалів (книги, відео, статті тощо) для звуження результатів.

3.2.2.3 Обробка (Processing)

- запит до бази даних: Система отримує запит користувача та виконує пошук у базі даних, використовуючи введені параметри;
- фільтрація: Система обробляє параметри пошуку, фільтруючи та сортуваючи результати відповідно до запиту користувача. Додатково, результати можуть бути відсортовані за рейтингом, популярністю або датою публікації;
- система відображає користувачу список матеріалів, які задовольняють його критеріям.

3.2.2.4 Вихідні Дані (Outputs)

- відображається список матеріалів, відповідних до запиту користувача, із зазначенням їх основної інформації (назва, автор, рейтинг);
- користувач має можливість перейти на сторінку конкретного матеріалу, де міститься більш детальна інформація, а також можливість завантаження.

3.2.2.5 Обробка Помилки (Error Handling)

- якщо за запитом не знайдено матеріалів, система повідомляє користувача про відсутність результатів за заданими критеріями;
- у разі виникнення помилок при зверненні до бази даних, система відображає повідомлення про неможливість завантажити каталог.

3.2.3 Завантаження книг в форматі PDF

3.2.3.1 Введення (Introduction)

Функція дозволяє користувачам завантажувати книги з програмування у форматі PDF на свої пристрої для подальшого офлайн перегляду. Це забезпечує зручний доступ до навчальних матеріалів у будь-який час.

3.2.3.2 Вхідні дані (Inputs)

- запит на завантаження: користувач вибирає книгу для завантаження, що є доступною у каталозі;
- ідентифікатор книги: система використовує унікальний ID книги, щоб знайти та завантажити відповідний файл.

3.2.3.3 Обробка (Processing)

- перевірка прав доступу: система перевіряє, чи користувач має права на завантаження вибраної книги, тобто чи є він зареєстрованим користувачем;
- завантаження файлу: після підтвердження прав, файл PDF завантажується з сервера на пристрій користувача;
- оновлення історії: система оновлює історію завантажень користувача, зберігаючи дані про нове завантаження.

3.2.3.4 Вихідні Дані (Outputs)

- завантажений файл: книга у форматі PDF завантажується на пристрій користувача, надаючи доступ до матеріалів без підключення до Інтернету;
- історія завантажень: система оновлює історію завантажень у базі даних, що дозволяє користувачеві легко відстежувати свої дії.

3.2.3.5 Обробка Помилки (Error Handling)

- відсутність прав на завантаження: якщо користувач не має прав на завантаження, тобто не авторизований, система відображає повідомлення про необхідність авторизації;
- файл не знайдено: якщо файл PDF не знайдено на сервері, система виводить повідомлення про помилку завантаження.

3.2.4 Оцінки та відгуки

3.2.4.1 Введення (Introduction)

Користувачі можуть оцінювати матеріали (книги, статті тощо) та залишати відгуки, що допомагає іншим користувачам обирати найкращі ресурси. Ця функція також дозволяє авторам отримувати зворотний зв'язок про їхні роботи.

3.2.4.2 Вхідні дані (Inputs)

- користувач вводить оцінку від 1 до 5 зірок для обраного матеріалу;
- користувач може написати текстовий відгук, що пояснює його думку про матеріал.

3.2.4.3 Обробка (Processing)

Користувач відкриває сторінку з відгуками та оцінками для певного матеріалу. Після чого вводить текстовий відгук та ставить оцінку. Наступним кроком користувач підтверджує свій відгук і система зберігає його в базі даних, пов'язуючи з відповідним матеріалом.

Середній рейтинг матеріалу оновлюється на основі нових оцінок, що допомагає іншим користувачам орієнтуватися в популярності ресурсу.

3.2.4.4 Вихідні Дані (Outputs)

- після внесення оцінки, система відображає новий середній рейтинг матеріалу;
- відгук з'являється на сторінці матеріалу для перегляду іншими користувачами, що сприяє формуванню загальної думки про ресурс.

3.2.4.5 Обробка Помилки (Error Handling)

- відсутність тексту відгуку, система буде надавати користувачеві відповідні повідомлення про помилки.
- проблеми з базою даних: Якщо з'єднання з базою даних неможливе, система повідомляє про тимчасову недоступність сервісу.

3.2.5 Можливість зберігати книги у «закладки», переглядати історію завантажень і читання

3.2.5.1 Введення (Introduction)

Ця функція дозволяє користувачам зберігати книги у «закладки», щоб мати швидкий доступ до обраних матеріалів, а також переглядати історію своїх завантажень і читання. Це робить взаємодію з платформою більш зручною та інтуїтивною.

3.2.5.2 Вхідні дані (Inputs)

- запит на додавання у «закладки»: користувач обирає книгу, яку бажає зберегти у «закладки»;

- запит на перегляд історії: користувач ініціює перегляд історії своїх завантажень і читання.

3.2.5.3 Обробка (Processing)

- збереження у «закладки»: система зберігає книгу у «закладках» користувача в базі даних, що дозволяє швидкий доступ до вибраних матеріалів.
- оновлення історії: після кожної дії користувача (завантаження чи відкриття книги), система оновлює історію завантажень і читання, фіксуючи відповідні дії.

3.2.5.4 Вихідні Дані (Outputs)

- список «закладок»: користувач отримує доступ до списку книг, збережених у «закладках»;
- історія завантажень і читання доступна користувачеві в особистому кабінеті, що дозволяє йому відстежувати свою діяльність.

3.2.5.5 Обробка Помилки (Error Handling)

- помилка додавання у «закладки»: якщо книга не може бути додана в «закладки», система відображає повідомлення про помилку;
- недоступність історії: у разі недоступності історії завантажень, система повідомляє про помилку з'єднання з базою даних.

3.2.6 Читання книг на сайті

3.2.6.1 Введення (Introduction)

Функція дозволяє користувачам читати книги у форматі PDF безпосередньо на платформі «DevLib», без необхідності завантаження на пристрій. Це забезпечує зручність та миттєвий доступ до матеріалів.

3.2.6.2 Вхідні дані (Inputs)

Користувач ініціює запит на відкриття книги для читання онлайн.

3.2.6.3 Обробка (Processing)

Система завантажує PDF файл з сервера і відкриває його в інтегрованому PDF-переглядачі, що забезпечує комфортний перегляд безпосередньо на сайті.

3.2.6.4 Вихідні Дані (Outputs)

Книга відкривається для перегляду в браузері, надаючи користувачеві можливість читати без завантаження.

3.2.6.5 Обробка Помилки (Error Handling)

Помилка завантаження: якщо файл не може бути завантажений (наприклад, через проблеми з сервером), система виводить відповідне повідомлення про помилку.

3.2.7 Пошук книг, сортування та фільтрація результатів

3.2.7.1 Введення (Introduction)

Ця функція дозволяє користувачам шукати книги та інші матеріали за ключовими словами і сортувати або фільтрувати результати пошуку за різними критеріями, що підвищує ефективність пошуку. Це дозволяє користувачам швидко знайти потрібні ресурси, не переглядаючи весь каталог.

3.2.7.2 Вхідні дані (Inputs)

1. Ключові слова: користувач вводить ключові слова для пошуку, які допомагають звужити результати.

2. Параметри сортування: користувач обирає параметри для сортування результатів:

- рейтинг: сортування матеріалів за середньою оцінкою користувачів.
- дата публікації: сортування матеріалів за датою їх випуску, від нових до старих або навпаки.
- популярність: сортування на основі кількості переглядів або завантажень матеріалу.
- автор: сортування за алфавітом імен авторів.

3. Критерії фільтрації: користувач може фільтрувати результати за такими параметрами:

- тип матеріалу: користувач вибирає, які типи матеріалів він хоче бачити (книги, статті, відео, tutorіали).
- тематика: фільтрація за тематичними категоріями (наприклад, веб-розробка, мобільна розробка, бази даних, машинне навчання).
- мова: вибір мови матеріалів (англійська, українська, російська тощо).
- ціна: фільтрація безкоштовних та платних матеріалів.

3.2.7.3 Обробка (Processing)

- система виконує пошук матеріалів у базі даних відповідно до введених ключових слів та критеріїв. Запит до бази даних враховує всі введені параметри пошуку, фільтрації та сортування.
- результати фільтруються та сортуються на основі вибраних параметрів, що дозволяє користувачеві швидше знаходити необхідні матеріали:
- якщо користувач вибрав фільтрацію за тематикою, то система обробляє запит і відображає лише ті матеріали, що відповідають обраній категорії.
- після фільтрації система застосовує обрані параметри сортування до залишених результатів.

3.2.7.4 Вихідні Дані (Outputs)

Відображається список книг або матеріалів, що відповідають критеріям пошуку, з основною інформацією про кожен ресурс, включаючи:

- назва матеріалу
- автор
- дата публікації
- середній рейтинг
- тип матеріалу (книга, стаття, відео)
- короткий опис або анотація
- посилання на матеріал (для завантаження або читання онлайн)

3.2.7.5 Обробка Помилки (Error Handling)

- якщо пошук не дає результатів, система повідомляє користувача про відсутність знайдених матеріалів за введеними критеріями, з можливістю спробувати інші ключові слова або скинути фільтри.
- якщо введені дані у полях пошуку мають неправильний формат (наприклад, заборонені символи в ключових словах), система відображає повідомлення про помилку та пропонує виправити введення.
- у разі проблем з підключенням до бази даних, система виводить повідомлення про технічні проблеми та пропонує повторити спробу пізніше.

3.2.8 Форум для будь-яких користувачів

3.2.8.1 Введення (Introduction)

Ця функція дозволяє будь-якому користувачеві системи переглядати форум без обов'язкової реєстрації або входу в обліковий запис. Користувач може переглядати інформацію, що розміщена на форумі, а також взаємодіяти з нею у відповідності з правилами і обмеженнями, встановленими системою.

3.2.8.2 Вхідні дані (Inputs)

Для цієї функції вхідні дані обмежені лише запитами користувачів на перегляд форуму.

3.2.8.3 Обробка (Processing)

Спершу система отримує запит від користувача на перегляд форуму спільноти. В подальшому відображає дані форуму користувачу. В свою чергу користувач може переглядати теми, повідомлення, коментарі та іншу інформацію.

3.2.8.4 Вихідні Дані (Outputs)

Вихідні дані цієї функції включають інформацію, що відображається користувачу на форумі та в спільноті.

3.2.8.5 Обробка Помилки (Error Handling)

Під час обробки цієї функції можуть виникнути технічні помилки пов'язані з проблемами на сервері або Інтернет з'єднанням. У разі виникнення такого роду помилок, система повинна відправити відповідне повідомлення користувачеві.

3.2.9 Форум для зареєстрованих користувачів

3.2.9.1 Введення (Introduction)

Ця функція дозволяє зареєстрованим користувачам системи створювати нові теми та взаємодіяти з іншими учасниками форуму шляхом розміщення власних дописів на форумі.

3.2.9.2 Вхідні дані (Inputs)

Запит на створення нової теми на форумі від зареєстрованого користувача. Текстовий вміст нової теми.

3.2.9.3 Обробка (Processing)

Зареєстрований користувач входить в систему та переходить на форум, на якому він бажає створити нову тему або написати новий допис. Користувач обирає опцію «Створити нову тему» та безпосередньо вводить текстовий вміст теми. Система зберігає нову тему на форумі.

3.2.9.4 Вихідні Дані (Outputs)

Нова тема або допис користувача на форумі.

3.2.9.5 Обробка Помилки (Error Handling)

Система обробляє можливі помилки, такі як відсутність текстового вмісту у темі/дописі або технічні проблеми при збереженні. У випадку помилок, система виводить відповідне повідомлення користувачеві та не зберігає некоректний допис.

3.3 Use Cases

3.3.1 Незареєстрований Користувач Use Case

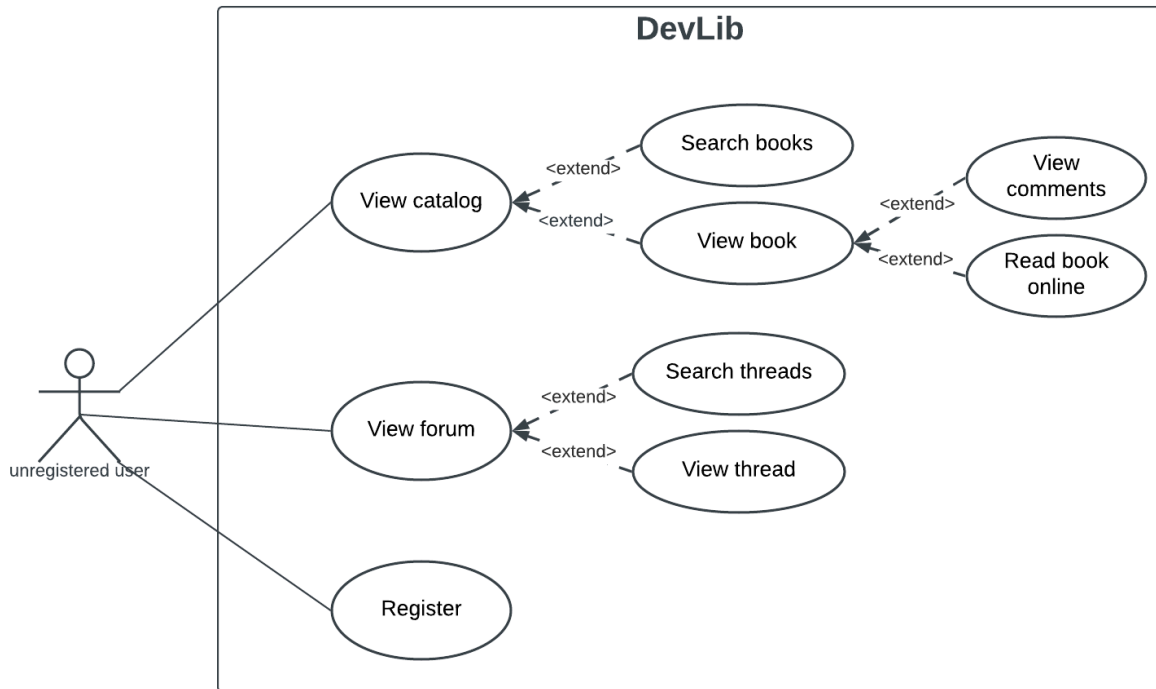


Рисунок 1 – Use Case діаграма незареєстрованого Користувача

3.3.2 Зареєстрований Користувач Use Case

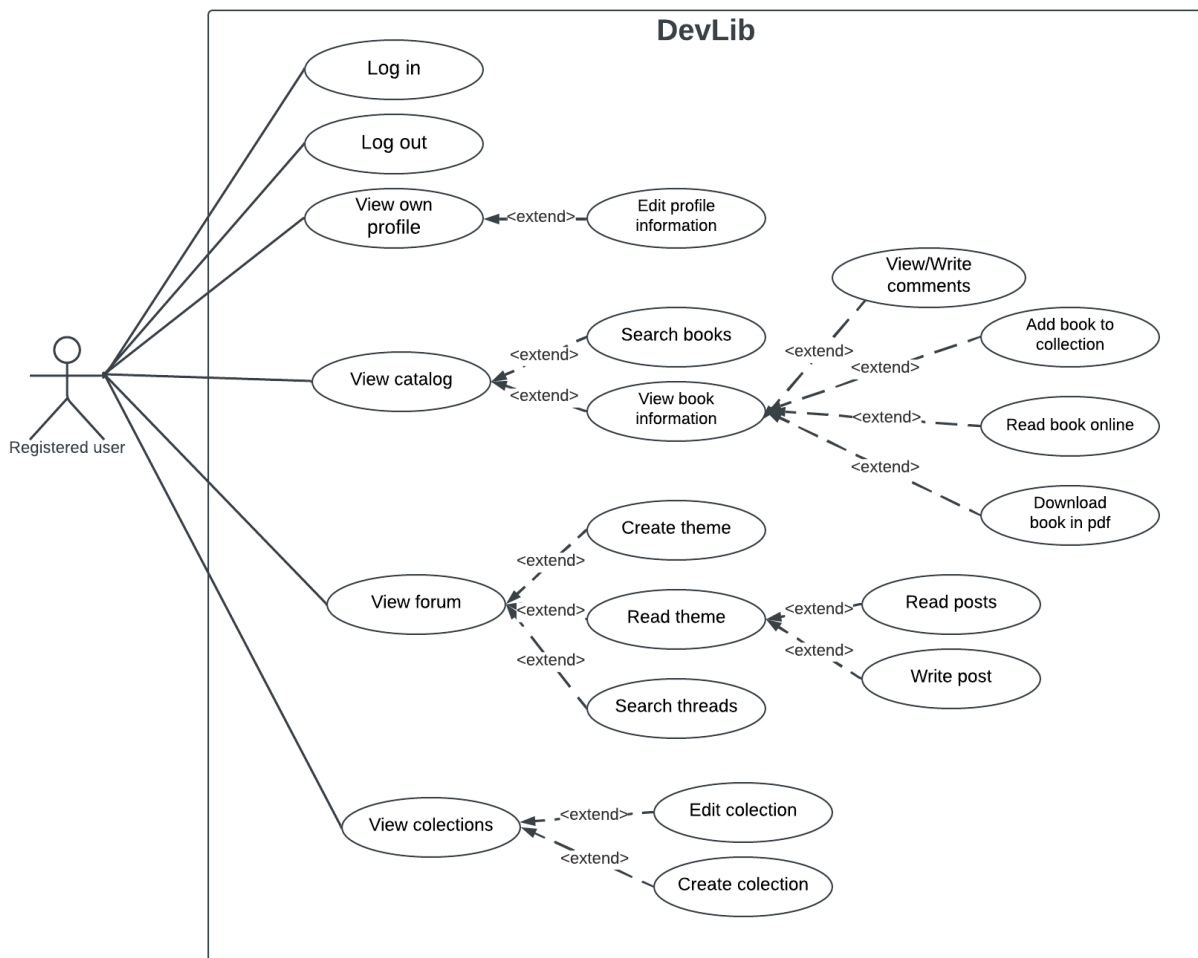


Рисунок 2 – Use Case діаграма зареєстрованого Користувача

3.3.2 Адміністратор Use Case

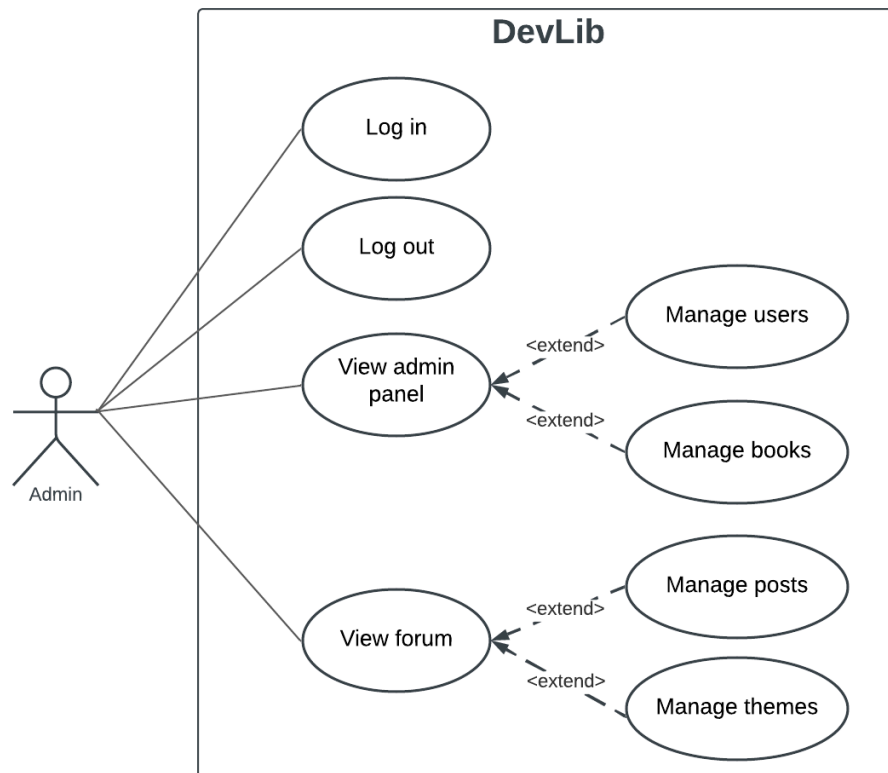


Рисунок 3 – Use Case діаграма Адміністратора

3.4 Класи/Об'єкти (Classes/Objects)

3.4.1 Клас User

Атрибути

id: унікальний ідентифікатор користувача

username: ім'я користувача

email: електронна адреса користувача

passwordHash: хеш пароля

role: роль користувача (звичайний користувач або адміністратор)

Методи

register(): метод для реєстрації нового користувача

login(): метод для авторизації користувача

logout(): метод для виходу з облікового запису

Посилання на функціональні вимоги

3.2.1 Реєстрація та авторизація

3.2.8 Форум для будь-яких користувачів

3.2.9 Форум для зареєстрованих користувачів

3.4.2 Клас (Catalog)

Атрибути

id: унікальний ідентифікатор каталогу

name: назва каталогу

Методи

getBooks(): метод для отримання списку книг у каталозі

filterBooks(): метод для фільтрації книг за різними параметрами (жанр, автор тощо)

sortBooks(): метод для сортування книг

Посилання на функціональні вимоги

3.2.2 Каталог матеріалів

3.2.7 Пошук книг, сортування та фільтрація результатів

3.4.3 Клас (Book)

Атрибути

id: унікальний ідентифікатор книги

title: назва книги

author: автор книги

format: формат файлу книги (PDF, ePub тощо)

rating: середній рейтинг книги

reviews: список відгуків користувачів

Методи

download(): метод для завантаження книги

readOnline(): метод для читання книги онлайн

rateBook(): метод для оцінки книги

addReview(): метод для написання відгуку

Посилання на функціональні вимоги

3.2.3 Завантаження книг у форматі PDF

3.2.4 Оцінки та відгуки

3.2.6 Читання книг на сайті

3.4.4 Клас (Forum)

Атрибути

id: унікальний ідентифікатор форуму

title: назва теми

author: автор теми

content: зміст теми або повідомлення

Методи

createThread(): метод для створення нової теми

postMessage(): метод для публікації повідомлення у темі

editMessage(): метод для редагування повідомлення

deleteMessage(): метод для видалення повідомлення

Посилання на функціональні вимоги

3.2.8 Форум для будь-яких користувачів

3.2.9 Форум для зареєстрованих користувачів

3.4.5 Клас (Bookmark)

Атрибути

id: унікальний ідентифікатор закладки

userId: ідентифікатор користувача, якому належить закладка

bookId: ідентифікатор книги, яку додано у закладки

dateAdded: дата, коли книга була додана у закладки

Методи

addBookmark(): метод для додавання книги у закладки

removeBookmark(): метод для видалення книги із закладок

getBookmarksByUser(): метод для отримання списку закладок користувача

Посилання на функціональні вимоги

3.2.5 Можливість зберігати книги у «закладки»

3.5 Нефункціональні вимоги (Non-Functional Requirements)

3.5.1 Продуктивність

- 1) 95% запитів на пошук книг або доступ до матеріалів повинні оброблятися протягом 3 секунд або менше.
- 2) Система повинна підтримувати до 1000 одночасних користувачів без значного зниження продуктивності.
- 3) Більшість сторінок повинні завантажуватися менше ніж за 3 секунди.
- 4) Система повинна бути здатною обробляти одночасні запити від багатьох користувачів і підтримувати високий рівень завантаження сервера.

3.5.2 Надійність

- 1) Система повинна мати можливість автоматичного відновлення після відмови або помилки.

3.5.3 Доступність

- 1) Платформа повинна бути доступною для користувачів з різних географічних регіонів без значного зниження швидкості.
- 2) Система повинна бути доступною 99.9% часу.

3.5.4 Безпека

- 1) Уся комунікація між клієнтами та сервером повинна проходити через HTTPS (SSL/TLS), що гарантує шифрування даних під час їх передачі.
- 2) Персональні дані користувачів повинні бути зашифровані і зберігатись відповідно до сучасних стандартів (AES-256).
- 3) Система повинна забезпечувати механізми аутентифікації та авторизації для забезпечення безпеки користувачів і контролю доступу

3.5.5 Обслуговування

- 1) Нові функції повинні додаватися до системи без необхідності зупинки її роботи.
- 2) Система повинна бути повністю протестована перед оновленням, з мінімальним відсотком помилок.
- 3) Вся документація по системі, включаючи API та архітектуру, повинна бути актуальною та підтримуватися впродовж життєвого циклу системи.
- 4) Система повинна бути легко розширюваною та оновлюваною для впровадження нових функцій та виправлення помилок.

3.5.6 Портативність

- 1) Платформа повинна коректно працювати на всіх популярних браузерях (Chrome, Firefox, Safari, Edge).
- 2) Користувачі повинні мати можливість використовувати платформу на мобільних пристроях з адаптивним дизайном.

3.6 Інверсні Вимоги (Inverse Requirements)

- 1) Веб-сервіс не повинний зберігати персональні дані користувачів у незашифрованому вигляді, будь то на сервері або в резервних копіях.
- 2) Веб-сервіс не має підтримувати незахищені протоколи передачі даних, такі як HTTP. Усі комунікації повинні здійснюватися тільки через HTTPS.
- 3) Система не повинна дозволяти незареєстрованим користувачам користуватися розширеними функціями платформи, такими як оцінки, відгуки, закладки або завантаження книг, без реєстрації та авторизації.

3.7 Обмеження дизайну (Design Constraints)

- 1) Платформа повинна відповідати сучасним веб-стандартам, таким як HTML5, CSS3, і ECMAScript (ES6 або новіші), для забезпечення сумісності з більшістю браузерів і платформ.
- 2) Використання open-source компонентів дозволяється, проте всі сторонні бібліотеки повинні бути ретельно перевірені на наявність ліцензій, сумісних з політиками компанії
- 3) Дизайн платформи повинен бути адаптивним для роботи на різних пристроях, включаючи комп'ютери, планшети і смартфони. Використання CSS Flexbox або Grid для забезпечення плавного адаптування елементів інтерфейсу до різних розмірів екрану.
- 4) Дизайн інтерфейсу повинен бути мінімалістичним та орієнтованим на користувача, з обмеженою кількістю складних анімацій та елементів, сучасним та зручним. Кольори повинні бути приємними для користувача та не викликати дискомфорт при тривалому використанні сервісу.
- 5) Всі сторінки та елементи інтерфейсу повинні дотримуватися єдиного стилю, включаючи колірну палітру, шрифти та розміщення елементів.
- 6) Використання однакових елементів керування для схожих завдань. Наприклад, кнопки для дій (завантаження, додавання до закладок, коментарі) повинні мати однаковий стиль і бути зрозумілими користувачам.
- 7) Система пошуку повинна бути помітною та легкою у використанні, з можливістю фільтрації та сортування результатів за різними параметрами (тип матеріалів, рейтинг, дата додавання).
- 8) Система повинна надавати зворотний зв'язок на будь-які дії користувача (наприклад, завантаження книги, додавання до закладок) за допомогою повідомлень або спливаючих вікон.
- 9) Помилки мають бути зрозумілими і пояснювати користувачу, що саме не вдалося і як це виправити.

3.8 Логічні вимоги до бази даних (Logical database requirements)

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

3.8.1 Формати даних:

- Користувацькі дані, такі як ім'я, електронна пошта, пароль, зберігаються у вигляді зашифрованого тексту.
- Інформація про матеріали (назва, автор, опис, формат файлу) зберігається в текстовому або файловому вигляді, залежно від типу даних.
- Оцінки, закладки та історія взаємодій користувачів зберігаються у вигляді зв'язаних записів у таблицях бази даних.

3.8.2 Можливості зберігання:

- База даних повинна підтримувати зберігання великого обсягу інформації (сотні тисяч книг і мільйони користувачів).
- Система повинна забезпечувати швидке зчитування даних при високих навантаженнях (до 1000 одночасних користувачів).

3.8.3 Цілісність даних:

- Всі транзакції повинні бути атомарними та підтримувати цілісність даних навіть у випадку системних збоїв.
- Резервні копії даних повинні створюватися щодня, з максимальним часом відновлення до 15 хвилин.

3.8.4 Зберігання даних:

- Персональні дані користувачів повинні зберігатися протягом всього часу активного використання платформи користувачем.
- Після деактивації облікового запису користувача, його дані повинні бути видалені або анонімізовані протягом 30 днів.

3.8.5 Захист даних:

- Всі дані повинні бути зашифровані за допомогою сучасних криптографічних стандартів (AES-256 для зберігання, TLS для передачі).

3.9 Інші вимоги (Other requirements)

Catchall section for any additional requirements.

3.9.1 Вимоги до користувацького інтерфейсу:

- Інтерфейс платформи повинен підтримувати багатомовність (мінімум дві мови: англійська, українська).
- Система повинна забезпечувати адаптивний дизайн для роботи на мобільних пристроях.

3.9.2 Захист від спаму та ботів:

- Для захисту від ботів та спаму необхідно реалізувати механізм CAPTCHA під час реєстрації та інших критичних дій користувачів.

3.9.3 Логування та аудит:

- Система повинна вести журнал дій користувачів та адміністраторів (авторизація, завантаження матеріалів, зміна профілю), що дозволить проводити аудит та моніторинг.

3.9.4 Підтримка API:

- Платформа повинна мати REST API для доступу до матеріалів, управління користувачами та їхньою активністю, а також для можливості інтеграції з мобільними додатками.

3.9.5 Відповідність:

- Система повинна відповідати регіональним стандартам захисту даних, таким як GDPR у ЄС.

4. Моделі аналізу

4.1 Діаграми послідовностей (Sequence Diagrams)

1. Реєстрація та авторизація

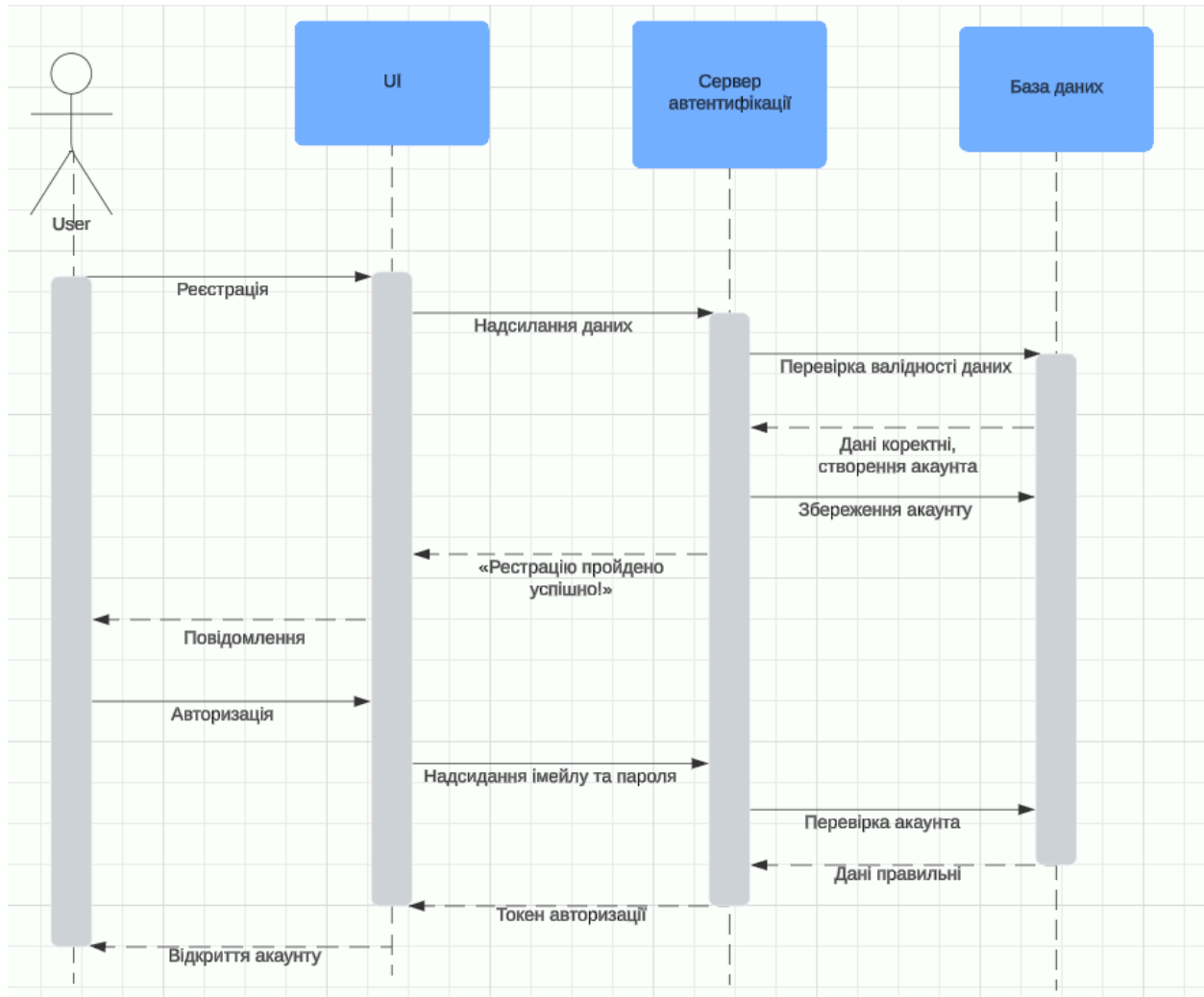
Учасники:

- користувач,
- інтерфейс (UI),
- сервер контенту,
- база даних (Database).

Процес:

1. Користувач взаємодіє з інтерфейсом і вибирає опцію "Реєстрація".
2. Інтерфейс запитує у користувача заповнити форму реєстрації (ім'я, email, пароль або використання соціальної мережі).
3. Користувач надсилає заповнені дані (ім'я, email, пароль) або підтверджує через соціальну мережу.
4. Інтерфейс передає дані на сервер автентифікації для обробки.
5. Сервер автентифікації перевіряє, чи існує вже такий email у базі даних.
 - Якщо email вже існує, повертається помилка "Користувач уже існує".

- Якщо email не існує, створюється новий запис у базі даних із відповідними даними користувача.
 - 6. Сервер автентифікації повертає результат реєстрації: успішно або з помилкою.
 - 7. Інтерфейс показує відповідне повідомлення користувачу ("Реєстрація успішна" або "Помилка").
- Після реєстрації:
- 8. Користувач переходить на форму авторизації та вводить email і пароль.
 - 9. Інтерфейс передає ці дані на сервер автентифікації.
 - 10. Сервер автентифікації перевіряє, чи існує користувач із такими даними в базі даних, і чи пароль правильний.
 - Якщо дані правильні, сервер генерує токен авторизації та повертає його на інтерфейс.
 - Якщо дані невірні, сервер відправляє помилку.
 - 11. Інтерфейс на основі отриманого токена надає користувачу доступ до захищених розділів сайту.



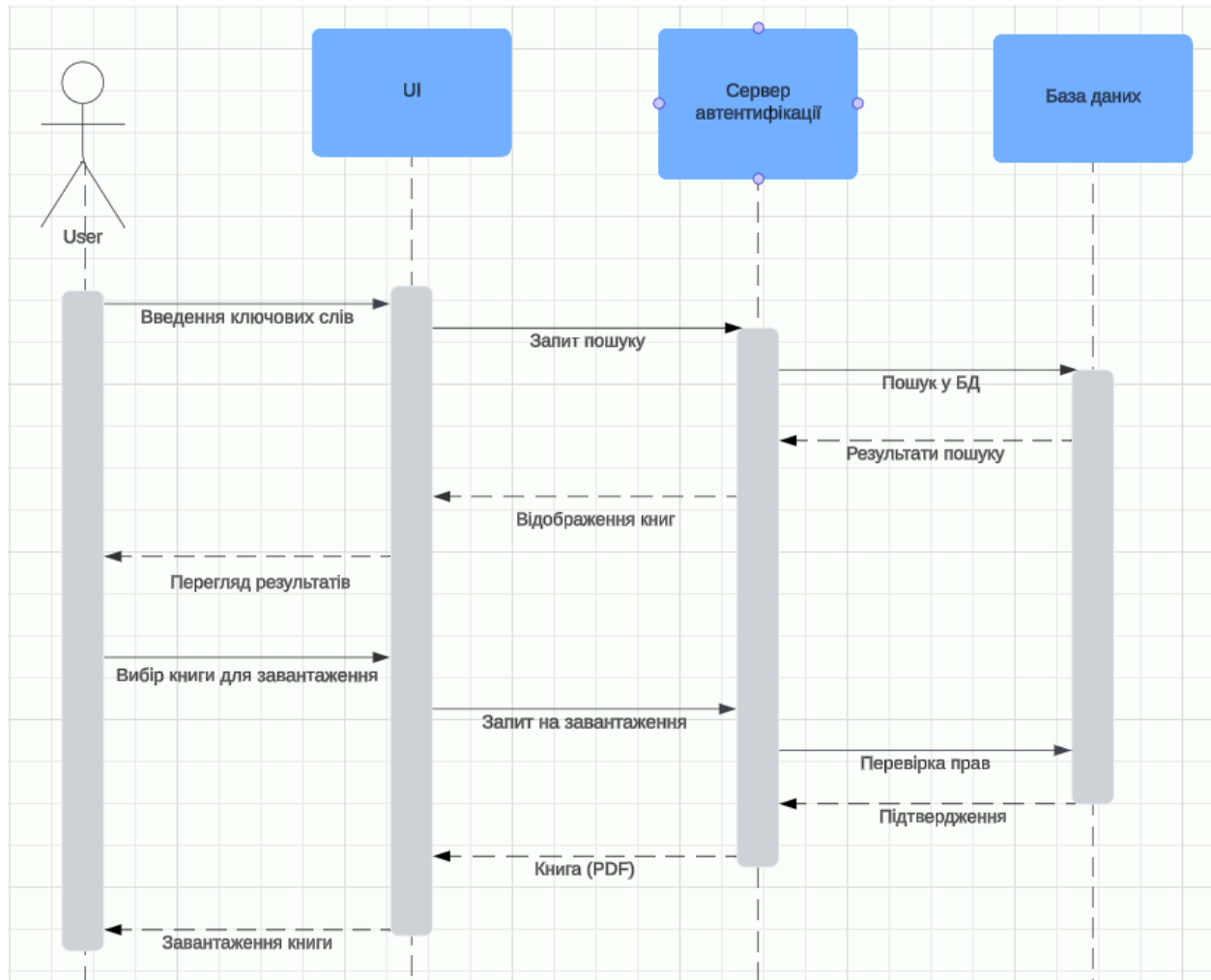
2. Пошук і завантаження книги

Учасники:

- користувач,
- інтерфейс (UI),
- сервер контенту,
- база даних (Database).

Процес:

1. Користувач заходить на платформу і вводить ключові слова в рядок пошуку.
2. Інтерфейс передає запит на сервер контенту.
3. Сервер контенту звертається до бази даних, де зберігається інформація про книги та матеріали, і отримує відповідні результати.
4. Сервер контенту повертає результати пошуку на інтерфейс.
5. Користувач переглядає результати та вибирає книгу для завантаження.
6. Інтерфейс відправляє запит на сервер контенту для завантаження книги.
7. Сервер контенту перевіряє права доступу користувача (на основі його авторизаційного токена) і, якщо права підтверджені, надсилає книгу у форматі PDF для завантаження.
8. Користувач завантажує книгу на свій пристрій.



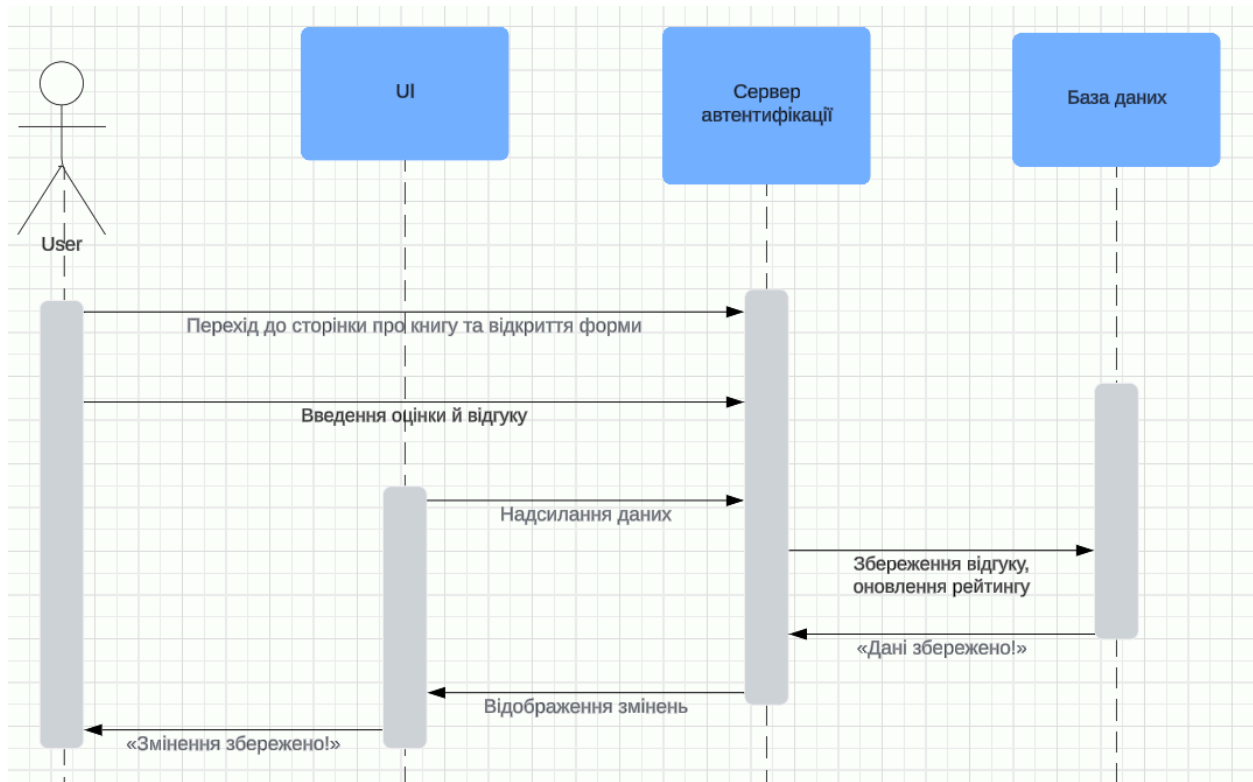
3. Оцінка та залишення відгуків

Учасники:

- користувач,
- інтерфейс (UI),
- сервер контенту,
- база даних (Database).

Процес:

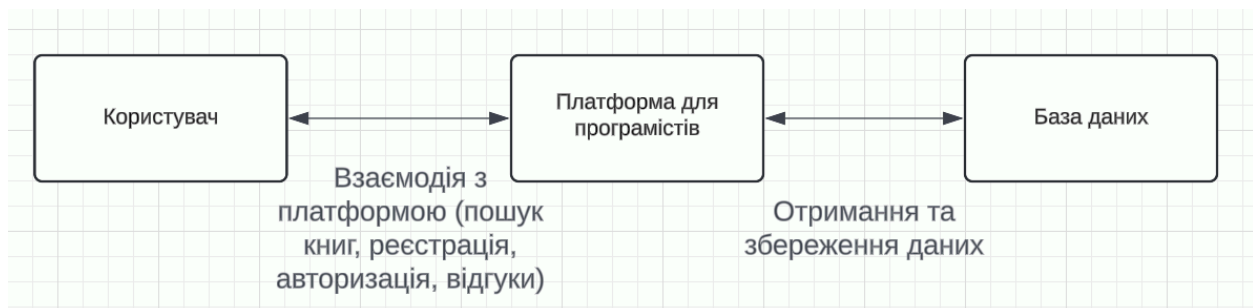
1. Користувач переглядає книгу і вирішує залишити відгук.
2. Інтерфейс пропонує форму для введення оцінки (зірки) та текстового відгуку.
3. Користувач надсилає оцінку та відгук.
4. Інтерфейс передає ці дані на сервер контенту.
5. Сервер контенту зберігає оцінку та відгук у базі даних, прив'язуючи їх до облікового запису користувача та відповідної книги.
6. Інтерфейс оновлює відображення середньої оцінки книги на основі нових даних з бази даних.



Загальна структура для всіх діаграм послідовності:

1. Користувач ініціює дію через інтерфейс (веб або мобільний додаток).
2. Інтерфейс надсилає запит на відповідний сервер.
3. Сервер перевіряє дані та звертається до бази даних, щоб отримати або зберегти необхідну інформацію.
4. Сервер повертає відповідь на інтерфейс.
5. Інтерфейс відображає результат для користувача.

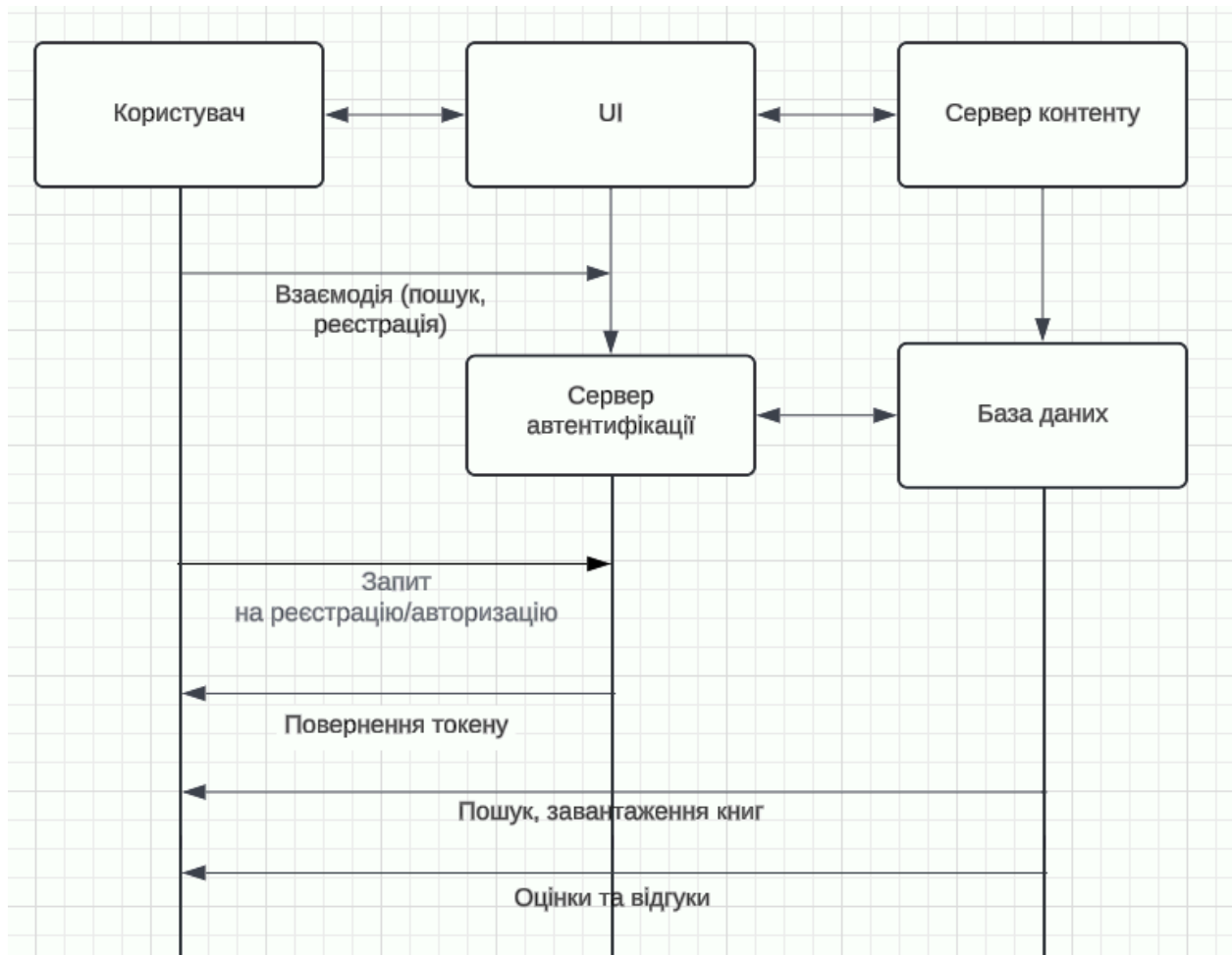
4.2 Діаграми потоків даних (Data Flow Diagrams, DFD)



Контекстна діаграма (DFD рівня 0)

Опис:

- **Користувач** взаємодіє з платформою, здійснюючи різні дії, такі як реєстрація, авторизація, пошук книг, завантаження та залишення відгуків.
- **Платформа для програмістів** обробляє запити користувача, взаємодіє з базою даних для отримання або збереження інформації, наприклад, для авторизації, пошуку книг або збереження відгуків.
- **База даних** містить інформацію про книги, користувачів, їхні відгуки, історію пошуку і завантажень.

**DFD рівня 1: розширена діаграма****Опис:**

1. Користувач взаємодіє з інтерфейсом платформи (UI) для виконання різних дій: реєстрація, авторизація, пошук і завантаження книг, а також залишення відгуків.

2. Інтерфейс UI:

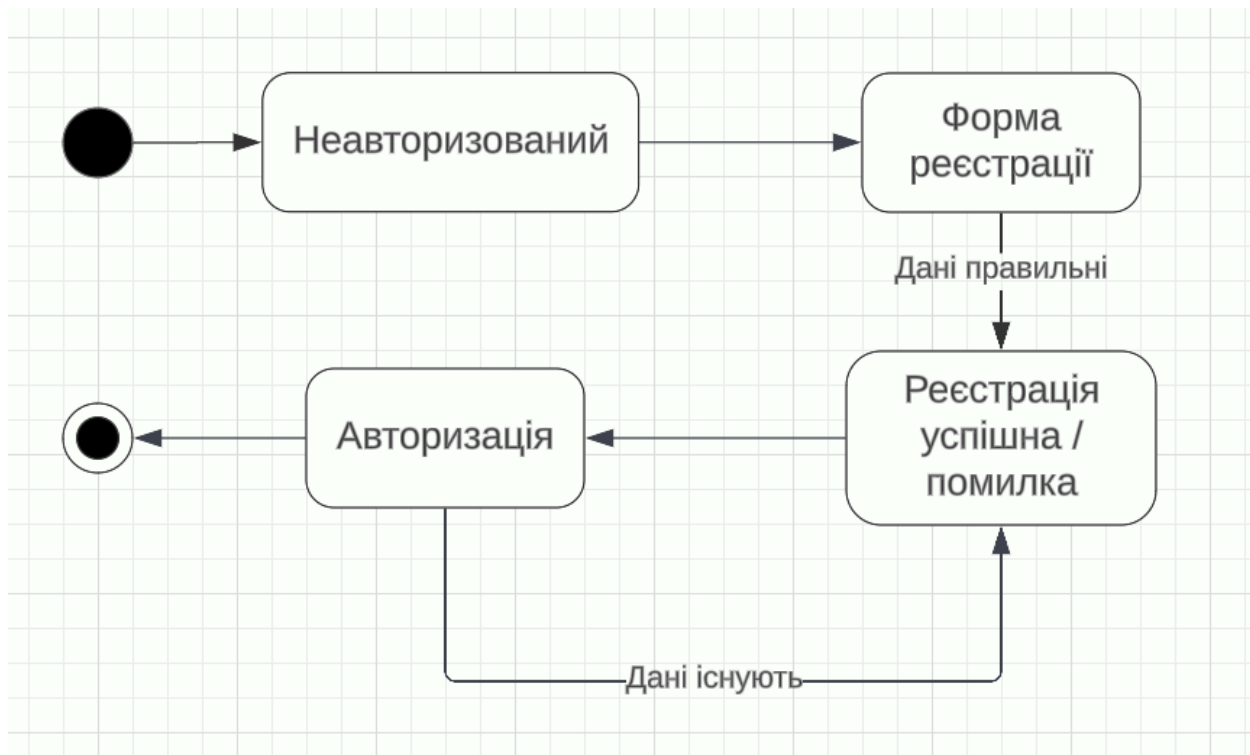
- Надсилає запити на сервер автентифікації для реєстрації або авторизації користувача. Сервер перевіряє дані в базі даних.
 - Для пошуку та завантаження книг або для відгуків надсилає запити на сервер контенту, який також працює з базою даних.
3. Сервер автентифікації відповідає за процеси реєстрації та авторизації, перевіряючи облікові дані користувача та зберігаючи нові облікові записи в базі даних.
 4. Сервер контенту обробляє пошукові запити користувачів, відповідає за доступ до книг, їх завантаження і роботу з відгуками. Він отримує дані з бази даних, яка містить інформацію про книги, користувачів та їх активність.

Компоненти:

- **Користувач** виконує дії на платформі.
- **Інтерфейс UI** взаємодіє з користувачем і передає запити на сервери.
- **Сервер автентифікації** обробляє реєстрацію та авторизацію.
- **Сервер контенту** обробляє запити на контент (книги) та відгуки.
- **База даних** зберігає інформацію про користувачів, книги, оцінки та інші дані.

4.3 Діаграми стану-переходу (State-Transition Diagrams, STD)

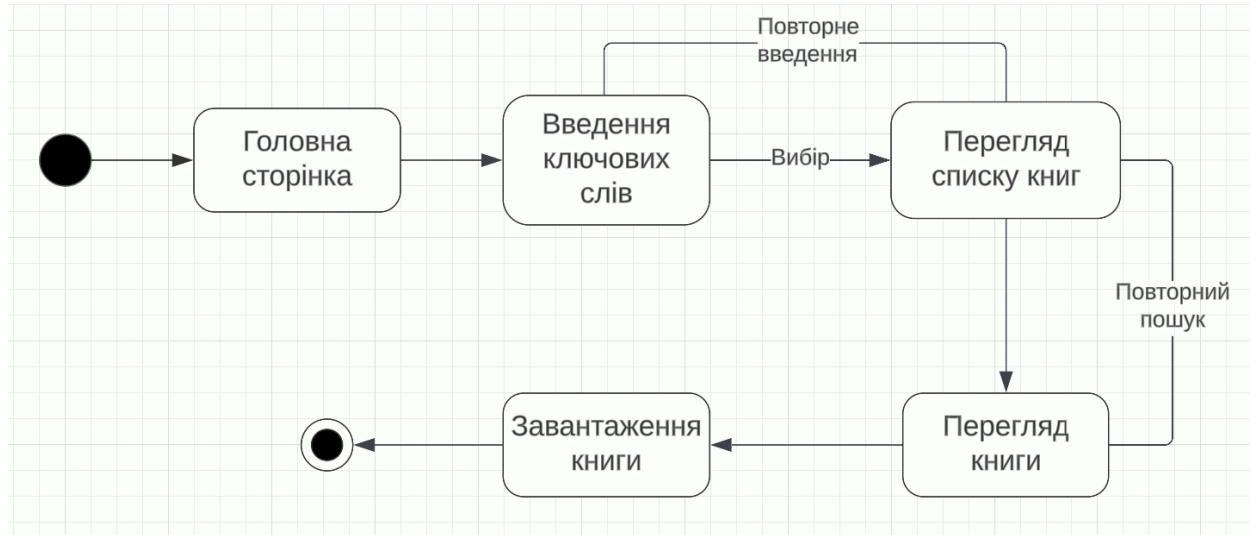
1. Діаграма станів для процесу реєстрації та авторизації



Опис:

- Неавторизований: Стан користувача до того, як він пройшов реєстрацію або авторизацію.
- Форма реєстрації: Користувач заповнює дані для реєстрації.
- Реєстрація: Якщо дані правильні і email ще не зареєстрований, користувач успішно зареєстрований; інакше показується помилка.
- Авторизація: Користувач вводить дані для входу. Користувач має доступ до захищених ресурсів платформи.

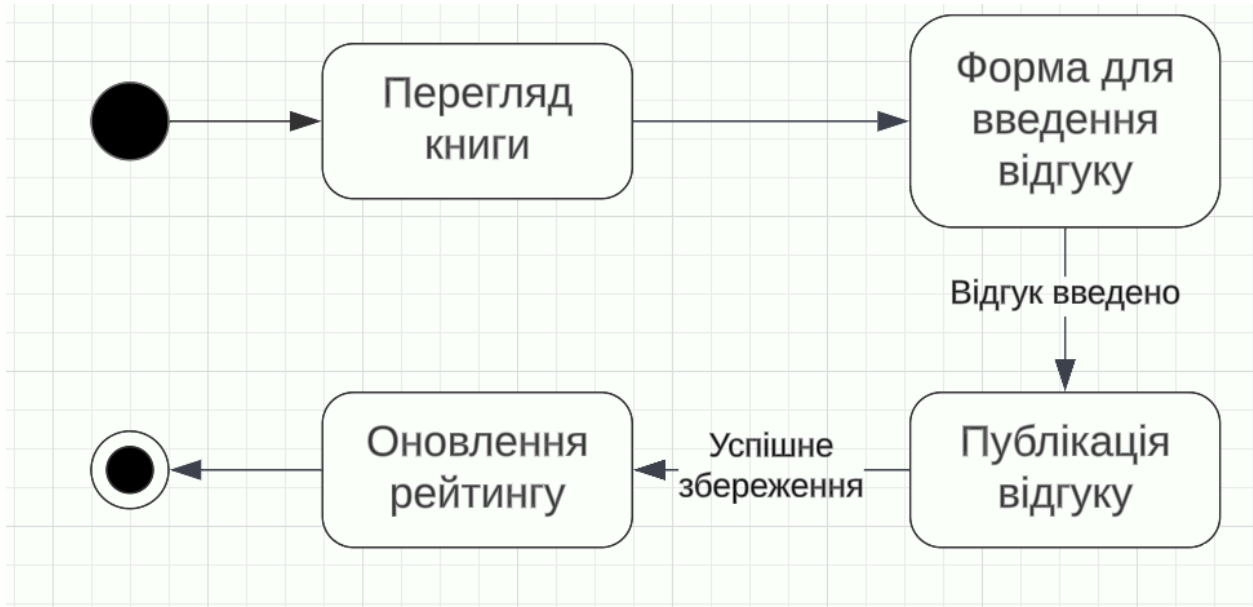
2. Діаграма станів для пошуку та завантаження книг



Опис:

- Головна сторінка: Стан, коли користувач знаходиться на головній сторінці.
- Введення ключових слів: Користувач вводить ключові слова для пошуку книг.
- Перегляд списку книг: Система відображає результати пошуку.
- Перегляд книги: Користувач вибирає книгу і може переглянути її деталі.
- Завантаження книги: Користувач завантажує книгу на свій пристрій.

3. Діаграма станів для процесу оцінки і залишення відгуків



Опис:

- Перегляд книги: Користувач переглядає книгу і вирішує залишити відгук.
- Форма для введення відгуку: Користувач заповнює поля для оцінки та текстового відгуку.
- Залишення відгуку: Система зберігає відгук.
- Оновлення рейтингу: Після успішного збереження відгуку рейтинг книги оновлюється.

Ці діаграми станів відображають основні етапи взаємодії користувача із системою на різних етапах її використання: реєстрація та авторизація, пошук та завантаження книг, а також залишення відгуків. Вони показують, як змінюються стани системи в залежності від дій користувача або внутрішніх процесів.

5. Процес управління змінами

Подання змін:

- Зміни до SRS можуть бути запропоновані будь-яким членом команди проекту та зацікавленими сторонами. Зміни оформлюються у вигляді запиту на зміну (Change Request).
- Запит на зміну повинен включати:
 - Опис пропонованої зміни.
 - Причини зміни (включаючи посилання на відповідні нові вимоги або проблеми).
 - Потенційний вплив на існуючі функції або системи.
 - Приблизні терміни впровадження та необхідні ресурси.

Оцінка змін:

- Усі подані запити розглядаються командою розробників та бізнес-аналітиками.
- Команда оцінює вплив змін на систему, графік розробки та ресурси.
- Після оцінки команда складає звіт, який містить аналіз наслідків та рекомендації щодо впровадження або відхилення запиту.

Затвердження змін:

- Після оцінки запит на зміну подається на розгляд керівнику проекту та зацікавленим сторонам для затвердження.
- Остаточне рішення щодо затвердження чи відхилення зміни приймається на спільній зустрічі за участю керівників, розробників та бізнес-аналітиків.

Впровадження змін:

- Після затвердження зміни команда розробників приступає до реалізації відповідних оновлень у проекті.
- Зміни вносяться в SRS, а також в інші відповідні документи проекту.
- Оновлені версії документів передаються усім зацікавленим сторонам.

Контроль версій:

- Кожна зміна в SRS супроводжується оновленням версії документа, яка відображає внесені зміни.
- Історія змін зберігається у вигляді журналу з детальним описом кожної внесеної зміни, включаючи дату, автора змін та затверджену версію.

А. Додатки

Д.1 Протоколи зустрічей:

Кожен протокол зустрічі включає такі основні компоненти:

- Дата та час зустрічі: Фіксація часу, коли відбулася зустріч, щоб забезпечити хронологічний порядок подій.
- Учасники: Повний список присутніх на зустрічі, включаючи представників замовника, членів команди розробників, бізнес-аналітиків та інших зацікавлених сторін.
- Порядок денний: Основні теми, які були обговорені під час зустрічі, включаючи нові вимоги, запити на зміни, уточнення функцій або технічні деталі проекту.
- Основні обговорені питання: Стислий виклад ключових моментів обговорення, наприклад, пропозиції щодо нових функцій, технічні вимоги або виправлення проблем у поточному стані проекту.
- Прийняті рішення: Конкретні рішення, які були ухвалені під час зустрічі, з зазначенням відповідальних за їх виконання.
- План дій: Перелік завдань та термінів для членів команди, що впливають із прийнятих рішень, а також зазначення осіб, відповідальних за кожну дію.
- Наступні кроки та дати: Опис наступних етапів проекту, зустрічей чи перевірок стану роботи, а також зазначення термінів для виконання поставлених завдань.

Приклади протоколів зустрічей включають узгодження загальних вимог до системи, обговорення функціональних можливостей інтерфейсу, а також перегляд нефункціональних вимог до безпеки та продуктивності системи.

Д.2 Технічна документація:

Основні компоненти технічної документації:

- Опис архітектури системи: Загальна структура системи DevLib, включаючи опис серверної та клієнтської частини, бази даних і взаємодії між ними.
- Технічні специфікації: Включає докладні описи всіх компонентів і модулів, які складають систему, такі як API, бібліотеки, середовища розробки та інструменти, що використовуються для створення системи.
- Інтеграційні схеми: Схеми, що демонструють, як різні частини системи DevLib взаємодіють одна з одною та з зовнішніми сервісами або інтерфейсами.
- Керівництво для розробників: Документація, що описує правила і найкращі практики для розширення та підтримки системи, включаючи інструкції з розгортання, налаштування і тестування.

Ця документація гарантує, що всі технічні аспекти проекту DevLib розуміються однозначно і що кожен член команди розробників має доступ до необхідної інформації для успішної реалізації проекту.