

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА»
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

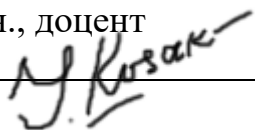
на тему

РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ПІДТРИМКИ ТА РОЗВИТКУ
ЧИТАЦЬКОЇ СПІЛЬНОТИ

здобувача Кулішової Єлизавети Сергіївни _____

Науковий керівник:

к.т.н., доцент

_____  Козак І.А.

Проект допущений до захисту
перед екзаменаційною комісією з
атестації здобувачів вищої освіти

завідувач кафедри:

к.е.н., доцент.

_____ Тішков Б.О.

Київ 2024

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО_ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:
Керівник проектної групи(гарант)
освітньо-професійної програми
_____Іванченко Г.Ф.
“ ____ ” _____ 2024 р.

ЗАТВЕРДЖУЮ:
Завідувач кафедри
_____Тішков Б.О.
“ ____ ” _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Кулішовій Єлизаветі Сергіївні

денної форми навчання

на підготовку кваліфікаційного бакалаврського проекту

на тему: **«Розробка веб-застосунку для підтримки та розвитку читачької спільноти»**

Тему затверджено наказом ректора Університету від «20» лютого 2023 р .№ 332-ст.

Кваліфікаційний бакалаврський проект виконується на матеріалах

навчальних посібників та матеріалів з мережі Інтернет. _____

План кваліфікаційного бакалаврського проекту

Розділ 1	Характеристика та аналіз предметної області (назва розділу)
Розділ 2	Проектування системи (назва розділу)

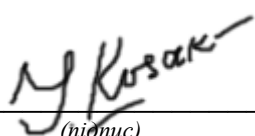
Розділ 3	Проектування та реалізація компонентів підсистеми (назва розділу)
----------	--

Об'єкт дослідження:	Процеси формування та функціонування читацької спільноти в умовах війни.
Предмет дослідження:	Розробка веб-застосунку для підтримки читачів, включаючи аналіз потреб аудиторії та створення інструментів доступності літературних ресурсів.
Мета кваліфікаційної бакалаврської роботи:	Розробка веб-застосунку, який посприє зручному пошуку та взаємодії з книжковим фондом, підвищує рівень особистого читацького досвіду та взаємодії з читачами.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі 1	<u>Надати характеристику та аналіз предметної області та об'єкту дослідження. Визначити сучасний стан проблеми, її актуальність та проаналізувати аналогічні веб-застосунки, розробити концепцію інформаційної системи.</u>
У розділі 2	<u>Надати фактологічну інформацію та її аналіз. Сформулювати вимоги до системи, підхід до моделювання, вхідну та вихідну інформацію та проектування інформаційної моделі задачі. Описати математичне забезпечення та алгоритм функціонування системи.</u>
У розділі 3	<u>Продемонструвати та описати різні види забезпечень, які необхідні для роботи інформаційної системи: інформаційне, технічне та програмне.</u>

Завдання підготував
науковий керівник


(підпис)

Козак Ірина Антонівна
(ініціали, прізвище)

« ____ » _____ 2024р.

Завдання одержав
здобувач

(підпис)

Кулішова Єлизавета Сергіївна
(ініціали, прізвище)

« ____ » _____ 2024р.

Відгук

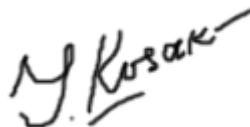
наукового керівника про кваліфікаційний бакалаврський проект
студента Навчально-наукового інституту
“Інститут інформаційних технологій в економіці”
галузі знань 12 “Інформаційні технології”
спеціальності 122 “Комп’ютерні науки”
Кулішової Єлизавети Сергіївни

на тему: « **РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ПІДТРИМКИ ТА
РОЗВИТКУ ЧИТАЦЬКОЇ СПІЛЬНОТИ**»

1. **Актуальність теми:** кваліфікаційний бакалаврський проект присвячений актуальній проблемі розробки ІС веб-застосунку для підтримки та розвитку читачької спільноти, яку пропонується розв’язувати та вдосконалювати сучасних інформаційних технологій.
2. **Позитивні риси кваліфікаційного бакалаврського проекту:** Студентка в процесі роботи над проектом показала себе як сформований фахівець, що може самостійно здійснювати ставити задачі та розробляти спроектовані інформаційні системи.
3. **Наявність самостійних розробок:** Студенткою самостійно здійснено проектування та розробку веб-додатку для зручного пошуку та взаємодії з книжковим фондом, підвищення рівня особистого читачького досвіду та взаємодії з читачами.
4. **Наявність недоліків:** система може бути в подальшому вдосконалена за рахунок покращення та розширення функціональних можливостей.
5. **Загальна оцінка кваліфікаційного бакалаврського проекту та його допущення до захисту перед Екзаменаційною комісією:** кваліфікаційний бакалаврський проект студентки Кулішової Є.С. відповідає встановленим вимогам, рекомендується до захисту і заслуговує високої позитивної оцінки.

Науковий керівник

доцент, кандидат економічних наук



Козак І. А.

В і д г у к
про проходження виробничої практики
здобувача навчально-наукового інституту
освітньо-професійної програми «Комп'ютерні науки»
Кулішової Єлизавети Сергіївни
В «Інституті телекомунікацій і глобального інформаційного простору
НАН України»

на тему «Розробка ІС веб-застосунку для розвитку та підтримки читацької спільноти»

Кулішова Єлизавета Сергіївна пройшла виробничу практику на базі «Інституту телекомунікацій і глобального інформаційного простору НАН України», яка тривала з 29-го квітня по 3-тє червня.

Протягом періоду практики виявила дисципліну та зацікавленість у роботі над проектом. Продемонструвала глибокі знання у сфері інформаційних технологій, а також здатність ефективно застосовувати їх на практиці.

Успішно виконала всі завдання, поставлені в рамках практики. Продемонструвала високий рівень професійних знань та навичок, здатність самостійно вирішувати складні задачі та ефективно застосовувати теоретичні знання на практиці.

Заслужовує високої оцінки за свою роботу під час проходження виробничої практики.

Керівник виробничої практики

Завідувач відділу досліджень навколишнього середовища

д.т.н., професор
(посада, учене звання, науковий ступінь)

Триснюк В.М.
(прізвище, ініціали)


(підпис)

«_3_» червня 2024р.

Власноручний підпис д.т.н., проф. Триснюка В.М. завіряю:
Вчений секретар Інституту телекомунікацій і глобального інформаційного простору НАН України



АНОТАЦІЯ

кваліфікаційного бакалаврського проекту студентки 4 курсу

Навчально-наукового інституту «Інститут інформаційних технологій в економіці»

Кулішової Єлизавети Сергіївни, виконаної на тему:

«Розробка веб-застосунку для підтримки та розвитку читацької спільноти»

Київ: кафедра інформаційних систем і економіки, 2024 р.

Кваліфікаційний бакалаврський проект присвячений розробці веб-застосунку для підтримки та розвитку читацької спільноти з використанням сучасних інформаційних технологій.

Робота складається з трьох розділів, логічно пов'язаних між собою.

В першому теоретико-методичному розділі дана характеристика та аналіз предметної області. Було визначено сучасний стан проблеми, її актуальність та проаналізовано аналогічні веб-застосунки, розроблено концепцію інформаційної системи.

Другий розділ є аналітико-дослідницьким, в якому міститься фактологічна інформація та її аналіз. Сформовано вимоги до системи, підхід до моделювання, вхідна та вихідна інформація та проектування інформаційної моделі задачі. Також описано математичне забезпечення та алгоритм функціонування системи.

Третій розділ – конструктивний. Демонструються різні види забезпечень, необхідні для роботи інформаційної системи: інформаційне, технічне та програмне.

У висновках міститься узагальнення результатів роботи у вигляді обґрунтованих підсумків та пропозицій.

РЕФЕРАТ

Кваліфікаційний бакалаврський проект містить 126 сторінок, 13 таблиць, 104 рисунки, список використаних джерел з 20 найменувань, додатки.

РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ПІДТРИМКИ ТА РОЗВИТКУ ЧИТАЦЬКОЇ СПІЛЬНОТИ

Перелік ключових слів: інформаційна система, веб-застосунок, книжки, читачі, проектування, розробка, вимоги, діаграми.

Предметом дослідження є розробка веб-застосунку для підтримки читачів, включаючи аналіз потреб аудиторії та створення інструментів доступності літературних ресурсів.

Об'єктом дослідження кваліфікаційного бакалаврського проекту є процеси формування та функціонування читацької спільноти в умовах війни.

Мета кваліфікаційного бакалаврського проекту полягає в розробці веб-додатку, який посприяє зручному пошуку та взаємодії з книжковим фондом, підвищує рівень особистого читацького досвіду та взаємодії з читачами.

Завданнями кваліфікаційного бакалаврського проекту є:

- проаналізувати потреби читачів та ринку літературних ресурсів для визначення ключових функцій веб-застосунку;
- дослідити існуючі аналогічні системи створені для читацької спільноти;
- сформулювати вимоги та основний функціонал веб-застосунку для читачів;
- спроектувати зручний інтерфейс веб-додатку, спрямованого на максимальне задоволення потреб користувачів;
- вибрати методології проектування веб-застосунку;
- визначити вхідні та вихідні дані, які є необхідними для веб-додатку;
- сформулювати математичне забезпечення та алгоритм функціонування системи;

- визначити забезпечення інформаційної системи та її опис;
- розробити тест-кейси до веб-додатку та потрасувати вимоги;
- виконати частину програмної реалізації веб-застосунку.

Апаратні засоби, що використовувались при проектуванні: Asus VivoBook 15.

Результати, досягнуті під час проектування: використані методи дослідження включали в себе аналіз існуючих інформаційних систем, розробку функціональної моделі та використання програмних середовищ для реалізації функціоналу. Новизна полягає у проектуванні веб-застосунку, який сприяє зручному пошуку та взаємодії з книжковим фондом, підвищує рівень особистого читацького досвіду та взаємодії з читачами з використанням сучасних методів моделювання. На даний момент система знаходиться на етапі розробки, і ступінь впровадження буде повніше визначений після завершення проекту.

Одержані результати можуть бути використані у сфері літературної взаємодії, надаючи читачам практичні та інноваційні інструменти для обміну враженнями від прочитаних творів. Спроековано веб-застосунок, який дозволяє здійснювати зручний пошук та взаємодію з книжковим фондом, збільшуючи рівень особистого читацького досвіду та комунікації з читачами. Рекомендації щодо практичного застосування отриманих результатів включають співпрацю з видавництвами та інтернет-магазинами, які пов'язані з довколакнижковою тематикою.

Рік виконання кваліфікаційного бакалаврського проекту – 2024.

Рік захисту кваліфікаційного бакалаврського проекту – 2024.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ	3
ВСТУП.....	5
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. 8	8
1.1. Характеристика предметної області та об'єкта дослідження	8
1.2. Дослідження існуючих інформаційних систем для обраної предметної області.....	10
1.3. Розробка концепції інформаційної системи.....	21
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ.....	46
2.1. Формування вимог	46
2.1.1. Бізнес-вимоги до системи.....	46
2.1.2. Функціональні вимоги до системи	47
2.1.3. Нефункціональні вимоги до системи	49
2.2. Постановка та алгоритм розв'язання задачі	52
2.2.1. Характеристика задачі	52
2.2.2. Вихідна інформація.....	54
2.2.3. Вихідна інформація.....	57
2.2.4. Алгоритм розв'язання задачі.....	60
2.3. Моделювання інформаційної підсистеми	71
2.3.1. Моделювання поведінки системи	71
2.3.2. Модель структури системи.....	84
2.3.3. Розподіл вимог за компонентами системи	88
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ ПІДСИСТЕМИ	93
3.1. Інформаційне забезпечення.....	93
3.2. Технічне забезпечення	109
3.3. Програмне забезпечення.....	113
ВИСНОВКИ	123
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	125

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ

- АРМ – автоматизоване робоче місце;
- БД – база даних;
- ІЗ – інформаційне забезпечення;
- ІС – інформаційна система;
- КТЗ – комплекс технічних засобів;
- ОС – операційна система;
- ПЗ – програмне забезпечення;
- ПК – персональний комп’ютер;
- СКБД – система управління базами даних;
- API – Application Programming Interface, інтерфейс прикладного програмування;
- Backlog – артефакт зібраних вимог до програмного продукту;
- CSS – Cascading Style Sheets, каскадні таблиці стилів;
- HTML – HyperText Markup Language, мова розмітки гіпертексту;
- JavaScript – динамічна, високорівнева, об’єктно-орієнтована прототипна мова програмування;
- М – кількість доданих книг;
- MVC – Model-View-Controller, схема розділення даних додатків і керуючої логіки на три окремі компоненти;
- N – кількість книг у списку після додавання;
- N_0 – кількість книг у списку до додавання;
- n_i – кількість користувачів, які оцінили i -ту книгу;
- phpMyAdmin – веб-інтерфейс для адміністрування СКБД MySQL;
- p_0 – кількість друзів користувача o ;
- R_i – оцінка i -тої книги;
- Scrum – фреймворк для впровадження інкрементального та ітераційного підходу до розробки;
- SSD – Solid-State Drive, твердотілий накопичувач;

- s_u – кількість книг у списку u ;
- SysML – The Systems Modeling Language, мова моделювання систем;
- UI/UX – User Experience/User Interface, взаємодія з користувачем/інтерфейс користувача;
- UML – Unified Modeling Language, уніфікована мова моделювання.

ВСТУП

У сучасному цифровому світі інформаційні системи веб-застосунків відіграють важливу роль у взаємодії та комунікації користувачів. Розвиток інтернет-технологій та соціальних мереж відкриває нові можливості для підтримки та розвитку різноманітних спільнот, включаючи читацьку. Важливість розробки такої системи для підтримки та розвитку читацької спільноти полягає в важливості та необхідності надання зручного та ефективного інструменту для обміну інформацією, рекомендаціями та взаємодією читачів.

У зв'язку з повномасштабним вторгненням в Україні з 2022 року, є нагальна потреба у створенні інформаційної системи веб-додатку для підтримки та розвитку читацької спільноти. Цей застосунок може стати важливим механізмом психологічної підтримки та виконувати функцію відволікання від відчуття тривожності для військових, ветеранів та цивільних. Більш того, слід враховувати, що війна сприяє єдності у громадській сфері, і створення читацького угруповування може стати важливим кроком у підтримці культурних зв'язків та просуванні соціальної солідарності.

Дослідження, проведене «Інститутом майбутнього» [1], вказує на значний вплив повномасштабної війни на читацькі звички українського населення. Отримані результати показали, що під час війни читання книг виявилось для багатьох людей не лише способом розваги, а й способом підтримки психологічного здоров'я та емоційного стану. Деякі респонденти відзначили, що в період кризи змінили свої читацькі пристрасті, надаючи перевагу тематиці, що відображає військові події, публіцистиці та історії України. За результатами дослідження можна підсумувати, що книжкова культура є важливою складовою українського суспільства, сприяючи не лише особистісному розвитку, а й формуванню національної ідентичності та покращенню власної культурної спадщини.

Мета цього кваліфікаційного бакалаврського проекту полягає в розробці веб-додатку, який спрямований на полегшення пошуку та взаємодії з книжковим фондом, з метою підвищення рівня особистого читацького досвіду та підтримки активного спілкування між читачами.

Завданням дослідження є проведення аналізу потреб читачів та формування характеристики ринку аналогічних літературних веб-сервісів, щоб визначити ключові функціональні вимоги до веб-додатку. Передбачається проведення аналізу існуючих систем, які спрямовані на підтримку читацької спільноти, для виявлення їхніх переваг та недоліків. Слід також сформулювати вимоги до функціоналу веб-додатку, який максимально задовольнятиме потреби користувачів. Важливим етапом є проектування зручного інтерфейсу, що спростить взаємодію з додатком та забезпечить комфортне користування. Необхідно обрати найбільш підходящу методологію розробки, що найліпшим чином відповідає поставленим завданням та цілям. Визначення вхідних та вихідних даних для оптимального функціонування системи є також важливим кроком у розробці. Розробка математичного забезпечення та алгоритмів роботи системи передбачає створення ефективного програмного продукту. Крім того, слід ретельно описати забезпечення інформаційної системи та функціональні можливості веб-додатку. Для перевірки відповідності розробленого додатку вимогам та тестування його функціоналу, необхідно розробити відповідні тест-кейси. І нарешті, виконання частини програмної реалізації веб-додатку є завершальним етапом дослідження.

Об'єктом дослідження є процес формування та функціонування читацької спільноти в умовах війни, зокрема, як війна впливає на її структуру, змінює читацькі вподобання та сприйняття літературних творів, а також взаємовідносини між членами спільноти. Предметом дослідження є розробка веб-застосунку для підтримки та розвитку читачів, яка вивчає потреби читацької аудиторії, розробляє інструменти для покращення доступності літературних ресурсів, а також забезпечує платформою для взаємодії та підтримки користувачів у складні часи.

Було визначено функціональну модель задачі, характеристику самої задачі, вихідну та вхідну інформацію, а також розглянуто математичне забезпечення та алгоритм функціонування системи. Обрано об'єктний підхід (діаграми прецедентів та послідовності мовою UML) до проектування веб-застосунку з метою забезпечення комплексного та систематичного підходу до розробки. Було розроблено рішення для ІС за різними видами забезпечення. Інформаційна система ґрунтується на передових веб-технологіях, таких як HTML, CSS, та JavaScript, з метою створення інтуїтивно зрозумілого та функціонального інтерфейсу для кінцевих користувачів. Використання бази даних phpMyAdmin забезпечує збереження всієї важливої для системи інформації, яка буде потрібна для якісного обслуговування веб-додатку.

Результати, отримані в ході проектування: використані методи дослідження включали аналіз вже існуючих інформаційних систем, розробку моделі та використання різних програмних середовищ для впровадження функціоналу. Спроектовано веб-додаток, який слугує зручному пошуку та взаємодії з книжковим фондом, підвищує та покращує рівень особистого читацького досвіду та взаємодії з читачами. На даному етапі система знаходиться на етапі розробки, а ступінь впровадження буде визначений після завершення проекту більш повним.

Отримані результати можуть бути використані у області довколалітературної взаємодії, надаючи читачам зручний та інноваційний інструмент для обміну враженнями від прочитаних творів. Основними техніко-експлуатаційними характеристиками є інтуїтивний інтерфейс, підтримка кросплатформенності, безпека даних, а також інтеграція з зовнішніми системами.

Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Характеристика предметної області та об'єкта дослідження

У сучасному світі, коли доступ до книг стає більш доступним завдяки електронним та аудіокнигам, існує потреба в централізованій системі, яка об'єднує читачів, надає їм доступ до літературного контенту та сприяє обговоренню, обміну враженнями та створенню особистих списків книг. Відсутність централізованої платформи призводить до того, що користувачам доводилось шукати книги на різних ресурсах, користуватися різними додатками для аудіокниг та електронних книг, а також обмежуватися малою можливістю обговорення та обміну думками.

В Україні зростає популярність електронних книг та аудіокниг, а з нею і важливість надання читачам не лише доступу до літературного контенту, а й інтегрованої платформи, що сприяє активній взаємодії та обміну ідеями між членами спільноти [2], [3]. Відповідно до статистики 2023 року, 17% респондентів читають книги щодня у вільний час, тоді як у 2020 році таких було лише 8%, а у 2018 році - 11%. Українці віддають перевагу друкованим книгам, але їхня популярність знижується: 39% респондентів читали лише паперові книги у 2020 році, тоді як у 2023 році цей показник знизився до 21%. Частка читачів, які надають перевагу як друкованим, так і електронним книгам одночасно, зросла з 27% у 2020 році до 33% у 2023 році; у 2023 році 12% читачів надаватимуть перевагу лише електронним форматам (5% у 2020 році); у 2023 році 62% респондентів не придбали жодної книги, 22% придбають 1-4 книги, а 12% придбають 5-12 книг. Електронні книги та аудіокниги придбали 3% та 0,5% респондентів відповідно. Споживання електронних та аудіокниг зростає [4].

Обрана предметна область інформаційної системи спрямована на покращення літературної взаємодії – полегшення доступу користувачів до різноманітного літературного контенту та створення умов для комунікації

читачів. Вона об'єднує у собі електронні та аудіокниги, надає можливість спілкування в спільнотах, чатах, отримання оновлень та новин зі світу літератури. Користувачі можуть створювати власні списки читання, слідкувати за улюбленими авторами та ділитися своїми враженнями.

Організаційна структура цієї предметної області включає різні класи суб'єктів, кожен із яких виконує конкретні функції. Адміністратори відповідають за управління та модерацію системи. Вони займаються управлінням користувацькими обліковими записами, редагуванням контенту в системі та розв'язанням технічних питань. Модератори виконують роль у контролі за якістю та безпекою інформаційного вмісту. Вони відповідають за контроль взаємодії в спільнотах, видалення контенту, який може бути недоречним, та підтримку користувачів. Звичайні користувачі, як основна маса, взаємодіють з системою для задоволення своїх читацьких потреб. Вони додають книги, створюють рецензії, обговорюють та рекомендують літературні твори, створюють та беруть участь в читацьких спільнотах. Організаційна структура веб-застосунку враховує різні ролі суб'єктів системи для забезпечення ефективного та цілеспрямованого функціонування платформи.

Веб-додаток забезпечує комплексну інформацію для прийняття рішень, надаючи користувачам зручний доступ до потрібних даних. Інформація для прийняття рішень включає аналіз популярності книг серед читачів, статистику читання, а також рецензії та огляди інших користувачів. Інформація для прийняття рішень у системі зосереджена на взаємодії та обміні враженнями між користувачами, забезпечуючи доступ до рецензій, відгуків про книги та оцінок до книг. Такий підхід дозволяє користувачам приймати обґрунтовані рішення на основі особистого досвіду та вражень від читання. Таким чином, система забезпечує простий та емоційно насичений підхід до прийняття рішень серед читачів.

Щоб забезпечити інформацією система включає в себе новини та події зі світу літератури, анонси нових книг, інтерв'ю з авторами та іншу контекстну інформацію, що покращує досвід читання. Структура інформації для

прийняття рішень охоплює модулі, які дозволяють користувачам робити зручний вибір книг на основі власних уподобань і динамічної взаємодії зі спільнотою.

1.2. Дослідження існуючих інформаційних систем для обраної предметної області

Щоб краще зрозуміти предметну область і обґрунтувати необхідність розробки інформаційної системи, був проведений аналіз та порівняння із існуючими сервісами, які виконують ідентичні функції в області читання та літературної діяльності. Такий етап аналізу є необхідним у процесі розробки веб-додатку, оскільки він допомагає визначити можливості для покращення взаємодії користувачів. Серед подібних систем, які були проаналізовані, можна відзначити такі платформи, як *Goodreads*, *Rork*, *LibraryThing*, *Google Play Books* та *Bookmate*. Під час аналізу були розглянуті такі аспекти, як зручність використання веб-сервісу (безпечність, наочність сприйняття інформації, інтуїтивна зручність інтерфейсу, кросплатформеність, доступ на всіх пристроях, зручність використання за допомогою смартфона, безкоштовна версія продукту що забезпечує необхідні потреби, автентифікація через сторонні сервіси), подання інформації (достатня кількість місця для подання необхідної інформації, структуроване подання інформації, позначення важливої інформації, зручний пошук, синхронізація між пристроями, обкладинки книг, можливість додавати власні книги, наявність метаданих, наявність рекомендацій, аудіокниги, електронні книги, доступність на різних мовах) та комунікація з іншими користувачами (чат для спілкування, форуми для спілкування, написання рецензій, можливість додавання у друзів, оцінювання книг, спільнота авторів і читачів).

Загальні характеристики обраних систем:

Характеристика Goodreads

Goodreads є однією з провідних сервісів для зятятих читачів. Цей веб-застосунок спеціалізується на створенні угруповування однодумців та надає широкий спектр можливостей для організації та співпраці в області літературної діяльності [5]. На рис. 1.2.1. подано скріншот платформи Goodreads, яка відома своєю функціональністю та широким спектром можливостей для своїх користувачів:

- Профілі користувачів і особисті кабінети: Надає можливість користувачам створювати власні облікові записи, де вони можуть вести свою читацьку активність, додавати книги до списків читання та писати відгуки.
- Читацькі списки: Платформа дозволяє користувачам створювати різні списки книг, такі як «Прочитані», «Хочу прочитати», «Улюблені», «Читаю зараз» тощо. Це допомагає читачам впорядковувати свою бібліотеку та ділитися нею з іншими.
- Рецензії і оцінки: Користувачі можуть залишити їх власні рецензії та оцінки книгам, які вони прочитали. Ця функція дозволяє іншим читачам отримувати інформацію про книгу від інших реальних осіб.
- Спільноти і групи: Сервіс має багато різних читацьких груп та спільнот, де користувачі можуть обговорювати книги, брати участь у обговореннях та цікавих заходах.
- Рекомендації: Система рекомендацій застосунку допомагає користувачам знаходити нові книги виходячи з їхнього читацького смаку та історії читання.
- Мобільний додаток: Goodreads має мобільний додаток для смартфонів та планшетів, що дозволяє користувачам отримувати доступ до функцій облікового запису сайту з різноманітних пристроїв.
- Пошук та каталог книг: Платформа має широку варіацію книг з можливістю пошуку за назвами, авторами, жанрами, роками видання тощо.
- Інтеграція з зовнішніми сервісами: Можливість інтеграції через сторонні платформи, такі як Facebook.

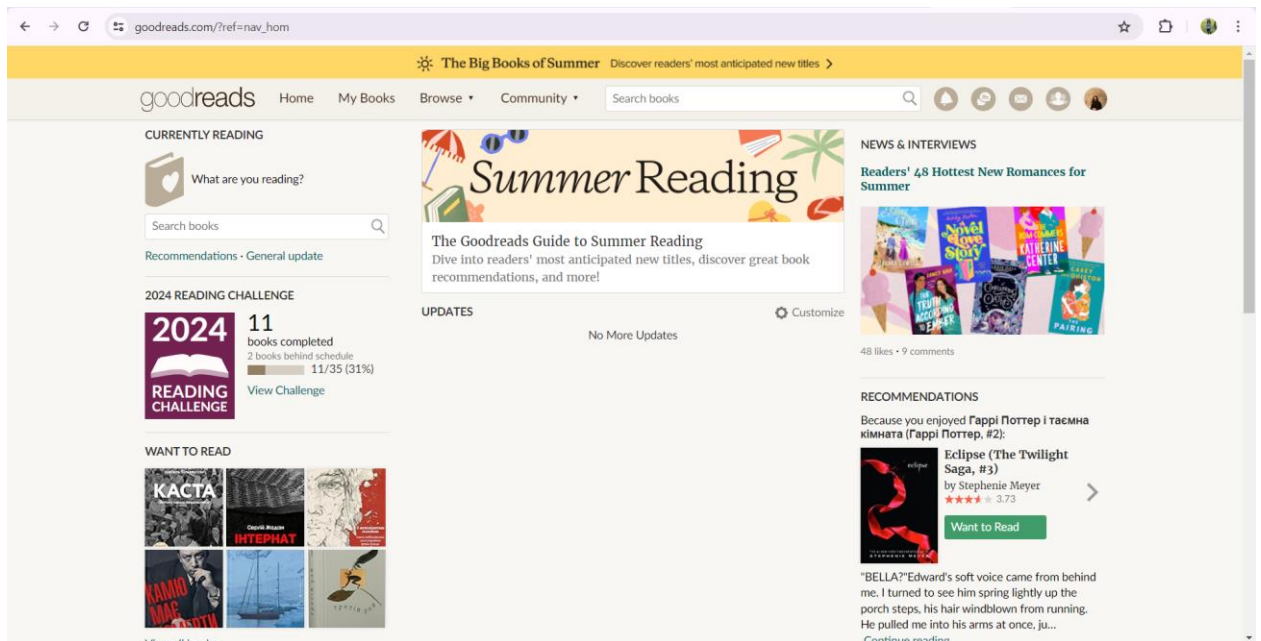


Рис. 1.2.1. – Веб-сервіс Goodreads

Загальна характеристика Rork

Мобільний додаток «Rork – Мистецтво читати» - це інноваційний читацький щоденник, який допомагає користувачам відстежувати своє просування у читанні та розвивати культуру [6]. На рис. 1.2.2. подано скріншот додатка Rork. Ось загальна характеристика цього додатка:

- **Трекінг читання:** Користувачі можуть відслідковувати кількість сторінок або час, який вони витрачають на читання кожної книги. Додаток розраховує, скільки часу ще потрібно, щоб завершити книгу, а також допомагає визначити швидкість читання.
- **Мінімальний час для читання:** Нагадує користувачам, що навіть невелика кількість часу на читання щодня може призвести до досягнення читацьких цілей.
- **Нотатки та цитати:** Додаток дозволяє користувачам робити замітки, виділяти цитати, що сприяє збереженню і організуванню важливі моменти з книги.
- **Приватність:** Користувачі можуть встановлювати книги як приватні, інші користувачі не бачитимуть назву, автора та обкладинку цих книг.
- **Спільні читання:** Користувачі можуть ділитися своїм читацьким прогресом та досягненнями з друзями у додатку.

- Створення списків і колекцій: Читачі можуть створювати структуровані списки книг, що допомагає краще планувати цілі.

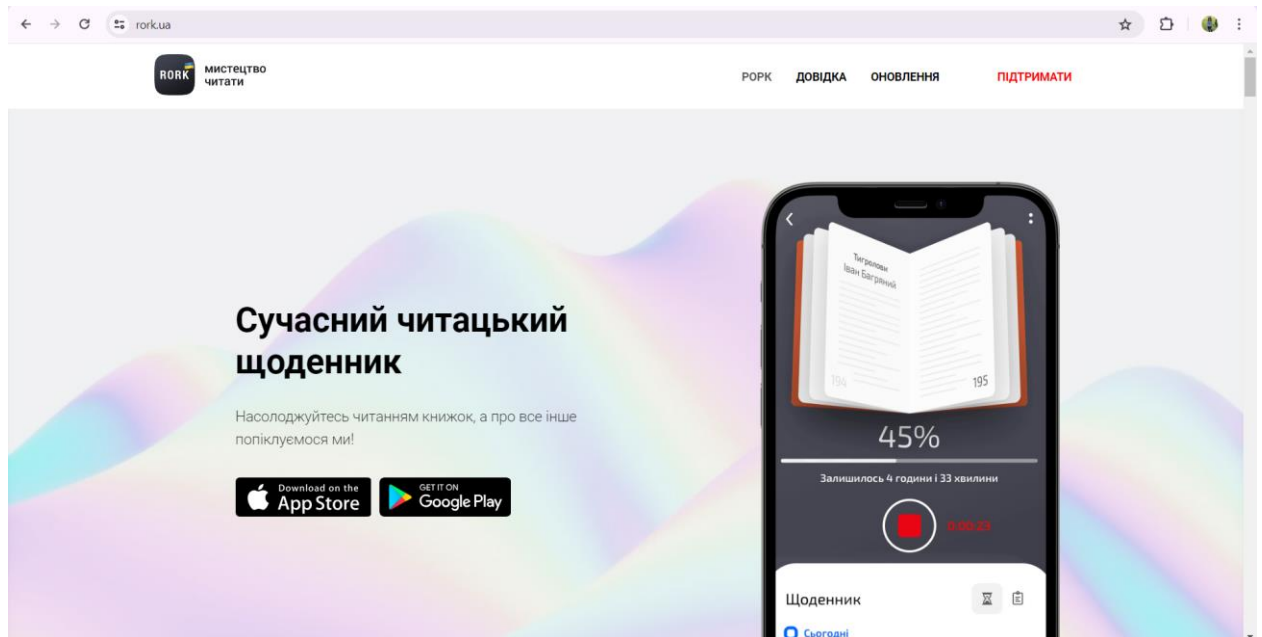


Рис. 1.2.2. – Веб-сайт додатку Rork

Загальна характеристика LibraryThing

LibraryThing – це веб-платформа для каталогізації та спільного обміну інформацією про книги. Вона надає користувачам можливість керувати своєю бібліотекою, спілкуватися з іншими читачами та отримувати рекомендації щодо книг [7]. На рис. 1.2.3. подано скріншот платформи LibraryThing. Основні характеристики включають наступне:

- Домашня сторінка: На домашній сторінці користувачі можуть переглядати рецензії, основні функції, новини та оновлення сайту.
- Профіль: Профіль користувача містить інформацію про нього і його бібліотеку, рекомендує інших користувачів зі схожими бібліотеками та дає можливість відстежувати активність друзів.
- Додавання книг: Користувачі можуть додавати книги до своєї бібліотеки.
- Списки: Додавання у списки, редагування та пошук всіх книг, які представлені у вигляді списку або обкладинок.
- Сторінки рецензій та авторів: Кожна книга, яку додає у список користувач, надає власну сторінку для книг та з'єднує з іншими користувачами,

які володіють цією книгою. Читачі також мають доступ до сторінок авторів до кожної рецензії, де надається цікава інформація про автора, події.

- Групи та форуми: Групи об'єднують користувачів через форуми обговорення різноманітних довколакнижкових тем.
- LibraryThing Local: Пов'язує читачів з місцевими книжковими магазинами, бібліотеками, а також з усіма книжковими подіями, які вони організовують.
- Zeitgeist: Збирає дані, складені з інформації про читачів та їхніх бібліотек, створює ще більше способів використання LibraryThing, включаючи можливість зовнішньої інтеграції зі своїми сторінками Facebook/Twitter.

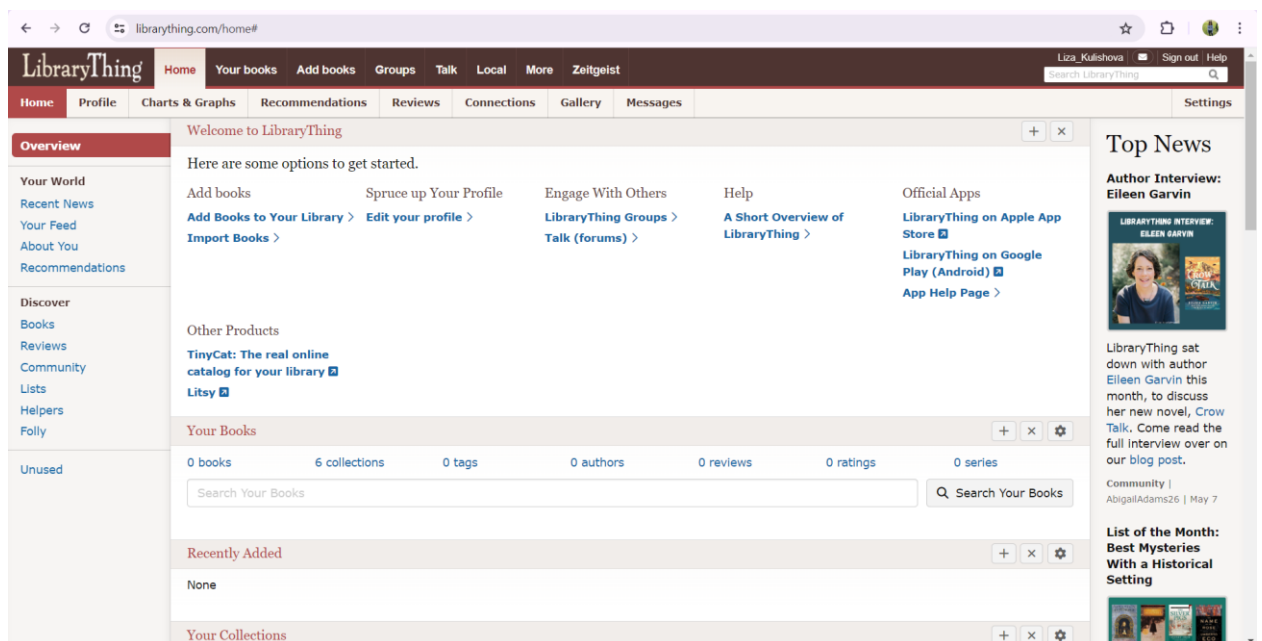


Рис. 1.2.3. – Веб-сервіс LibraryThing

Загальна характеристика Google Play Books

Google Play Books – це веб-платформа та мобільний додаток для читання електронних книг та аудіокниг, розроблений компанією Google. Цей сервіс пропонує великий вибір електронних книг і аудіокниг для читання та прослуховування на різних пристроях [8]. На рис. 1.2.4. подано скріншот додатку Google Play Books. Основні характеристики Google Play Books включають:

- Великий вибір книг: Google Play Books має широкий вибір електронних книг та аудіокниг різних жанрів.

- Доступ на різних пристроях: Можливість читати свої книги на різних пристроях, таких як смартфони, планшети, комп'ютери та електронні читалки. Прогрес читання синхронізується між пристроями, що дозволяє продовжувати читання з того місця, де користувач зупинився.
- Адаптивний дизайн: Підлаштовує вигляд тексту до розміру вашого екрану та налаштувань для максимальної зручності читання.
- Офлайн-режим: Можливість завантажити книги та аудіокниги для читання або прослуховування в офлайн-режимі, що дозволяє користуватися контентом навіть без доступу до Інтернету.
- Зручна інтеграція з Google: Google Play Books інтегровано з іншими послугами Google, такими як Google Drive, що дозволяє легко додавати свої електронні книги та зберігати їх у хмарі.
- Виділення тексту та нотатки: Можливість виділяти текст, робити замітки та позначати важливі місця в книзі, щоб повернутися до них пізніше.
- Спільний доступ: Користувач може ділитися своїми книгами та аудіокнигами з іншими користувачами або надавати доступ до них в режимі читання або слухання.
- Підказки та рекомендації: Сервіс надає персоналізовані рекомендації щодо книг на основі вашого читацького смаку.
- Підтримка багатьох мов і форматів: Google Play Books підтримує багато мов та різні формати електронних книг.
- Відслідковування прогресу читання: Сервіс веде статистику прогресу читання та надає інформацію про те, скільки часу залишилося до завершення книги.
- Покупка та безкоштовні книги: Можливість придбати книги або знайти безкоштовні варіанти.

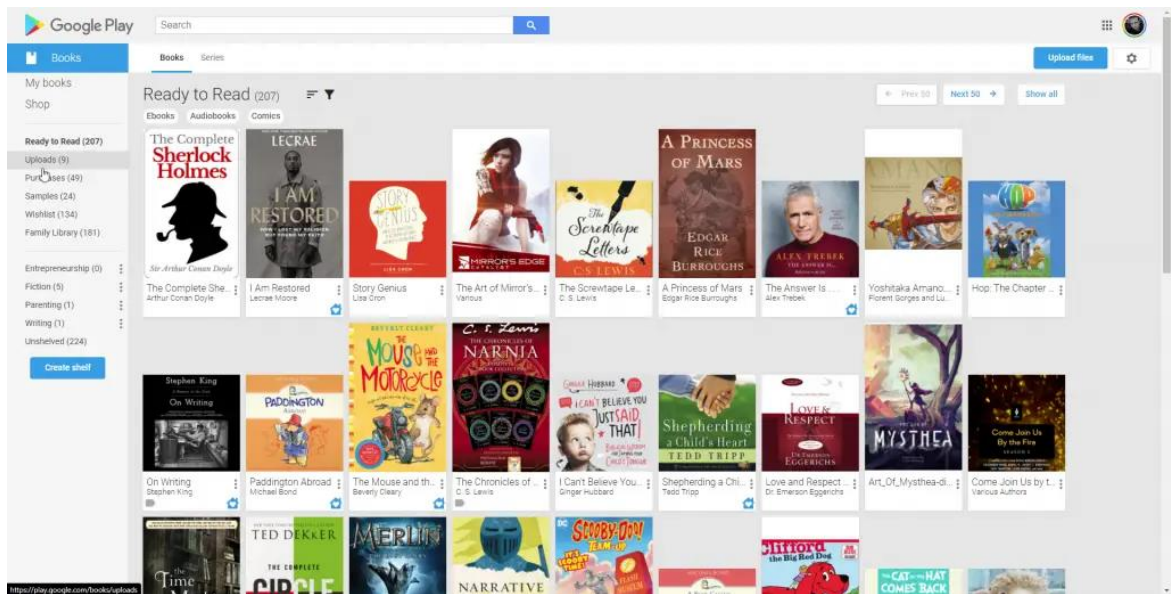


Рис. 1.2.4. – Веб-сервіс Google Play Books

Загальна характеристика Bookmate

Bookmate – це веб-платформа та мобільний додаток для читання електронних книг та прослуховування аудіокниг. Сервіс надає користувачам можливість мати доступ до великої кількості літературних творів різних жанрів і форматів [9]. На рис. 1.2.5. подано скріншот платформи Bookmate. Основні характеристики включають:

- Великий вибір: Сервіс пропонує більше мільйона книг та аудіокниг різних жанрів та авторів.
- Аудіокниги: У Bookmate є великий вибір аудіокниг, які можна слухати в режимі онлайн або завантажити для відтворення в режимі офлайн.
- Доступ на різних пристроях: Можливість читати або слухати книги на різних пристроях, включаючи смартфони, планшети, комп'ютери, тощо. Прогрес читання синхронізується між пристроями.
- Персоналізовані рекомендації: Додаток надає рекомендації щодо книг на основі читацького смаку та історії читання.
- Спільний доступ та обговорення: Можливість ділитися своїми враженнями від книг з іншими користувачами, обговорювати книги в коментарях, читати відгуки.
- Зручний пошук і фільтрація: Сервіс дозволяє швидко знаходити книги за ключовими словами, авторами та іншими параметрами.

- Безкоштовні варіанти і пробні періоди: Bookmate надає безкоштовний доступ до деяких книг і аудіокниг. Крім того, можна використовувати пробний період для оцінки сервісу перед підпискою.

Типи підписок:

- Преміум: придбання такої підписки надає можливість доступу до всіх електронних книжок. Також користувач отримує доступ до аудіокнижок та подкастів. Ця послуга поширюється на 1 обліковий запис.
- Книжки: за такою підпискою користувач має доступ тільки до електронних книжок. Ця послуга поширюється на 1 обліковий запис.
- Аудіо: за такою підпискою користувач має доступ тільки до аудіокнижок та подкастів. Ця послуга поширюється на 1 обліковий запис.
- Виділення тексту і збереження цитат: Можливість позначати важливі місця в книзі, виділяти та зберігати цитати.
- Офлайн-режим: Книги та аудіокниги можна завантажити для читання або прослуховування без доступу до Інтернету.
- Соціальна активність: Можна додавати друзів у Bookmate, взаємодіяти з ними та ділитися власними досягненнями у читанні.
- Безпека і конфіденційність: Застосунок забезпечує захист персональних даних та забезпечує конфіденційність усіх користувачів.

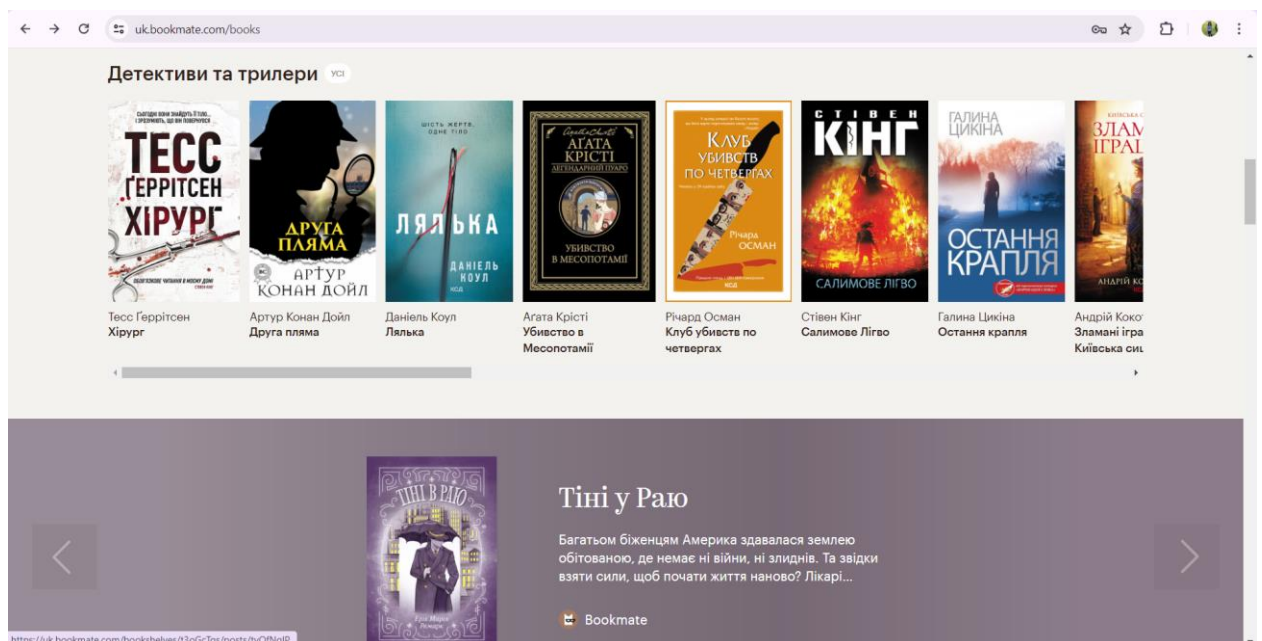


Рис. 1.2.5. – Веб-сервіс Bookmate

У таблиці 1.2.1. подано порівняння п'яти аналогічних систем, з метою визначити найкращу, та проаналізувати помилки конкурентів.

Таблиця 1.2.1. – Таблиця порівняння систем

№	Можливості	Goodreads	Rork	Library Thing	Google Play Books	Bookmate
1	2	3	4	5	6	7
Зручність використання						
1	Безпечність	так	так	так	так	так
2	Наочність сприйняття інформації	так	так	так	так	так
3	Інтуїтивна зручність інтерфейсу	ні	так	ні	ні	так
4	Кросплатформеність	так	так	так	так	так
5	Доступ на всіх пристроях	так	ні	так	так	так
6	Зручність використання за допомогою смартфона	так	так	так	так	так
7	Безкоштовна версія продукту що забезпечує необхідні потреби	так	так	так	так	так
8	Автентифікація через сторонні сервіси	так	ні	так	ні	так
Подання інформації						
9	Достатня кількість місця для подання необхідної інформації	так	так	так	так	так
10	Структуроване подання інформації	так	так	так	так	так
11	Позначення важливої інформації	так	так	так	так	так
12	Зручний пошук	так	так	ні	так	так
13	Синхронізація між пристроями	так	так	так	так	так
14	Обкладинки книг	так	так	так	так	так
15	Можливість додавати власні книги	ні	так	так	ні	так
16	Наявність метаданих	так	так	так	так	так
17	Наявність рекомендацій	так	ні	так	так	так
18	Аудіокниги	ні	ні	ні	ні	так
19	Електронні книги	ні	ні	ні	так	так

Продовження таблиці 1.2.1.

1	2	3	4	5	6	7
20	Доступність на різних мовах	так	ні	так	так	так
Спілкування						
21	Чат для спілкування	так	ні	так	ні	ні
22	Форуми для спілкування	так	ні	так	ні	ні
23	Написання рецензій	так	ні	так	так	так
24	Можливість додавання у друзів	так	так	так	ні	так
25	Оцінювання книг	так	так	так	так	так
26	Спільнота авторів і читачів	так	ні	так	ні	ні

Висновки за результатами аналізу програмних систем:

Отже, розглянувши таблицю можна побачити, що найкращою інформаційною системою для підтримки і розвитку читацької спільноти є Bookmate. Веб-застосунок має усі необхідні функції, які забезпечують зручне використання та комфортну подачу інформації. Ця платформа гарантує безпеку та чіткість розпізнавання інформації, має інтуїтивно зрозумілий інтерфейс та високий рівень кросплатформенності, що робить її доступною на різних пристроях. Можливість автентифікації через сторонні сервіси додатково підсилюють зручність використання. Стосовно подання інформації, Bookmate надає достатньо місця для зберігання важливої інформації, забезпечує її структурованість та надає можливість виділення важливих даних. Важливим також є зручний пошук і можливість синхронізації між пристроями. Присутність обкладинок книг, можливість додавати власні книги та наявність рекомендацій дозволяють користувачам легко знаходити та взаємодіяти з книгами різних жанрів та мов. У сфері комунікації користувачів Bookmate також має переваги, бо надає можливість писати рецензії, додавати друзів і оцінювати книги. Однак відсутність чату та форумів може бути недоліком для тих, хто шукає більше спілкування з іншими людьми. В цілому, Bookmate відповідає вимогам інформаційної системи для читачів і надає багатий функціонал для задоволення потреб книголюбів.

Друге місце розділяють Goodreads та LibraryThing. Обидві системи надають безпечну та доступну для користувачів інформаційну платформу. Вони зручно подають інформацію, а також забезпечують можливість автентифікації через зовнішні сервіси. Обидві платформи надають користувачам можливість використання функціоналу на смартфонах, а також мають безкоштовні версії продукту, що забезпечують необхідні потреби. Що стосується відображення інформації, обидві системи мають достатньо місця для зберігання необхідних даних та структурованого подання. Вони виділяють важливу інформацію та забезпечують можливість зручного пошуку та синхронізації між пристроями. Обкладинки книг, наявність рекомендацій сприяють приємному користуванню обома платформами. У сфері спілкування як Goodreads, так і LibraryThing пропонують користувачам засоби для обговорення книг і взаємодії з іншими людьми, включаючи чати, форуми, написання рецензій та можливість додавання у друзі. Це дозволяє користувачам оцінювати книги та приєднуватися до спільнот авторів і читачів.

Третє місце належить застосунку Google Play Books. Ця інформаційна система є захищеною і доступ до неї можна отримати з різних пристроїв. Вона дозволяє користувачам зручно сприймати інформацію та забезпечує достатньою кількістю місця для збереження необхідної інформації. Має добре структурований інтерфейс, де виділена вся важлива інформація, що полегшує користувачам пошук та навігацію. Однак, є деякі обмеження, серед яких відсутність можливості автентифікації за допомогою зовнішніх сервісів та відсутність можливості додавати інших користувачів у друзі, або спілкуватися у форматі чату та форумів. Крім того, система не дозволяє завантажувати власні аудіокниги та ділитися ними з іншими читачами. В області подання інформації застосунок надає користувачам обкладинки книг, наявність рекомендацій, а також можливість доступу до електронних книг різними мовами. Оцінювання книг та написання рецензій є частиною функціоналу. Загалом, Google Play Books є корисною інформаційною системою для користувачів, які хочуть отримати доступ до електронних книг різними мовами

та можливістю легко читати книги. Однак, в порівнянні з іншими системами, вона має обмеження щодо спілкування та соціальної взаємодії.

Останнє місце посів мобільний додаток «Rork – Мистецтво читати». Ця інформаційна система відзначається обмеженою доступністю на різних пристроях, таких як комп'ютери чи мобільні телефони. Що стосується подання інформації, застосунок пропонує структуроване відображення даних, але в ньому відсутні такі ключові функції: рекомендації та доступ до аудіокниг, або електронних книг. Це робить його менш зручним та доступним для користувачів, які шукають розширені можливості читання та взаємодії з іншими людьми. У сфері спілкування Rork не має чату, форумів чи можливості написання рецензій, що обмежує спілкування та взаємодію користувачів. Однак, є можливість додавання користувачів у друзі та оцінювання книг. Загалом, мобільний додаток «Rork – Мистецтво читати» характеризується простотою використання та структурованою подачею інформації, але при цьому він обмежений в області комунікативних функцій та наданні розширених можливостей користувачам.

1.3. Розробка концепції інформаційної системи

В якості архітектурного патерну для розробки ІС було обрано Model-View-Controller (MVC). Цей патерн забезпечить ефективне розділення компонентів системи, дозволяючи відокремити логіку програми (Модель), представлення даних та взаємодію з користувачем (Вид) і управління подіями та взаємодією між Моделлю та Видом (Контролер). Це дозволить досягти високого рівня модульності та розширюваності системи.

Система для підтримки та розвитку читацької спільноти буде спроектована з використанням сучасних веб-технологій для забезпечення ефективного та інтуїтивно зрозумілого інтерфейсу користувача. Використання бази даних phpMyAdmin забезпечує зберігання всієї необхідної інформації для якісного обслуговування веб-застосунку. Стандартизована мова розмітки

HTML (Hypertext Markup Language) використовуватиметься для створення структури сторінок, CSS (Cascading Style Sheets) буде використано для стилізації та оформлення веб-елементів, забезпечуючи естетичний вигляд інтерфейсу. JavaScript, як сучасна мова програмування для веб-розробки, забезпечить динамічність та інтерактивність веб-сторінок, дозволяючи користувачам ефективно взаємодіяти з системою без перезавантаження сторінок. Використання фреймворку на базі моделі MVC допоможе впорядкувати код, роблячи його легко зрозумілим та розширюваним. Такий підхід також сприяє швидкій розробці та внесенню змін у систему. Розробка системи буде вестися з дотриманням найновіших стандартів та рекомендацій у сфері веб-розробки.

Для початку треба було сформувати *Backlog* проекту. Backlog – це список завдань, функціональностей або ідей, які потенційно можуть бути включені в майбутні версії продукту або проекту. Даний функціонал розписаний у табл. 1.3.1. за пріоритетністю і всього має 10 пунктів.

Таблиця. 1.3.1. – Backlog

Priority	User Story	Результат
1	2	3
1	Користувач хоче створити новий обліковий запис веб-додатку, або увійти у вже створений обліковий запис	ІС містить всі необхідні засоби аби користувач мав змогу створити обліковий запис, або увійти у систему
2	У кожного користувача має бути особистий профіль	Користувачі можуть переглядати та редагувати свої профілі, включаючи фото профілю, ім'я та інші деталі
3	У веб-додатку користувач хоче знайти докладну інформацію про книгу: - Назва книги - Автор книги - Обкладинка - Короткий опис сюжету - Рецензії (написання рецензії) - Оцінка та рейтинг (можливість оцінювати прочитану книгу) - Цитати (можливість занотувувати та цитувати уривки з книг) - Наявність електронної книги - Наявність аудіокниги - Можливість додати книгу до списку - Відмітити книгу як вже прочитану/читаю зараз/планую читати або видалити статус	Розроблена сторінка з докладною інформацією про кожну книгу, включаючи назву книги, його автора, обкладинку, короткий опис сюжету, рецензії, оцінки та рейтинг, цитати, наявність електронної або аудіокниги, можливість додати книгу до списку користувача, відмітити книгу як вже прочитану/читаю зараз/планую читати або видалити статус. Функціонал розроблений для зручного користування

Продовження таблиці 1.3.1.

1	2	3
4	Користувач хоче додати книгу до свого списку	Було додано можливість додавати нові книги до власного списку користувача
5	Користувач хоче мати змогу створити новий список для книг	Користувач може створити власний список за бажанням, надавши йому власну назву та колір
6	Користувач хоче поділитись своїми враженнями від прочитаної книги, або знайти однодумців та нових друзів у локальних спільнотах	Створення для користувачів середовища для спілкування у вигляді приватних чатів, можливістю написання рецензій та постів, спільнот. Існує можливість додати іншого користувача у друзі, читати блоги улюблених авторів
7	Користувач хоче прочитати книгу в електронному форматі, або послухати книгу	Створення можливості прочитання книги у веб-додатку в електронному форматі та також наявність плеєра для комфортного прослуховування з можливістю керування записом (кнопки програвання, слайдер звуку)
8	Користувач хоче мати змогу змінювати налаштування та приватність	Додати розділ з налаштувань для користувачів та можливість керування приватністю профілю
9	Користувач хоче прочитати останню інформацію про оновлення веб-додатка	Додати розділ із повідомленнями про оновлення та відповіді на питання користувачів
10	Користувач хоче розширити свої можливості у використанні веб-додатка та мати безлімітний доступ до всіх постів, електронних та аудіокниг	Користувач отримує доступ до розширеного функціоналу, що дозволяє йому зручно взаємодіяти з додатком

Після формування Backlog було створено прототип інформаційної системи. З аналізу поданих вимог було прийнято рішення створити веб-додаток для полегшення та покращення свого читацького досвіду. Під час створення прототипу були задіяні знання зі сфери UI-UX дизайну. Метою було розробити сучасний та креативний інтерфейс.

- Первинний описовий екран для ознайомлення з веб-додатком

Цей екран створений для ознайомлення користувача з головними ідеями додатка (рис. 1.3.1 – 1.3.7.). Він складається із семи частин, а також включає у себе форми для входу або створення нового облікового запису.



Рис. 1.3.1. – Первинний екран (перша частина)

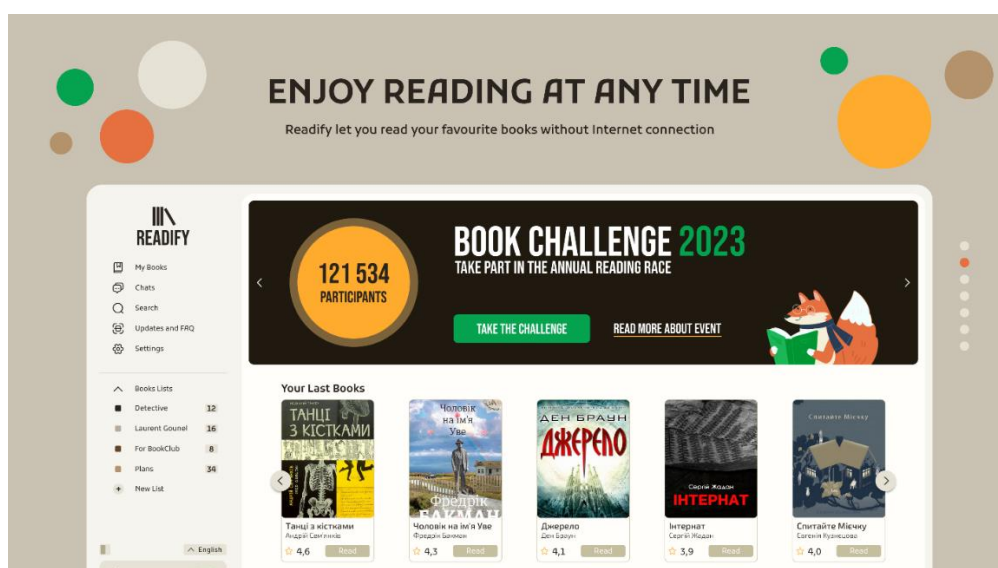


Рис. 1.3.2. – Первинний екран (друга частина)

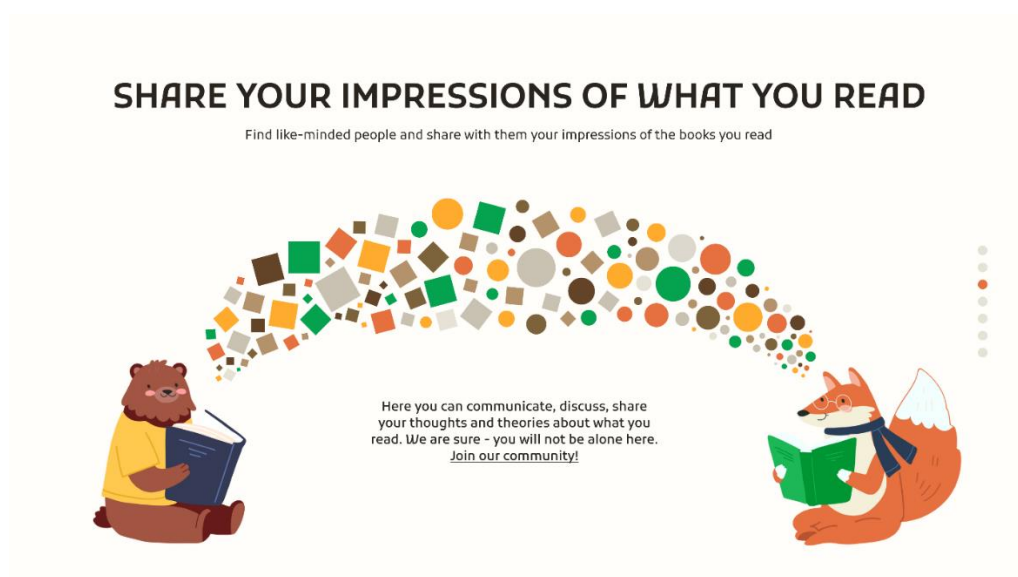


Рис. 1.3.3. – Первинний екран (третья часть)



Рис. 1.3.4. – Первинний екран (четверта частина)

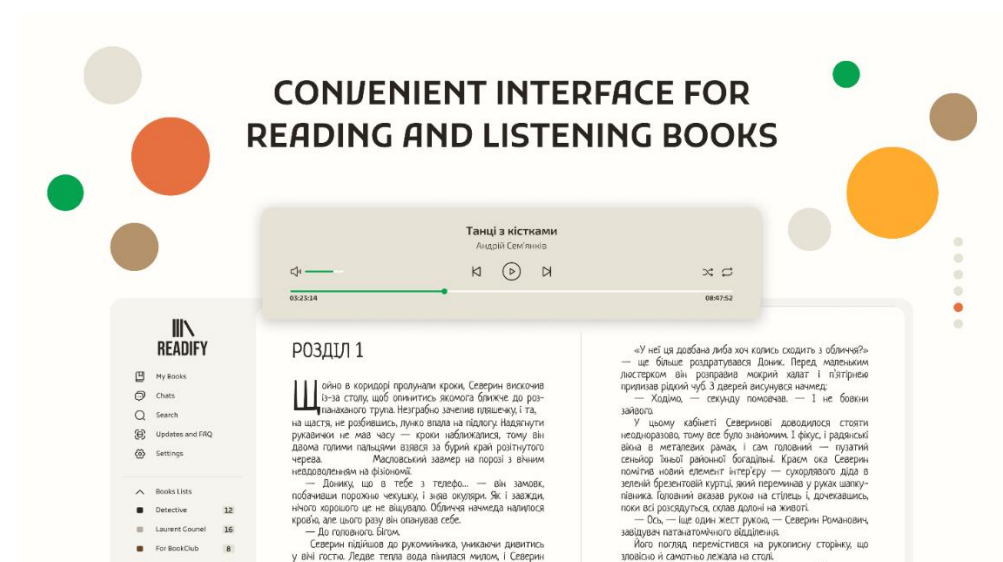


Рис. 1.3.5. – Первинний екран (п'ята частина)

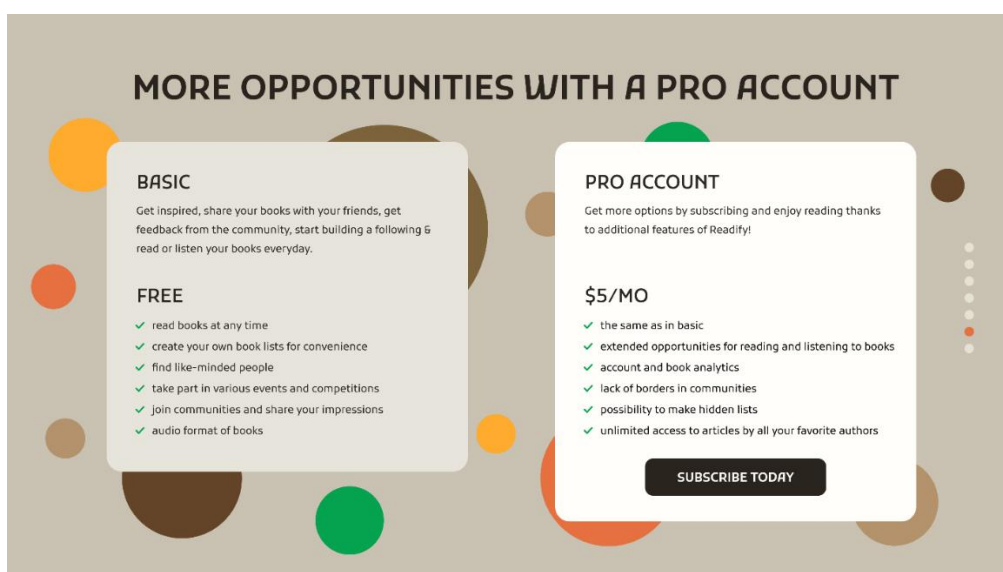


Рис. 1.3.6. – Первинний екран (шоста частина)

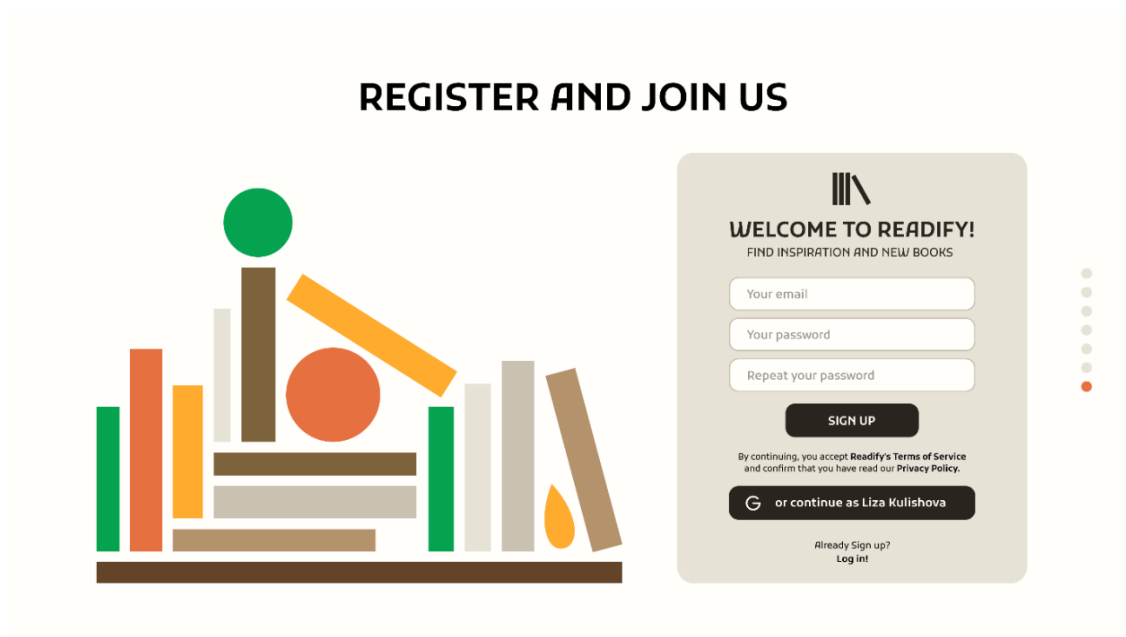


Рис. 1.3.7. – Первинний екран (сьома частина)

Нижче подано загальний футер веб-застосунку (рис. 1.3.8) та три екранні форми входу в існуючий обліковий запис, реєстрації на сайті та відправки форми зворотного зв'язку (рис. 1.3.9. – 1.3.11.).

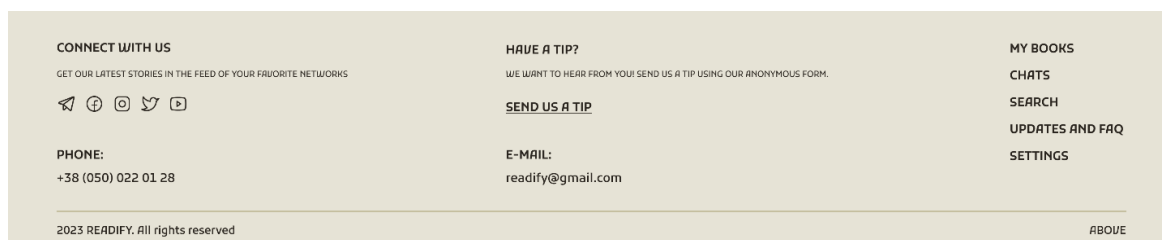


Рис. 1.3.8. – Футер веб-додатку

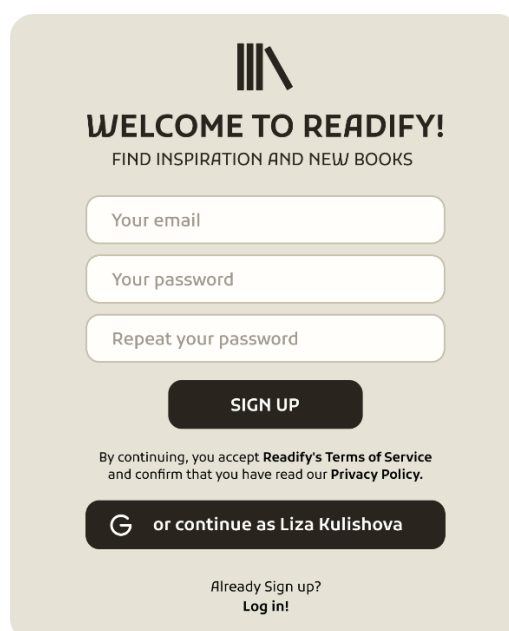


Рис. 1.3.9. – Форма реєстрації користувача

The image shows a login form for 'Readify!'. At the top is a logo consisting of three vertical bars of increasing height followed by a stylized 'N'. Below the logo, the text 'WELCOME TO READIFY!' is displayed in bold, followed by 'FIND INSPIRATION AND NEW BOOKS' in a smaller font. There are two input fields: 'Your email' and 'Your password'. Below these is a dark button labeled 'LOG IN'. Further down, a line of text states: 'By continuing, you accept Readify's Terms of Service and confirm that you have read our Privacy Policy.' Below this is a dark button with a Google 'G' logo and the text 'or continue as Liza Kulishova'. At the bottom, there is a link that says 'Don't have an account yet? Sign up!'.

Рис. 1.3.10. – Форма входу в існуючий обліковий запис

The image shows a form titled 'SEND US A TIP'. It has a light beige background. At the top, the title 'SEND US A TIP' is in bold. Below it is a 'Your email' input field. Underneath that is a larger 'Your Message' text area. At the bottom of the form is a dark button labeled 'SEND'.

Рис. 1.3.11. – Форма зворотного зв'язку

- Головний екран після входу в обліковий запис або реєстрації

Даний екран створений для основної орієнтації по веб-застосунку. Користувач має зручну навігаційну панель зліва, яка містить у собі посилання на інші сторінки веб-додатка та списки книг користувача. Також можна змінити мову інтерфейсу, тему, приховати список книг та саму панель. Справа розташоване велике інформаційне поле на якому виносяться останні новини, участь у книжковому виклику, останні книги, які були додані до списків та рекомендовані книги, корисні дописи читацької спільноти, рецензії, блоги авторів та різноманітні тести для розваги. Зовнішній вигляд головного екрану в неповному варіанті можна побачити на рис. 1.3.12., а на рис. 1.3.13. екран у повному варіанті.

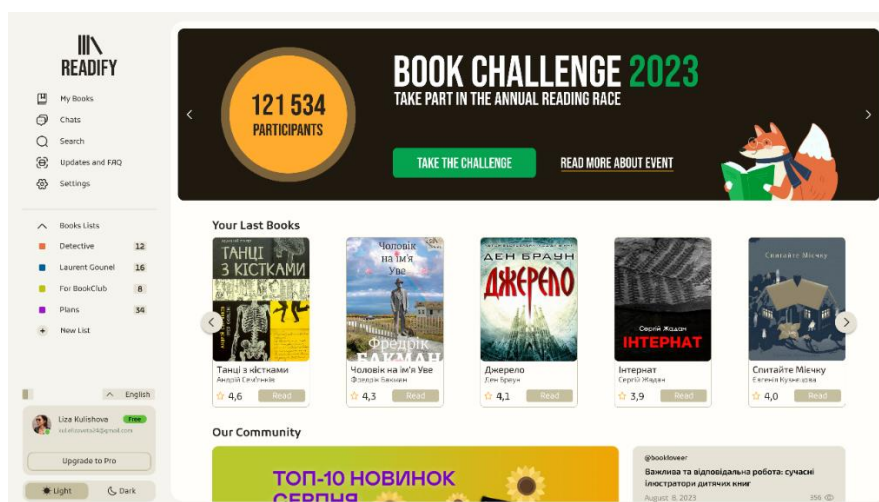


Рис. 1.3.12. – Зовнішній вигляд головного екрану (неповний варіант)

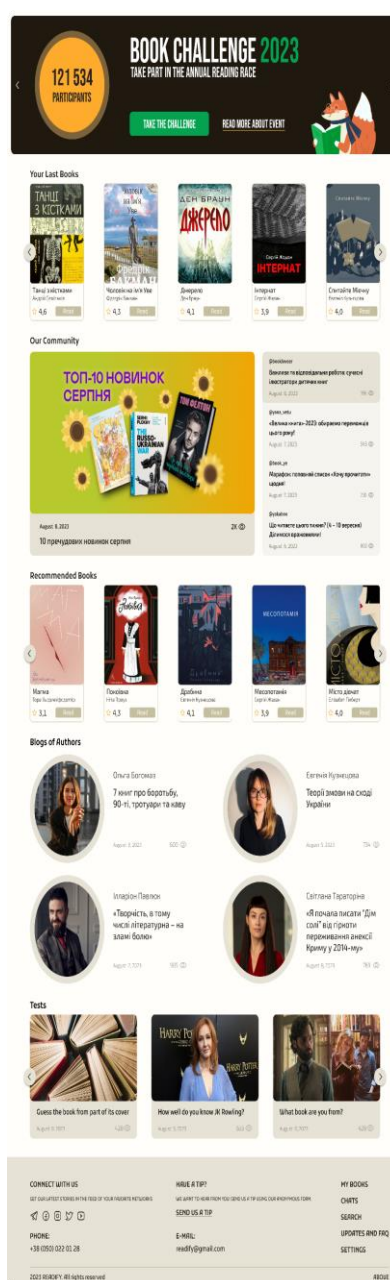


Рис. 1.3.13. – Зовнішній вигляд головного екрану (повний варіант)

Було показано анімацію деяких елементів на головній сторінці. На рис. 1.3.14 показано відкриття панелі, яка змінює мову у веб-додатку при натисканні. На рис. 1.3.15. показано процес закриття списків книг користувача. Закриття навігаційної панелі за допомогою кнопки виглядає як на рис. 1.3.16., а заміна теми веб-додатку продемонстрована на рис. 1.3.17.

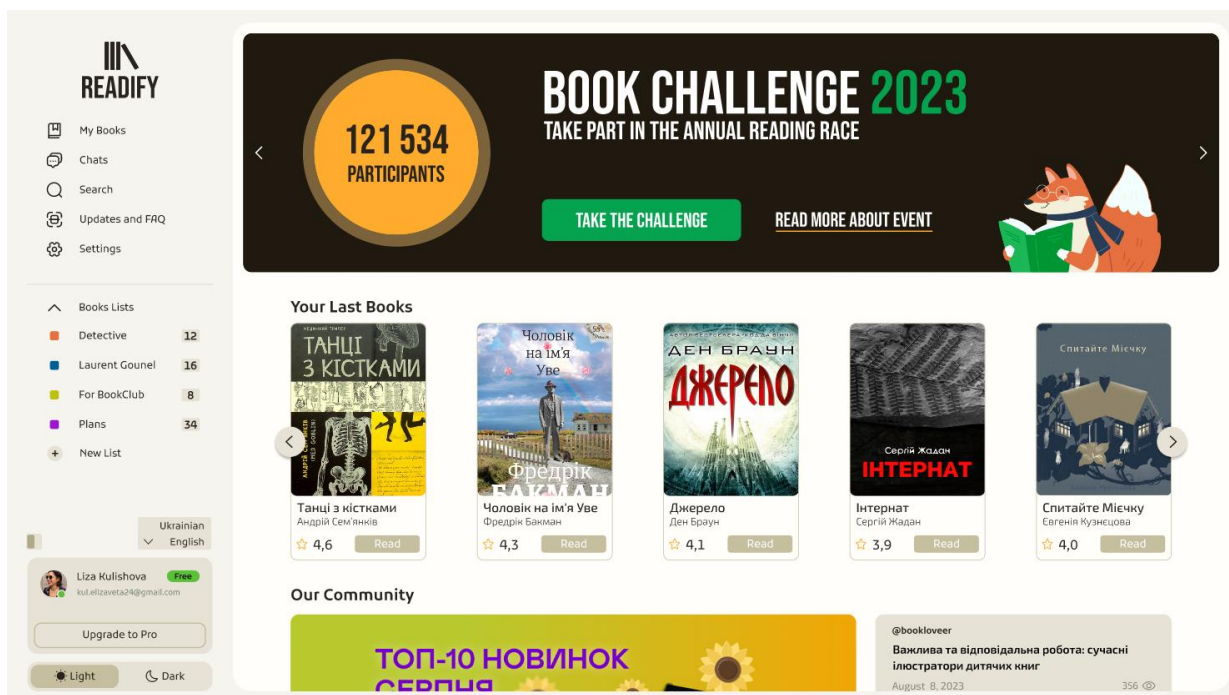


Рис. 1.3.14. – Відкриття панелі зміни мови веб-додатку при натисканні

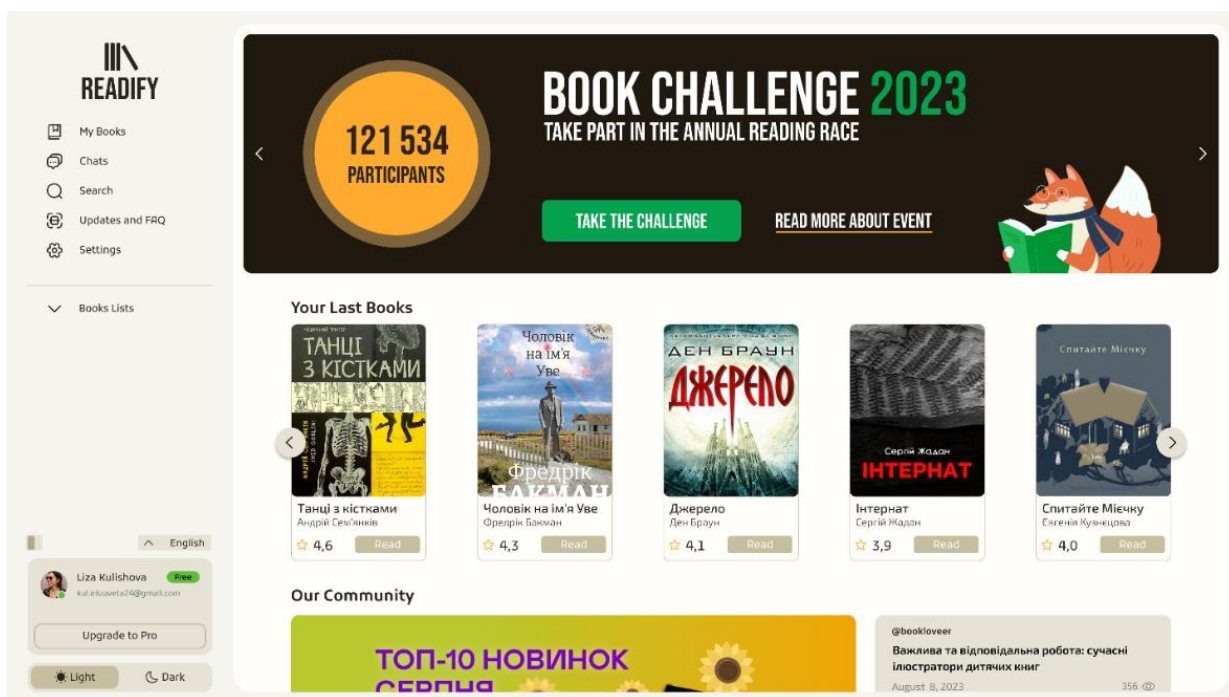


Рис. 1.3.15. – Закриття списків книг користувача

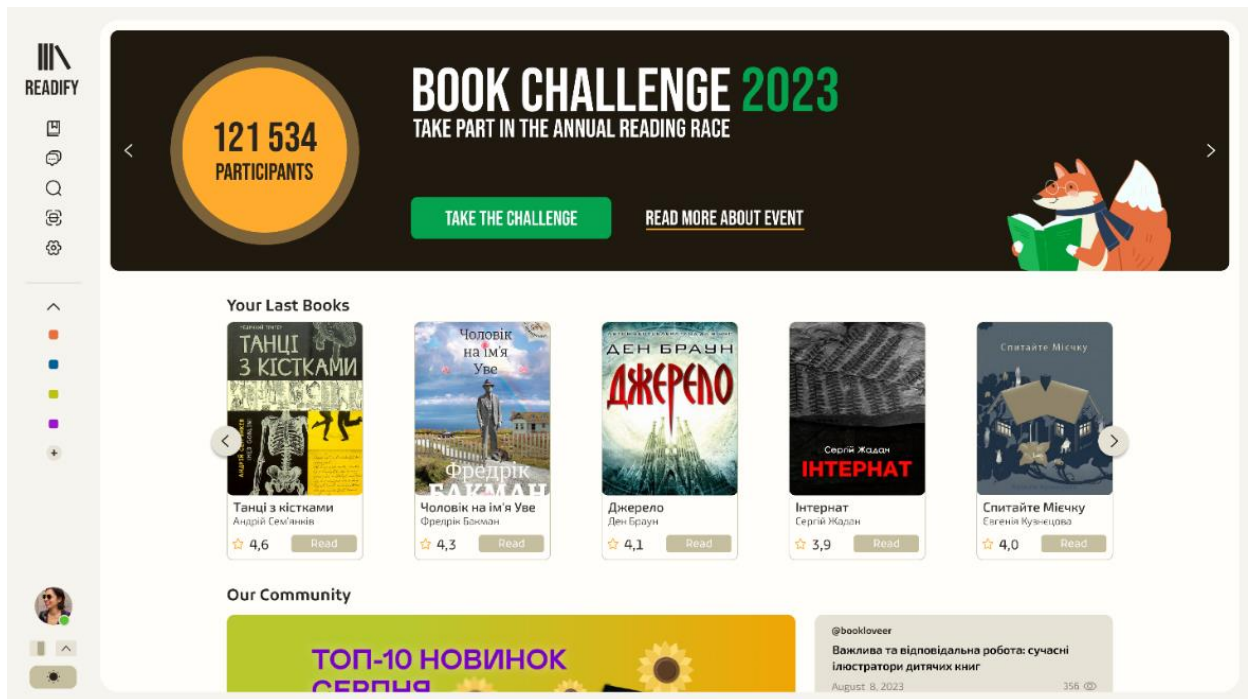


Рис. 1.3.16. – Закриття навігаційної панелі

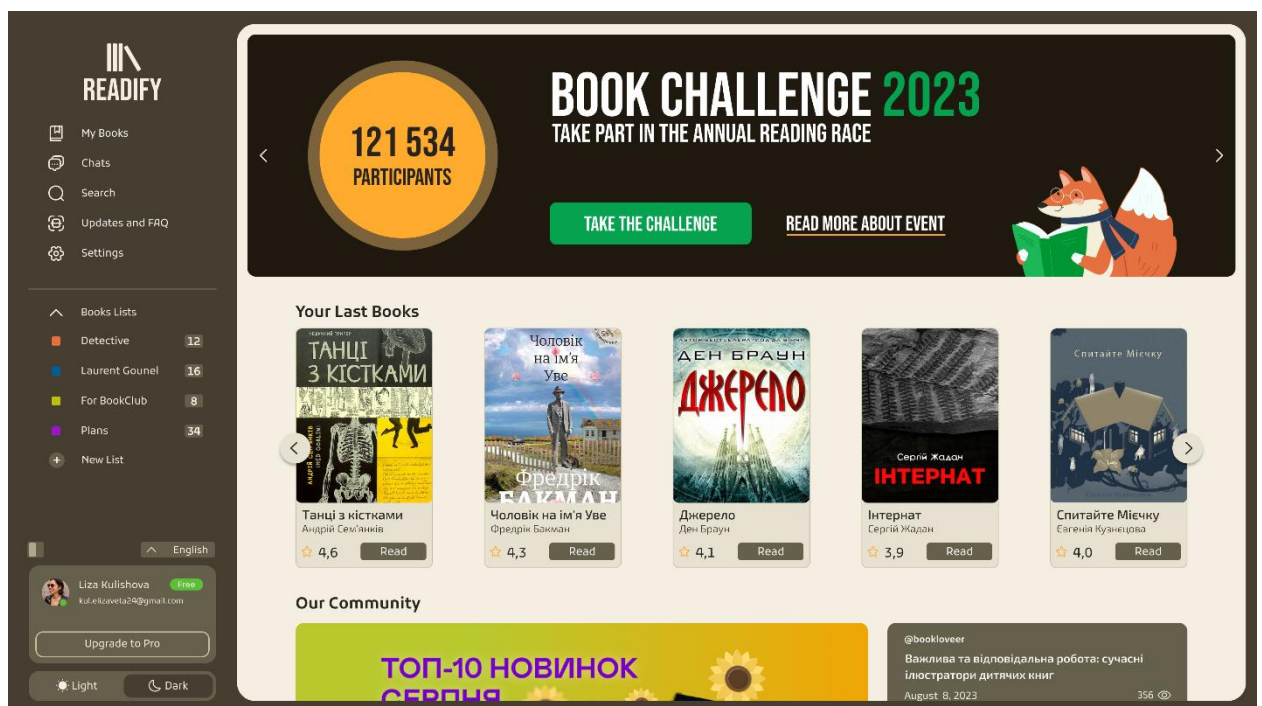


Рис. 1.3.17. – Зміна теми веб-додатку зі світлої на темну

- «My Books» («Мої книги») та його додаткові екрани

При переході у навігаційній панелі на сторінку «My Books» користувач може переглянути всі свої списки книг та їхню кількість в кожній (рис. 1.3.18.). Кожний список має свою кольорову гаму, яку можна задати при створенні (рис. 1.3.19.). Також на екрані розміщений пошук, за яким можна знайти конкретний список. При натисканні на будь-який список буде відкриватись

нова сторінка з розміщеними в ньому книжками (рис. 1.3.20.). Там також можна буде здійснити пошук. Звісно, при натисканні на будь яку книгу відкриється ще один новий екран з докладною інформацією про неї (рис. 1.3.21.). Книгу можна відкрити в аудіо форматі (рис. 1.3.22. – 1.3.23.), або в електронному варіанті. Також там розміщені рецензії, цитати інших користувачів та можна додати свої. Книгу можна додати до списку або видалити з нього, а також оцінити її.

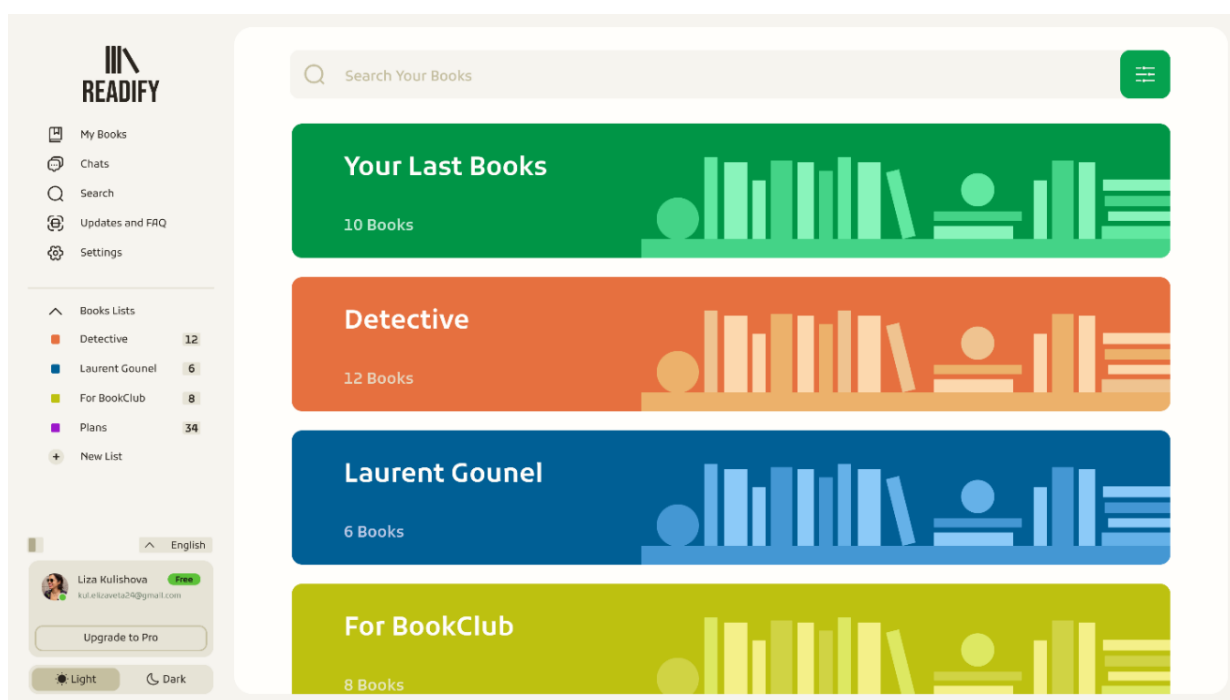


Рис. 1.3.18. – Сторінка «My Books»

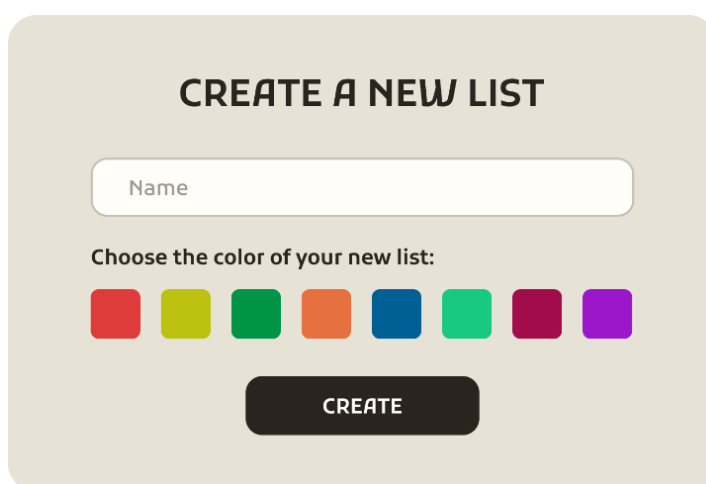


Рис. 1.3.19. – Панель створення нового списку

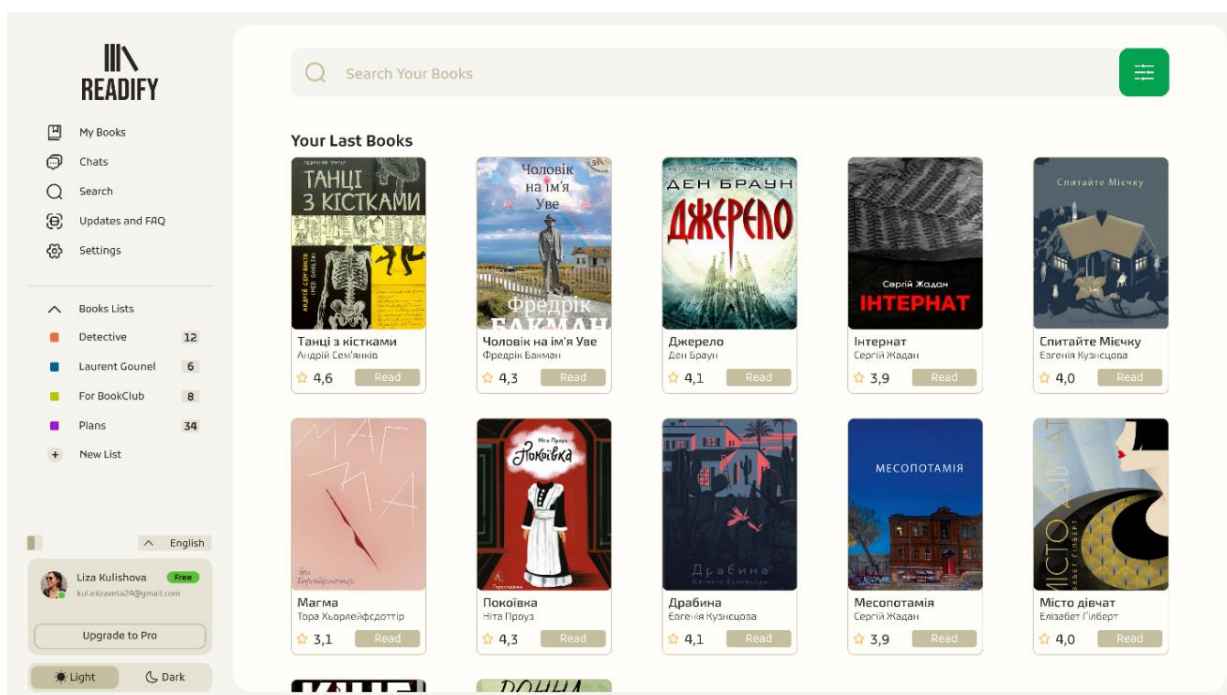


Рис. 1.3.20. – Відображення книг зі списку

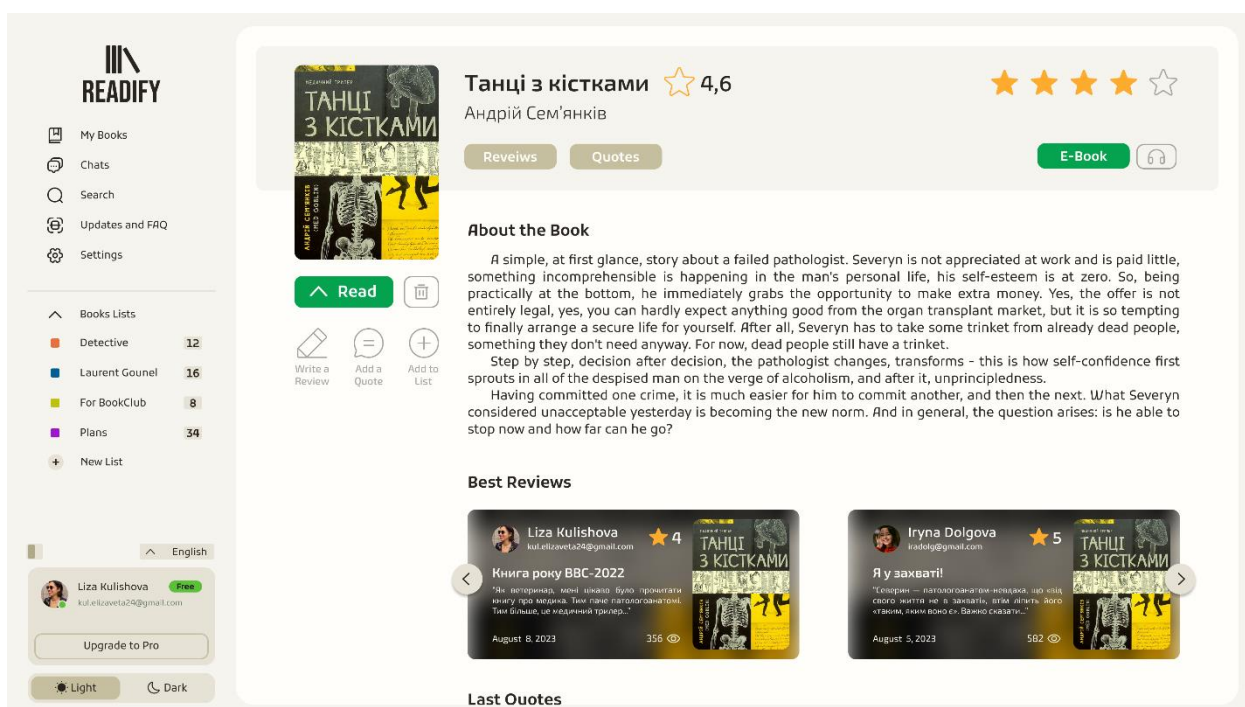


Рис. 1.3.21. – Відображення докладної інформації про книгу

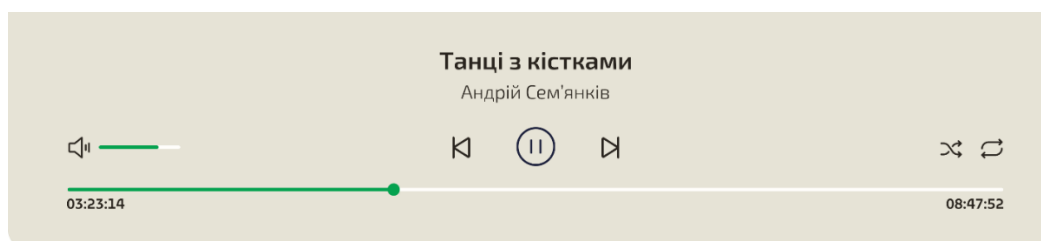


Рис. 1.3.22. – Розгорнутий плеєр для прослуховування аудіо формату книги
(запис йде)

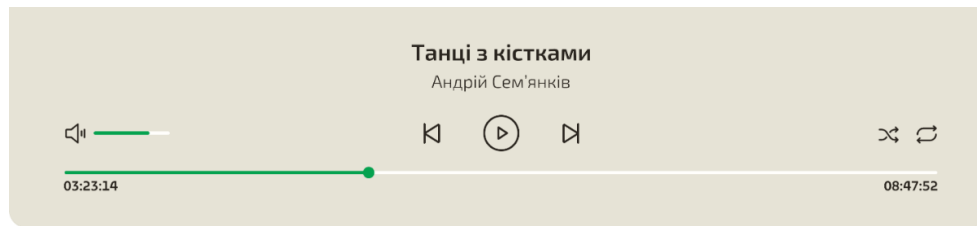


Рис. 1.3.23. – Розгорнутий плеєр для прослуховування аудіо формату книги (запис зупинено)

В електронному форматі книгу можна гортати натискаючи на ліву або праву частину робочої зони (рис. 1.3.24 – 1.3.25.).

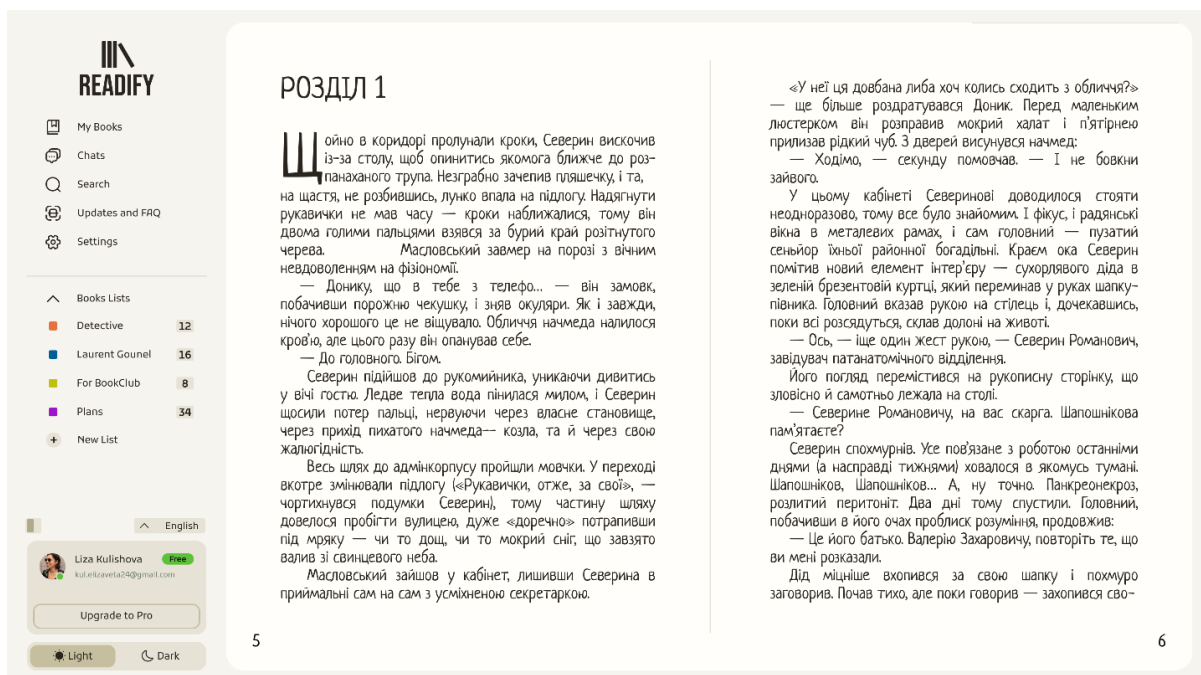


Рис. 1.3.24. – Відображення електронної книги (5-6 сторінки)

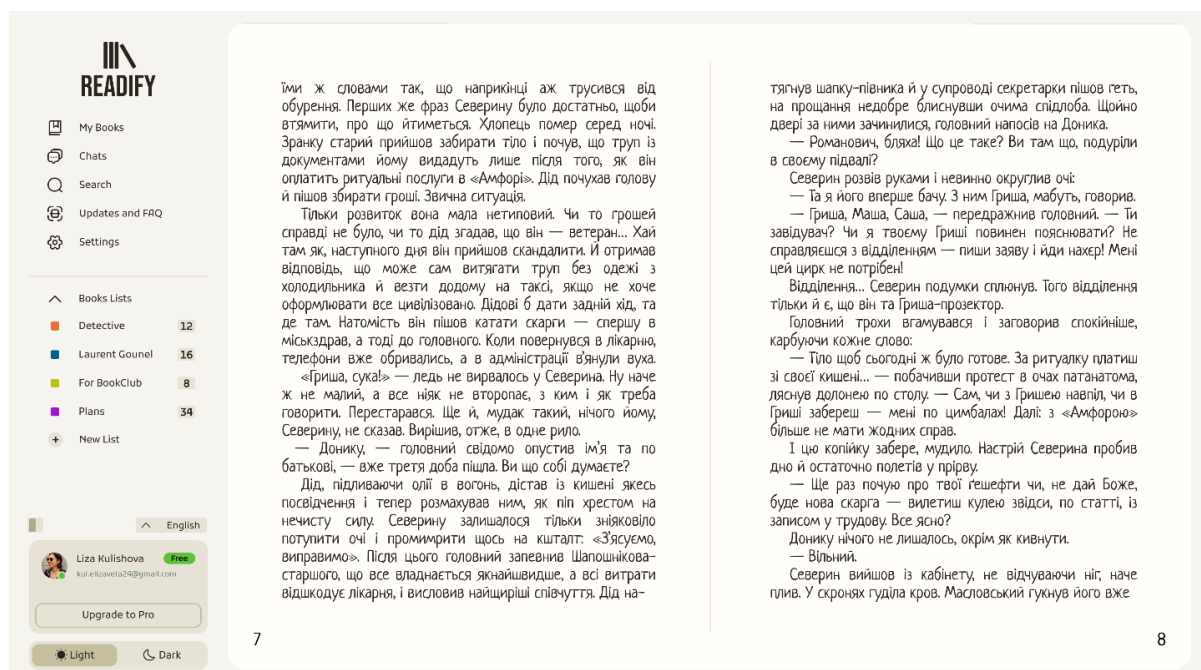


Рис. 1.3.25. – Відображення електронної книги (7-8 сторінки)

- «Chats» («Чати») та його додаткові екрани

При переході на цю сторінку відображаються всі приватні чати користувача (рис. 1.3.26.). Зверху розміщений пошук, для зручності. На кожному чаті видно статус онлайн/ офлайн користувача та статус останнього повідомлення та час його відправки. Також видно ім'я співрозмовника, фото його облікового запису та вміст останнього повідомлення. При виборі конкретного приватного чату – відкривається новий екран з цим чатом (рис. 1.3.27.). Зображення на фоні стоїть на замовчєнні. Внизу розміщений рядок для написання тексту. Також у чаті передбачено його налаштування (при натисканні на три крапки у правому верхньому куті) – зміна зображення на фоні, очистити чат та видалення чату (рис. 1.3.28.).

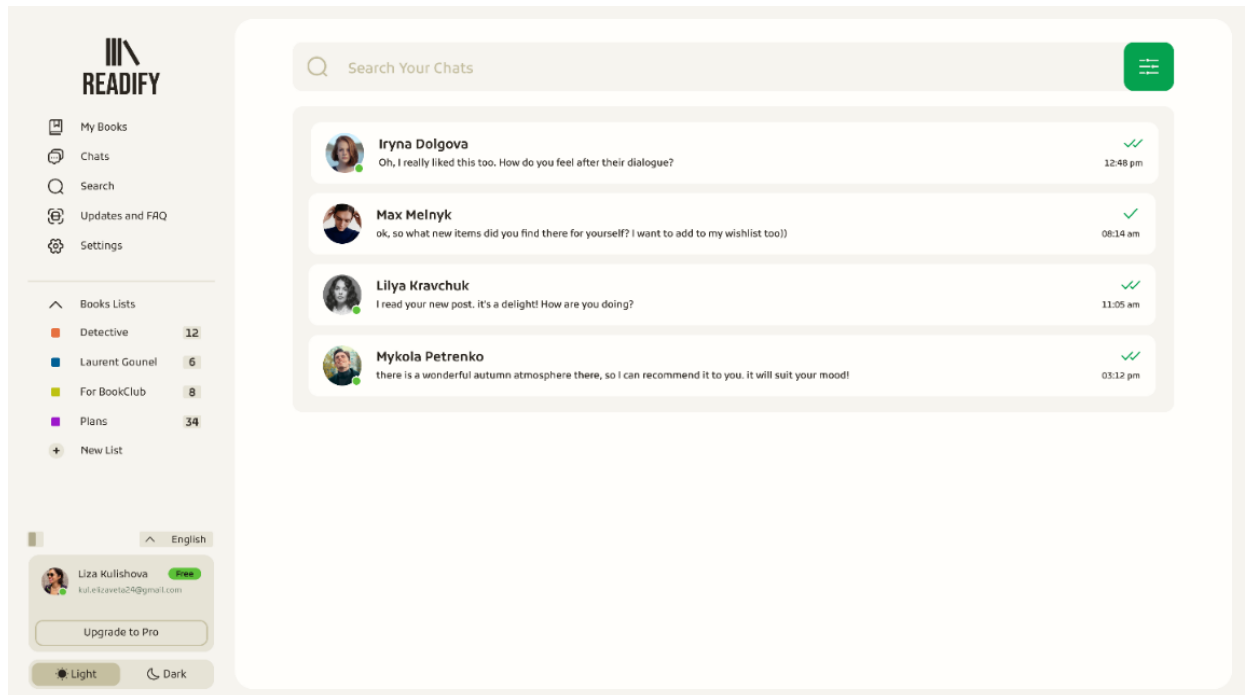


Рис. 1.3.26. – Сторінка переліку приватних чатів

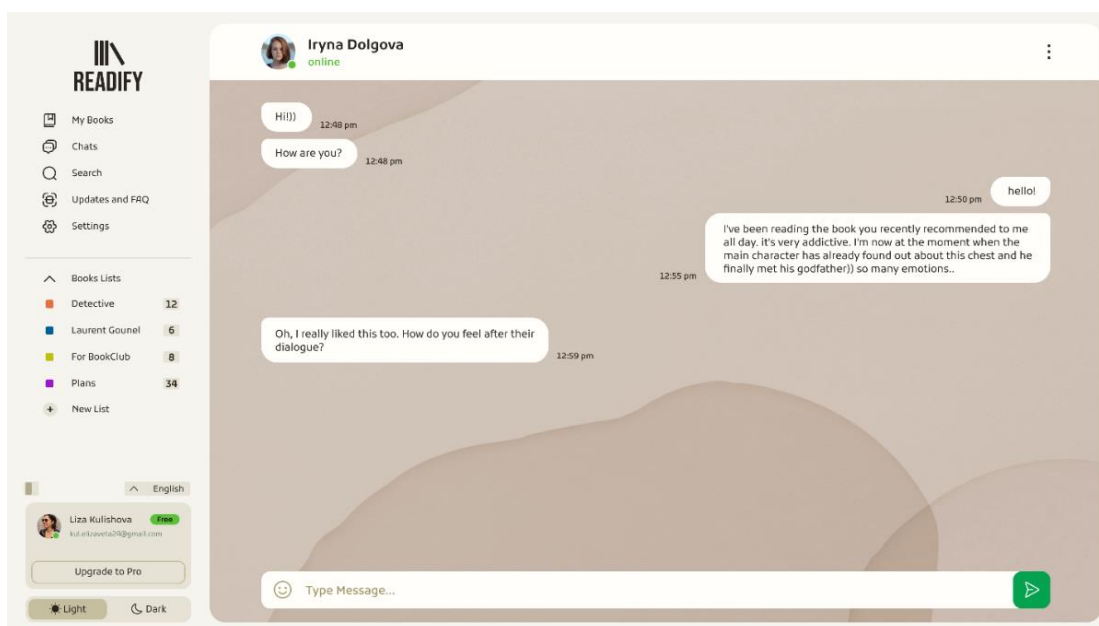


Рис. 1.3.27. – Відкритий приватний чат

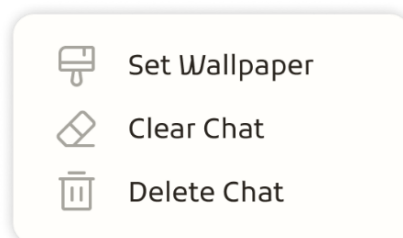


Рис. 1.3.28. – Панель налаштувань чату

- Сторінка «Search» («Пошук»)

На цій сторінці можна знайти книгу за назвою. Трохи нижче відображений рекомендований та швидкий пошук (те, що користувачі найчастіше шукають). Екран показаний на рис. 1.3.29.

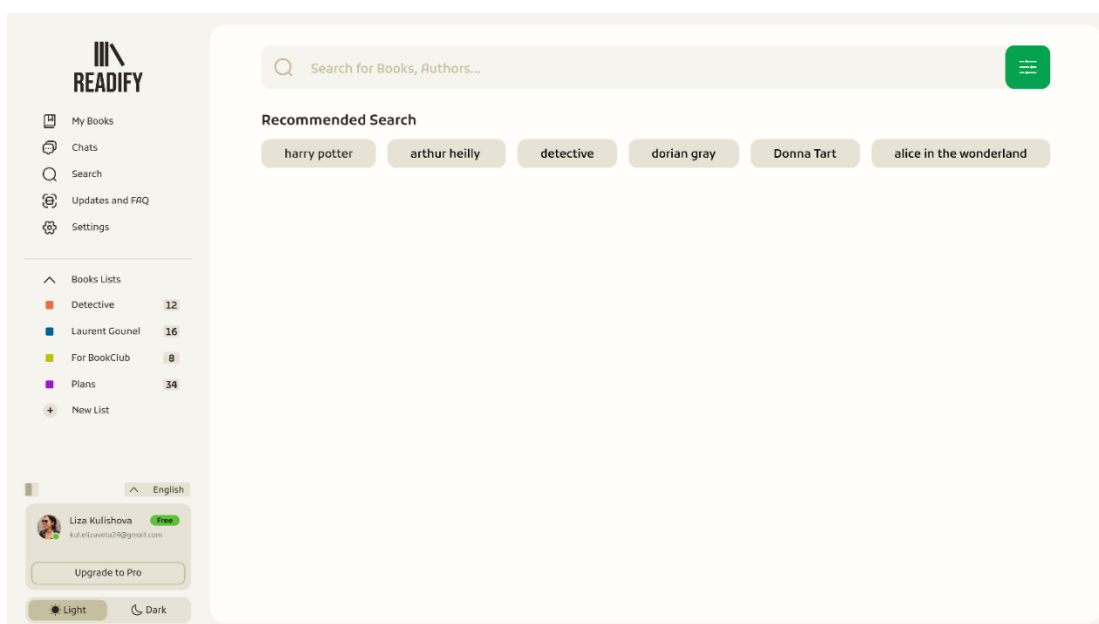


Рис. 1.3.29. – Сторінка пошуку книг

- Сторінка «Updates & FAQ» («Оновлення та відповіді на питання»)

Цей екран містить інформацію про останні та попередні оновлення веб-додатку та їх дати (рис. 1.3.30.). Також тут можна знайти відповіді на питання, які часто задавали інші користувачі. При натисканні на «+» розкривається палень, де написана відповідь на задане питання (рис. 1.3.31.).

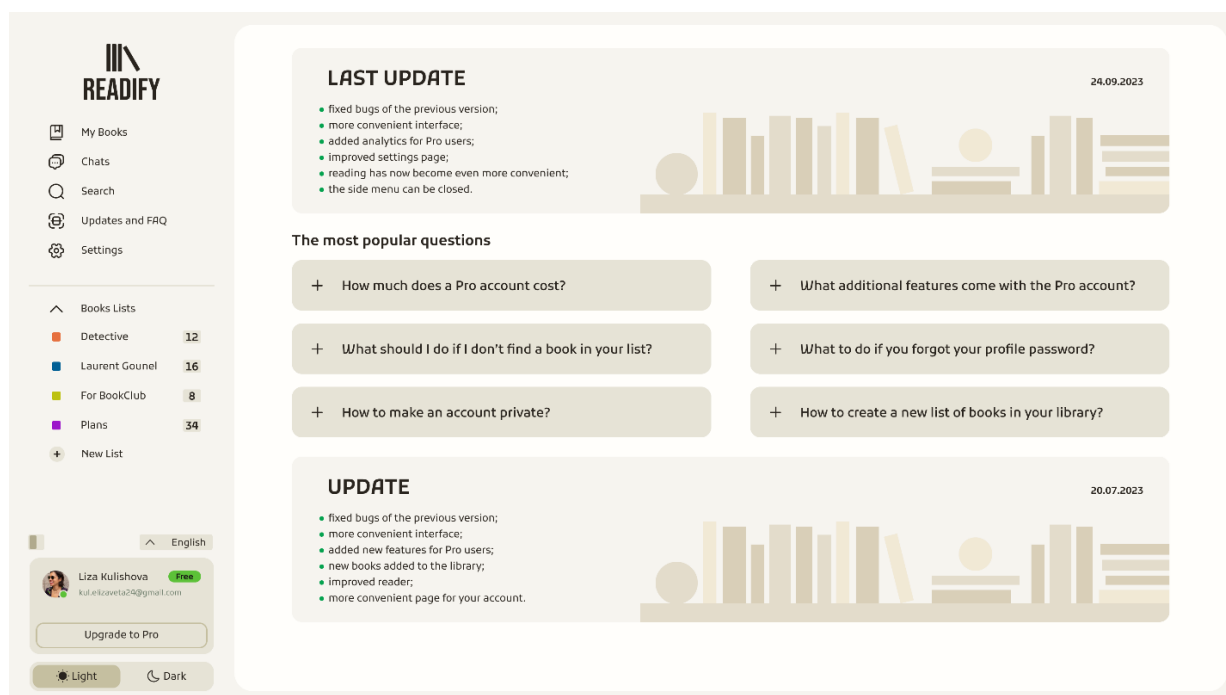


Рис. 1.3.30. – Сторінка оновлень та відповідей на питання

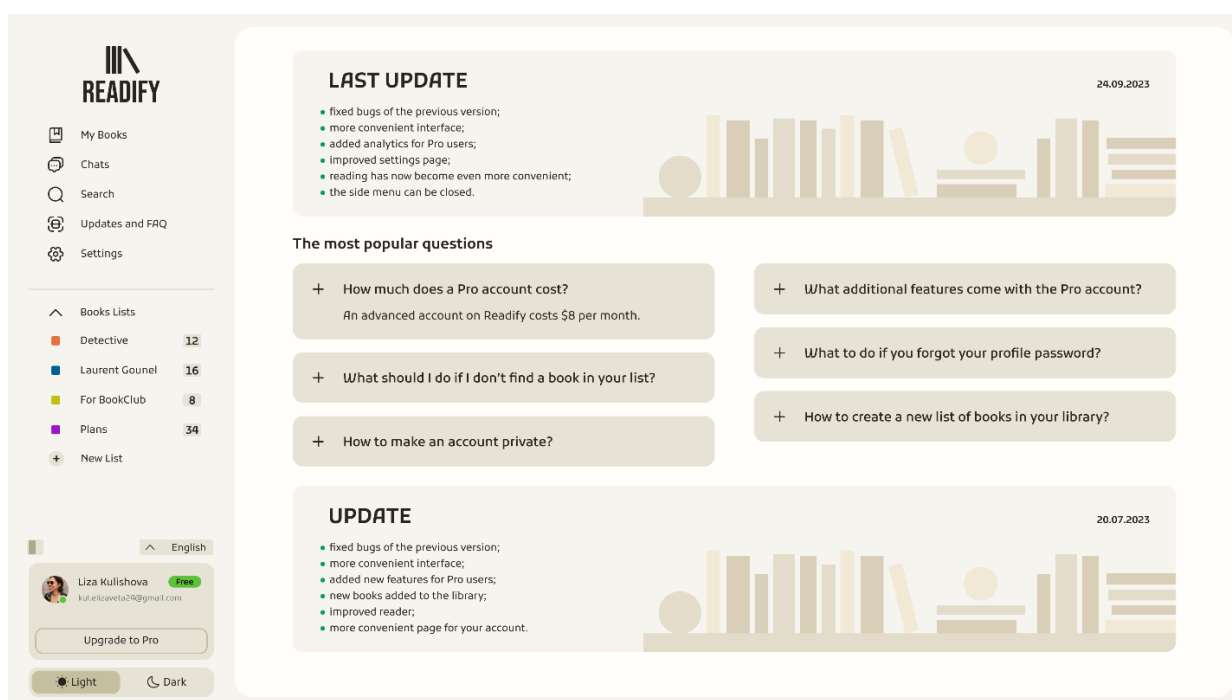


Рис. 1.3.31. – Сторінка оновлень та відповідей на питання (розкрита відповідь)

- «Settings» («Налаштування») та його додаткові екрани

Тут користувач може налаштувати свій профіль та його приватність (рис. 1.3.32.). Налаштування мають такі розділи: Мій обліковий запис, Повідомлення та звуки, Конфіденційність, Розширенні налаштування, Мова, Преміум веб-додаток, Оновлення та відповіді на питання та Написати питання. При переході на розділ з налаштуваннями облікового запису відображається інформація показана на рис. 1.3.33.

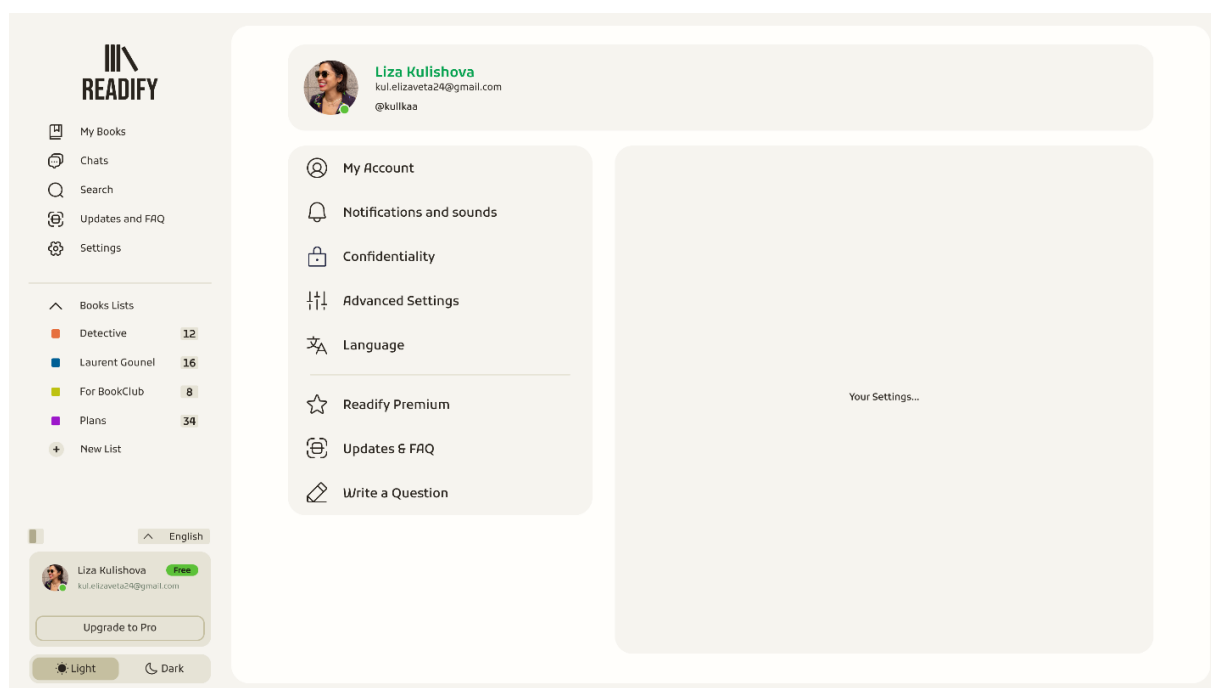


Рис. 1.3.32. – Сторінка налаштувань

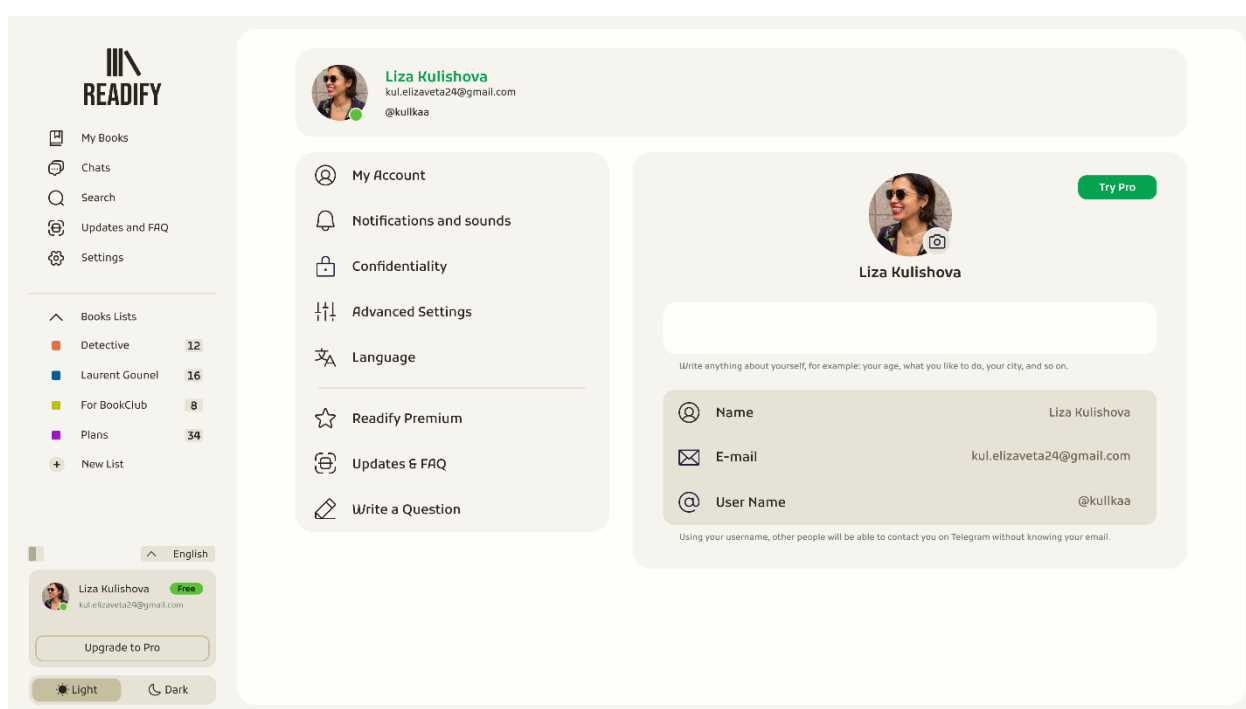


Рис. 1.3.33. – Відображення налаштувань профілю користувача

- Сторінка «My Profile» («Мій обліковий запис»)

На цій сторінці відображається обліковий запис користувача (рис. 1.3.34.). Тут можна побачити ім'я та фотографію читача, його фонове зображення та кількість його друзів. Праворуч розташовані дві кнопки: пропозиція перейти на платний обліковий запис та змінити профіль (рис. 1.3.35.). Внизу показана остання активність користувача (його дописи та рецензії).

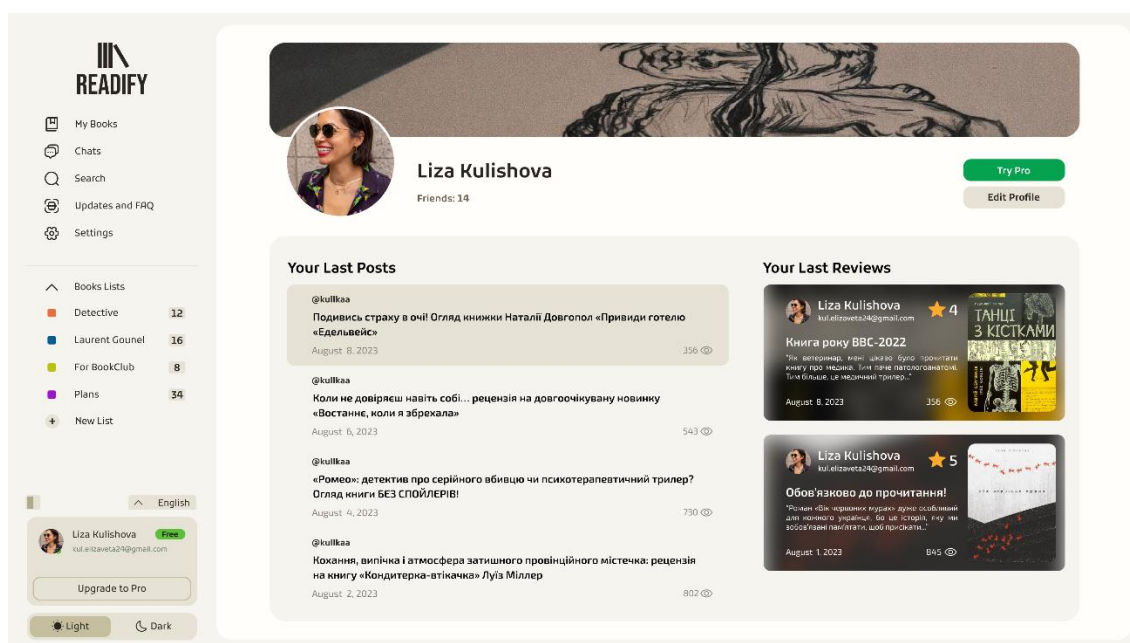


Рис. 1.3.34. – Сторінка облікового запису користувача

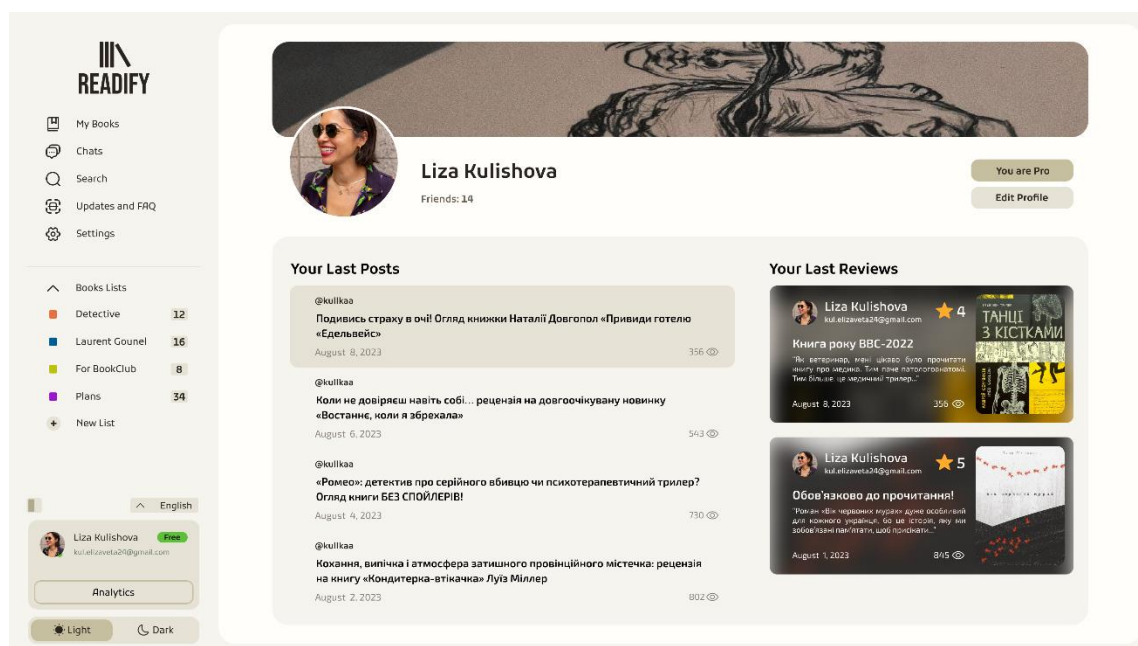


Рис. 1.3.35. – Відображення облікового запису користувача, при придбанні платного облікового запису

- Відображення дописів

Дописи – чудовий спосіб, щоб підкреслити для себе щось нове, або поділитись з кимось своїми думками. При відкритті дописів буде відображено назву поста та обкладинку. Ліворуч розташована кнопка, щоб повернутися на попередню сторінку. У дописі розміщена інформація, яка може містити текст, картинки, посилання та відео. Також міститься додаткова інформація у вигляді дати поширення допису. Зовнішній вигляд дописів можна побачити на рис. 1.3.36. – 1.3.38.

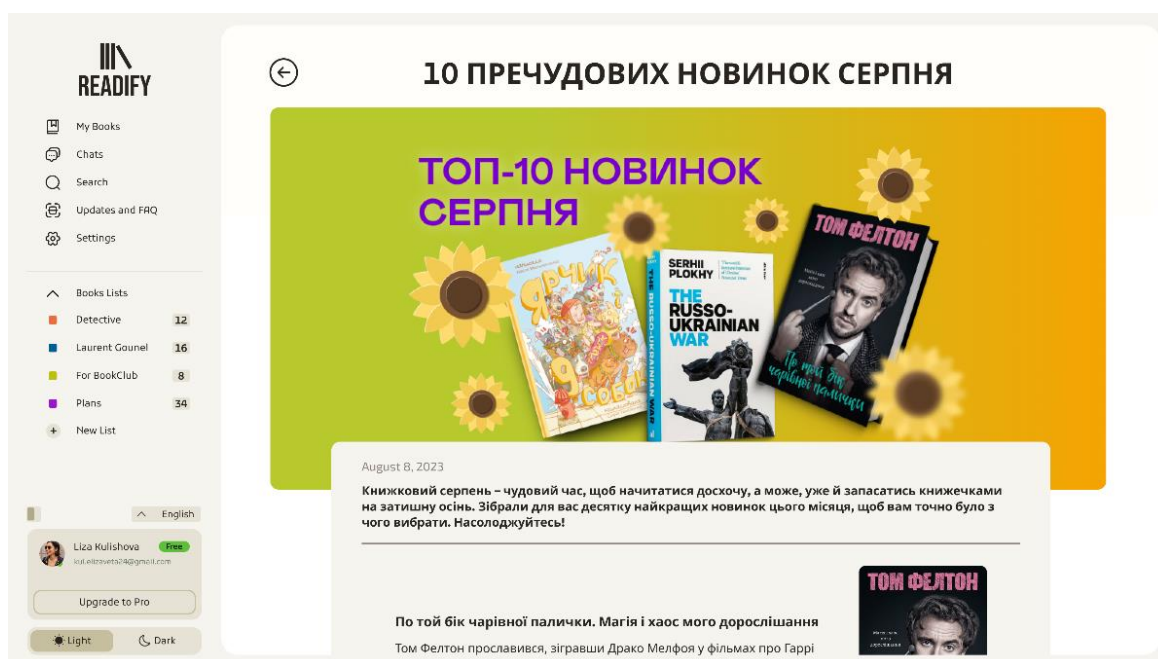


Рис. 1.3.36. – Сторінка відображення допису

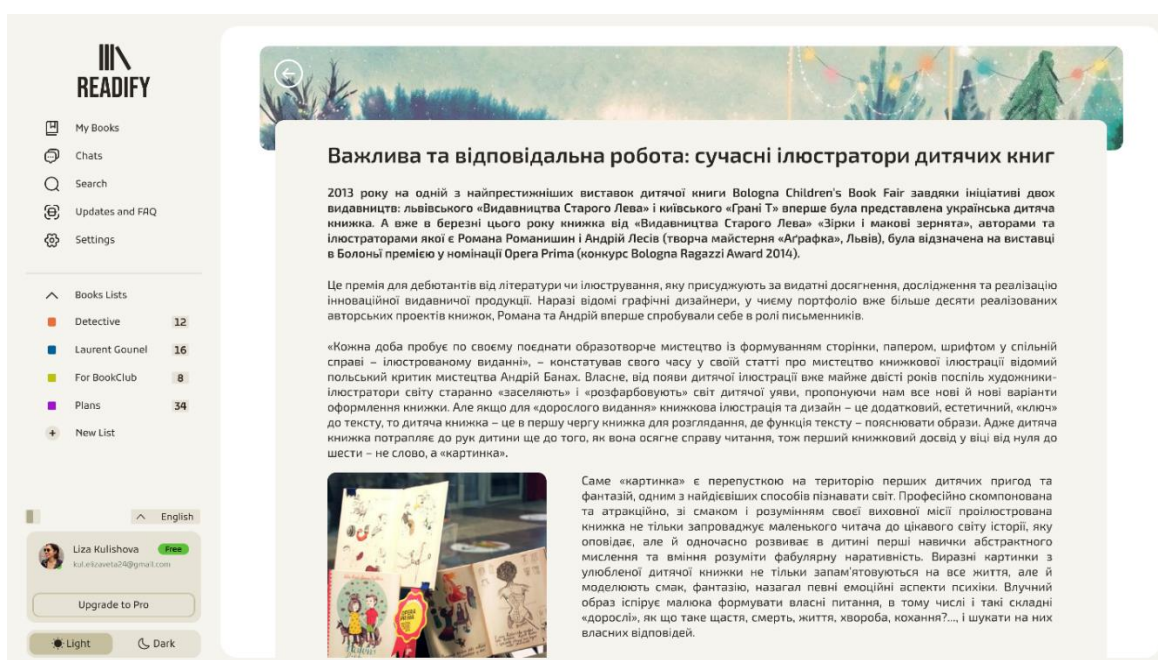


Рис. 1.3.37. – Приклад відображення допису (неповна версія)



CONNECT WITH US GET OUR LATEST STORIES IN THE FEED OF YOUR FAVORITE NETWORKS PHONE: +38 (050) 022 01 28	HAVE A TIP? WE WANT TO HEAR FROM YOU! SEND US A TIP USING OUR ANONYMOUS FORM. SEND US A TIP E-MAIL: readify@gmail.com	MY BOOKS CHATS SEARCH UPDATES AND FAQ SETTINGS
--	---	--

2023 READIFY. All rights reserved

ABOUT

Рис. 1.3.38. – Приклад відображення допису (повна версія)

- Відображення рецензій

Для поширення своїх думок стосовно прочитаних книг у веб-додатку є можливість написання рецензій. Окрім написання, також можна прочитати враження інших користувачів від книги. У рецензії може міститись текст. Також відображається дата поширення, кількість переглядів, автор рецензії, та оцінка книги. Приклад відображення рецензії продемонстровано на рис. 1.3.39.

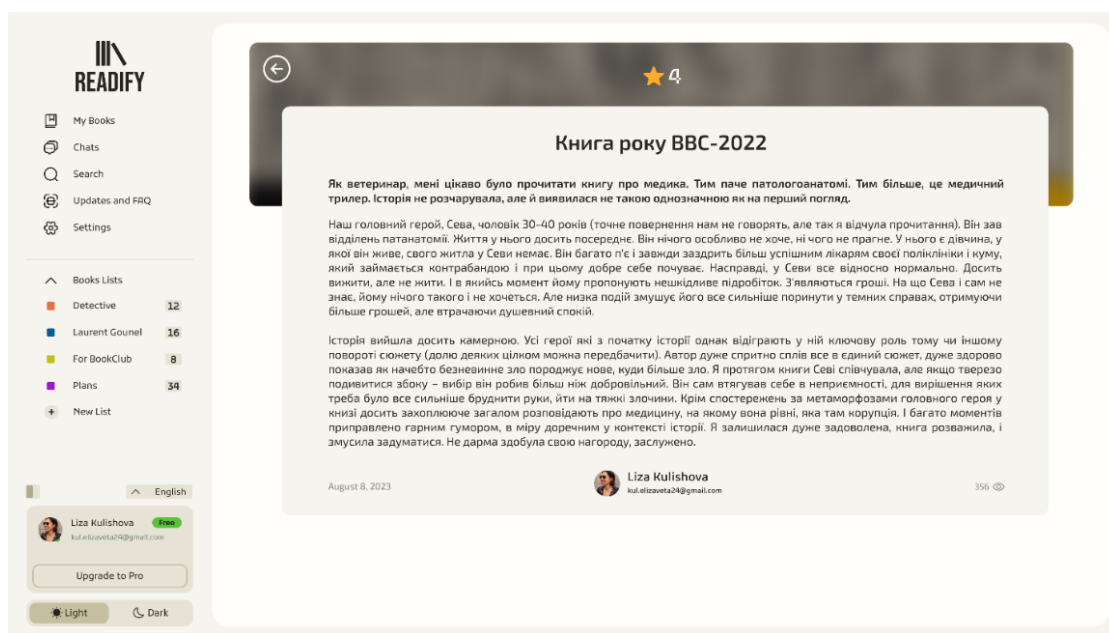


Рис. 1.3.39. – Приклад відображення рецензії

- Відображення цитат

При прочитанні користувач може зустріти фрагмент, який йому потрібно буде зберегти там, де він зможе його знайти у швидкому доступі. Цитати інших читачів також можна переглядати, щоб віднайти для себе щось пропущене або те, що не помітив. На сторінці цитат відображена обкладинка книги, її назва та автор, рейтинг та сама цитата (рис. 1.3.40.). Додатковою інформацією є ім'я користувача, який її додав, дату поширення та кількість переглядів.

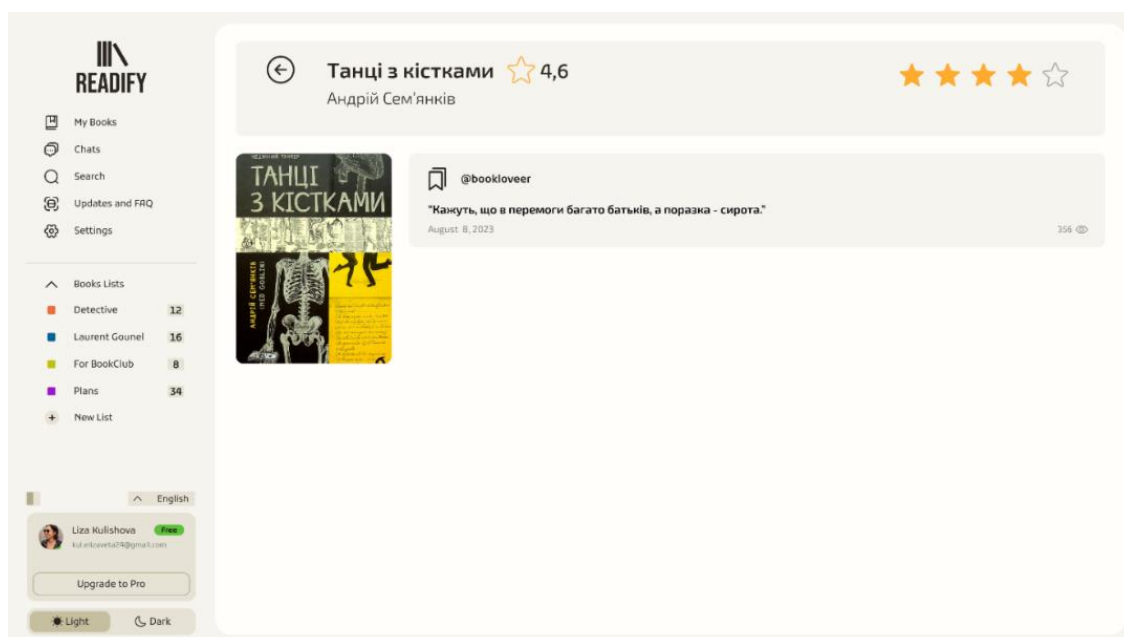


Рис. 1.3.40. – Приклад відображення рецензії

- Зовнішній вигляд прототипа у середовищі Figma

Для перегляду всього проекту у фігма достатньо змінити масштаб і бути на вкладинці Design (рис. 1.3.41.). Щоб побачити зв'язки між екранами треба перейти на вкладинку Prototype (1.3.42.).

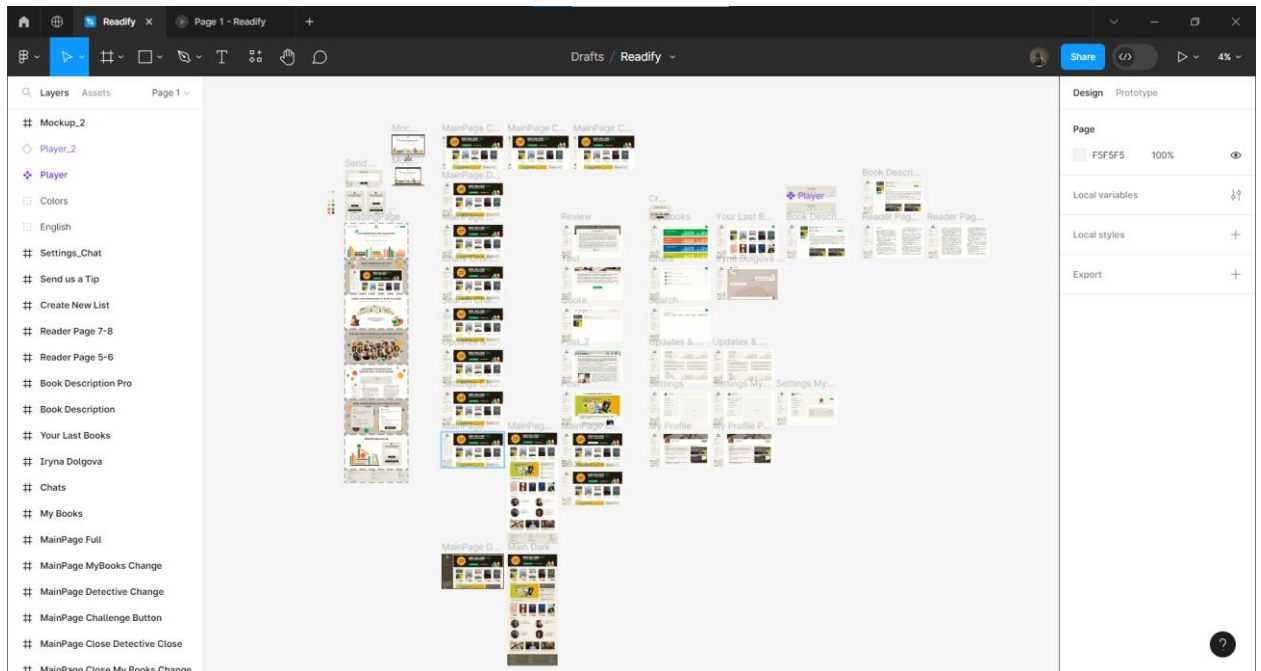


Рис. 1.3.41. – Зовнішній вигляд прототипа у режимі Design

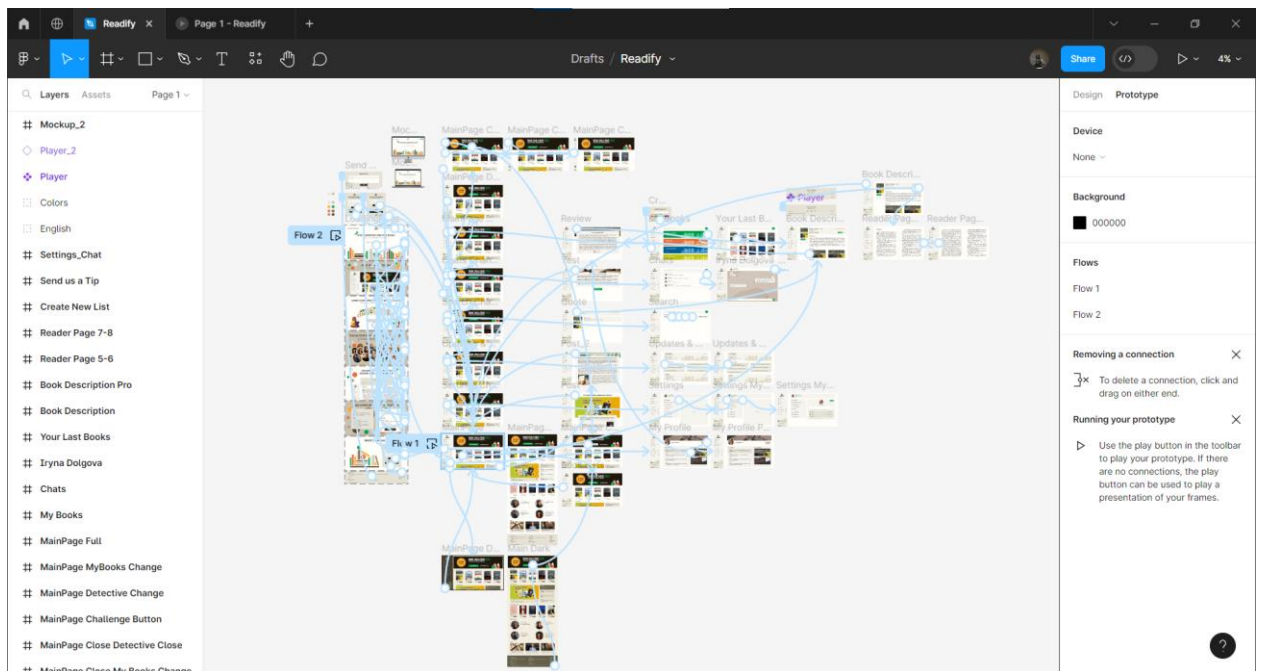


Рис. 1.3.42. – Зовнішній вигляд прототипа у режимі Prototype

- Допоміжні засоби у створенні прототипу

При створенні клікабельного прототипу було задіяно готові іконки (рис. 1.3.43.) та плагіни, які полегшують процес роботи (рис. 1.3.44.)

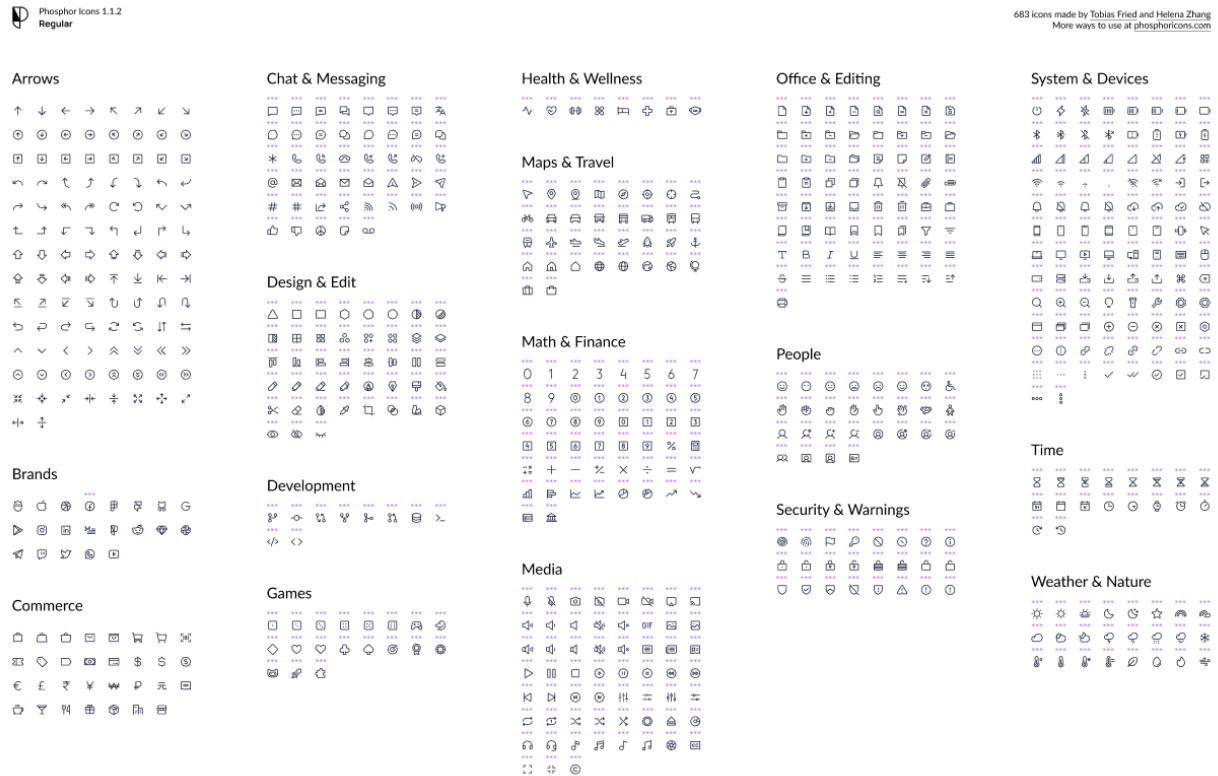


Рис. 1.3.43. – Використання готових іконок для проекту

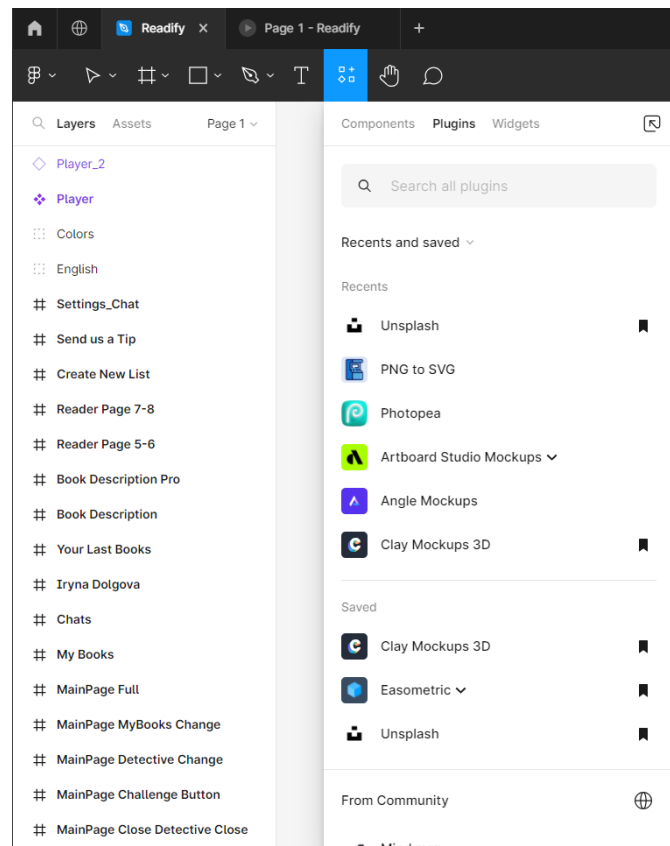


Рис. 1.3.44. – Використання плагінів для проекту

- Відображення веб-додатку на пристроях (москур)

Для демонстрації веб-застосунку на різних пристроях було використано техніку Москур. До прикладу зовнішній вигляд сервісу на ноутбучі (рис. 1.3.45.) та на екрані монітора (рис. 1.3.46.).

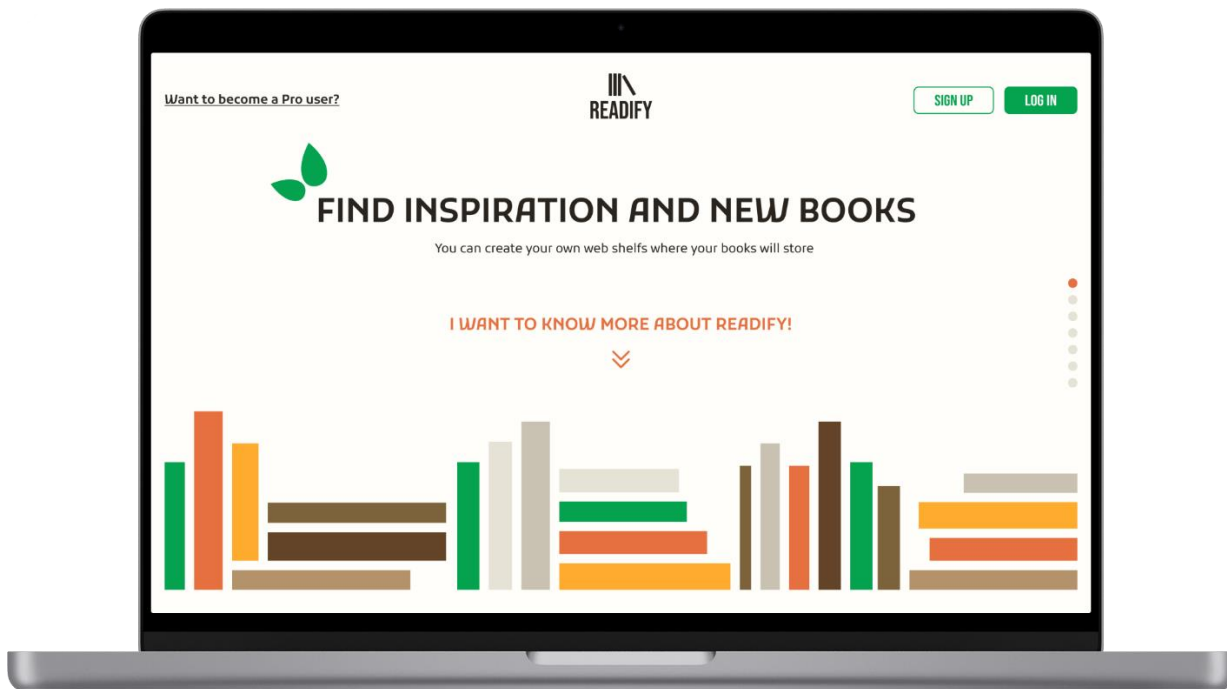


Рис. 1.3.45. – Використання техніки Москур (ноутбук)



Рис. 1.3.46. – Використання техніки Москур (монітор)

Для зберігання даних використовується система, яка надає інтерфейс для адміністрування СКБД (системи керування базами даних) MySQL. Було обрано веб-застосунок з відкритим кодом *phpMyAdmin* у якості надійної та ефективної реляційної бази даних. Він забезпечує структуроване та швидке зберігання інформації про користувачів, книги, списки читання, форму зворотного зв'язку та інші необхідні дані.

У роботі над проектом використані наступні API:

1. **Google API:** Це API дозволяє користувачам реєструватися та автентифікуватися за допомогою облікових записів Google, забезпечуючи простий та безпечний доступ.

2. **Платіжна API:** Інтеграція цього API дозволяє здійснювати оплату та взаємодіяти з платіжною системою, забезпечуючи надійний контроль та облік транзакцій.

Ці API впроваджені для оптимізації роботи з проектом та спрощення взаємодії з ключовими функціональними елементами зовнішніх систем.

Обраний підхід та методології сприяють створенню високоякісної, функціональної та ефективної інформаційної системи, яка відповідає сучасним стандартам розробки. Було обрано об'єктний підхід до проектування інформаційної системи з метою забезпечення комплексного та систематичного підходу до розробки.

Вибір об'єктно-орієнтованого підходу до проектування ІС обґрунтовується кількома ключовими перевагами. Діаграми прецедентів та послідовності дозволяють зосередитися на функціональності системи з точки зору користувача, полегшуючи розуміння для всіх акторів. Для об'єктного моделювання було обрано програмне забезпечення Enterprise Architect.

Фреймворк Scrum було обрано для впровадження ітераційного та інкрементального підходу до розробки. Scrum дозволяє ефективно керувати змінами в процесі розробки, сприяє взаємодії замовника та розробників, а також швидкому впровадженню нових функцій та змін у виробництво.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ

2.1. Формування вимог

2.1.1. Бізнес-вимоги до системи

Інформаційна система веб-додатку для читацької спільноти може значно полегшити доступ до літературних ресурсів. Однак, для успішного розвитку та впровадження такого сервісу необхідно чітко визначити бізнес-вимоги, які він має вирішувати. Створення діаграми бізнес-вимог є обов'язковим етапом в розробці інформаційної системи, який дозволить визначити вимоги до системи, забезпечити її відповідність потребам користувачів та бізнес-потребам, а також створити зручний та ефективний інтерфейс для користувачів.

Було визначено такі **бізнес-вимоги**:

1. Стимулювання активності користувачів та збільшення кількості платників сервісу – система має спонукати користувачів до активного використання, а також збільшувати кількість користувачів, які готові платити за сервіс або продукт.
2. Розробка нового та вдосконалення існуючого функціоналу – розширення можливостей веб-додатку шляхом впровадження нового функціоналу та вдосконалення існуючого. Цим може бути додавання нових функцій, виправлення помилок або оптимізацію інтерфейсу користувача.
3. Покращення швидкості завантаження веб-застосунку та його оптимальної продуктивності, з урахуванням підтримки різних пристроїв та браузерів – завдання зробити веб-додаток швидшим та ефективнішим у використанні. Йдеться про оптимізацію завантаження сторінок, підтримку різних пристроїв і браузерів, а також забезпечення оптимальної продуктивності навіть при великому навантаженні.
4. Розробка та впровадження стратегій цифрового маркетингу для збільшення обізнаності про застосунок серед цільової аудиторії читацької

спільноти – розробка та виконання маркетингових стратегій для просування веб-додатку серед цільової аудиторії. У цю бізнес-вимогу можуть входити рекламні кампанії, використання соціальних мереж, контент-маркетинг, пошукова оптимізація та інші інструменти маркетингу.

5. Забезпечення швидкого та ефективного вирішення запитань і проблем користувачів, використовуючи інструменти для моніторингу та взаємодії з клієнтами – система має надати користувачам можливість швидко отримувати відповіді на свої запитання та вирішувати проблеми.

Визначені бізнес-вимоги було створено в середовищі Enterprise Architect і показано на рис. 2.1.1.1.

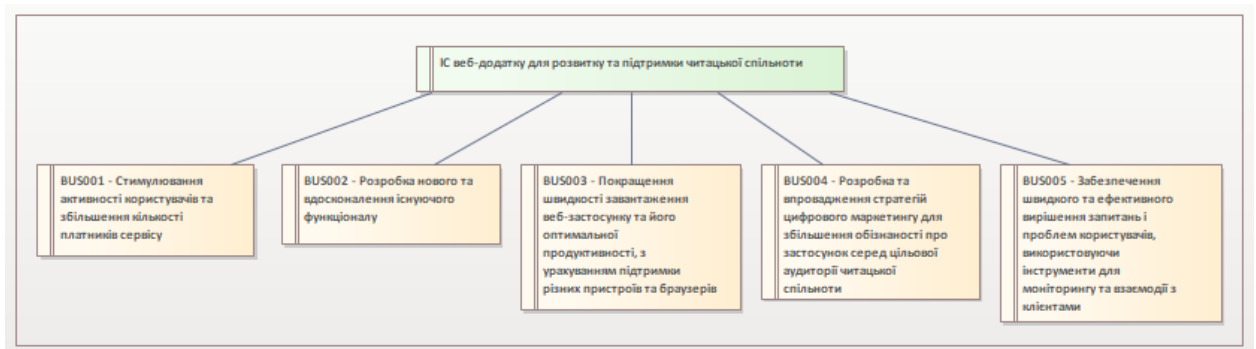


Рис. 2.1.1.1. – Діаграма бізнес-вимог для ІС веб-додатку для розвитку та підтримки читачької спільноти

2.1.2. Функціональні вимоги до системи

Створення функціональних вимог для інформаційної системи також є важливим етапом в процесі розробки. Такі вимоги описують функції, які система повинна виконувати, або операції, які користувачі можуть виконувати у системі. Вони визначають, як система повинна вести себе у відповідь на конкретні вхідні дані або події, і які результати мають бути згенеровані після виконання цих дій.

Потреба у створенні функціональних вимог виникає через необхідність збалансувати очікування користувачів та розробників щодо функціональності системи. Вони визначають обсяг та властивості програмного забезпечення, а також формують основи для подальшого проектування та розробки системи.

Функціональні вимоги є основою для тестування системи та оцінки її відповідності вимогам користувачів.

На рис. 2.1.2.1 подано модулі функціональних вимог до ІС веб-додатку для підтримки та розвитку читацької спільноти.



Рис. 2.1.2.1. – Модулі функціональних вимог для ІС веб-додатку для розвитку та підтримки читацької спільноти

Для демонстрації ієрархії та всіх зв'язків було побудовано діаграму, де розміщено всі бізнес-вимоги, функціональні та нефункціональні вимоги, а також логічні зв'язки між ними (рис. 2.1.2.2.).

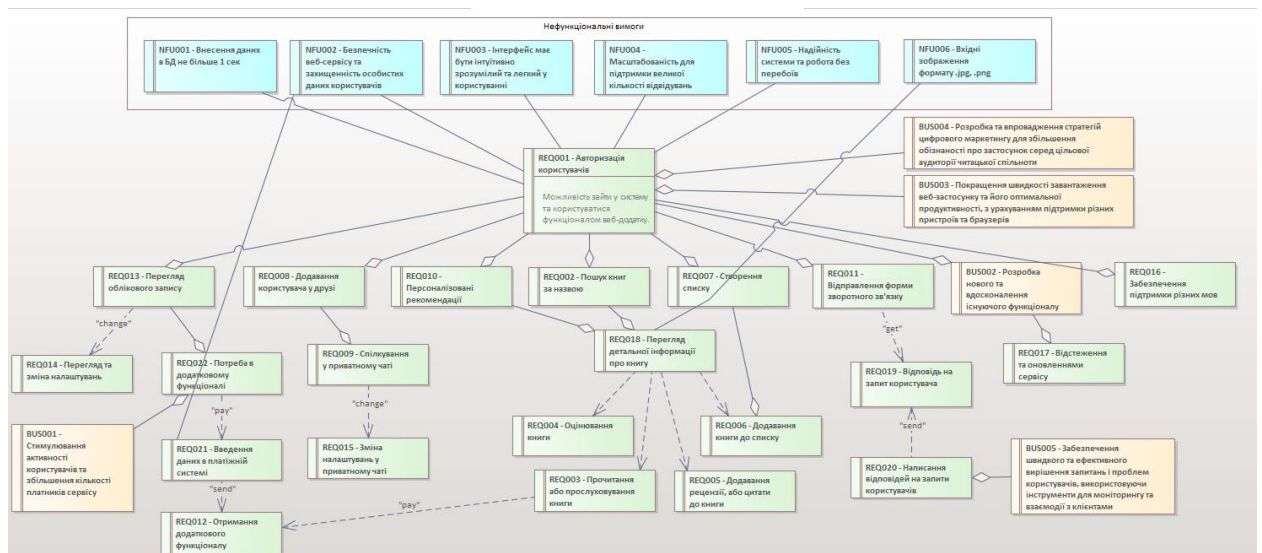


Рис. 2.1.2.2. - Діаграма вимог з поданням ієрархії вимог

Також було створено специфікацію вимог, підготовлену за допомогою Specification Manager (менеджера специфікацій). Відповідна таблиця містить такі відомості: ідентифікатор вимоги, стисла назва вимоги, повна назва вимоги, тип, статус, складність, пріоритетність, ризикованість та примітки (рис. 2.1.2.3.).

Item	Alias	All Indicators	Type	Status	Difficulty	Priority	R
☒ REQ001 - Авторизація користувачів Можливість зайти у систему та користуватися функціоналом веб-додатку.	REQ001	→	Requirement	Proposed	Medium	Medium	—
☒ REQ002 - Пошук книг за назвою	REQ002	→	Requirement	Proposed	Medium	Medium	
☒ REQ003 - Прочитання або прослуховування книги	REQ003	→	Requirement	Proposed	Medium	Medium	
☒ REQ004 - Оцінювання книги	REQ004	→	Requirement	Proposed	Medium	Medium	
☒ REQ005 - Додавання рецензії, або цитати до книги	REQ005	→	Requirement	Proposed	Medium	Medium	
☒ REQ006 - Додавання книги до списку	REQ006	→	Requirement	Proposed	Medium	Medium	
☒ REQ007 - Створення списку	REQ007	→	Requirement	Proposed	Medium	Medium	
☒ REQ008 - Додавання користувача у друзі	REQ008	→	Requirement	Proposed	Medium	Medium	

Рис. 2.1.2.3. – Специфікація вимог у Specification Manager

2.1.3. Нефункціональні вимоги до системи

Нефункціональні вимоги визначають якісні характеристики системи, такі як продуктивність, надійність, безпека, ефективність використання та інші аспекти, які не відображають конкретну функціональність системи, але впливають на її поведінку та сприйняття користувачами.

Потреба у створенні нефункціональних вимог виникає через необхідність забезпечення якості програмного забезпечення та задоволення потреб користувачів щодо його використання. Вимоги до продуктивності можуть включати, наприклад, кількість потрібного часу для відповіді системи на запити користувачів, або обсяг даних, який вона може обробляти за певний період часу.

На рис. 2.1.3.1. наведено перелік нефункціональних вимог ІС веб-додатку для підтримки та розвитку читацької спільноти.

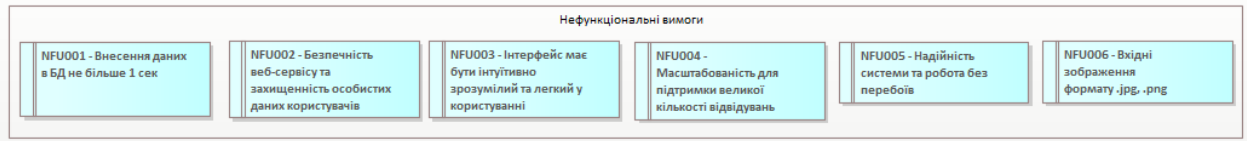


Рис. 2.1.3.1. – Діаграма нефункціональних вимог до ІС веб-додатку для підтримки та розвитку читацької спільноти

Джерелами вимог у ІС веб-додатку для розвитку та підтримки читацької спільноти є різні сторони та учасники. Користувачі, фахівці з маркетингу, розробники, менеджери зі зворотного зв'язку виступають як джерела вимог. Користувачі виражають свої потреби та очікування від системи, менеджери зі зворотного зв'язку допомагають у вирішенні проблем користувачів. Фахівці з маркетингу розробляють стратегії просування платформи серед аудиторії. Розробники створюють та підтримують функціональність веб-сервісу.

Деталізація опису пакетів специфікацій вимог до ІС веб-додатку для розвитку та підтримки читацької спільноти у Додатку А.

Нижче на рис. 2.1.3.2. подано діаграму підсумкової кількості вимог до ІС, що виконана засобами *Dashboard* в середовищі *Enterprise Architect*. Всього система має 33 вимог, з них 5 – бізнес-вимоги, 22 – функціональних і 6 – нефункціональних вимог.

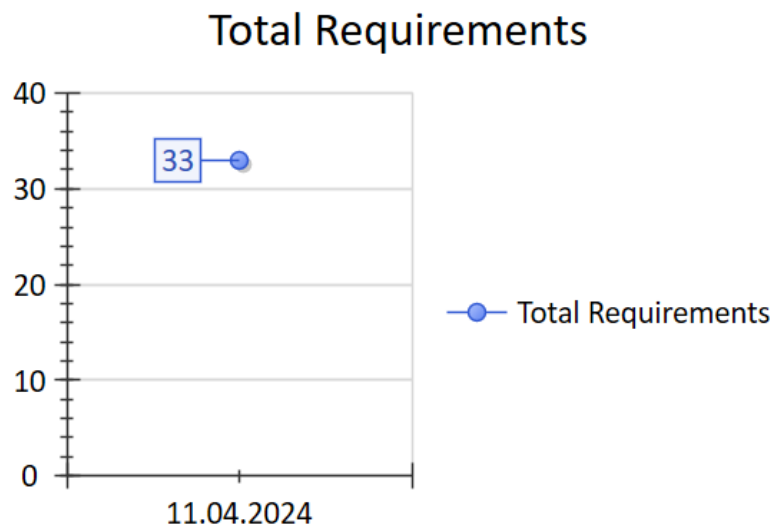


Рис. 2.1.3.2. – Діаграма підсумкової кількості вимог

Було створено вертикальну гістограму, яка показує статус реалізації вимог до інформаційної системи (рис. 2.1.3.3.). За статусом вимога може бути: approved – затверджена, proposed – запропонована, validated – підтверджена.

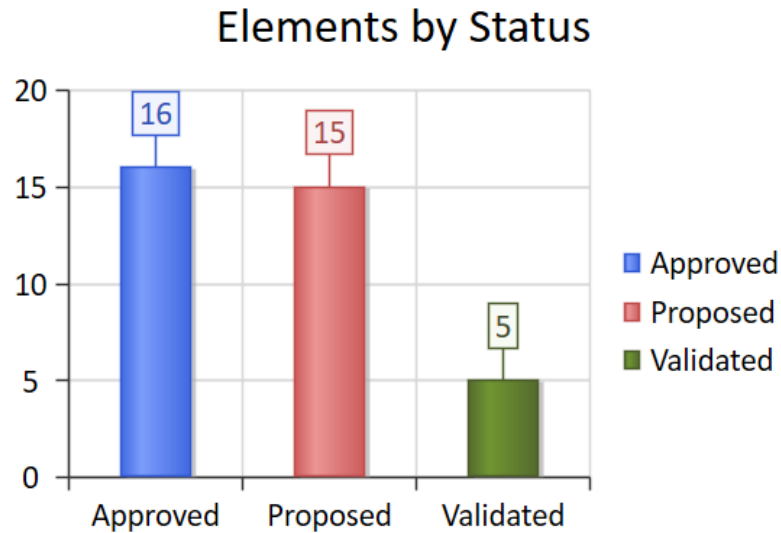


Рис. 2.1.3.3. – Діаграма розподілу всіх вимог за статусом

Також було представлено діаграму розподілу вимог за пріоритетом (рис. 2.1.3.4.). Пріоритет може бути High - високим, Medium - середнім, Low - низьким. Частка «No Value» є частиною самої діаграми. Система буде нарощувати свій функціонал починаючи з вимог з високим пріоритетом, потім середнім.

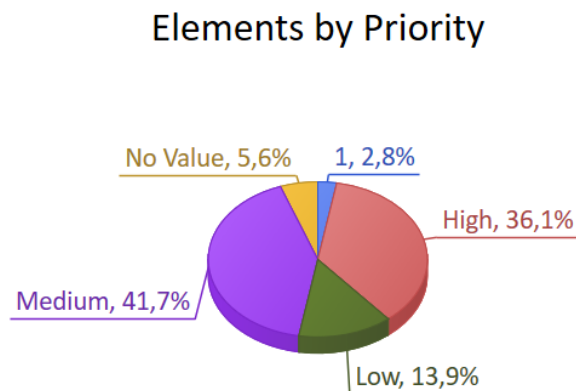


Рис. 2.1.3.4. – Діаграма розподілу всіх вимог за пріоритетом

2.2. Постановка та алгоритм розв’язання задачі

2.2.1. Характеристика задачі

Розглядаючи економічний зміст задач оброблення інформації, першочерговим завданням є вивчення потреб цільової аудиторії. Даний веб-застосунок спрямований на людей, які читають книжки та бажають знайти спільноту однодумців. Аналіз їхніх потреб дозволяє ефективно визначити функціонал та ціну за нього. Монетизація сервісу буде здійснюватися за допомогою Професійного-акаунта, який буде містити у собі необхідні для користувачів функції. Враховуючи зібрані дані, можна встановити доступну ціну на платний обліковий запис у розмірі 5 доларів на місяць, що є середньою на ринку серед конкурентів. Аналіз економічного потенціалу наших користувачів допоможе оптимізувати стратегію ціноутворення та максимізувати прибутковість.

В інформаційній моделі задачі на рис. 2.2.1.1. відображені вихідні та вхідні повідомлення, а також довідники та масиви системи.

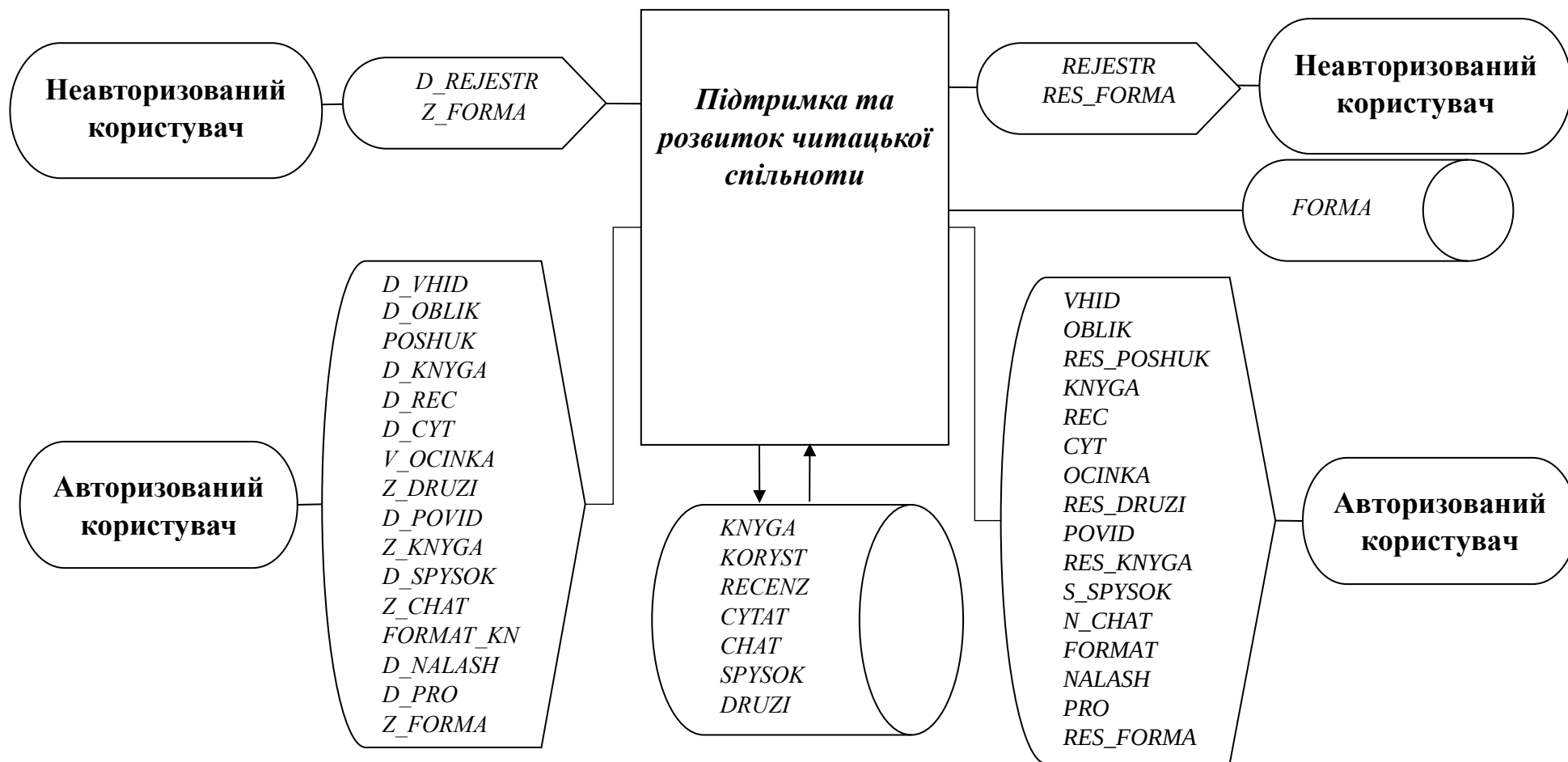


Рис. 2.2.1.1. – Інформаційна модель задачі (макет)

2.2.2. Вихідна інформація

Вихідна інформація включає результати взаємодії користувача з системою. Це може бути інформація про вхід або реєстрацію в обліковий запис, подробиці облікового запису користувача, результати пошуку книги за назвою, детальна інформація про книги, підтвердження додавання користувача у друзі, можливість надсилати приватні повідомлення, збереження налаштувань чату та інші взаємодії з функціоналом додатку. Частина даних була проаналізована для того, щоб описати наслідки вирішення проблеми та ідентифікувати отримувача цієї інформації:

- Отримання доступу до облікового запису користувача здійснюється за допомогою введення логіну та паролю. Успішна авторизація в системі включає взаємодію користувача та системи для підтвердження та управління обліковим записом.
- Інформація про обліковий запис користувача є результатом взаємодії, одержувачами якої є сам користувач, а система зберігає та надає доступ до цих даних.
- Результати пошуку містять відповіді на пошукові запити, одержувачами яких є користувач та інші користувачі, які можуть бачити результати.
- Докладна інформація про книгу, яка виводиться в результаті взаємодії користувача, стає доступною користувачу, який шукав інформацію про книгу.
- Користувач може бачити цитати та рецензії до книги, що представляє собою результат взаємодії інших користувачів з системою.
- Користувач має можливість взаємодіяти з системою для оновлення своєї оцінки конкретної книги.
- Підтвердження додавання у друзі є результатом взаємодії, одержувачами якої є користувач, який додає друга, та сам доданий друг.
- Можливість відправлення приватних повідомлень визначається як результат взаємодії, одержувачами якої є користувачі, що обмінюються повідомленнями.

- Користувач може взаємодіяти з системою, створюючи власні списки та додаючи до них книги, що дозволяє організувати свою бібліотеку та виражати свій смак.
- Під час взаємодії з системою користувач може внести зміни в налаштування свого чату та особистого профілю, що дозволяє персоналізувати його досвід використання платформи.
- Користувач, взаємодіючи з системою, може відтворити формат конкретної книги, забезпечуючи комфортне читання та відображення тексту у відповідному стилі.
- Користувач отримує доступ до нового функціоналу, спілкуючись з системою, що розширює можливості та функціональність платформи.
- Користувач може взаємодіяти з системою, відправляючи форму зворотного зв'язку, що дозволяє надати власні думки, зауваження чи пропозиції, що поліпшують якість обслуговування чи функціоналу платформи.

Важливо зазначити, що ця вихідна інформація не тільки дозволяє виконувати певні дії користувача, але й використовується для аналізу та вдосконалення продукту, забезпечуючи якісне обслуговування та реагування на потреби користувачів.

Перелік і опис вихідних повідомлень наведено у табл. 2.2.2.1.

Таблиця 2.2.2.1. – Перелік і опис вихідних повідомлень

Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність видання	Термін видання і допустимий час затримки	Користувач інформації
1	2	3	4	5	6
Підтвердження входу в систему	<i>VHID</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Підтвердження реєстрації	<i>REJESTR</i>	Екранна форма	За потребою	Відразу	Неавторизований користувач
Інформація про обліковий запис	<i>OBLIK</i>	Екранна форма	За потребою	Відразу	Авторизований користувач

Продовження таблиці 2.2.2.1.

1	2	3	4	5	6
Результати пошуку	<i>RES_POSHUK</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Докладна інформація про книгу	<i>KNYGA</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Відображення рецензії до книги	<i>REC</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Відображення цитати до книги	<i>CYT</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Оновлена оцінка книги	<i>OCINKA</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Підтвердження додавання в друзі	<i>RES_DRUZI</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Надіслане повідомлення	<i>POVID</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Підтвердження додавання книги до списку	<i>RES_KNYGA</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Створення списку книг	<i>S_SPYSOK</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Збереження змін налаштувань чату	<i>N_CHAT</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Вибір формату книги	<i>FORMAT</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Збереження змін налаштувань профілю	<i>NALASH</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Відкриття додаткового функціоналу	<i>PRO</i>	Екранна форма	За потребою	Відразу	Авторизований користувач
Відправлена форма зворотного зв'язку	<i>RES_FORMA</i>	Екранна форма	За потребою	Відразу	Авторизований користувач, неавторизований користувач
Масив форм зворотного зв'язку	<i>FORMA</i>	Файл	По мірі надходження форм	По мірі надходження інформації	ІС

2.2.3. Вхідна інформація

У якості **вхідної інформації** веб-застосунок для підтримки та розвитку читацької спільноти передбачає введення різноманітних даних, пов'язаних із користувачами, зокрема тексту, чисел і зображень. Слід зазначити, що обсяг інформації є великим через різні аспекти роботи додатку – від персональних даних до вибору контенту та взаємодії з іншими користувачами. Джерела персональних даних користувача реалізуються під час реєстрації або при вході в особистий кабінет, тоді як банківські дані надходять від платіжної системи.

Персональні дані користувача повинні бути повними і точними для забезпечення безпеки та ідентифікації, а пошукові дані повинні повністю розуміти запит користувача для надання точних результатів.

Термінові операції з даними, такі як пошукові запити, вимагають швидкої обробки для надання актуальних результатів. Точність також особливо важлива, наприклад при обробці банківських даних або персональних даних користувачів, щоб запобігти помилкам і забезпечити конфіденційність.

У веб-додатку існують вхідні дані, які необхідні для входу в систему (облікові дані користувача), персональні дані, що містять інформацію про обліковий запис користувача, а також пошукові запити для пошуку конкретних книг. Користувачі також можуть вказати свої вподобання, обираючи книги для додавання до списку та формат читання. Ця вхідна інформація важлива для визначення споживчих потреб користувача та забезпечення персоналізованого досвіду роботи у веб-додатку. Крім того, інформація про вподобання користувача щодо зміни налаштувань чату, рецензій, цитувань, рейтингів книг, а також про те, чи має він акаунт з розширеним функціоналом, може допомогти налаштувати функціональність додатку до конкретних потреб кожного користувача.

Перелік і опис вхідних повідомлень наведено у табл. 2.2.3.1.

Таблиця 2.2.3.1. – Перелік і опис вхідних повідомлень

Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
1	2	3	4	5
Дані для входу в систему	<i>D_VHID</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Дані для реєстрації в системі	<i>D_REJESTR</i>	Екранна форма	За потребою, миттєва обробка	Неавторизований користувач
Запит на виведення особистих даних користувача	<i>D_OBLIK</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Пошуковий запит	<i>POSHUK</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Запит на отримання додаткової інформації про книгу	<i>D_KNYGA</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Введення даних для створення рецензії	<i>D_REC</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Введення даних для створення цитати	<i>D_CYT</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Виставлення оцінки книзі	<i>V_OCINKA</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Запит на додавання в друзі	<i>Z_DRUZI</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Введення текстового повідомлення	<i>D_POVID</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Запит на додавання книги в список	<i>Z_KNYGA</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Введення даних для створення списку книг	<i>D_SPYSOK</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Внесення змін в налаштування чату	<i>Z_CHAT</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Запит на відтворення формату книги	<i>FORMAT_KN</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач

Продовження таблиці 2.2.3.1.

1	2	3	4	5
Зміна особистих даних	<i>D_NALASH</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Введення банківських даних	<i>D_PRO</i>	Екранна форма	За потребою, миттєва обробка	Авторизований користувач
Заповнення форми зворотного зв'язку	<i>Z_FORMA</i>	Екранна форма	За потребою, миттєва обробка	Неавторизований користувач, авторизований користувач
Довідник книг	<i>KNYGA</i>	Файл	Оновлення 1 раз на тиждень	Інтернет
Довідник користувачів	<i>KORYST</i>	Файл	По мірі реєстрації користувачів	Авторизований користувач
Довідник рецензій	<i>RECENZ</i>	Файл	По мірі додавання рецензій	Авторизований користувач
Довідник цитат	<i>CYTAT</i>	Файл	По мірі додавання цитат	Авторизований користувач
Довідник чатів	<i>CHAT</i>	Файл	По мірі надходження інформації	Авторизований користувач
Довідник списків книг	<i>SPYSOK</i>	Файл	По мірі додавання списків	Авторизований користувач
Довідник друзів	<i>DRUZI</i>	Файл	По мірі додавання друзів	Авторизований користувач

2.2.4. Алгоритм розв'язання задачі

Було наведено масиви використаної та результативної інформації веб-застосунку у табл. 2.2.4.1. Масив форм зворотного зв'язку зберігає у собі контактну інформацію користувача та його повідомлення до технічної підтримки. У масиві книг зберігається вся детальна інформація про книгу, яка виведена на сайті. Масив користувачів зберігає в собі повну інформацію про користувача. Масиви рецензій і цитат містять в собі сам зміст та скорочену інформацію про книгу, до якої було написано рецензію, або цитату. У масиві чатів зберігаються повідомлення та налаштування чату. Масив списків книг зберігає в собі всі створені списки користувачем, їх назви та кольорову тему. Масив друзів містить у собі кількість доданих друзів у профілі користувача. Кожний масив має свій ідентифікатор та максимальну кількість записів.

Таблиця 2.2.4.1. – Перелік масивів використаної та результатної інформації

Масив	Ідентифікатор	Максимальна кількість записів
1	2	3
Масив форм зворотного зв'язку	FORMA	Без обмежень
Масив книг	KNYGA	Без обмежень
Масив користувачів	KORYST	Без обмежень
Масив рецензій	RECENZ	Без обмежень
Масив цитат	CYTAT	Без обмежень
Масив чатів	CHAT	Без обмежень
Масив списків книг	SPYSOK	Без обмежень
Масив друзів	DRUZI	Без обмежень

Математичне забезпечення інформаційної системи відіграє велику роль у забезпеченні точності та коректності операцій, що проводяться в веб-застосунку. Воно включає в себе різноманітні математичні моделі, формули та методи, які застосовуються для обробки даних, розрахунків та аналізу результатів. Таке забезпечення допомагає вирішувати різноманітні завдання, наприклад, розрахунок середнього значення оцінок, визначення кількості елементів у списку, аналіз змін у кількості друзів користувача, тощо. Воно дозволяє автоматизувати та оптимізувати процеси обробки та аналізу даних, забезпечуючи ефективну роботу інформаційної системи.

1) Розрахунок оцінки книги **(1)**:

- Нехай R_i - оцінка i -тої книги (в діапазоні від 1 до 5 зірочок).
- Нехай n_i - кількість користувачів, які оцінили i -ту книгу.
- Тоді середнє значення оцінки книги i може бути визначено як

$$\bar{R}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} R_i$$

2) Розрахунок кількості книг у списку після додавання книги до списку **(2)**:

- Нехай кількість книг у списку до додавання як N_0 .
- Нехай кількість доданих книг як M .
- Нехай кількість книг у списку після додавання як N .
- Тоді кількість книг у списку після додавання книги до списку може бути визначено як

$$N = N_0 + M$$

3) Додавання користувача до друзів **(3)**:

- Нехай p_o - кількість друзів користувача o .
- Коли користувач o додає користувача e у друзі – буде збільшено кількість друзів для обох користувачів, тоді

$$p_o = p_o + 1$$

$$p_e = p_e + 1$$

5) Додавання книги до списку **(5)**:

- Нехай s_u - кількість книг у списку u .
- Коли користувач додає книгу до списку – буде збільшено кількість книг у цьому ж списку, тоді

$$s_u = s_u + 1$$

Таким чином, математичне забезпечення є важливою складовою будь-якої інформаційної системи, оскільки воно забезпечує точність, надійність та ефективність її функціонування, дозволяючи здійснювати різноманітні обчислення та аналіз даних.

Побудова *алгоритму розв'язання задачі* для інформаційної системи є важливою складовою її успішної реалізації та функціонування. Цей процес вкрай важливий з різних причин. По-перше, ефективний алгоритм дозволяє системі автоматизувати та оптимізувати виконання різноманітних операцій та процесів, що значно полегшує роботу користувачів та підвищує продуктивність роботи системи в цілому. По-друге, побудова алгоритму забезпечує точність та надійність виконання завдань, що є критично важливим у багатьох галузях. Окрім того, побудова алгоритму дозволяє підвищити масштабованість системи, тобто забезпечити її працездатність навіть у випадку великої кількості даних та операцій. Це робить систему більш гнучкою та адаптивною до змін у вихідних умовах та обсягах завдань.

Треба зазначити, що зрозумілість алгоритмів роботи системи сприяє її ефективній розробці, супроводженню та покращенню в майбутньому. Тому побудова алгоритму розв'язання задачі є обов'язковою частиною процесу розробки інформаційної системи, що допомагає забезпечити її успішне функціонування та відповідність потребам користувачів.

На рис. 2.2.4.1. відображено основу алгоритму розв'язку задачі. Від цієї основи можна рухатися по модулях алгоритму (створення списку, виведений список книг, виведення детальної інформації про книгу, виведення цитати, виведення рецензії, оцінювання книги, вибір формату книги, виведення списку чатів, комунікація з іншим користувачем, пошук книги, зміна налаштувань, оплата додаткового функціоналу, форма зворотного зв'язку), які показані на рис. 2.2.4.2. – 2.2.4.14. Також в алгоритмі показано бази даних, вхідну та вихідну інформацію, які пов'язані з певними блоками.

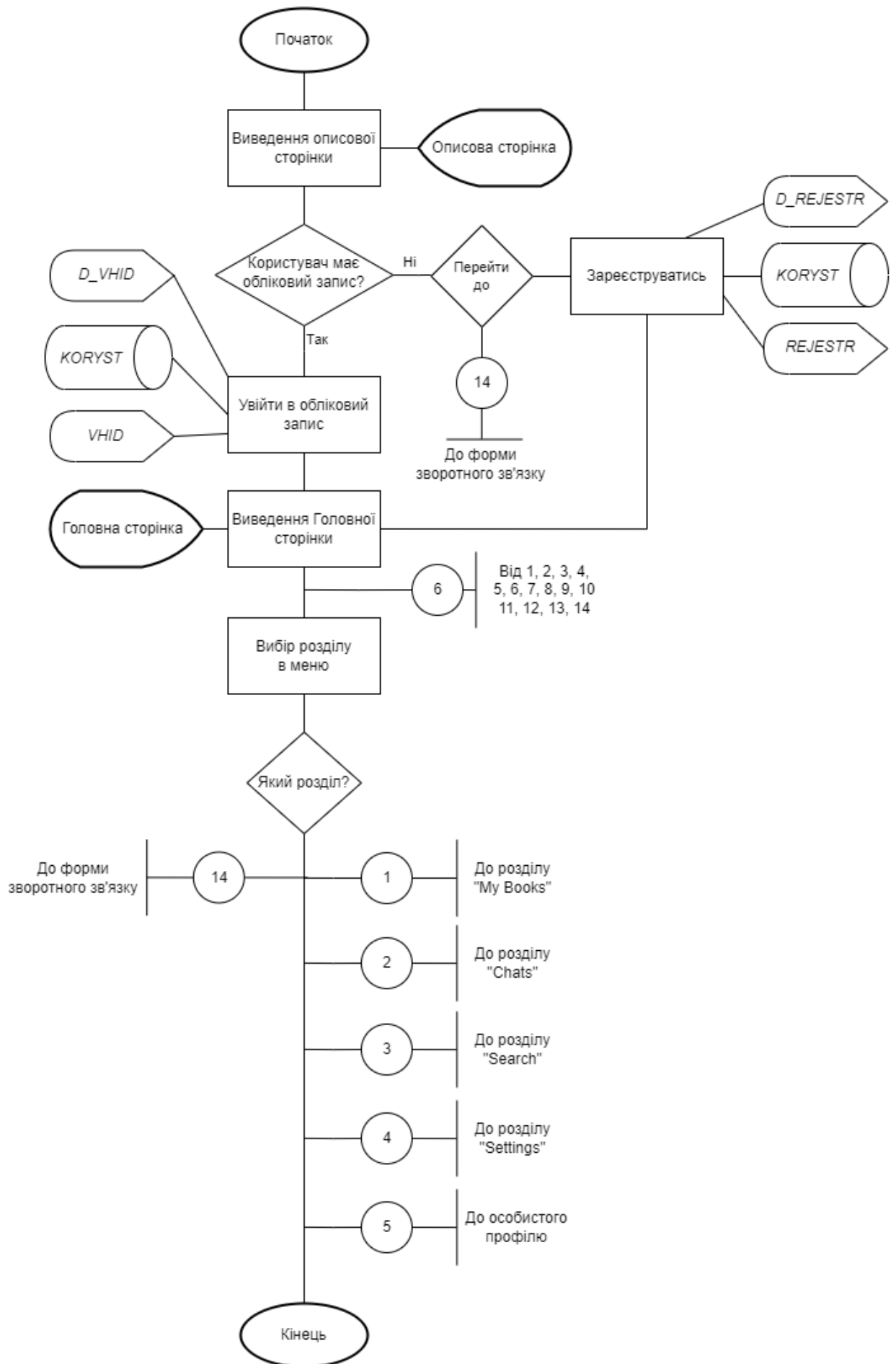


Рис. 2.2.4.1. – Алгоритм – Основа алгоритму

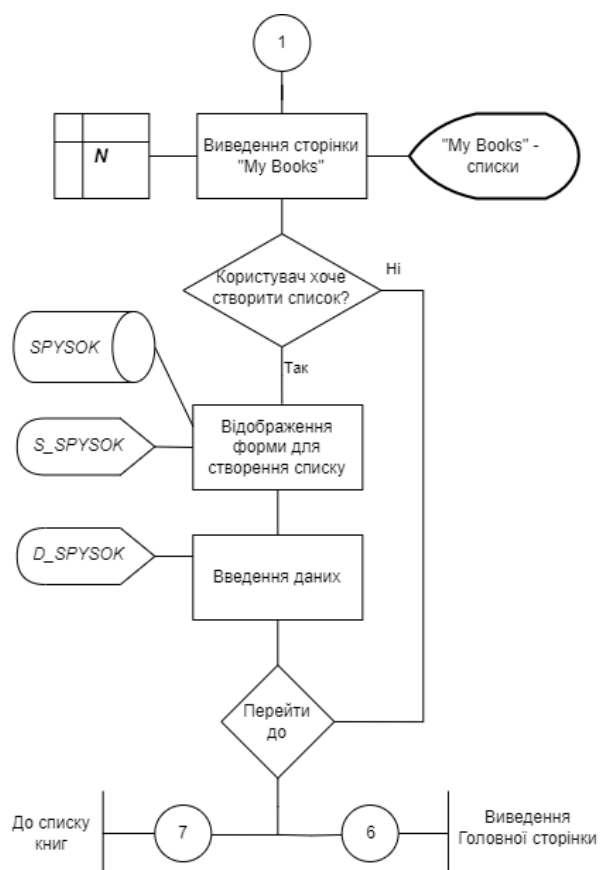


Рис. 2.2.4.2. – Алгоритм – Створення списку

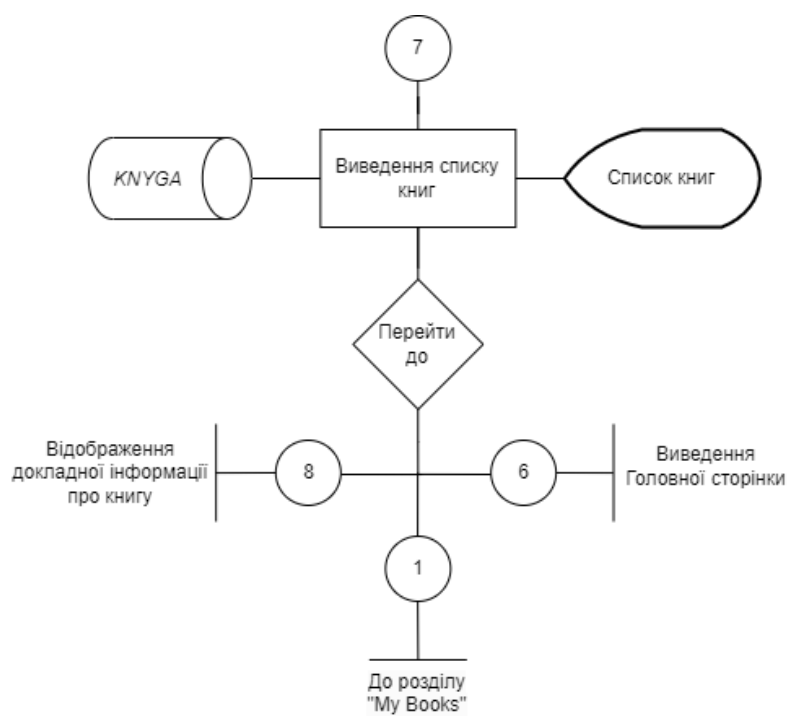


Рис. 2.2.4.3. – Алгоритм – Виведений список книг

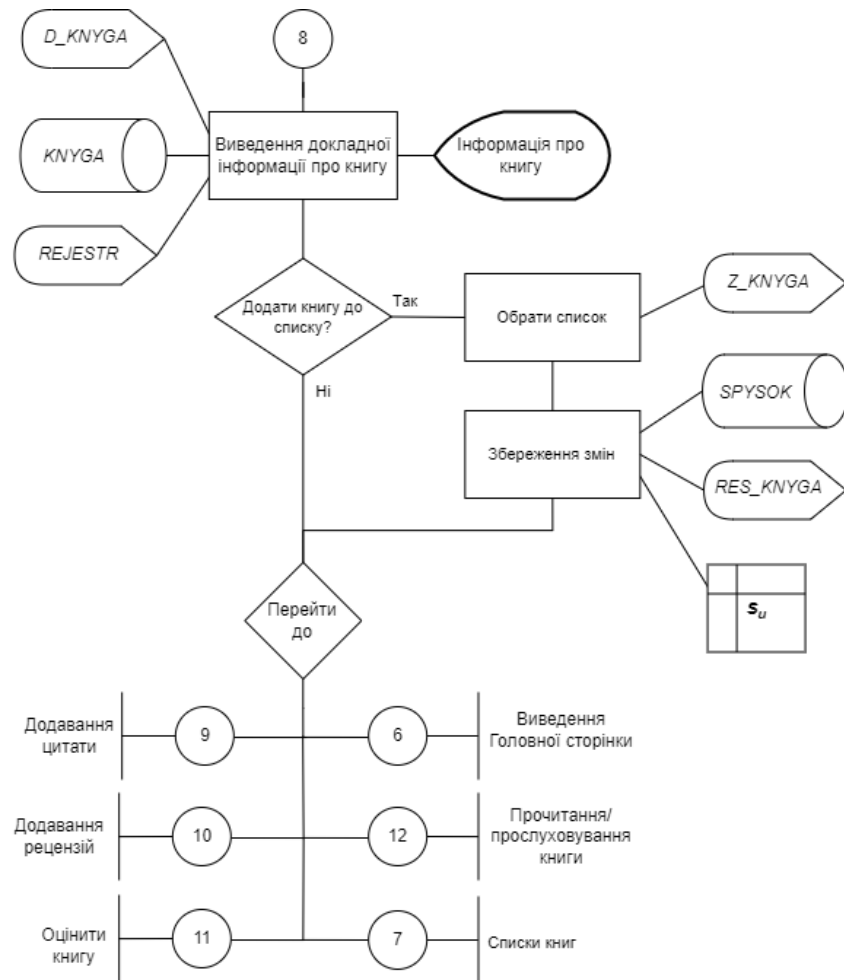


Рис. 2.2.4.4. – Алгоритм – Виведення детальної інформації про книгу

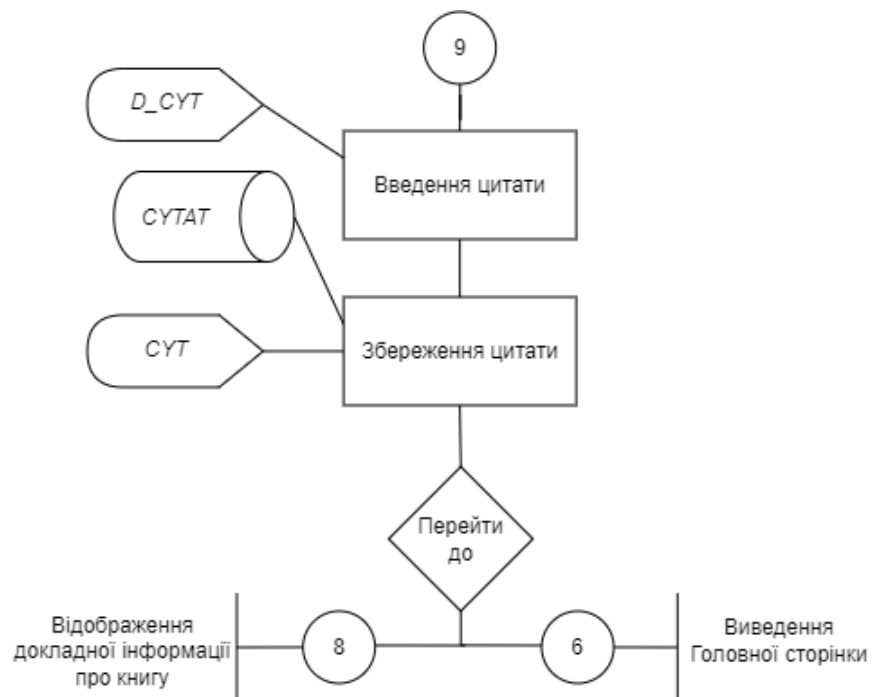


Рис. 2.2.4.5. – Алгоритм – Виведення цитати

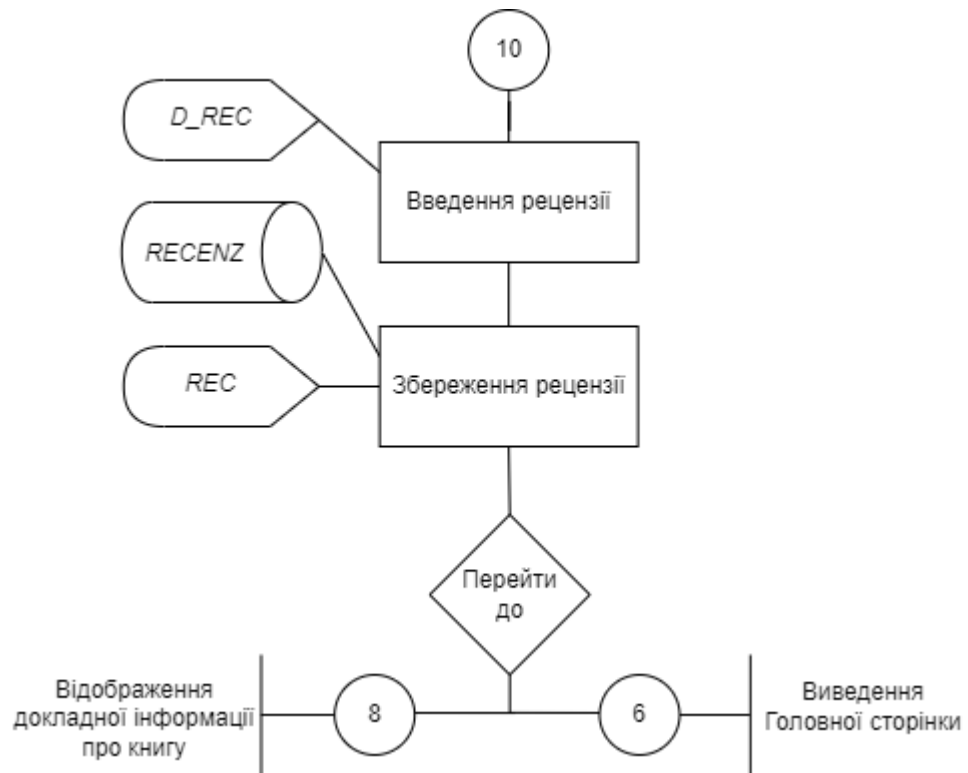


Рис. 2.2.4.6. – Алгоритм – Виведення рецензії

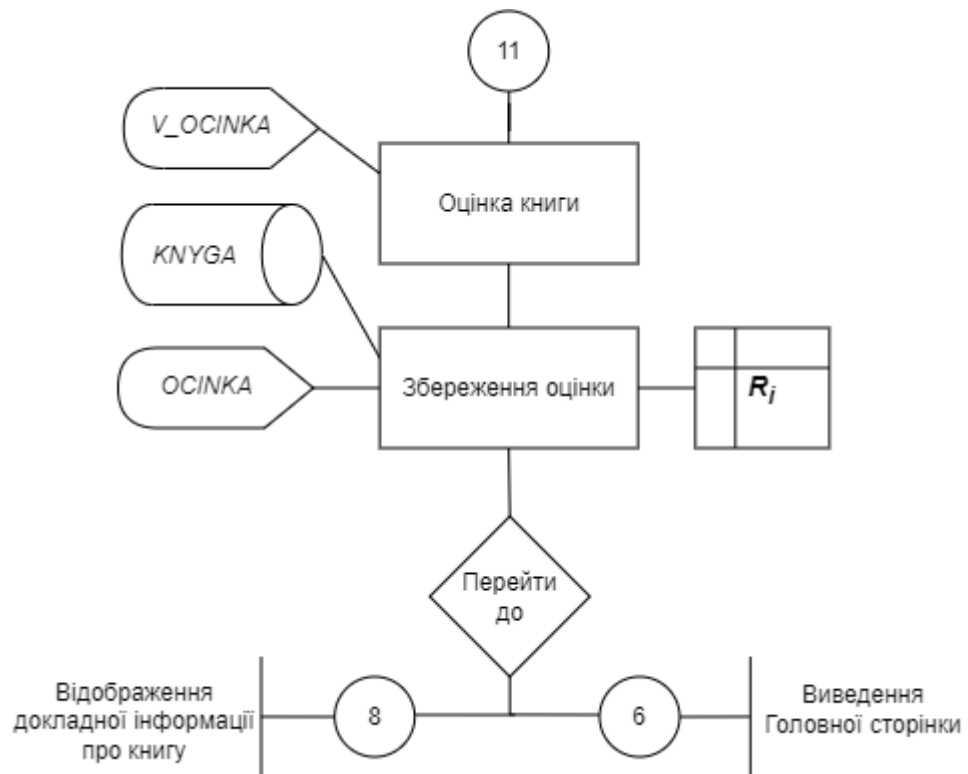


Рис. 2.2.4.7. – Алгоритм – Оцінювання книги

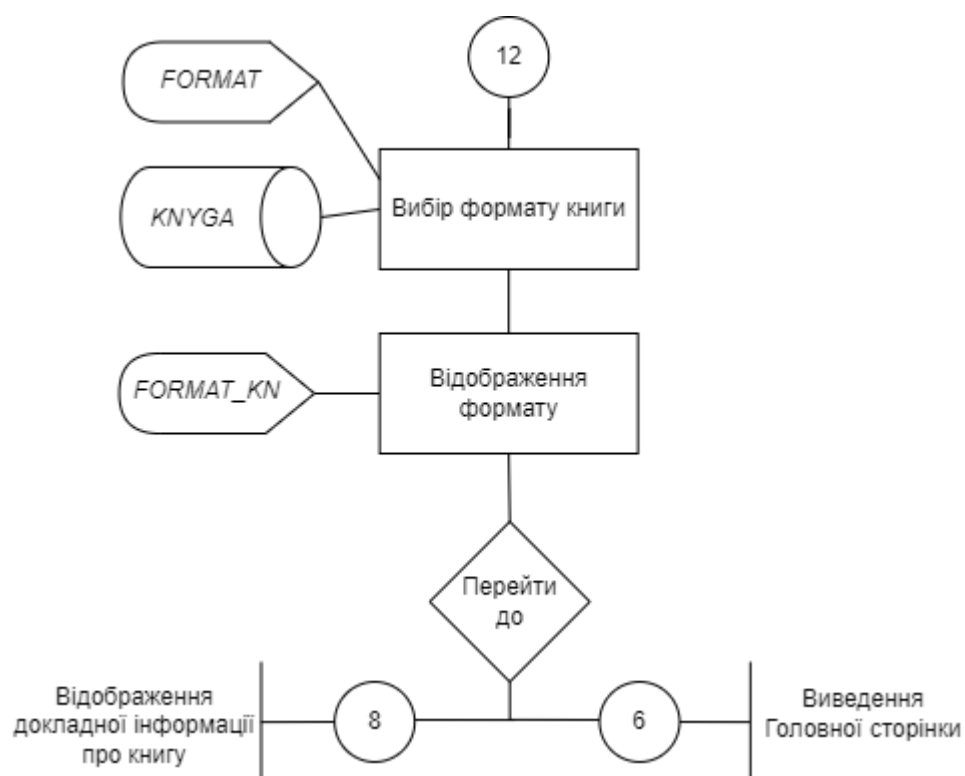


Рис. 2.2.4.8. – Алгоритм – Вибір формату книги

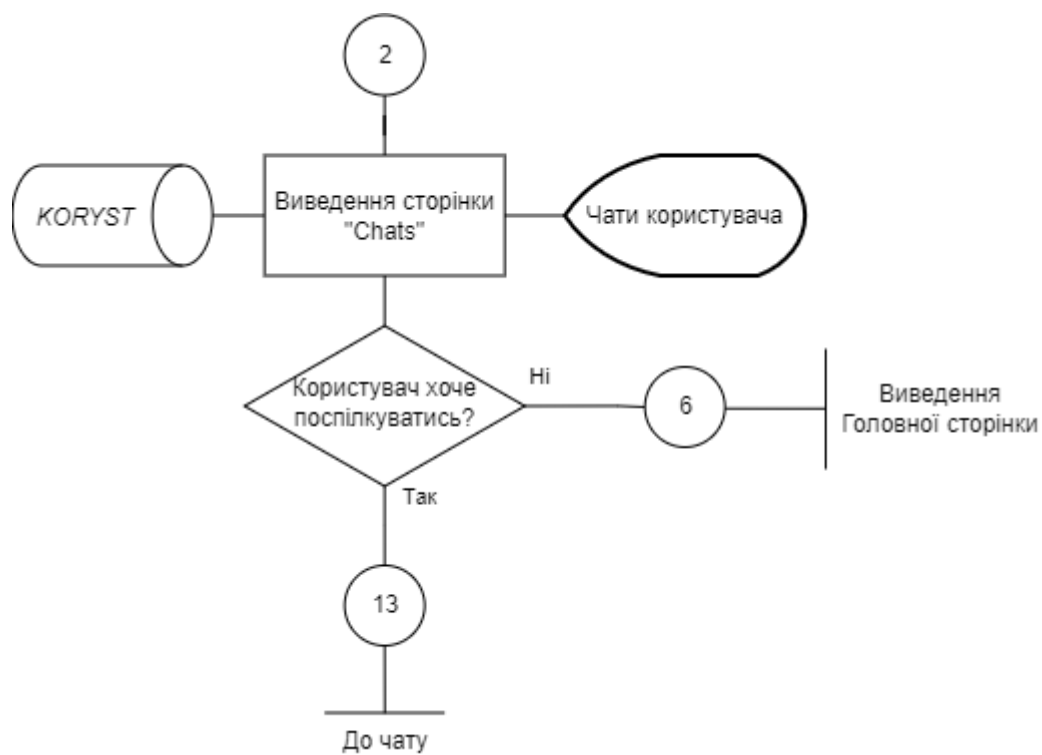


Рис. 2.2.4.9. – Алгоритм – Виведення списку чатів

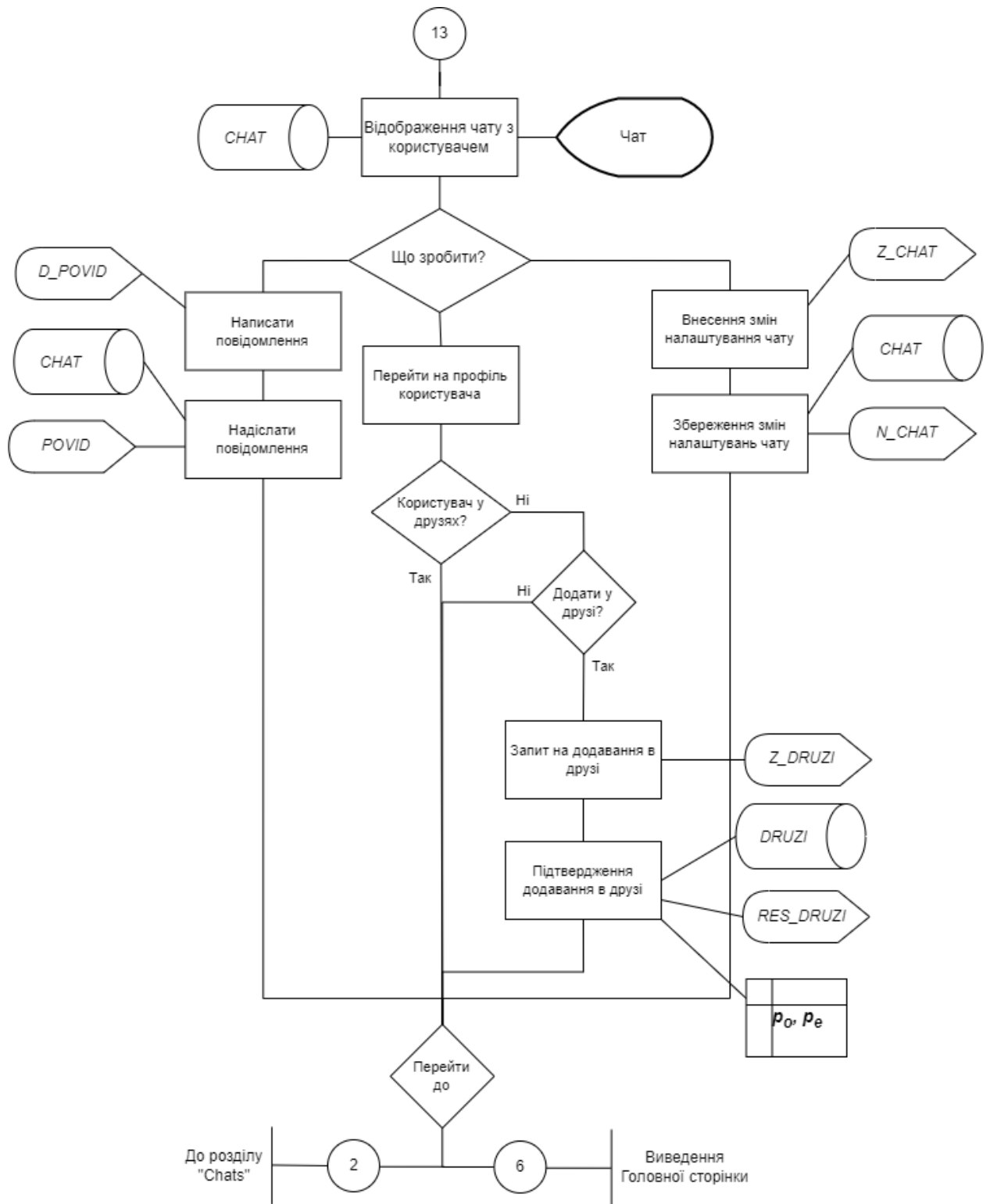


Рис. 2.2.4.10. – Алгоритм – Комунікація з іншим користувачем

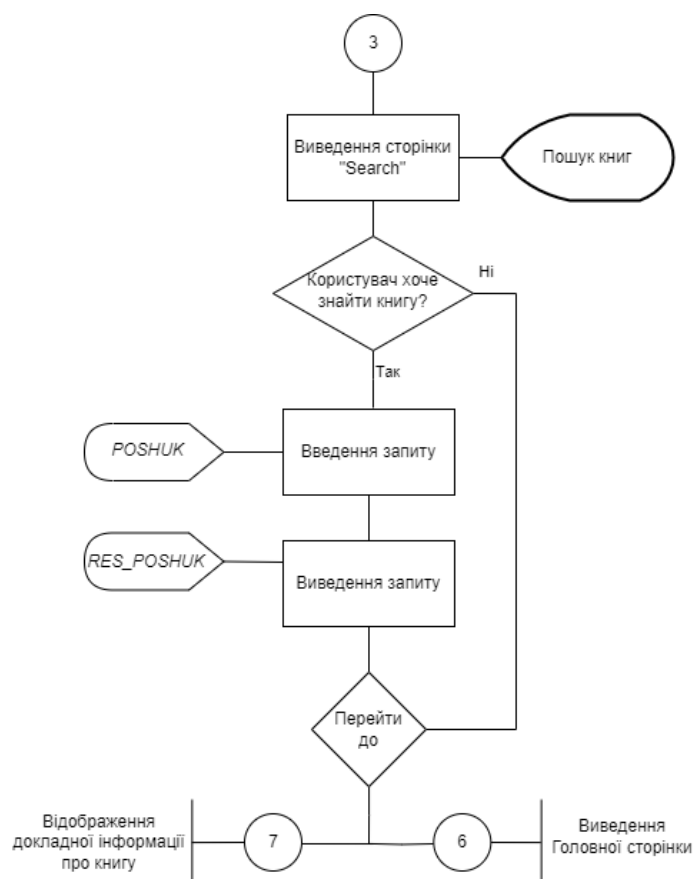


Рис. 2.2.4.11. – Алгоритм – Пошук книги

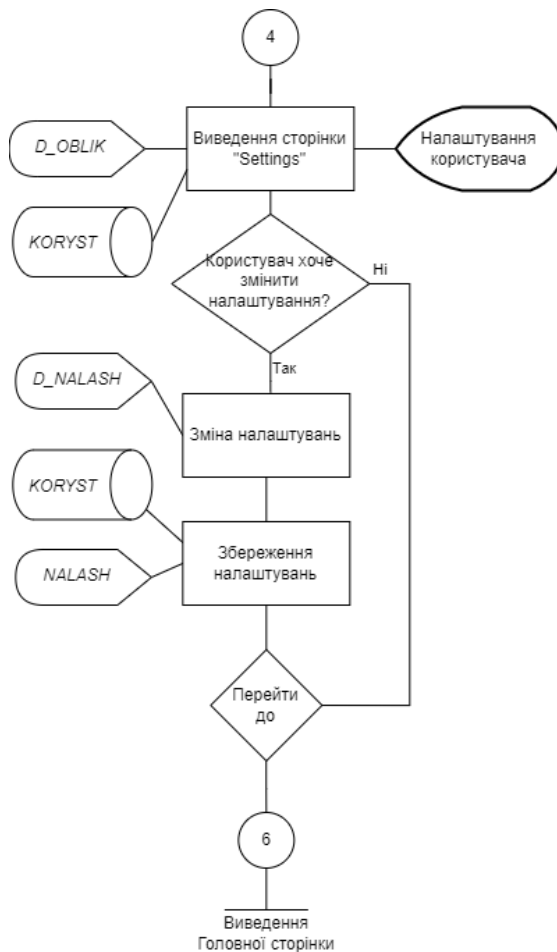


Рис. 2.2.4.12. – Алгоритм – Зміна налаштувань

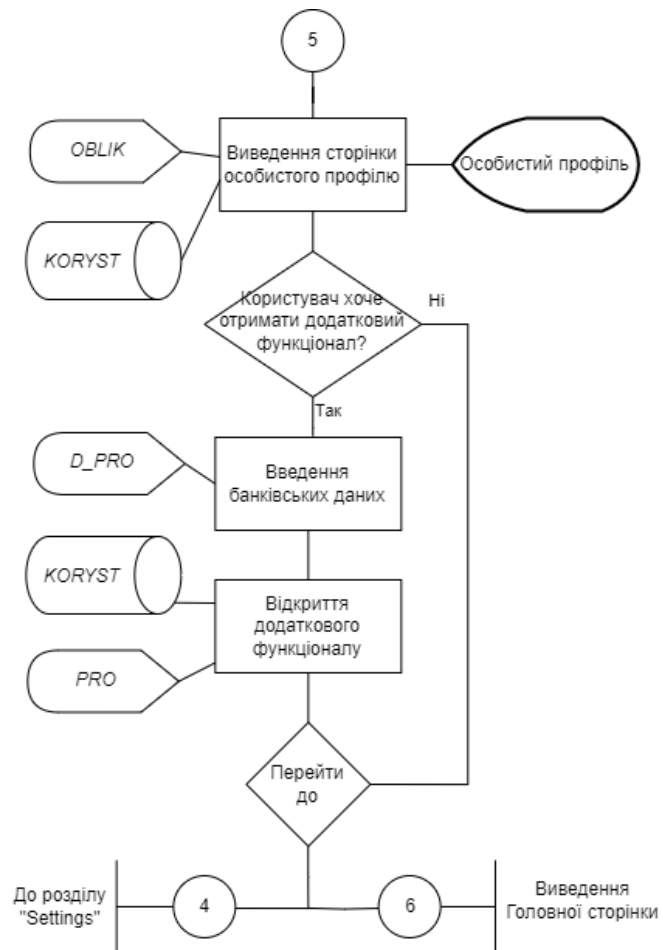


Рис. 2.2.4.13. – Алгоритм – Оплата додаткового функціоналу

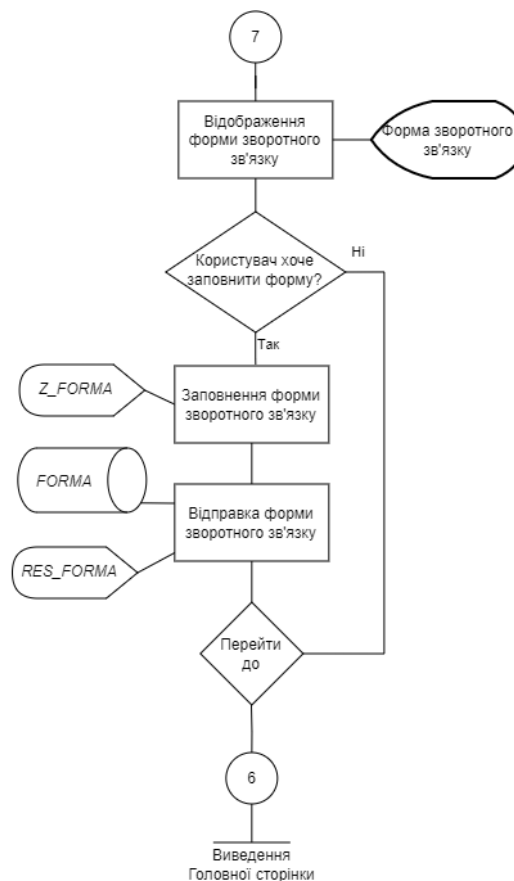


Рис. 2.2.4.14. – Алгоритм – Форма зворотного зв'язку

2.3. Моделювання інформаційної підсистеми

2.3.1. Моделювання поведінки системи

Задля реалізації моделювання інформаційної підсистеми було обрано об'єктний підхід, який допоможе зрозуміти складну систему за допомогою спрощення шляхом абстрагування. Під час моделювання було використано програмне середовище Enterprise Architect.

Діаграма варіантів використання, або прецедентів (Use-Case Diagrams) – це графічне представлення поведінки системи – варіантів її використання. Було побудовано модель поведінки системи для підтримки і розвитку читацької спільноти, яку подано в нотації UML, діаграмою прецедентів, що описує доступні дії і випадки на рис. 2.3.1.1.

Для моєї інформаційної системи було створено два актори – неавторизований користувач, авторизований користувач. Кожен з них має свої прецеденти: неавторизований користувач (увійти в обліковий запис; зареєструватися; відправити форму зворотного зв'язку), авторизований користувач (відправити форму зворотного зв'язку; пошук книг за назвою; взаємодіяти з книгами, що включає в себе написати рецензію/цитату до книги, оцінити книгу, обрати формат читання – E-Book, Audio-Book; додати книгу до списку; комунікація з іншими користувачами, що включає в себе спілкування в чаті і можливість зміни налаштувань у чаті, додавання у друзі; створити список книг; оплата додаткового функціоналу, який включає в себе отримання додаткових функцій та є розширенням під час прочитання/прослуховування книги, бо повну версію книги можна отримати лише оплативши додатковий функціонал). Отже, щоб користуватися функціоналом веб-застосунку треба увійти в існуючий обліковий запис, або зареєструватися.

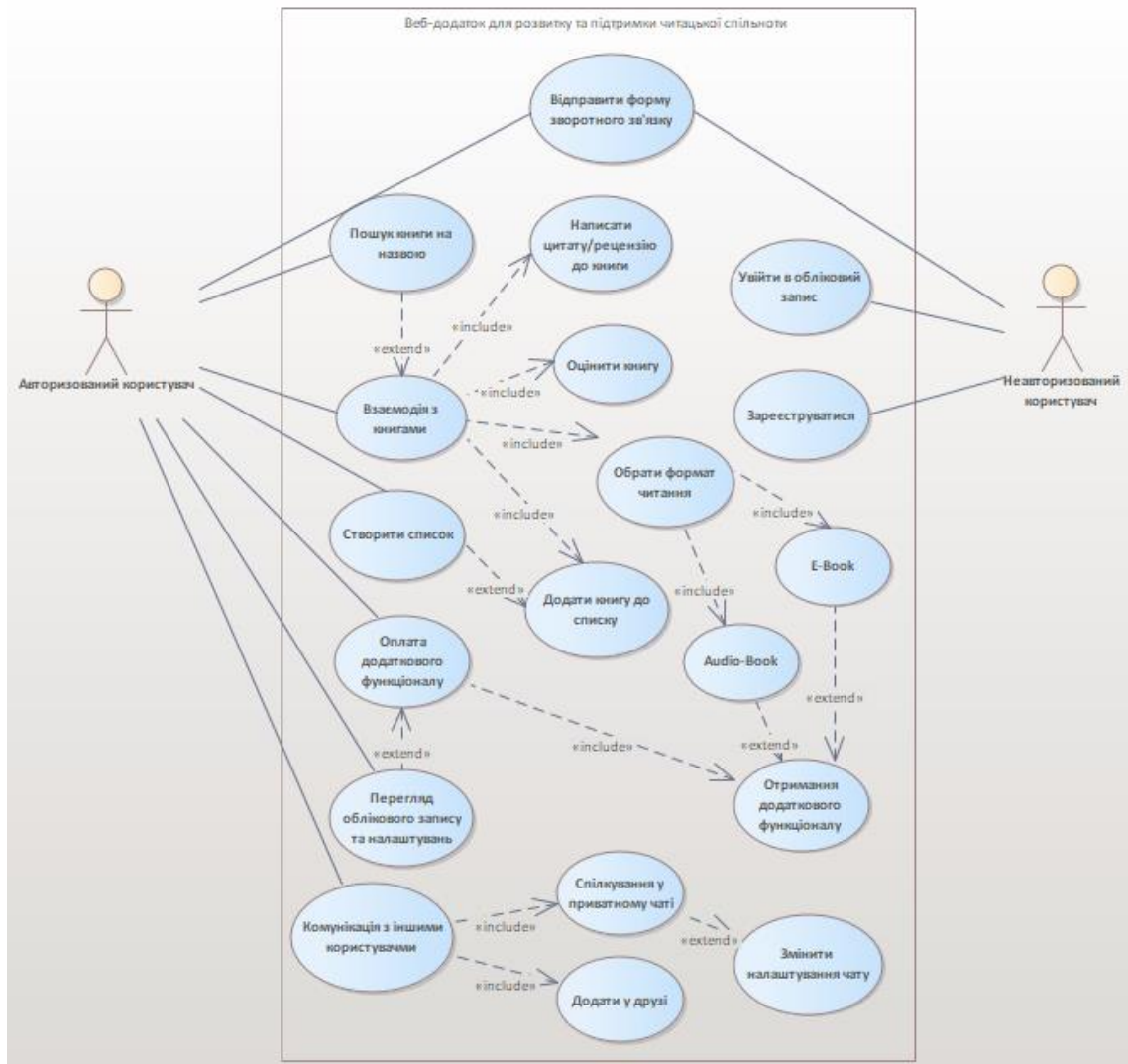


Рис. 2.3.1.1. – Діаграма варіантів використання

Діаграма послідовності (Sequence Diagram) – це модель, яка описує поведінку взаємодіючих груп об'єктів в рамках одного варіанту використання.

Діаграми послідовності було створено на кожний варіант використання діаграми прецедентів.

Неавторизований користувач:

1) Реєстрація неавторизованого користувача показана на рис. 2.3.1.2.: у виведену форму для введення персональних даних користувач має ввести e-mail, пароль та повторення паролю. Ці дані пересилаються у функції `sign()` та перевіряються функцією `validate()` на правильність введення. Після внесення даних у базу даних користувача інформація повертається до контролера, потім до в'юшки форми реєстрації, яка оновлює свій зовнішній вигляд та до користувача.

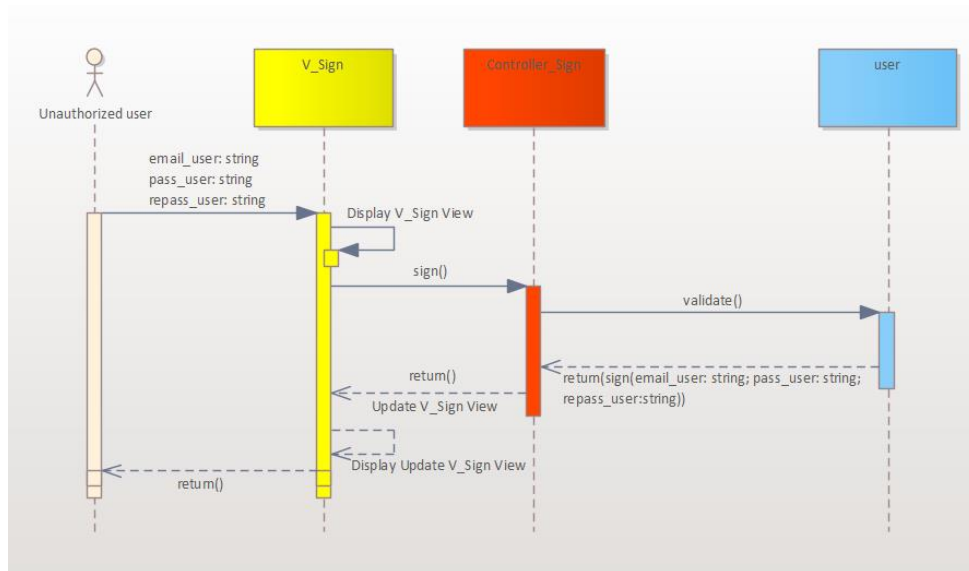


Рис. 2.3.1.2. – Діаграма послідовності – Регістрація неавторизованого користувача

1) Вхід в обліковий запис неавторизованого користувача показаний на рис. 2.3.1.3.: у виведену форму для введення персональних даних користувач має ввести e-mail та пароль. Ці дані пересилаються у функції `log()` та перевіряються функцією `validate()` на правильність введення. Після внесення даних у базу даних користувача інформація повертається до контролера, потім до в'юшки форми входу в обліковий запис, яка оновлює свій зовнішній вигляд та до користувача.

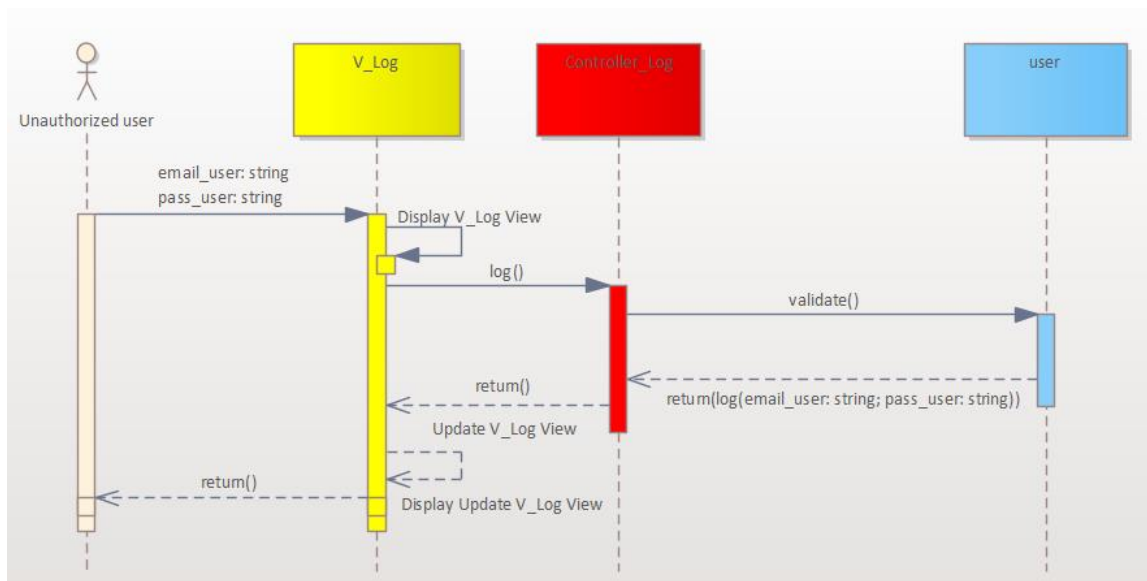


Рис. 2.3.1.3. – Діаграма послідовності – Вхід в обліковий запис неавторизованого користувача

2) Відправлення неавторизованим користувачем форми зворотного зв'язку показано на рис. 2.3.1.4.: у виведену форму для введення даних у форму користувач має ввести e-mail та своє повідомлення технічній підтримці. Ці дані пересилаються у функції `sendform()` та перевіряються функцією `formval()` чи всі комірки були заповнені. Після внесення даних у базу даних форми інформація повертається до контролера, потім до в'юшки форми зворотного зв'язку, яка оновлює свій зовнішні вигляд та показує повідомлення, що форма відправлена, та до неавторизованого користувача.

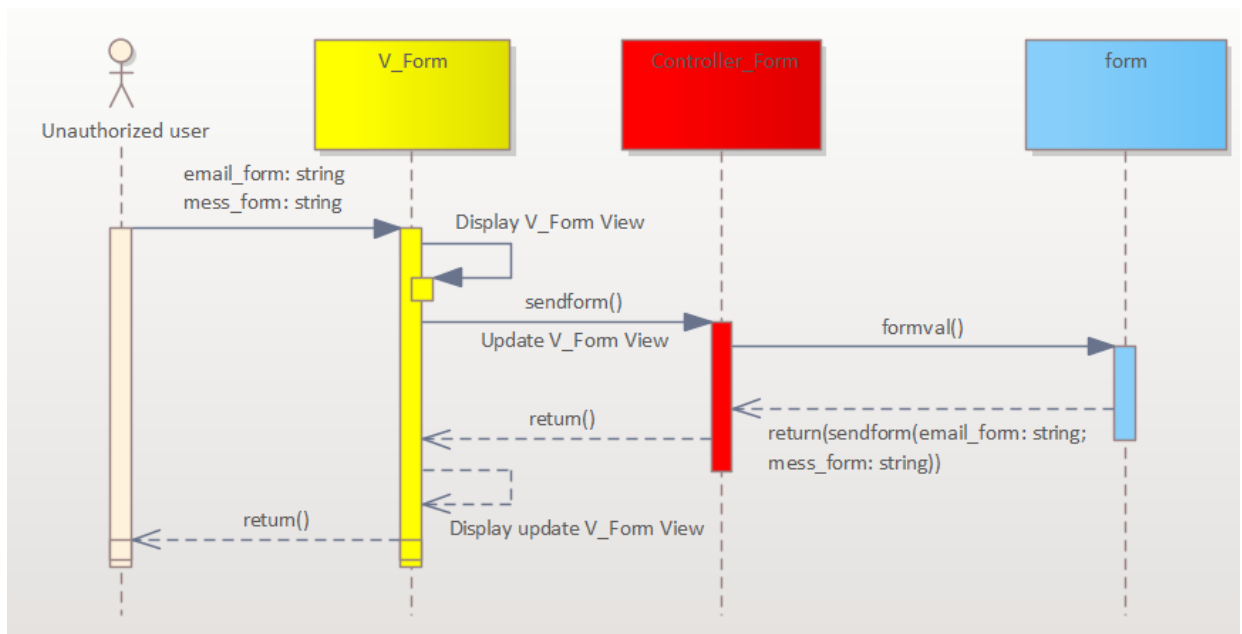


Рис. 2.3.1.4. – Діаграма послідовності – Відправлення неавторизованим користувачем форми зворотного зв'язку

Авторизований користувач:

1) Відправлення авторизованим користувачем форми зворотного зв'язку (так само як і у варіанті з неавторизованим користувачем) показано на рис. 2.3.1.5.: у виведену форму для введення даних у форму користувач має ввести e-mail та своє повідомлення технічній підтримці. Ці дані пересилаються у функції `sendform()` та перевіряються функцією `formval()` чи всі комірки були заповнені. Після внесення даних у базу даних форми інформація повертається до контролера, потім до в'юшки форми зворотного зв'язку, яка оновлює свій зовнішні вигляд та показує повідомлення, що форма відправлена, та до авторизованого користувача.

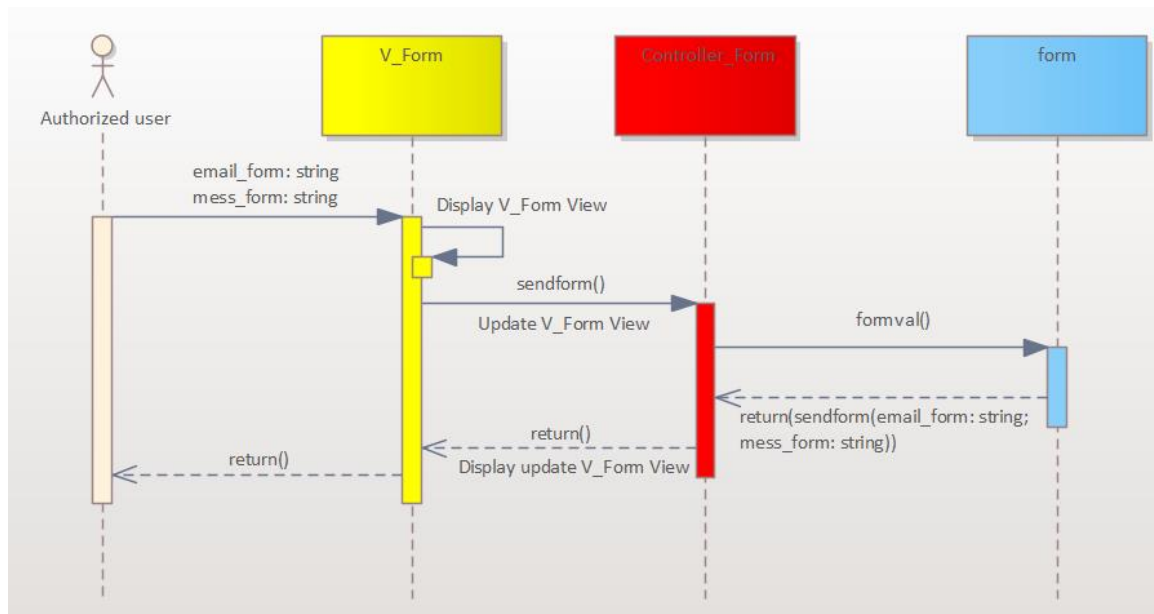


Рис. 2.3.1.5. – Діаграма послідовності – Відправлення авторизованим користувачем форми зворотного зв'язку

2) Пошук книг, або авторів авторизованим користувачем показаний на рис. 2.3.1.6.: перейшовши на вкладку пошуку у виведену форму для введення запиту користувач має ввести книгу. Ці дані пересилаються у функції `searchBooks()`. Після внесення даних у базу даних книги інформація повертається до контролера `searchResult()`, потім до в'юшки вкладки пошуку, яка оновлює свій зовнішній вигляд та показує результати запиту, та до авторизованого користувача.

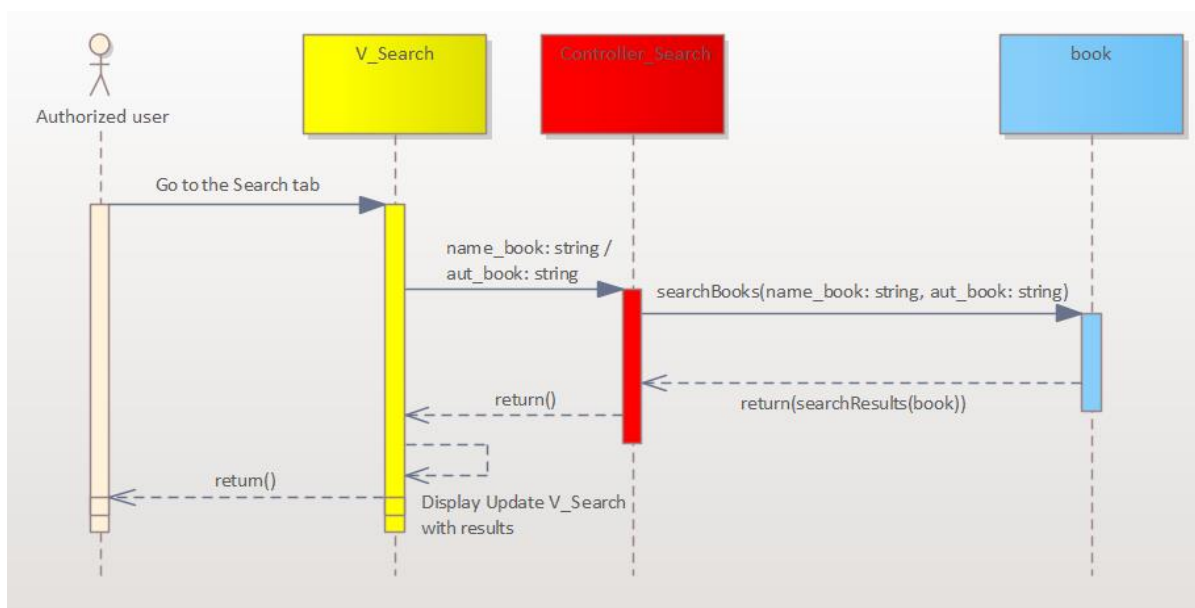


Рис. 2.3.1.6. – Діаграма послідовності – Пошук книг

3) Взаємодія з книгами – написання рецензії/цитати до книги авторизованим користувачем показано на рис. 2.3.1.7.: на сторінці відображення повної інформації про книгу користувач обирає між написанням рецензії, або цитати. У будь-якому варіанті він має ввести назву рецензії/цитати та її зміст. Якщо це варіант рецензії, то дані пересилаються у функції `createReview()`. Після внесення даних у базу даних рецензій інформація повертається до контролера `reviewResult()`, потім до в'юшки рецензій, яка оновлює свій зовнішній вигляд, та до авторизованого користувача. У випадку якщо це варіант цитати, то дані пересилаються у функції `createQuote()`. Після внесення даних у базу даних цитат інформація повертається до контролера `quoteResult()`, потім до в'юшки цитат, яка оновлює свій зовнішній вигляд, та до авторизованого користувача.

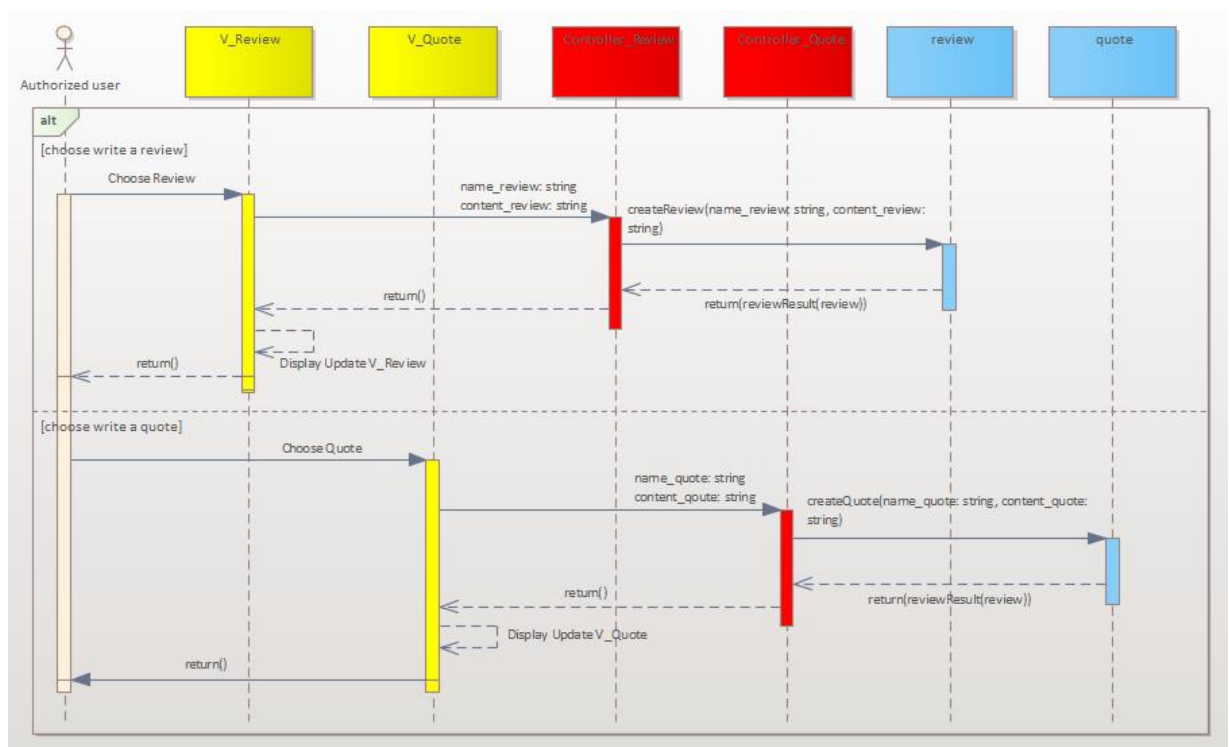


Рис. 2.3.1.7. – Діаграма послідовності – Взаємодія з книгами – написання рецензії/цитати до книги авторизованим користувачем

4) Взаємодія з книгами – оцінювання книги авторизованим користувачем показано на рис. 2.3.1.8.: перейшовши на вкладку детальної інформації про книгу користувач хоче її оцінити. Він надсилає свою оцінку контролеру, який передає її базі даних у функції `rate()`. Після внесення даних у базу даних книги інформація повертається до контролера, який розраховує

середню оцінку по книзі серед користувачів, потім до в'юшки вкладки детальної інформації про книгу, яка оновлює свій зовнішній вигляд та показує оновлене оцінювання з урахуванням голосу користувача, та до авторизованого користувача.

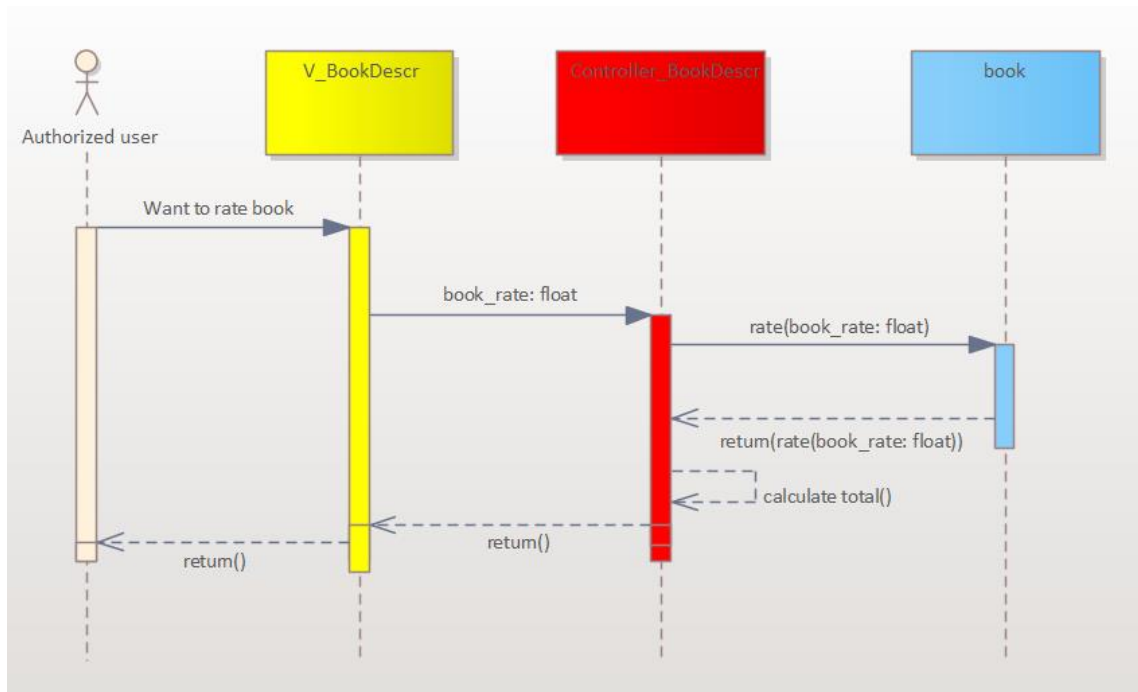


Рис. 2.3.1.8. – Діаграма послідовності – Взаємодія з книгами – оцінювання книги авторизованим користувачем

5) Взаємодія з книгами – прочитання книги в обраному форматі читання (E-Book, Audio-Book) авторизованим користувачем показано рис. 2.3.1.9.: перейшовши на вкладку детальної інформації про книгу користувач хоче її прочитати/прослухати. У варіанті вибору E-Book, система надсилає запит з перевіркою `check()` у базі даних книги та користувача (його статус на наявність додаткового функціоналу), яку версію книги він має (повну чи неповну). Якщо книга неповного варіанту, то і відповідно відображається демоверсія, а якщо користувач має розширений функціонал, то йому відобразиться повний варіант книги. Результат відображення повертається авторизованому користувачу.

Такий самий принцип у варіанті вибору Audio-Book. Система надсилає запит з перевіркою `check()` у базі даних книги та користувача (його статус на наявність додаткового функціоналу), яку версію книги він має (повну чи неповну). Якщо книга неповного варіанту, то і відповідно відображається

демоверсія Audio-Book, а якщо користувач має розширений функціонал, то йому відобразиться повний варіант Audio-Book. Результат відображення повертається авторизованому користувачу.

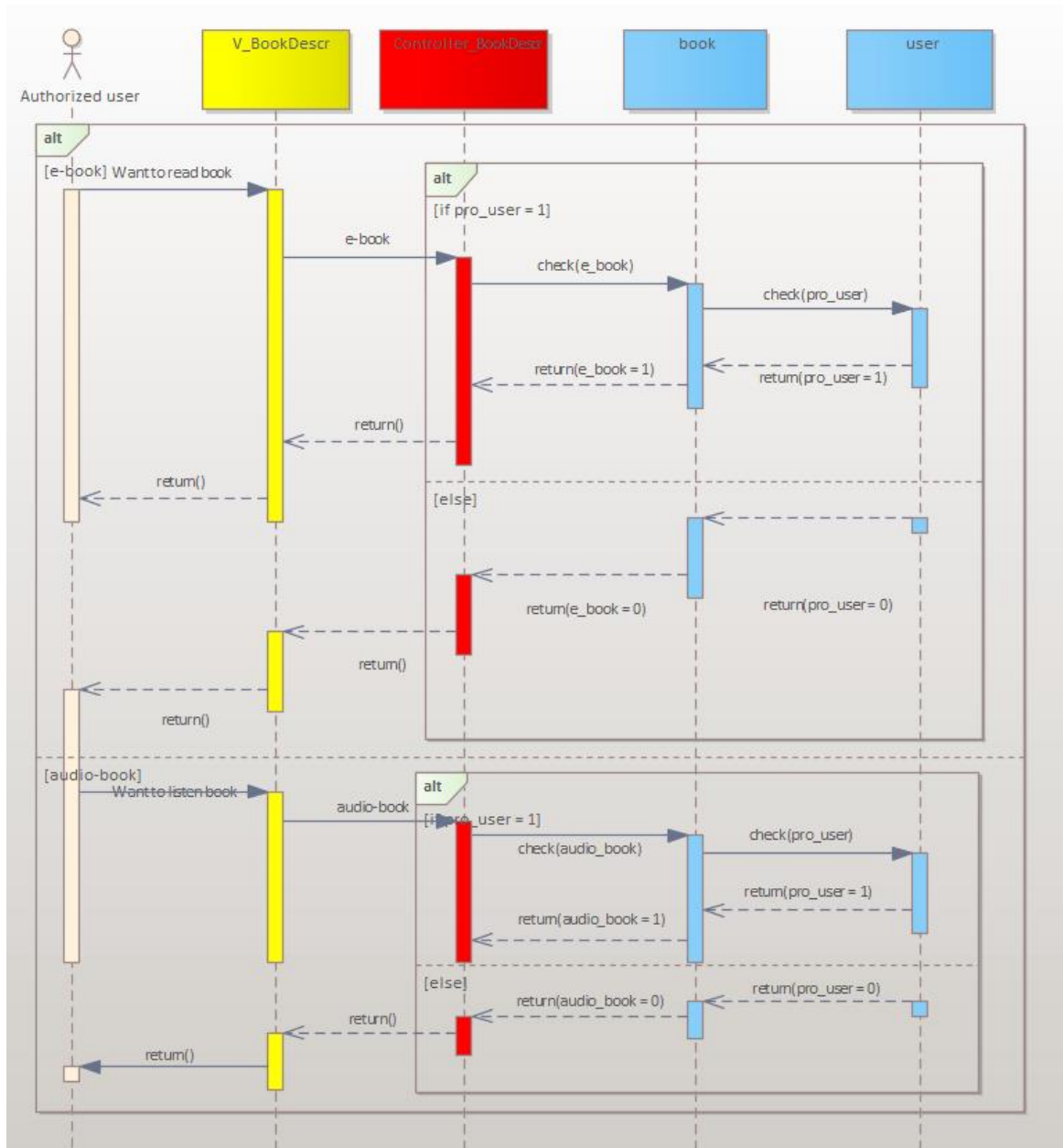


Рис. 2.3.1.9. – Діаграма послідовності – Взаємодія з книгами – прочитання книги в обраному форматі читання (E-Book, Audio-Book) авторизованим користувачем

б) Створення списку книг авторизованим користувачем показано рис. 2.3.1.10.: перейшовши на вкладку всіх списків книг користувача, він хоче створити ще один. Натискаючи на форму створення нового списку користувач вводить необхідні дані для створення: назва списку та вибір кольору серед запропонованих варіантів. Контролер передає цю інформацію базі даних

списків у функції `createList()`. Після внесення даних у базу даних списків інформація повертається до контролера, потім до в'юшки вкладки усіх списків книг, яка оновлює свій зовнішній вигляд – всі списки з урахуванням нового, та до авторизованого користувача.

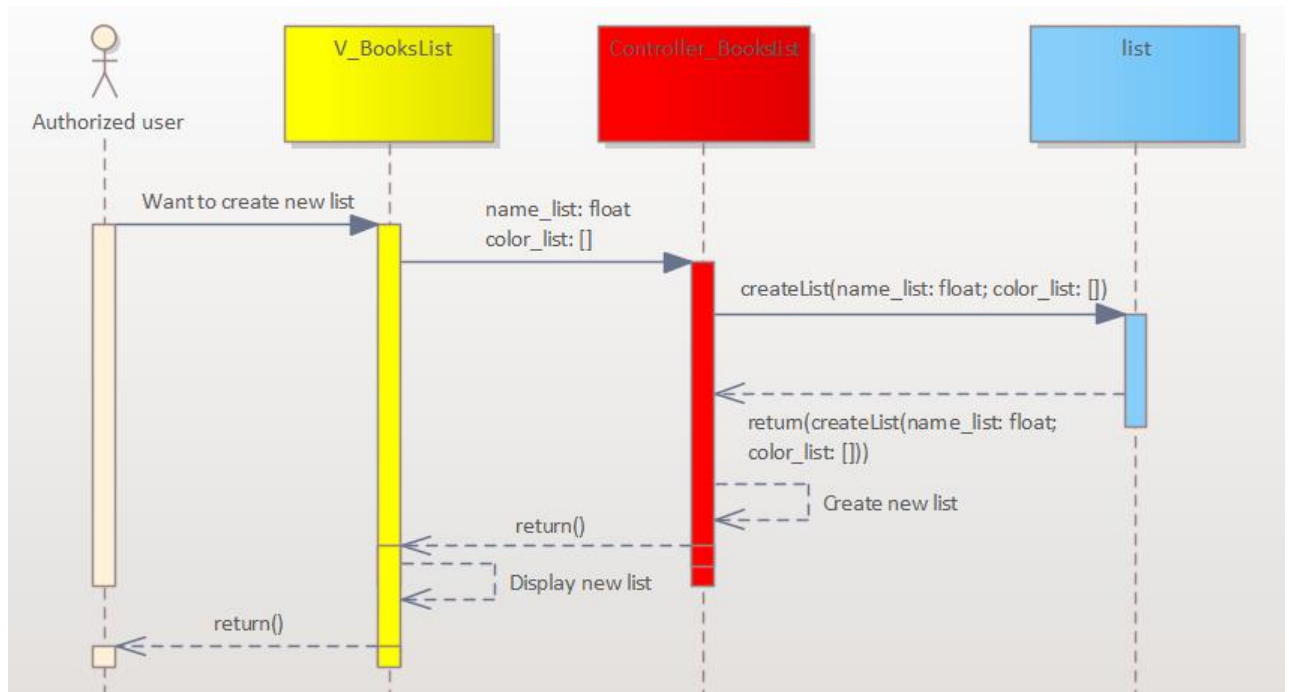


Рис. 2.3.1.10. – Діаграма послідовності – Створення списку книг авторизованим користувачем

7) Комунікація з іншими користувачами – спілкування у чаті, зміна налаштувань чату авторизованим користувачем показано рис. 2.3.1.11.: перейшовши на вкладку конкретного чату він хоче відправити повідомлення іншому користувачеві платформи, або змінити якісь налаштування чату. Якщо відбувається процес відправки повідомлення, то контролер відправляє до бази даних чату функцію `sendMess()` у якій міститься зміст повідомлення та його статус (поки що невідправлене). База даних оновлюється та змінює статус повідомлення на відправлене. Ця інформація повертається контролеру, потім в'юшці відображення чату, який виглядає оновлено, та до авторизованого користувача.

Якщо користувач хоче змінити налаштування обраного чату, то він повідомляє про це систему, обравши певну функцію. Контролер передає його запит базі даних чату: `changeWalp()` – зміна фону чату, `clear()` – очищення чату від повідомлень, `delete()` – видалення чату. У всіх випадках інформація

оновлюється та передається контролеру, потім відображається оновлений варіант чату, та до авторизованого користувача.

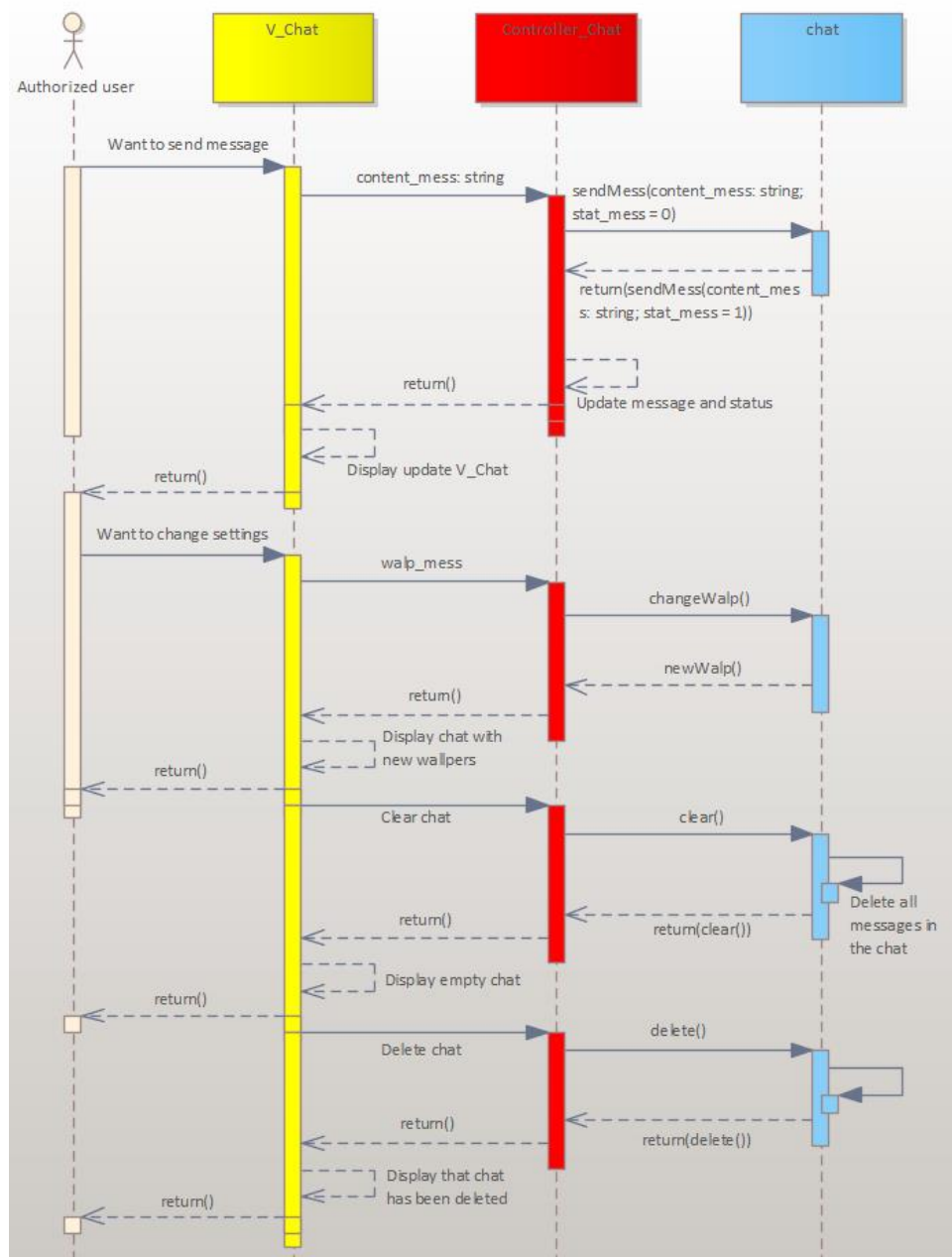


Рис. 2.3.1.11. – Діаграма послідовності – Комунікація з іншими користувачами – спілкування у чаті, зміна налаштувань чату авторизованим користувачем

8) Комунікація з іншими користувачами – додавання у друзі авторизованим користувачем показано рис. 2.3.1.12.: перейшовши на вкладку відображення профілю конкретного користувача надсилається запит на додавання у друзі. Ці дані пересилаються у функції `getFriends()`, де статус друзів = 0. Після внесення даних у базу даних друзів інформація повертається до контролера `setFriends()`, де статус друзів вже = 1, потім до в'юшки вкладки

профілю, яка оновлює свій зовнішній вигляд та показує, що обраний користувач доданий у друзі, та до авторизованого користувача.

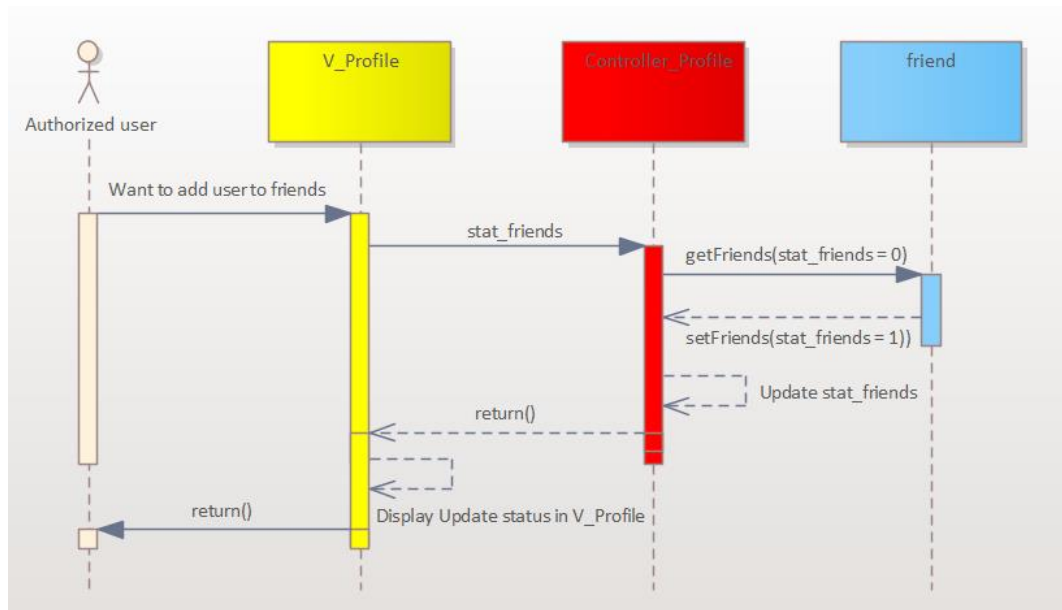


Рис. 2.3.1.12. – Діаграма послідовності – Комунікація з іншими користувачами – додавання у друзі авторизованим користувачем

9) Зміна налаштувань облікового запису авторизованим користувачем показано рис. 2.3.1.13. – перейшовши на вкладку відображення налаштувань профілю користувача можна внести зміни. Змінені дані пересилаються у функції `getSett()`. Після внесення змінених даних у базу даних користувача інформація повертається до контролера `setSett()`, потім до в'юшки вкладки налаштувань профілю, яка оновлює свій зовнішній вигляд та показує змінені налаштування, та до авторизованого користувача.

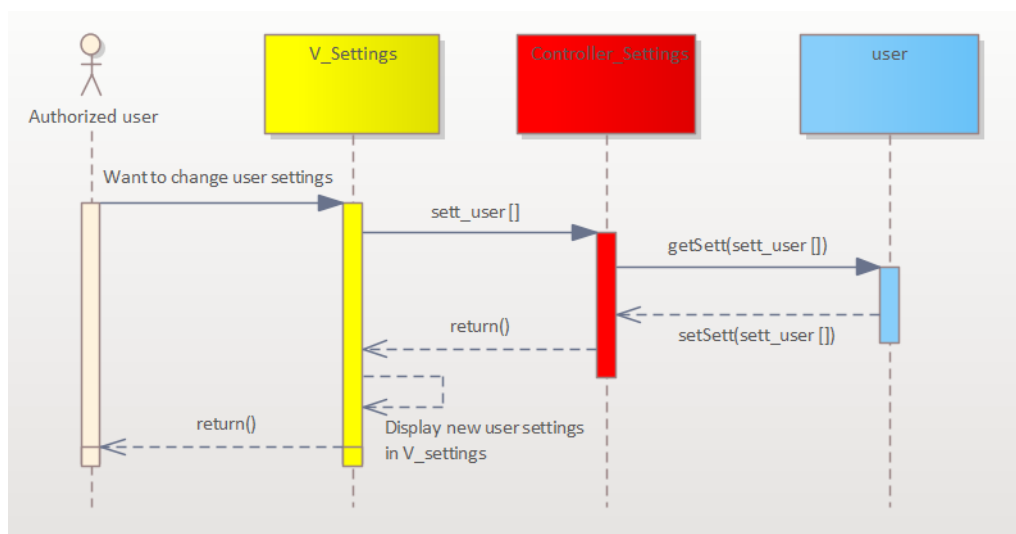


Рис. 2.3.1.13. – Діаграма послідовності – Зміна налаштувань облікового запису

10) Оплата додаткового функціоналу – отримання додаткових функцій авторизованим користувачем показано рис. 2.3.1.14.: перейшовши, наприклад, на вкладку відображення облікового запису користувача можна отримати, при оплаті через платіжну систему, профіль з розширеними можливостями. Користувач передає свої персональні дані платіжній системі, вона їх опрацьовує та передає результат контролеру, який передає функцію `getPro()` базі даних користувача. База даних повертає результат функцією `setPro()`, користувач отримує більше функцій, ця інформація оновлюється на його сторінці, та повертається авторизованому користувачу.

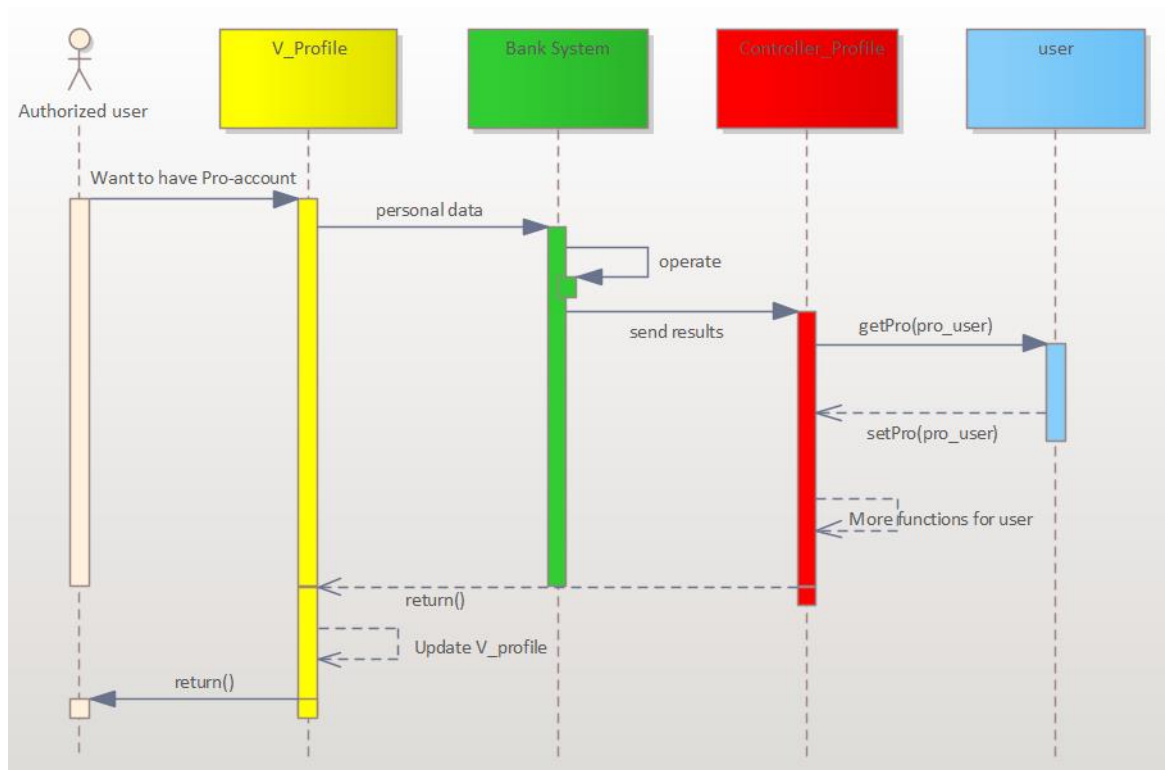


Рис. 2.3.1.14. – Діаграма послідовності – Оплата додаткового функціоналу – отримання додаткових функцій авторизованим користувачем

Діаграма дій (Activity Diagram) – показує динаміку роботи системи, дії з об'єктами, які включають в себе альтернативні і паралельні.

Було відображено процес відправлення форми зворотного зв'язку (рис. 2.3.1.15.). Користувач входить на сайт та обирає функцію відправки форми зворотного зв'язку технічній підтримці. Він вносить дані та підтверджує відправку форми. Потім ця форма надходить до технічної підтримки, де вони отримують, обробляють, пишуть та відправляють відповідь користувачеві. У

цей самий час користувач може або виконувати інші дії на сайті, або чекати на відповідь. Також реалізовано вихід з веб-застосунку.

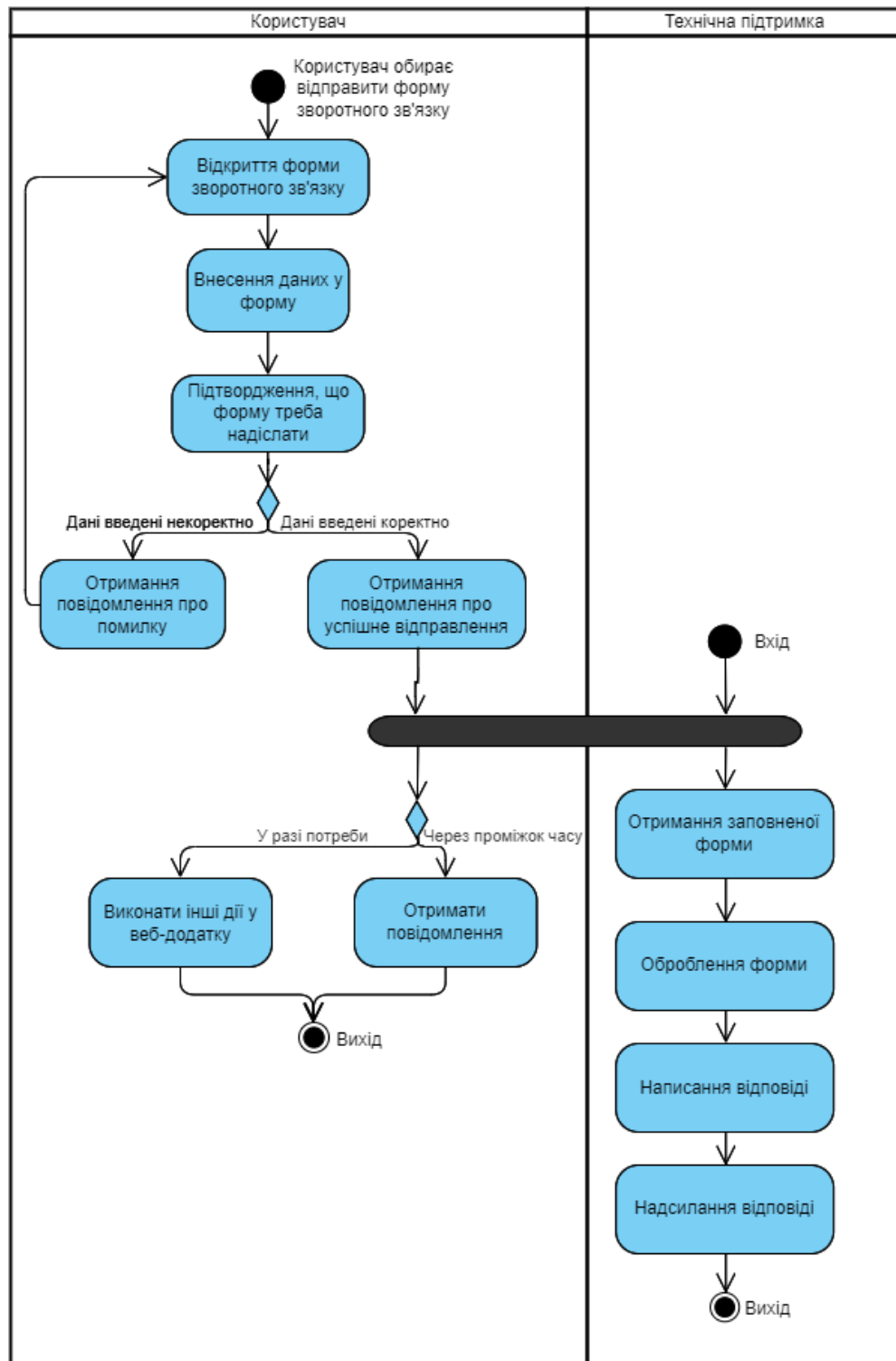


Рис. 2.3.1.15. – Activity Diagram

2.3.2. Модель структури системи

Діаграма класів (Class Diagrams) демонструє типи об'єктів системи, їх властивості (атрибути і операції) та різноманітні статичні відношення, які між ними є.

Було обрано патерн MVC (Model – View – Controller), який можна використовувати для побудування інформаційної системи.

- Модель (Model) відповідає на команди, які дає контролер, надаючи дані і змінюючи свій стан.
- Подання (View) демонструє дані моделі користувачам, реагуючи на зміни моделі.
- Контролер (Controller) пояснює дії користувача та повідомляє модель про необхідність змін.

Перелік конкретних елементів, які будуть використані при функціонуванні ІС:

- Моделі: form, review, quote, list, user, book, friends, chat.
- Подання: V_BookList, V_List, V_Search, V_Settings, V_Profile, V_ChatList, V_Chat, V_BookDescr, V_Form, V_Sign, V_Log, V_Rewiev, V_Quote.
- Контролер: Control_BookList, Control_List, Control_Search, Control_Settings, Control_Profile, Control_ChatList, Control_Chat, Control_BookDescr, Control_Form, Control_Sign, Control_Log, Control_Rewiev, Control_Quote.

Для візуальної зрозумілості класи різних типів показані трьома кольорами на рис. 2.3.2.1.: жовтий для подання, червоний для контролерів та синій для моделей.

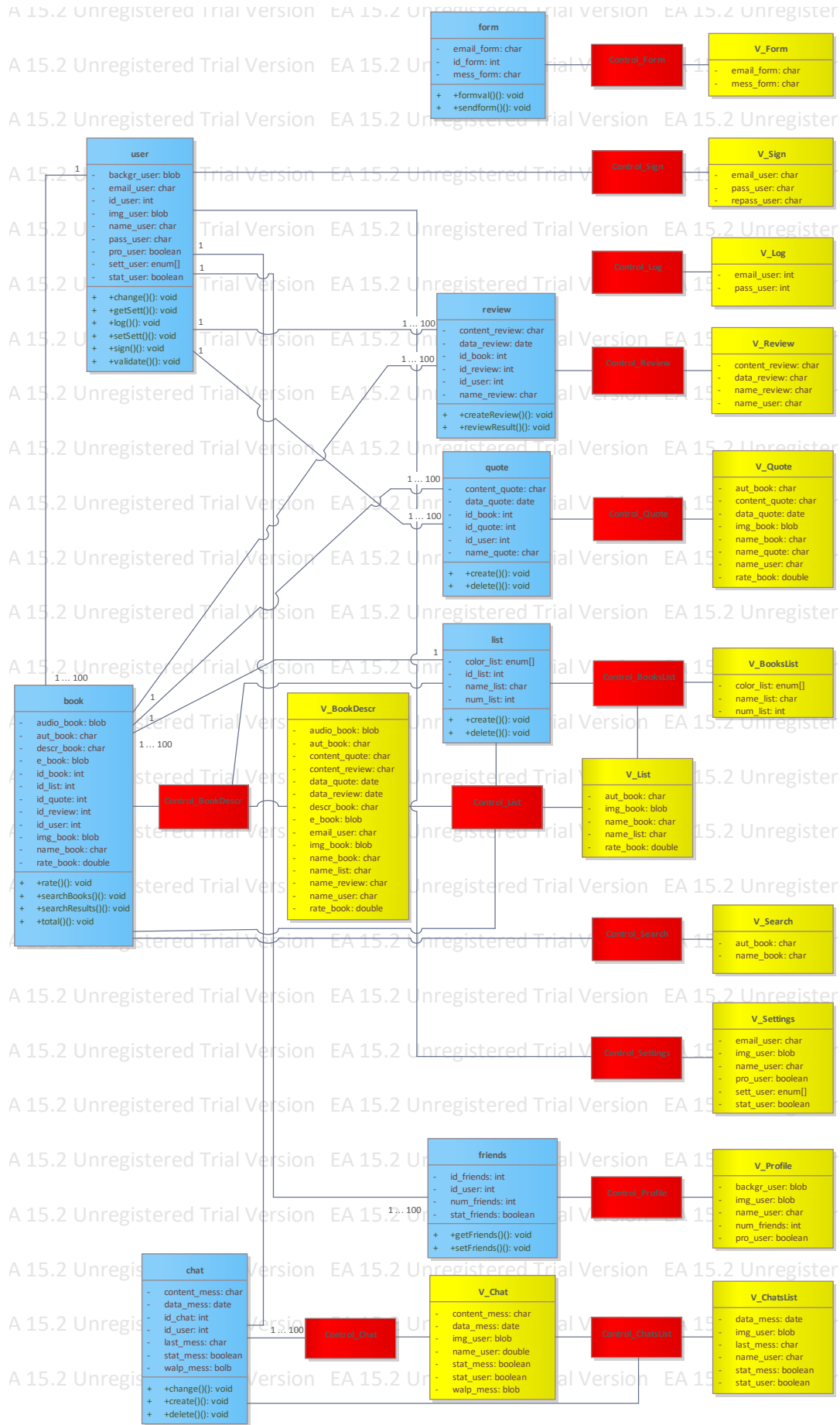


Рис. 2.3.2.1. – Діаграма класів

У якості структурної діаграми SysML було створено *діаграму визначення блоків*, яка показує внутрішню структуру системи у вигляді ієрархії.

ІС було умовно поділено на 4 складові: Frontend, Backend, Бізнес-логіку і Технічне забезпечення. *Frontend* має такі складові: інтерфейс користувача (відображають різні елементи інтерфейсу користувача, такі як кнопки, поля вводу, таблиці, списки тощо), CSS (зовнішній вигляд, стилі), маршрутизація (маршрутизацію користувача між різними сторінками або відображеннями вашого додатку), запити до сервера (взаємодія з сервером, відправляючи запити до API або інших віддалених сервісів для отримання або відправлення даних) та клієнтські бібліотеки та фреймворки (взаємодія клієнтської частини веб-додатків з користувачем та сервером). До блоку *Backend* відносяться: система кешування (кешування даних для покращення швидкодії веб-додатку та зменшення навантаження на сервер), система логування (збір та зберігання журналів подій та помилок, що відбуваються в системі для подальшого аналізу та відладки), серверна логіка (логіка, яка виконується на сервері та відповідає за обробку запитів від клієнтів, роботу з базою даних та виконання бізнес-логіки), API (надання програмного інтерфейсу для взаємодії з іншими системами або клієнтами через HTTP-запити), сервіс авторизації та автентифікації (компонент системи зовнішньої інтеграції, який пов'язаний із веб-сервером) та модуль виконання запитів (обробка запитів користувача). У блоку *Бізнес-логіки* такі складові: система Авторизації (компонент системи зовнішньої інтеграції, який пов'язаний із веб-сервером), Платіжна система (компонент платіжної системи, який пов'язаний із веб-сервером) та База Даних (система організації та зберігання структурованої інформації, яка дозволяє ефективно здійснювати пошук, зміну та керування даними). Останнім блоком є блок *Технічного забезпечення* який складається з Веб-серверу (містить веб-інтерфейс, Log-файл; інтерфейс для роботи з базою даних) та серверу Баз Даних (містить у собі компонент бази даних, який пов'язаний із веб-сервером).

На рис. 2.3.2.2. подано схему зі зв'язками на діаграмі визначення блоків:

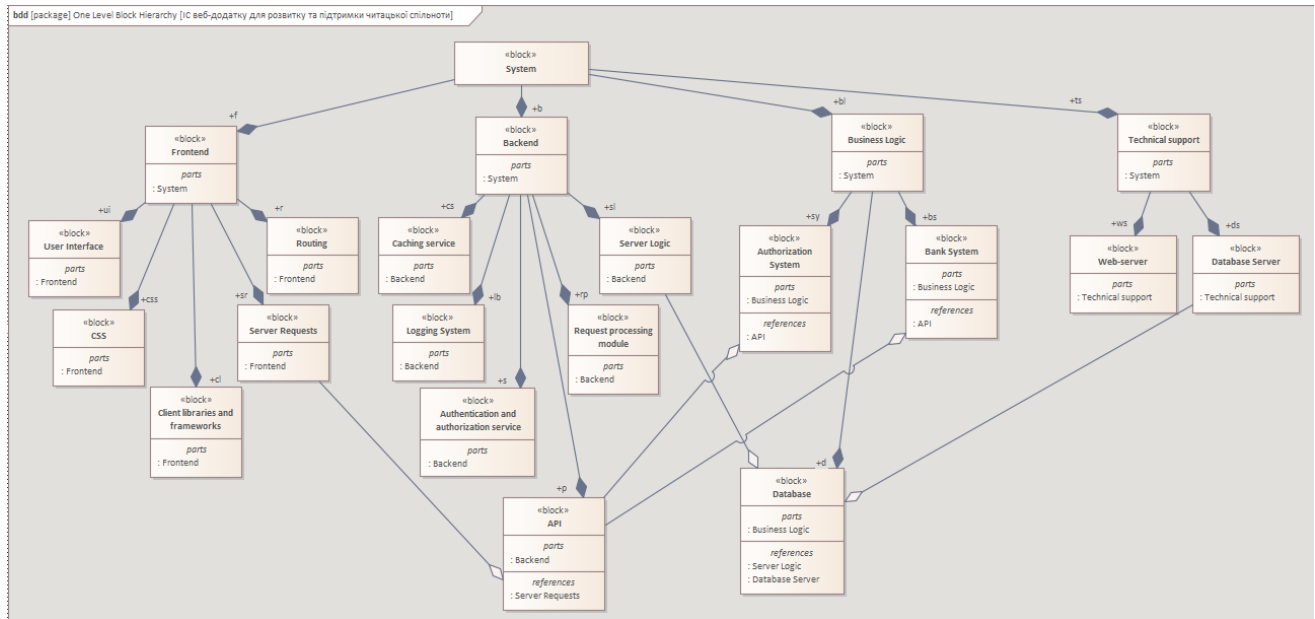


Рис. 2.3.2.2. – Діаграма визначення блоків

У якості діаграми *внутрішніх блоків*, які подаються як частини системи, було продемонстровано два фрагменти: оплата додаткового функціоналу (рис 2.3.2.3.) та авторизація користувача (рис. 2.3.2.4.). На діаграмах видно з'єднання між собою безпосередньо через порти і потоки, що існують між частинами і портами.

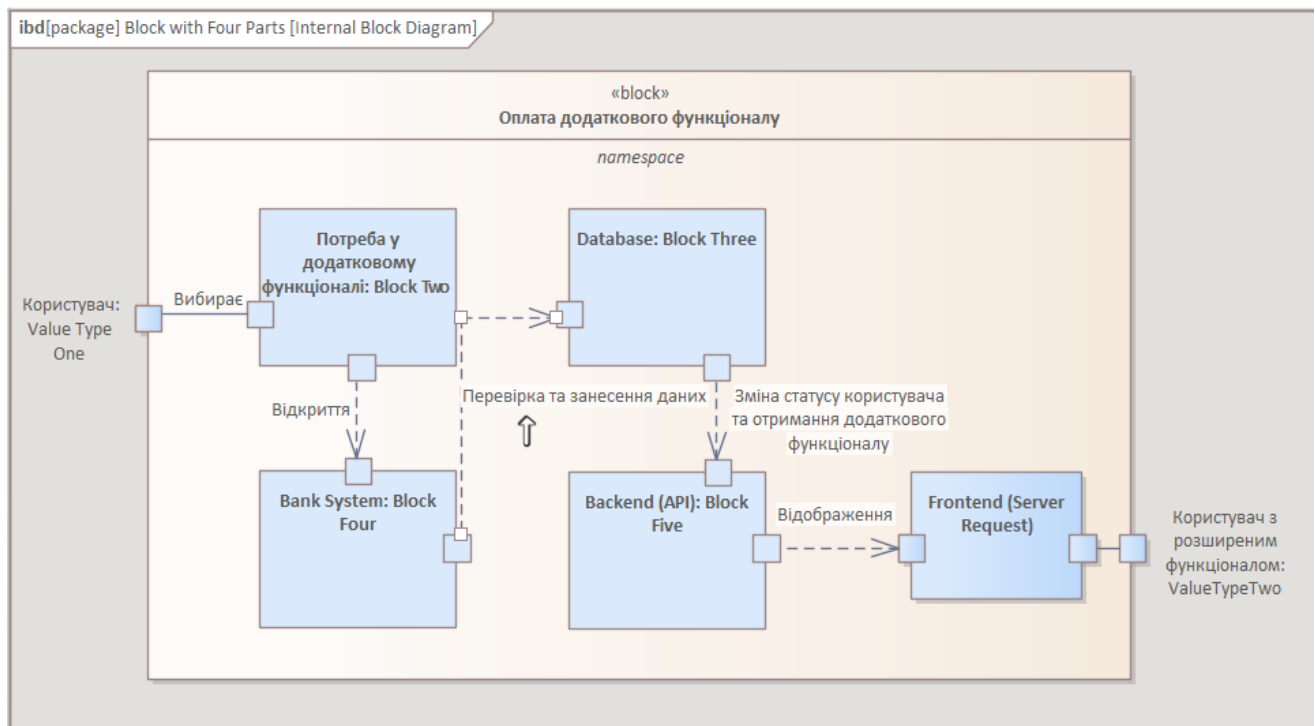


Рис. 2.3.2.3. – Діаграма внутрішніх блоків «Оплата додаткового функціоналу»

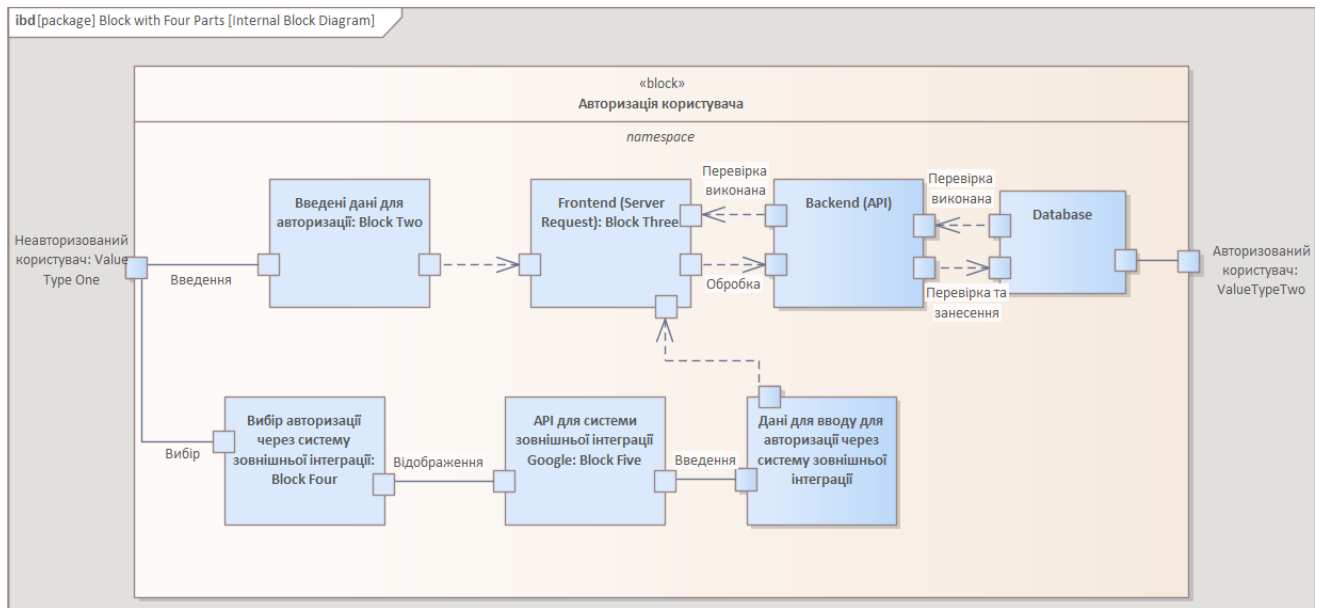


Рис. 2.3.2.4. – Діаграма внутрішніх блоків «Авторизація користувача»

2.3.3. Розподіл вимог за компонентами системи

Для проекту було спроектовано та розроблено тест-кейси для кожного прецедента. Нижче розписаний стислий опис тестування, а документація, створена засобами Enterprise Architect, наведена у Додатку Б.

Неавторизований користувач:

1. **Вхід у обліковий запис:** Користувач вводить свій валідний email та пароль, система перевіряє ці дані та видає повідомлення про успішний вхід, якщо дані правильні, або повідомлення про помилку вводу, якщо дані невірні.

2. **Реєстрація:** Користувач вводить унікальний email та пароль, система перевіряє ці дані на унікальність та коректність, після чого користувач отримує повідомлення про успішну реєстрацію.

3. **Відправлення форми зворотного зв'язку:** Користувач переходить на форму зворотного зв'язку, вводить свій валідний email та повідомлення, після чого натискає кнопку "Відправити". Після успішної відправки, користувач отримує повідомлення про успішне надіслання повідомлення.

Авторизований користувач:

1. **Відправлення форми зворотного зв'язку:** Користувач переходить на форму зворотного зв'язку, вводить свій валідний email та повідомлення, після чого натискає кнопку "Відправити". Після успішної відправки, користувач отримує повідомлення про успішне надіслання повідомлення.

2. **Пошук книги за назвою:** Користувач вводить назву книги в поле пошуку, система аналізує введені дані та відображає результати, відповідні введеній назві.

3. **Додавання книги до списку:** Користувач обирає книгу та додає її до свого списку бажань. Після успішного додавання, користувач отримує підтвердження про додавання книги до списку.

4. **Оцінка книги:** Користувач вибирає оцінку для книги, після чого система зберігає оцінку у базі даних.

5. **Створення списку книг:** Користувач створює персональний список книг, вводить назву та вибирає колір списку. Після успішного створення, користувач отримує повідомлення про успішне створення та збереження списку.

6. **Обрання формату читання книги:** Користувач відкриває сторінку обраної книги та обирає бажаний формат читання (Е-Book або Audio-Book). Після обрання, користувач переходить до інтерфейсу обраного формату для зручного читання або прослуховування книги.

7. **Оплата додаткового функціоналу:** Користувач переходить до розділу "Оплата додаткового функціоналу" у своєму особистому кабінеті, де коректно вводить дані своєї картки для оплати обраного додаткового функціоналу. Після успішної оплати, обрані додаткові можливості активуються для користувача.

8. **Написання цитати або рецензії до книги:** Користувач відкриває сторінку обраної книги, обирає розділ для написання цитати або рецензії та вводить текст. Після збереження, написане успішно зберігається та відображається на сторінці книги.

9. **Перегляд облікового запису та налаштувань:** Користувач переходить у свій особистий кабінет та відкриває розділ "Мій профіль" або "Налаштування", де може переглянути та змінити особисті дані та налаштування облікового запису.

10. **Додавання у друзі користувача:** Користувач переходить до профілю іншого користувача, обирає опцію "Додати до друзів" та додає користувача у свій список друзів.

11. **Спілкування у приватному чаті:** Користувач переходить до переліку чатів, обирає іншого користувача для спілкування, вводить текст повідомлення та надсилає його. Повідомлення успішно надіслано та відображено у приватному чаті.

12. **Зміна налаштувань приватного чату:** Користувач переходить до налаштувань приватного чату, де змінює фон чату, виконує опцію очищення чату або видаляє чат. Після виконання змін, вони успішно зберігаються та відображаються коректно на сторінці.

Верифікацію проектних рішень було проведено шляхом *трасування* вимог до системи та їх співвіднесення з проектними компонентами та іншими артефактами, що показано на рис. 2.3.3.1. На діаграмі присутні прецеденти, вимоги, елементи користувацького інтерфейсу. Між елементами діаграми трасування проведені відповідні зв'язки.

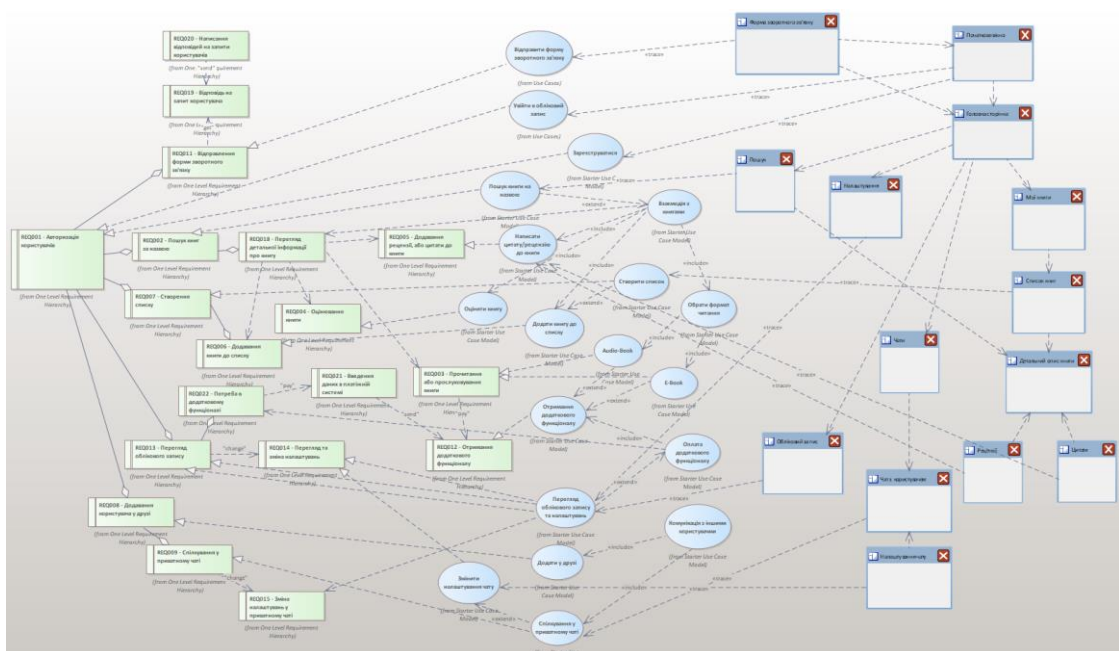


Рис. 2.3.3.1. – Діаграма трасування

2) Для доповнення діаграми трасування було побудовано *матрицю взаємозв'язків* (рис. 2.3.3.2.). Матриця показує зв'язки між прецедентами та вимогами системи. Так як матриця вийшла велика, то рисунок відображає тільки певну частину зв'язків.

Source \ Target	One Level Requirement Hierarchy: RFU001 - Внес	One Level Requirement Hierarchy: RFU002 - Блан	One Level Requirement Hierarchy: RFU004 - Магн	One Level Requirement Hierarchy: RFU005 - Назад	One Level Requirement Hierarchy: RFU006 - Бюджет	One Level Requirement Hierarchy: REQ001 - Авториз	One Level Requirement Hierarchy: REQ002 - Пошук	One Level Requirement Hierarchy: REQ003 - Прог	One Level Requirement Hierarchy: REQ004 - Опці	One Level Requirement Hierarchy: REQ005 - Додат	One Level Requirement Hierarchy: REQ006 - Додат	One Level Requirement Hierarchy: REQ007 - Створ	One Level Requirement Hierarchy: REQ008 - Додат	One Level Requirement Hierarchy: REQ009 - Чинит	One Level Requirement Hierarchy: REQ010 - Пере	One Level Requirement Hierarchy: REQ011 - Відтис	One Level Requirement Hierarchy: REQ012 - Отри	One Level Requirement Hierarchy: REQ013 - Пере	One Level Requirement Hierarchy: REQ014 - Пере	One Level Requirement Hierarchy: REQ015 - Змін	One Level Requirement Hierarchy: REQ016 - Забес	One Level Requirement Hierarchy: REQ017 - Бюджет	One Level Requirement Hierarchy: REQ018 - Пере	One Level Requirement Hierarchy: REQ019 - Відтис	One Level Requirement Hierarchy: REQ020 - Напи	One Level Requirement Hierarchy: REQ021 - Введ	One Level Requirement Hierarchy: REQ022 - Торг
Starter Use Case Model: E-Book																											
Starter Use Case Model: Взаємодія з книгами																											
Starter Use Case Model: Додати книгу до списку																											
Starter Use Case Model: Додати у друзі																											
Starter Use Case Model: Зареєструватися																											
Starter Use Case Model: Змінити налаштування чату																											
Starter Use Case Model: Комунікація з іншими користувачами																											
Starter Use Case Model: Написати цитату/рецензію до книги																											
Starter Use Case Model: Обрати формат читання																											
Starter Use Case Model: Оплата додаткового функціоналу																											
Starter Use Case Model: Отримання додаткового функціоналу																											
Starter Use Case Model: Оцінити книгу																											
Starter Use Case Model: Перегляд облікового запису та налаштувань																											
Starter Use Case Model: Пошук книги на назвою																											
Starter Use Case Model: Спілкування у приватному чаті																											
Starter Use Case Model: Створити список																											
Use Cases: Відправити форму зворотного зв'язку																											
Use Cases: Увійти в обліковий запис																											

Рис. 2.3.3.2. – Матриця трасування

У вигляді експерименту було побудовано ще декілька діаграм трасування (рис. 2.3.3.3 – 2.3.3.5.), але які демонструють лише певні процеси. На них розміщені актори, прецеденти, елементи користувацького інтерфейсу, вимоги та внутрішні блоки. Було побудовано матрицю взаємозв'язків (рис. 2.3.3.6.).

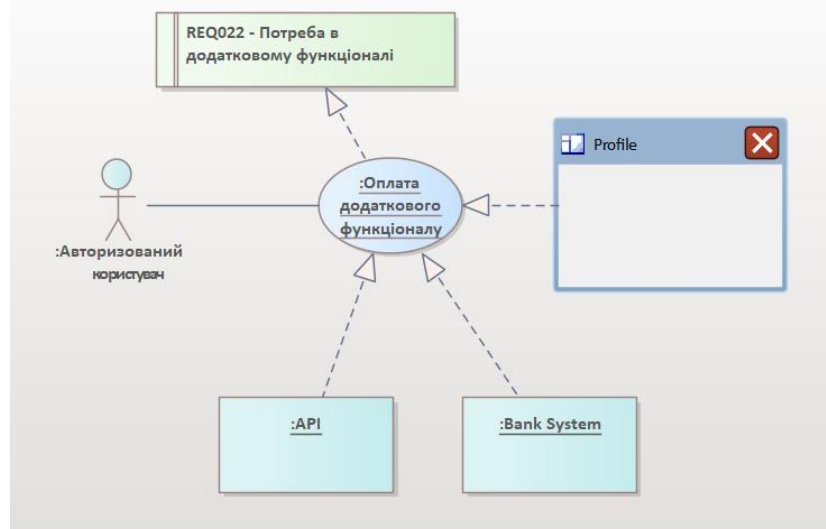


Рис. 2.3.3.3. – Діаграма трасування «Оплата додаткового функціоналу»

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ ПІДСИСТЕМИ

3.1. Інформаційне забезпечення

Було надано *загальну характеристику інформаційного забезпечення (ІЗ)*. В складі міститься інтерфейс користувача для введення та відображення інформації, база даних для зберігання інформації phpMyAdmin, API платіжної системи та авторизації за допомогою сервісу Google.

Структура інформаційного забезпечення поділяється на:

База даних:

- 1) Таблиця form (Форма зворотного зв'язку):
 - id_form: int – Ідентифікатор форми зворотного зв'язку;
 - email_form: varchar(20) – Електронна пошта користувача;
 - mess_form: varchar(200) – Повідомлення.
- 2) Таблиця review (Рецензії)
 - id_review: int – Ідентифікатор рецензій;
 - name_review: varchar(45) – Назва рецензії;
 - data_review: date – Дата публікації рецензії.
 - content_review: varchar(800) – Зміст рецензії;
 - id_user: int – Зв'язок з таблицею користувачів;
- 3) Таблиця quote (Цитати)
 - id_quote: int – Ідентифікатор цитат;
 - name_quote: varchar(45) – Назва цитати;
 - data_quote: date – Дата публікації цитати;
 - content_quote: varchar(100) – Зміст цитати;
 - id_user: int – Зв'язок з таблицею користувачів;
- 4) Таблиця list (Списки книг)
 - id_list: int – Ідентифікатор списків;

- name_list: varchar(30) – Назва списку;
- color_list: enum[] – Перелік кольорів для списку;
- num_list: int – Кількість об'єктів у списку.

5) Таблиця user (Користувачі)

- id_user: int – Ідентифікатор користувача;
- email_user: varchar(20) – Пошта користувача;
- pass_user: varchar(20) – Пароль користувача;
- name_user: varchar(25) – Ім'я користувача;
- img_user: blob – Фото користувача в обліковому записі;
- backgr_user: blob – Фото заднього плану в обліковому записі;
- stat_user: boolean – Статус користувача у мережі: онлайн/офлайн;
- sett_user: enum[] – Налаштування користувача;
- pro_user: boolean – У користувача є розширений функціонал, чи ні.

6) Таблиця book (Книги)

- id_book: int – Ідентифікатор книги.
- name_book: varchar(40) – Назва книги;
- aut_book: varchar(35) – Автор книги;
- img_book: blob – Обкладинка книги;
- rate_book: double – Оцінка книги;
- descr_book: varchar(1000) – Опис книги;
- e_book: blob – Електронний варіант книги;
- audio_book: blob – Аудіо-книга;
- id_review: int – Зв'язок з таблицею рецензій;
- id_quote: int – Зв'язок з таблицею цитат;
- id_user: int – Зв'язок з таблицею користувачів;
- id_list: int – Зв'язок з таблицею списків книг.

7) Таблиця friends (Друзі)

- id_friends: int – Ідентифікатор друзів;
- num_friends: int – Кількість друзів;
- stat_friends: boolean – Статус користувача у друзях чи ні;

- `id_user: int` – Зв’язок з таблицею користувачів

8) Таблиця chat (Чатів)

- `id_chat: int` – Ідентифікатор чатів;
- `content_mess: varchar(1000)` – Зміст повідомлення;
- `last_mess: varchar(30)` – Останнє повідомлення;
- `data_mess: date` – Дата відправки повідомлення;
- `stat_mess: boolean` – Статус чи відправилось повідомлення

користувачу;

- `walp_mess: blob` – Фон чату;
- `id_user: int` – Зв’язок з таблицею користувачів.

Інтерфейс користувача:

- Веб-сторінки: описова, головна, список книг, пошук книг, оновлення та питання користувачів, налаштування, облікового запису, цитат та рецензій, детальна інформація про книгу, чат з користувачами.
- Форми входу та реєстрації в обліковий запис, зворотного зв’язку, редагування облікового запису, додавання цитати/рецензії, пошуку книги.
- Елементи навігації: кнопки, модальні вікна, інтерактивні списки, форми.
- Візуальні елементи: зображення, іконки, текстові елементи, звукові елементи, візуалізація статистики оцінки книг.

Зовнішні інтеграції API:

- API платіжної системи: Для обробки платежів та покупок користувачів.
- API авторизації Google: Для спрощеної реєстрації та входу користувачів через облікові записи Google.

Серед головних принципів організації інформаційного забезпечення є:

- Доступність – забезпечення зручного та швидкого доступу до інформації для користувачів веб-застосунку.
- Безпека – дані користувачів захищаються від несанкціонованого доступу інших людей та зловживань.

- Масштабованість – розширення бази даних і функціоналу без значних змін у структурі веб-застосунку.
- Надійність – забезпечення стабільної роботи з мінімальним ризиком втрати даних.

Для зберігання даних використовується електронний тип носія даних – серверні жорсткі диски (SSD), який дозволяє зберігати великі обсяги інформації та забезпечує швидкий доступ до неї. Також можуть бути використані хмарні сховища для резервного копіювання даних.

Було *обрано phpMyAdmin* у якості веб-інтерфейсу для адміністрування СКБД та визначено такі переваги веб-додатку:

- Зручний та інтуїтивно зрозумілий інтерфейс – графічний інтерфейс, що дозволяє швидко і легко виконувати різноманітні завдання з адміністрування баз даних.
- Вільно поширюваний програмний продукт - програмним забезпеченням з відкритим вихідним кодом, яке розповсюджується під ліцензією GNU General Public License.
- Безкоштовний сервіс – є безкоштовним для завантаження і використання. Відсутність ліцензійних витрат дозволяє знизити загальні витрати на розробку та підтримку системи, що є особливо важливим для невеликих компаній та проектів з обмеженим бюджетом.
- Масштабованість – підтримує роботу з великими обсягами даних і великою кількістю користувачів.
- Безпечність – забезпечує високий рівень безпеки завдяки використанню найкращих практик захисту даних.

Основними характеристиками phpMyAdmin є: створення нових баз даних та таблиць; змінення структури існуючих таблиць, додаючи, змінюючи чи видаляючи стовпці; виконання SQL запити для видалення, оновлення, вставки або видалення даних; імпортування даних із зовнішніх джерел, таких як файли CSV або SQL; експортування даних до різних форматів, таких як SQL, CSV або XML; керування користувачами та привілеями доступу до баз даних; відслідковування за станом та параметрами сервера MySQL.

Було виділено такі методи контролю інформації:

- Автентифікація користувачів: перевірка облікових даних користувачів для доступу до системи.
- Логування подій: запис дій користувачів і системних подій для моніторингу і аналізу.
- Валідація даних: перевірка введених даних на коректність та відповідність вимогам.

З вимог до надійності і достовірності інформації можна відзначити такі:

- Надійність: система має забезпечувати безперервну роботу з мінімальними перебоями.
- Достовірність: система має забезпечити точність та актуальність даних, що зберігаються і обробляються.

На рис. 3.1.1. продемонстровано узагальнену схему інформаційного забезпечення веб-застосунку:

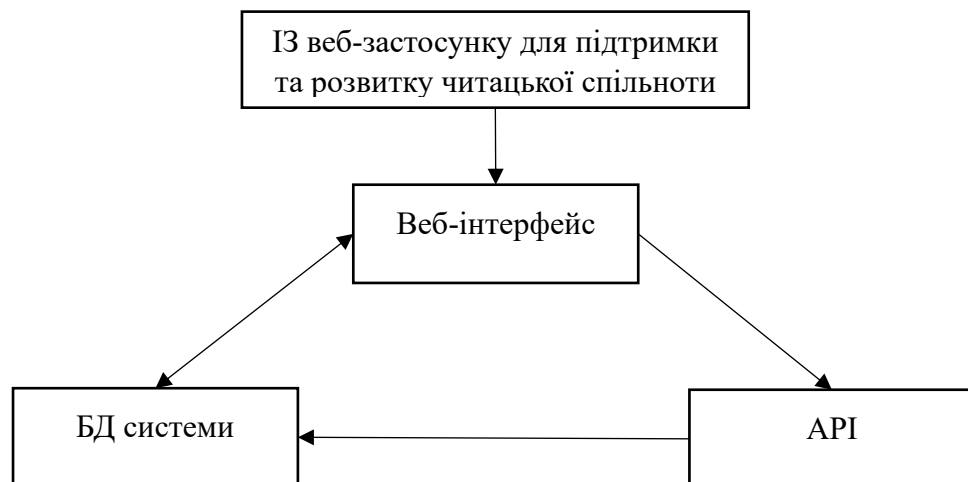


Рис. 3.1.1. – Загальна схема ІЗ

Описано *організацію збору і передавання первинної інформації* такими критеріями: перелік джерел і носіїв інформації та опис загальних вимог.

Перелік джерел і носіїв інформації:

1. Внутрішні підрозділи організації:

Адміністрація системи – надає дані про користувачів, налаштування ІС, нові дописи. Оновлення даних відбувається постійно та в реальному часі. Дані вводяться через інтерфейс. Носіями слугують сервер бази даних.

2. Зовнішні підрозділи організації:

До таких можна віднести видавництва, бібліотеки, які містять дані про нові книжки, авторів. Дані оновлюються по мірі надходження нової інформації та надаються через електронну пошту. Носіями таких даних є файли формату XML, JSON, електронні листи.

3. Інтернет:

Використовуються інформаційні ресурси веб-сайтів видавництв, новин літературного світу. Оновлюється така інформація постійно та в реальному часі. Дані отримуються шляхом веб-скрапінгу, а носіями є файли формату JSON та HTML.

4. База даних:

Внутрішня база даних містить в собі інформацію про книги, користувачів, форми зворотного зв'язку, рецензії та цитати, списки книг, чати та друзів користувачів. Оновлення відбувається постійно та в реальному часі. Інформація надається через введення даних через веб-інтерфейс, а носієм є сервер бази даних.

Опис загальних вимог:

1. Точність і своєчасність даних: вчасний та точний збір даних з усіх потрібних джерел.

2. Безперервність збору даних: постійний збір та оновлення інформації, задля отримання тільки актуальної інформації користувачами.

3. Безпека: захист даних від втрат.

4. Контроль якості даних: використання методів контролю якості даних для забезпечення їх надійності і достовірності.

5. Масштабованість: гнучкість системи, яка може обробляти великий обсяг даних та підтримувати нові джерела інформації.

Для забезпечення ефективного функціонування інформаційної системи веб-застосунку для підтримки та розвитку читацької спільноти, була

розроблена система класифікації об'єктів. Основні об'єкти класифікації включають: форми зворотного зв'язку, рецензії, цитати, списки книг, користувачів, книги, друзів та чати.

1. *Таблиця form (Форма зворотного зв'язку)*
 - id_form: Ідентифікатор форми зворотного зв'язку
 - Метод кодування: Автоматичне збільшення (auto-increment)
 - Структура і довжина: INT, довжина до 11 цифр
2. *Таблиця review (Рецензії)*
 - id_review: Ідентифікатор рецензій
 - Метод кодування: Автоматичне збільшення (auto-increment)
 - Структура і довжина: INT, довжина до 11 цифр
3. *Таблиця quote (Цитати)*
 - id_quote: Ідентифікатор цитат
 - Метод кодування: Автоматичне збільшення (auto-increment)
 - Структура і довжина: INT, довжина до 11 цифр
4. *Таблиця list (Списки книг)*
 - id_list: Ідентифікатор списків
 - Метод кодування: Автоматичне збільшення (auto-increment)
 - Структура і довжина: INT, довжина до 11 цифр
5. *Таблиця user (Користувачі)*
 - id_user: Ідентифікатор користувача
 - Метод кодування: Автоматичне збільшення (auto-increment)
 - Структура і довжина: INT, довжина до 11 цифр
6. *Таблиця book (Книги)*
 - id_book: Ідентифікатор книги
 - Метод кодування: Автоматичне збільшення (auto-increment)
 - Структура і довжина: INT, довжина до 11 цифр
7. *Таблиця friends (Друзі)*
 - id_friends: Ідентифікатор друзів
 - Метод кодування: Автоматичне збільшення (auto-increment)
 - Структура і довжина: INT, довжина до 11 цифр

8. Таблиця chat (Чати)

- id_chat: Ідентифікатор чатів
- Метод кодування: Автоматичне збільшення (auto-increment)
- Структура і довжина: INT, довжина до 11 цифр

Методи кодування об'єктів класифікації:

- Автоматичне збільшення (auto-increment): використовується для ідентифікаторів усіх таблиць. Це дозволяє автоматично генерувати унікальні значення для кожного нового запису.

- Структура і довжина: для всіх ідентифікаторів використовується тип даних INT з максимальною довжиною до 11 цифр. Це забезпечує можливість зберігання великих обсягів даних.

Нижче подано *структуру інформаційних масивів*. По кожному масиву наведено таблицю (табл. 3.1.1 – 3.1.8.) з описом структури масиву.

Таблиця 3.1.1. – Масив форми зворотного зв'язку

Опис масиву form (Форма зворотного зв'язку)

1. Найменування масиву: Форма зворотного зв'язку.
2. Ідентифікатор масиву: form.
3. Найменування носія інформації: сервер phpMyAdmin.
4. Максимальний об'єм масиву: без обмежень.
5. Довжина запису: 230 байтів.
6. Метод організації записів: послідовний.
7. Ключі упорядкування: id_form (первинний ключ).

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9

Продовження таблиці 3.1.1.

1	2	3	4	5	6	7	8	9
Ідентифікатор форми зворотного зв'язку	id_form	If	int(11)	PK	-	Так	-	-
Електронна пошта користувача	email_form	Ef	varchar(20)	-	-	Так	ІНД	-
Повідомлення	mess_form	Mf	varchar(200)	-	-	Так	-	-

Таблиця 3.1.2. – Масив рецензій

Масив review (Рецензії)

1. Найменування масиву: Рецензії.
2. Ідентифікатор масиву: review.
3. Найменування носія інформації: сервер phpMyAdmin.
4. Максимальний об'єм масиву: без обмежень.
5. Довжина запису: 900 байтів.
6. Метод організації записів: послідовний.
7. Ключі упорядкування: id_review (первинний ключ).

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні чи семантичні
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9
Ідентифікатор рецензій	id_review	Ir	int(11)	PK	-	Так	-	-
Назва рецензії	name_review	Nr	varchar(45)	-	-	Так	ІНД	-
Дата публікації рецензії	data_review	Dr	date	-	-	Так	-	-
Зміст рецензії	content_review	Cr	varchar(800)	-	-	Так	-	-
Зв'язок з таблицею користувачів	id_user	Iu	int(11)	FK	-	Так	-	user

Таблиця 3.1.3. – Масив цитат

Масив quote (Цитати)

1. Найменування масиву: Цитати.
2. Ідентифікатор масиву: quote.
3. Найменування носія інформації: сервер phpMyAdmin.
4. Максимальний об'єм масиву: без обмежень.
5. Довжина запису: 200 байтів.
6. Метод організації записів: послідовний.
7. Ключі упорядкування: id_quote (первинний ключ).

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9
Ідентифікатор цитат	id_quote	Iz	int(11)	PK	-	Так	-	-
Назва цитати	name_quote	Nz	varchar(45)	-	-	Так	ІНД	-
Дата публікації цитати	data_quote	Dz	date	-	-	Так	-	-
Зміст цитати	content_quote	Zz	varchar(100)	-	-	Так	-	-
Зв'язок з таблицею користувачів	id_user	Iu	int(11)	FK	-	Так	-	user

Таблиця 3.1.4. – Масив списків книг

Масив list (Списки книг)

1. Найменування масиву: Списки книг.
2. Ідентифікатор масиву: list.
3. Найменування носія інформації: сервер phpMyAdmin.

4. Максимальний об'єм масиву: без обмежень.
5. Довжина запису: 80 байтів.
6. Метод організації записів: послідовний.
7. Ключі упорядкування: id_list (первинний ключ).

Найменування	Ідентифікатор у програмі	Умовне позначення у формулах	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9
Ідентифікатор списків	id_list	И	int(11)	PK	-	Так	-	-
Назва списку	name_list	N1	varchar(30)	-	-	Так	ІНД	-
Перелік кольорів для списку	color_list	P1	enum[]	-	-	Так	-	-
Кількість об'єктів у списку	num_list	K1	int(11)	-	-	Так	-	-

Таблиця 3.1.5. – Масив користувачів

Масив user (Користувачі)

1. Найменування масиву: Користувачі.
2. Ідентифікатор масиву: user.
3. Найменування носія інформації: сервер phpMyAdmin.
4. Максимальний об'єм масиву: без обмежень.
5. Довжина запису: 100 байтів.
6. Метод організації записів: послідовний.
7. Ключі упорядкування: id_user (первинний ключ).

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9
Ідентифікатор користувача	id_user	Iu	int(11)	PK	-	Так	-	-
Пошта користувача	email_user	Eu	varchar(20)	-	-	Так	ІНД	-
Пароль користувача	pass_user	Pau	varchar(20)	-	-	Так	-	-
Ім'я користувача	name_user	Kl	varchar(20)	-	-	Ні	ІНД	-
Фото користувача в обліковому записі	img_user	Imu	blob	-	-	Ні	-	-
Фото заднього плану і обліковому записі	backgr_user	Bu	blob	-	-	Ні	-	-
Статус користувача у мережі: онлайн/офлайн	stat_user	Stu	boolean	-	-	Так	-	-
Налаштування користувача	sett_user	Seu	enum[]	-	-	Так	-	-
У користувача є розширений функціонал, чи ні	pro_user	Pru	boolean	-	-	Так	-	-

Таблиця 3.1.6. – Масив книг

Масив book (Книги)

1. Найменування масиву: Книги.
2. Ідентифікатор масиву: book.
3. Найменування носія інформації: сервер phpMyAdmin
4. Максимальний об'єм масиву: без обмежень.

5. Довжина запису: 1500 байтів.
6. Метод організації записів: послідовний.
7. Ключі упорядкування: id_book (первинний ключ).

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9
Ідентифікатор книги	id_book	Ib	int(11)	PK	-	Так	-	-
Назва книги	name_book	Nb	varchar(40)	-	-	Так	ІНД	-
Автор книги	aut_book	Ab	varchar(35)	-	-	Так	-	-
Обкладинка книги	img_book	Ib	blob	-	-	Так	-	-
Оцінка книги	rate_book	Rb	double	-	-	Hi	-	-
Опис книги	descr_book	Db	varchar(1000)	-	-	Hi	-	-
Електронний варіант книги	e_book	Eb	blob	-	-	Так	-	-
Аудіо-книга	audio_book	Aub	blob	-	-	Hi	-	-
Зв'язок з таблицею рецензій	id_review	Ir	int(11)	FK	-	Так	-	review
Зв'язок з таблицею цитат	id_quote	Iq	int(11)	FK	-	Так	-	quote
Зв'язок з таблицею користувачів	id_user	Id	int(11)	FK	-	Так	-	user
Зв'язок з таблицею списків книг	id_list	Il	int(11)	FK	-	Так	-	list

Таблиця 3.1.7. – Масив друзів

Масив friends (Друзі)

1. Найменування масиву: Друзі.
2. Ідентифікатор масиву: friends.
3. Найменування носія інформації: сервер phpMyAdmin.
4. Максимальний об'єм масиву: без обмежень.
5. Довжина запису: 20 байтів.
6. Метод організації записів: послідовний.
7. Ключі упорядкування: id_friends (первинний ключ).

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9
Ідентифікатор друзів	id_friends	If	int(11)	PK	-	Так	-	-
Кількість друзів	num_friends	Nf	int(11)	-	-	Так	-	-
Статус користувача у друзях чи ні	stat_friends	Sf	boolean	-	-	Так	-	-
Зв'язок з таблицею користувачів	id_user	Iu	int(11)	FK	-	Так	-	user

Таблиця 3.1.8. – Масив чатів

Масив chat (Чати)

1. Найменування масиву: Чати.
2. Ідентифікатор масиву: chat.
3. Найменування носія інформації: сервер phpMyAdmin.
4. Максимальний об'єм масиву: без обмежень.
5. Довжина запису: 1,100 байтів.
6. Метод організації записів: послідовний.

7. Ключі упорядкування: id_chat (первинний ключ).

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9
Ідентифікатор чатів	id_chat	Ic	int(11)	PK	-	Так	-	-
Зміст повідомлення	content_mess	Cm	varchar(1000)	-	-	Так	-	-
Останнє повідомлення	last_mess	Lm	varchar(30)	-	-	Ні	-	-
Дата відправки повідомлення	data_mess	Dm	date	-	-	Так	-	-
Статус чи відправилось повідомлення користувачу	stat_mess	Sm	boolean	-	-	Так	-	-
Фон чату	walp_mess	Wm	blob	-	-	Ні	-	-
Зв'язок з таблицею користувачів	id_user	Iu	int(11)	FK	-	Так	-	user

Було представлено *інфологічну модель бази даних* для веб-застосунку для підтримки та розвитку читацької спільноти на рис. 3.1.2. БД нормалізована відповідно до стандартних нормалізаційних форм. Інфологічна модель включає основні сутності, їх атрибути та взаємозв'язки. Основними сутностями є Користувачі, Книги, Рецензії, Цитати, Списки книг, Друзі, Чати та Форми зворотного зв'язку.

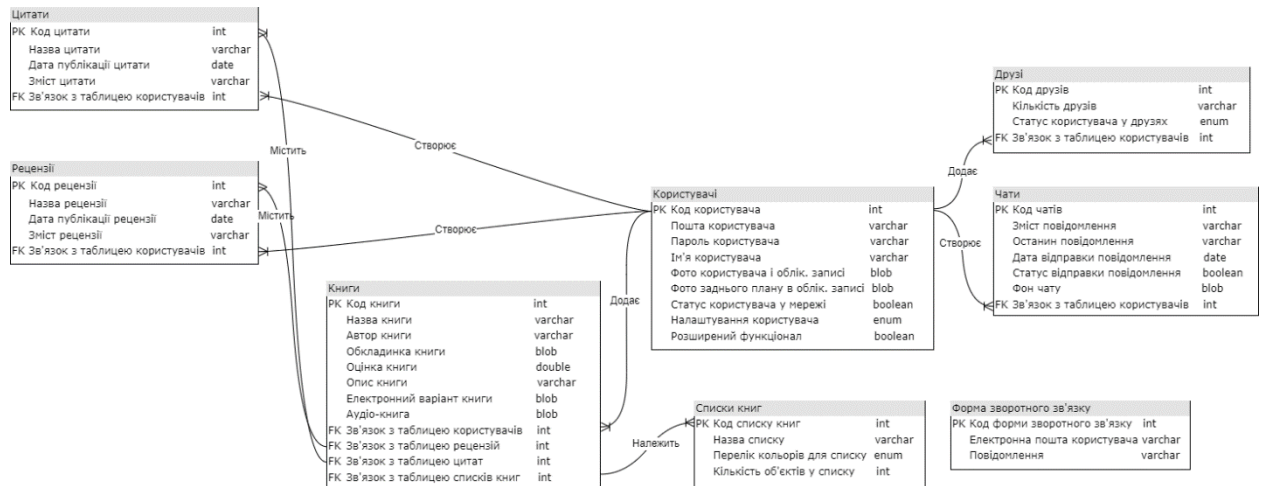


Рис. 3.1.2. – Інфологічна модель бази даних

Також було побудовано **даталогічну модель бази даних**, створену в phpMyAdmin, веб-застосунку для підтримки та розвитку читацької спільноти. Модель включає основні сутності, їх атрибути та взаємозв'язки (рис. 3.1.3.), що забезпечує цілісність та ефективність обробки даних. У результаті, добре спроектована даталогічна модель бази даних не лише підтримує поточні вимоги застосунку, але й готова до майбутніх змін та доповнень.

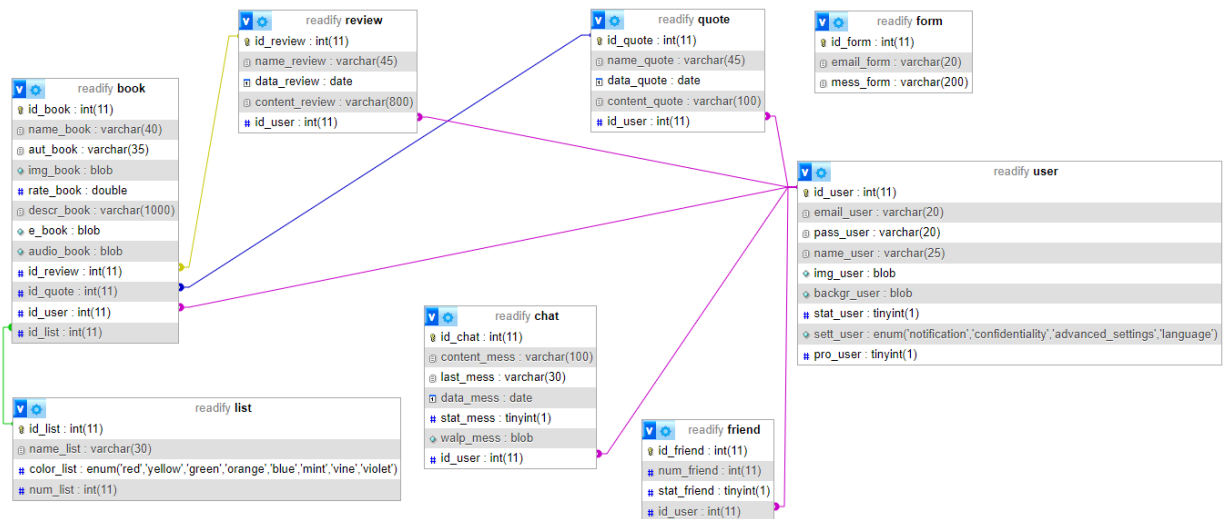


Рис. 3.1.3. – Даталогічна модель бази даних

3.2. Технічне забезпечення

Для забезпечення ефективного функціонування веб-застосунку, важливо ретельно обґрунтувати та організувати технічне забезпечення системи. Технічне забезпечення включає комплекс технічних засобів (КТЗ), який складається з серверного обладнання, робочих станцій користувачів, мережевої інфраструктури та периферійних пристроїв. Основною метою автоматизації є забезпечення стабільної та безперебійної роботи веб-застосунку, висока продуктивність та безпека даних.

Розписано *загальні положення таким чином*: інформаційна система для підтримки та розвитку читацької спільноти буде функціонувати на віддаленому сервері, розміщеному в захищеному дата-центрі, або використовувати орендовані потужності хмарного провайдера. Сервер буде містити ядро системи, базу даних та всі необхідні файли для забезпечення роботи веб-застосунку. Доступ користувачів до системи буде здійснюватись через використання веб-сайту з доступом до мережі Інтернет.

Загальна схема автоматизації подана на рис. 3.2.1. У схемі показано зв'язок між різними групами технічних засобів, які об'єднані певними логічними ознаками. Користувачі під'єднуються до веб-сервера через Інтернет-адаптер на Front-office. В свою чергу веб-сервер приймає HTTP-запити від користувачів і поєднує сервер бази даних (адміністрованим через phpMyAdmin) для доступу до даних користувачів, книг, рецензій та цитат, тощо, файловий сервер. Крім цього, сервер додатків інтегрується з API зовнішніх сервісів, таких як платіжна система та сервіс авторизації Google, для забезпечення додаткових функцій. На Back-office знаходяться керівництво, розробники, маркетологи та технічна підтримка, які забезпечують безперебійну роботу сервісу. Розробники відповідають за створення та підтримку системи, маркетологи використовують систему для аналізу та просування, а технічна підтримка обробляє запити та відгуки користувачів.

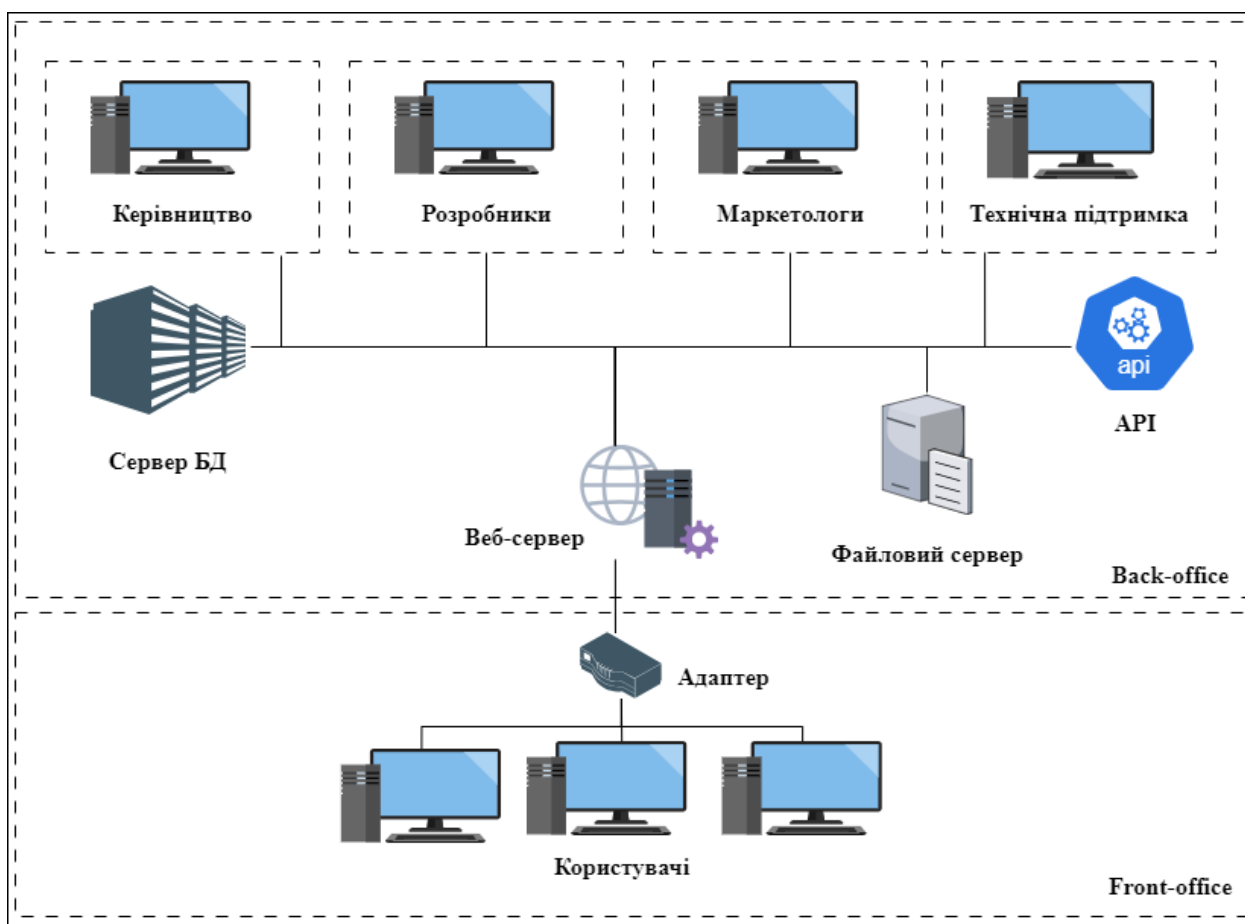


Рис. 3.2.1. – Загальна схема автоматизації

Описано **структуру комплексу технічних засобів**. Мінімальною допустимою кількістю автоматизованих робочих місць (АРМ) для кожного відділу є 1 місце. Для забезпечення ефективної роботи інформаційної системи підтримки та розвитку читачкої спільноти було обрано комплекс технічних засобів, що включає використання серверів та обладнання хостинг-провайдера HOSTiQ. Основними критеріями вибору були висока надійність, масштабованість, безпека та можливість інтеграції з іншими автоматизованими системами. HOSTiQ є провідним хостинг-провайдером, що пропонує послуги розміщення серверів у декількох датацентрах по всьому світу: Нідерланди, Роттердам; США, штат Джорджія, Атланта; Франція, Страсбург та Україна, Львів.

При виборі датацентру «Кінескоп» в Україні наведено такі характеристики хостинг провайдера: акумулятори для забезпечення безперебійного живлення, 2 окремі вводи електрики та канали з резервними шляхами на більшість провайдерів України з пропускнуою здатністю 1.5 Gbps

та 4 Gbps. Надається цілодобова технічна підтримка та прямий доступ до мережевого та серверного обладнання. При цьому електрична щільність на стійку становить близько 3 кВт.

Для забезпечення безпеки технічних засобів використовується комплексний підхід, який охоплює різноманітні аспекти захисту:

- Фізичний захист забезпечується шляхом розміщення обладнання у спеціально обладнаних приміщеннях, де всі параметри температури і вологості контролюються. А щоб обмежити доступ до обладнання використовуються спеціальні ключ-картки.
- Регулярні резервні копіювання зберігаються на віддалених серверах, для забезпечення безпеки даних.

При виборі типу персонального комп'ютера (ПК) було досліджено нефункціональні вимоги системи, потенційні навантаження та вартість транзакцій. Ці фактори впливають на ефективність, надійність та вартість проекту. Сервери та ПК мають бути з потужними процесорами та великою кількістю оперативної пам'яті для швидкої обробки великої кількості даних. Було оцінено вартість володіння та експлуатації, де враховано не тільки собівартість обладнання, а й витрати на підтримку, енергоспоживання та оновлення. В результаті, обрано сервер та ПК з високою продуктивністю, надійністю та масштабованістю, які відповідають вимогам ІС.

Технічні засоби включають в себе різноманітні компоненти, що забезпечують функціонування та взаємодію системи: датчики контролю температури та вологості та освітлення, різноманітні комунікаційні пристрої, апаратне забезпечення для зберігання та обробки даних, апаратні засоби для шифрування та захисту інформації.

При виборі периферійних технічних засобів для отримання, контролю, підготовки, збору, реєстрації, зберігання і відображення інформації важливо враховувати потреби та вимоги системи. Для збереження великого обсягу даних, що збираються з різних датчиків та пристроїв, можуть бути використані системи зберігання даних, такі як сервери або хмарні сховища. Для отримання і збору різних типів даних можуть бути використані датчики, такі як

температурні, вологості, руху тощо. Щоб відстежувати події та безпеку приміщень можуть бути встановлені камери відеоспостереження. Камери мають бути з високою роздільною здатністю та можливістю нічного бачення. Кількість персоналу буде залежати від обсягу робіт та специфіки системи, проте загалом можна вказати, що для ефективного функціонування ІС може знадобитися команда з кількох спеціалістів, здатних виконувати різноманітні завдання з підтримки, налагодження та безпеки.

Автоматизоване робоче місце (АРМ) у системі має бути оптимізоване для ефективної роботи з інформацією та забезпечення зручного доступу до неї. Обґрунтування вибору технічних засобів та периферійних пристроїв АРМ відповідає вимогам ІС і мінімальним системним вимогам для коректної роботи.

У виборі серед технічних засобів було обрано такі характеристики: процесор – маркування процесору Intel Core i5 або еквівалент; оперативна пам'ять – мінімум 8 ГБ RAM для оптимальної продуктивності; постійна пам'ять на жорсткому диску - обсяг не менше 256 ГБ SSD для швидкого доступу до даних та запобігання затримкам у роботі.

Серед периферійних технічних засобів було обрано: монітори - IPS-панель роздільною здатністю не менше Full HD для чіткого відображення інформації та забезпечення комфортної роботи; клавіатура та миші – безшумні та ергономічні; веб-камера та гарнітура – для зв'язку з колегами.

Кількість технічних засобів АРМ визначається кількістю працівників, які потребують доступ до системи. Наприклад, якщо є 4 працівника, то буде необхідно 4 комплекта АРМ. Підключення кожного з АРМ до мережі здійснюється за допомогою Ethernet для стабільного та надійного з'єднання. Для кожного робочого місця обрано комплект технічних засобів, що відповідає мінімальним вимогам ІС та забезпечує зручну та ефективну роботу.

3.3. Програмне забезпечення

Описано *структуру програмного забезпечення (ПЗ)*:

У якості базового ПЗ використано операційну систему (ОС) Windows 10, яке керує апаратним обладнанням комп'ютера та забезпечує основне середовище для виконання програм. Системне ПЗ включає в себе драйвери (системні програми, що взаємодіють з апаратним обладнанням і дозволяють йому працювати на операційній системі) та системні утиліти (програми для управління та налаштування ОС).

Прикладне, або спеціалізоване ПЗ вимагає створення *текстових сценаріїв* для кожного варіанта використання – щонайменше один, "нормальний", а за необхідності – ще один або кілька альтернативних, що описують відхилення від нього. Для підготовки сценаріїв було використано розробник сценаріїв (Scenario Builder). Нижче подано текстові сценарії і фрагмент (рис. 3.3.1.) із середовища Enterprise Architect.

1. Увійти в обліковий запис.

Нормальний сценарій:

1. Користувач відкриває веб-застосунок.
2. Система відображає сторінку входу в обліковий запис.
3. Користувач вводить свій логін та пароль.
4. Система перевіряє введені дані.
5. Якщо вони вірні, система авторизує користувача та перенаправляє його на особистий кабінет.

6. Якщо вони невірні, система виводить повідомлення про помилку.

Альтернативний сценарій (невірні дані):

1. Система відображає повідомлення про помилку та просить користувача перевірити введені дані.

2. Користувач повторно вводить логін та пароль.

Альтернативний сценарій (невірний пароль):

1. Після введення пароля користувач помиляється.

2. Система виводить повідомлення про помилку.
3. Користувач може повторно ввести пароль або скористатися функцією відновлення паролю.

2. Реєстрація.

Нормальний сценарій:

1. Користувач відкриває веб-застосунок.
2. Система відображає сторінку реєстрації.
3. Користувач вводить необхідну інформацію (логін, пароль, електронну адресу тощо).
4. Система перевіряє унікальність логіну та електронної адреси.
5. Якщо дані унікальні, система створює новий обліковий запис.
6. Система автоматично авторизує користувача та перенаправляє його на особистий кабінет.

Альтернативний сценарій (конфлікт із вже існуючим обліковим записом):

1. Система виявляє, що введений логін або електронна адреса вже використовуються.
2. Система виводить повідомлення про необхідність вибору іншого логіну або електронної адреси.
3. Користувач вводить нові дані та повторює спробу реєстрації.

3. Відправка форми зворотного зв'язку.

Нормальний сценарій:

1. Система відображає сторінку для відправки форми зворотного зв'язку.
2. Користувач вводить своє повідомлення у відповідні поля форми.
3. Система отримує введені дані та надсилає їх технічній підтримці.
4. Технічна підтримка обробляє отримане повідомлення та надає відповідь користувачеві.

Альтернативний сценарій (відміна відправки форми):

1. Користувач вирішує не відправляти форму зворотного зв'язку.
2. Користувач закриває сторінку без відправки повідомлення.

Альтернативний сценарій (неповне заповнення полів):

1. Користувач вводить часткові дані у відповідні поля форми.
2. Користувач намагається надіслати форму зворотного зв'язку.
3. Система перевіряє, чи всі обов'язкові поля для введення даних заповнені.
4. Система виводить повідомлення про помилку та наголошує, що всі поля повинні бути заповнені.

4. Пошук книг за назвою:

Нормальний сценарій:

1. Користувач вводить назву книги у поле пошуку.
2. Система виконує пошук за назвою книги в базі даних.
3. Знайдені результати відображаються на сторінці.
4. Користувач обирає потрібну книгу зі списку результатів.

Альтернативний сценарій (неможливий пошук):

1. Користувач вводить назву книги у поле пошуку.
2. Система не знаходить жодної книги, що відповідає введеному запиту.
3. Система виводить повідомлення про відсутність результатів для введеного запиту.

5. Взаємодія з книгами:

Нормальний сценарій:

1. Користувач вибирає книгу для взаємодії.
2. Система відображає сторінку книги з доступними опціями: рецензія, цитата, оцінка, вибір формату читання.
3. Користувач вибирає необхідну опцію та здійснює відповідну дію.

Альтернативний сценарій (неможливість взаємодії):

1. Користувач намагається вибрати книгу для взаємодії.
2. Система повідомляє про неможливість взаємодії з книгою наразі (наприклад, через технічні проблеми або відсутність доступу до книги).

6. Додавання книги до списку:

Нормальний сценарій:

1. Користувач обирає книгу, яку бажає додати до свого списку.
2. Система надає опцію "Додати до списку" на сторінці книги.
3. Користувач підтверджує додавання книги до свого списку.

Альтернативний сценарій (неуспішне додавання до списку):

1. Користувач обирає книгу, яку бажає додати до свого списку.
2. Система повідомляє про неможливість додавання книги до списку

(наприклад, через технічні проблеми або обмеження системи).

7. Комунікація з іншими користувачами (спілкування в чаті):

Нормальний сценарій:

1. Користувач вибирає опцію «Чат» у веб-застосунку.
2. Система відкриває список доступних чатів або дозволяє створити новий чат.
3. Користувач обирає чат та взаємодіє з іншими користувачами.

Альтернативний сценарій (помилка з'єднання):

1. Користувач спробував відкрити чат, але система не може підключитися до сервера.
2. Система повідомляє про технічні проблеми і рекомендує спробувати пізніше або звернутися до підтримки.

8. Оплата додаткового функціоналу:

Нормальний сценарій:

1. Користувач обирає опцію оплати додаткового функціоналу.
2. Система надає форму для введення даних картки користувача та інформацію про доступний функціонал.
3. Користувач вводить дані картки та підтверджує оплату.
4. Після успішної оплати система надає доступ до додаткового функціоналу.

Альтернативний сценарій (неуспішна оплата):

1. Користувач вводить некоректні дані картки та підтверджує оплату.
2. Система повідомляє про помилку під час оплати та рекомендує перевірити введені дані картки або спробувати іншу картку.

9. Написання рецензії/цитати до книги:

Нормальний сценарій:

1. Користувач обирає книгу, до якої бажає написати рецензію або цитату.
2. Система надає можливість написання рецензії або цитати на сторінці книги.
3. Користувач вводить текст рецензії або цитати та зберігає їх.

Альтернативний сценарій (помилка збереження):

1. Користувач вводить текст рецензії або цитати.
2. Система не може зберегти введений текст через технічні проблеми або обмеження.
3. Система повідомляє користувача про неможливість збереження та рекомендує спробувати пізніше.

10. Оцінка книги:*Нормальний сценарій:*

1. Користувач відкриває сторінку книги, яку бажає оцінити.
2. Система надає можливість поставити оцінку книзі.
3. Користувач обирає оцінку за шкалою і підтверджує свій вибір.

Альтернативний сценарій (неможливість поставити оцінку):

1. Користувач обирає оцінку книги.
2. Система не може прийняти оцінку через технічні проблеми або обмеження.
3. Система повідомляє користувача про неможливість поставити оцінку та рекомендує спробувати пізніше.

11. Вибір формату читання (E-Book, Audio-Book):*Нормальний сценарій:*

1. Користувач відкриває сторінку книги, яку бажає прочитати або прослухати.
2. Система надає можливість вибору формату читання: E-Book або Audio-Book.
3. Користувач обирає бажаний формат і підтверджує свій вибір.

Альтернативний сценарій (неможливість вибору):

1. Користувач обирає формат читання книги.
2. Система не може обробити вибір через технічні проблеми або обмеження.
3. Система повідомляє користувача про неможливість змінити формат читання

12. Створення списку:

Нормальний сценарій:

1. Користувач обирає опцію "Створити список книг".
2. Система дозволяє користувачу ввести назву нового списку та обрати колір та зберігає його в базі даних.
3. Після збереження користувач може додавати книги до цього списку, вибираючи відповідну опцію на сторінці кожної книги.

Альтернативний сценарій (відмова системи в створенні списку):

1. Користувач намагається створити новий список книг у веб-застосунку.
2. Система повідомляє про технічні проблеми або обмеження, що перешкоджають створенню нового списку.
3. Користувач отримує повідомлення про неможливість створення нового списку.

13. Додавання користувача в друзі:

Нормальний сценарій:

1. Користувач переходить у профіль користувача, якого він бажає додати в друзі.
2. На сторінці профілю користувача знаходиться кнопка «Додати в друзі».
3. Користувач натискає на цю кнопку, і система додає обраного користувача в список друзів.

Альтернативний сценарій (відмова в додаванні в друзі):

1. Користувач спробував додати іншого користувача в друзі.

2. Система повідомляє про те, що запрошення в друзі не може бути відправлене, оскільки користувач вже є у списку друзів або заборонив додавання в друзі.

3. Користувач отримує повідомлення про неможливість додати обраного користувача в друзі.

14. Перегляд облікового запису та налаштувань:

Нормальний сценарій:

1. Користувач переходить у розділ «Мій обліковий запис» або «Налаштування».

2. В цьому розділі він може переглянути та змінити особисті дані, пароль, налаштування приватності та інші параметри облікового запису.

3. Після зміни параметрів користувач зберігає зміни, а система оновлює його обліковий запис з урахуванням нових налаштувань.

Альтернативний сценарій (відмова в зміні налаштувань):

1. Користувач намагається змінити налаштування облікового запису у веб-застосунку.

2. Система повідомляє про те, що зміни не можуть бути збережені через технічні проблеми або обмеження.

3. Користувач отримує повідомлення про неможливість змінити налаштування.

15. Зміна налаштувань чату:

Нормальний сценарій:

1. Користувач переходить у приватний чат.

2. Він може змінити параметри чату, такі як тему оформлення, очищення чату від повідомлень, або видалення чату.

3. Після зміни параметрів користувач зберігає зміни, а система оновлює налаштування чату згідно нових параметрів.

Альтернативний сценарій (відмова в зміні налаштувань чату):

1. Користувач намагається змінити налаштування чату у веб-застосунку.

2. Система повідомляє про те, що зміни не можуть бути збережені через технічні проблеми або обмеження.

3. Користувач отримує повідомлення про неможливість змінити налаштування чату.

Type: Scenario: Basic Path Взаємодія з книгами

</

Рис. 3.3.1. – Розробник сценаріїв (взаємодія з книгами)

Для забезпечення функціонування комп'ютера та комп'ютерних мереж у ІС були обрані наступні комплекси **системного програмного забезпечення**:

1. Операційна система: Windows обрано як основну операційну систему серверів для надійності та сумісності з великою кількістю програм.

2. RAD-засоби розробки проекту: Drawio та Enterprise Architect для моделювання бази даних; phpMyAdmin для розробки та управління базами даних.

3. Основа для сайту: HTML, PHP, CSS, JavaScript для розробки власних веб-сайтів з використанням різноманітних веб-технологій.

4. Сервісні програми: CCleaner для оптимізації роботи системи, Avast для захисту системи.

5. Програми технічного обслуговування: AIDA64 для діагностики та тестування обладнання на продуктивність та стабільність роботи.

6. Програми машинної графіки: Figma, пакет Adobe, CorelDRAW для редагування та створення графіки.

7. Інструментальне ПЗ: Visual Studio Code для редагування вихідного коду.

Для *прикладного ПЗ* описано сукупність програм:

1. ПЗ загального призначення: пакет програм для загальних завдань офісної роботи Microsoft Office.

2. ПЗ для роботи глобальних мереж: програми, які використовуються для доступу до даних через мережу Інтернет. Це можуть бути звичайні браузери (Opera, Google Chrome, Mozilla Firefox, Microsoft Edge), так і програми для спілкування та листування (Gmail, Zoom, Google Meet).

3. ПЗ для організації (адміністрування) обчислювального процесу: Laragon для налаштування та керування локальними веб-серверами, що забезпечує середовище розробки та тестування веб-додатків, phpMyAdmin для адміністрування баз даних MySQL, що дозволяє виконувати операції з базами даних через веб-інтерфейс.

Програмна документація включає в себе різні пункти для детального опису ІС, забезпечує всебічну підтримку користувачів та системних адміністраторів, гарантуючи ефективне та безпечне функціонування інформаційної системи для підтримки та розвитку читацької спільноти. Це дозволяє створювати, обробляти та зберігати дані, взаємодіяти з користувачами та забезпечувати стабільність і безпеку роботи системи. У Додатку В наведено фрагменти коду для веб-застосунку. Формуляр складає:

Назва – Веб-застосунок для підтримки та розвитку читацької спільноти.

Розробник – Кулішова Єлизавета Сергіївна.

Версія – 1.0.

Склад програмного забезпечення – Laragon, Visual Studio Code, Figma, phpMyAdmin.

Інформація про експлуатацію: Щомісячні оновлення, постійна підтримка з боку розробника.

Серед відомостей про призначення програми можна виділити:

- Управління контентом та спільнотою читачів;
- Моніторинг серверів та систем;

- Обробка та управління базами даних;
- Забезпечення безпеки і захисту даних.

А області виконання наведено такі:

- Спілкування та взаємодія учасників спільноти;
- Новини про нові книги та заходи від видавництв;
- Організація блогів авторів.

Не накладаються обмеження на застосування мінімальної конфігурації технічних засобів, так як сама система для користувачів не є вибагливою.

ВИСНОВКИ

У кваліфікаційному бакалаврському проєкті «Розробка веб-застосунку для підтримки та розвитку читацької спільноти» було виявлено значну актуальність теми розробки інформаційної системи для підтримки та розвитку читацької спільноти в сучасному інформаційному середовищі. Було проведено аналіз сучасних тенденцій у читацькій активності, що підтвердив зростання популярності електронних та аудіокниг, а також зменшення популярності друкованих видань. Результати свідчать про актуальність розробки інформаційної системи для підтримки читацької спільноти. Розробка нової системи має на меті вирішити цю проблему, забезпечуючи зручний та інтуїтивно зрозумілий інтерфейс для користувачів.

У вступі було розкрито суть і актуальність теми, останніх досліджень і публікацій. Було описано мету та завдання дослідження, об'єкт та предмет дослідження, методи дослідження. Також описано практичне значення отриманих результатів у ході аналізу, перелік використаних інструментів та структуру роботи.

У першому розділі було проведено дослідження предметної області. Було розкрито сучасний стан наукової думки щодо проблем даної тематики, проведено аналіз аналогічних систем (рис. 1.2.1. – 1.2.5.), (табл. 1.2.1.), висловлено та обґрунтовано авторську позицію, а також викладено методики, які будуть використані в наступних розділах кваліфікаційного бакалаврського проєкту. Розміщено таблицю Backlog, яка містить в собі список завдань, функціональностей або ідей (табл. 1.3.1.). Була описана робота веб-додатку для розвитку та підтримки читацької спільноти, а також представлений прототип інтерфейсу системи (рис. 1.3.1. – 1.3.46.).

У другому розділі були визначені основні вимоги: бізнес-вимоги (рис. 2.1.1.1.), функціональні вимоги (рис. 2.1.2.1. – 2.1.2.3.) та нефункціональні вимоги (рис. 2.1.3.1.) до ІС за допомогою мови SysML. Було продемонстровано діаграми, які відображають кількість, статус та

пріоритетність вимог до системи (рис. 2.1.3.2. – 2.1.3.4.). Було побудовано інформаційну модель задачі (рис. 2.2.1.1.). Сформовано вихідні (табл. 2.2.2.1.) та вхідні (табл. 2.2.3.1.) повідомлень. Також в цьому розділі було перелічено масиви використаної та результатної інформації (табл. 2.2.4.1.) виконано математичне забезпечення та побудовано алгоритм функціонування системи (рис. 2.2.4.1. - 2.2.4.14.). Змодельовано інформаційну підсистему: поведінку системи за допомогою діаграми варіантів використання (рис. 2.3.1.1.), діаграм послідовності (рис. 2.3.1.2. – 2.3.1.14.) та діаграми дій (рис. 2.3.1.15.); структуру системи за допомогою діаграми класів (рис. 2.3.2.1.), діаграми визначення блоків (рис. 2.3.2.2.) та діаграми внутрішніх блоків (рис. 2.3.2.3. – 2.3.2.4.); розподіл вимог за компонентами системи з допомогою діаграми трасування (рис. 2.3.3.1.), матриці трасування (рис. 2.3.3.2.), уточнюючих матриць трасування (рис. 2.3.3.3. – 2.3.3.5.) та матриці трасування по ним (рис. 2.3.3.6.).

У третьому розділі були описані забезпечення інформаційної системи такі як: інформаційне з загальною схемою ІЗ (рис. 3.1.1.), таблицями масивів і їх описом структури (табл. 3.1.1. – 3.1.8.), а також інфологічною (рис. 3.1.2.) та даталогічною моделями БД (рис. 3.1.3), технічне забезпечення, де зображено загальну схему автоматизації (рис. 3.2.1.) і програмне забезпечення де показано фрагмент з розробника сценаріїв (рис. 3.3.1.).

У додатках наведені звіти з Enterprise Architect по тестуванню кейсів, повна специфікація вимог до системи, програмний код.

Підсумовуючи, можна визначити, що розроблені концепції та моделі веб-застосунку відповідають вимогам предметної області. Використання сучасних інформаційних технологій та програмних середовищ сприяло створенню функціонального та ефективного інструменту для підтримки та розвитку читацької спільноти. Надалі рекомендується акцентувати увагу на розвитку гейміфікаційних елементів та персоналізованих рекомендацій для підвищення взаємодії користувачів із системою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. LB.ua. Інститут майбутнього дослідив, як повномасштабна війна вплинула на читання книг українцями. URL: https://lb.ua/society/2023/04/26/553158_institut_maybu_tnogo_doslidiv_yak.html. (дата звернення 16.04.2024).
2. The page. Книжки заходять у цифру: українські видавці про тренд на електронні книги. URL: <https://thepage.ua/ua/news/oglyad-rinku-elektronnih-knizhok-v-ukrayini-u-2023-roci>. (дата звернення: 16.04.2024).
3. ЧИТОМО. За час війни українці більше читають електронні й аудіокнижки. URL: <https://chytomo.com/za-chas-vijny-ukraintsi-bilshe-chytaiut-elektronni-j-avdioknyzhky/>. (дата звернення: 16.04.2024).
4. Forbes. Як читають українці. Під час вторгнення частка людей, які читають книги щодня, зросла удвічі. URL: <https://forbes.ua/news/yak-chitayut-ukraintsi-pid-chas-vtorgnennya-chastka-lyudey-yaki-chitayut-knigi-shchodnya-zrosla-vdvichi-doslidzhennya-uik-12102023-16629>. (дата звернення: 16.04.2024).
5. Goodreads. URL: <https://www.goodreads.com/>. (дата звернення: 09.04.2024).
6. Rork. URL: <https://rork.ua/>. (дата звернення: 09.04.2024).
7. LibraryThing. URL: <https://www.librarything.com/>. (дата звернення: 09.04.2024).
8. Google Play Books. URL: <https://play.google.com/store/apps/details?id=com.google.android.apps.books&hl=ru&gl=US>. (дата звернення: 09.04.2024).
9. Bookmate. URL: <https://bookmate.com/?lang=uk>. (дата звернення: 09.04.2024).
10. Автоматизоване проектування інформаційних систем: навч. посіб. / О.О. Денісова. – К.: КНЕУ, 2011. – 412 с.
11. Проектування баз і сховищ даних: Навч.-метод. посіб. для самост. вивч. дисц. / Ситник Н.В., Краснюк М.Т. — К.: КНЕУ, 2005. — 264 с.
12. Проектування інформаційних систем: през. курсу / Козак І.А.: КНЕУ

13. Документація для роботи з Enterprise Architect [Електронний ресурс] – Режим доступу до ресурсу: <https://sparxsystems.com/>
14. Документація для роботи з Figma [Електронний ресурс] – Режим доступу до ресурсу: <https://help.figma.com/hc/en-us/categories/360002042553-Figma-design>
15. Документація для роботи з UML діаграмами [Електронний ресурс] – Режим доступу до ресурсу: <https://www.uml-diagrams.org/>
16. Інформаційні технології віртуальних організацій: Навч.-метод. посіб. для самот. вивч. дисц. / Козак І.А., Мелашенко О.Б. – К.: КНЕУ, 2005. – 154 с.
17. Logos IT Academy UA. Урок 9. UI UX Як створити прототип сайту, або додатку? Що таке прототип?. YouTube URL: https://www.youtube.com/watch?v=fSa8Mh3TL-0&ab_channel=LogosITAcademyUA. (дата звернення: 16.04.2024)
18. Документація для роботи з Visual Paradigm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visual-paradigm.com/support/documents/vpuserguide.jsp>
19. Blog.imena.ua. Найкращі програми для діагностики комп'ютера. URL: <https://www.imena.ua/blog/best-programs-for-computer-diagnostics/>. (дата звернення: 18.05.2024)
20. JavaRush. Підхід MVC. URL: <https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture02>. (дата звернення: 18.05.2024)

ДОДАТКИ

Додаток А

Специфікація бізнес-вимог до ІС веб-додатку для розвитку та підтримки читацької спільноти

Ідентифікатор	Стисла назва вимоги	Повна назва вимоги	Тип	Статус	Складність	Пріоритетність	Ризикованість
BUS001	Збільшення активності	Стимулювання активності користувачів та збільшення кількості платників сервісу	Бізнес-вимога	Запропоновано	Висока	Висока	Середня
BUS002	Розробка та вдосконалення функціоналу	Розробка нового та вдосконалення існуючого функціоналу	Бізнес-вимога	Запропоновано	Висока	Висока	Низька
BUS003	Оптимізація швидкості та продуктивності	Покращення швидкості завантаження веб-застосунку та його оптимальної продуктивності, з урахуванням підтримки різних пристроїв та браузерів	Бізнес-вимога	Затверджено	Середня	Середня	Середня
BUS004	Цифровий маркетинг для збільшення обізнаності	Розробка та впровадження стратегій цифрового маркетингу для збільшення обізнаності про застосунок серед цільової аудиторії читацької спільноти	Бізнес-вимога	Запропоновано	Висока	Висока	Середня
BUS005	Підтримка користувачів	Забезпечення швидкого та ефективного вирішення запитань і проблем користувачів, використовуючи інструменти для моніторингу та взаємодії з клієнтами	Бізнес-вимога	Затверджено	Середня	Середня	Середня
REQ001	Авторизація користувачів	Можливість зайти у систему та користуватися функціоналом веб-додатку	Функціональна вимога	Затверджено	Середня	Середня	Середня

Ідентифікатор	Стисла назва вимоги	Повна назва вимоги	Тип	Статус	Складність	Пріоритетність	Ризикованість
REQ002	Пошук книг за назвою	Пошук книг в системі за їхніми назвами для швидкого доступу до конкретної літератури	Функціональна вимога	Підтверджено	Висока	Висока	Середня
REQ003	Прочитання або прослуховування книги	Зчитувати текст книги або прослуховувати її у вигляді аудіокниги через веб-застосунок	Функціональна вимога	Запропоновано	Висока	Висока	Низька
REQ004	Оцінювання книги	Ставити оцінку книзі і виражати свою думку про неї, що допомагає іншим користувачам у виборі літератури	Функціональна вимога	Підтверджено	Середня	Середня	Середня
REQ005	Додавання рецензії, або цитати до книги	Додавати свої рецензії або цитати з книги до системи для обміну враженнями та думками з іншими читачами	Функціональна вимога	Підтверджено	Середня	Середня	Середня
REQ006	Додавання книги до списку	Додавати обрані книги до персонального списку для подальшого зручного доступу та організації читачьких переліків	Функціональна вимога	Підтверджено	Середня	Середня	Середня
REQ007	Створення списку	Створювати власні списки книг за певними критеріями	Функціональна вимога	Підтверджено	Низька	Низька	Низька
REQ008	Додавання користувача у друзі	Можливість користувача додавати інших користувачів до свого списку друзів	Функціональна вимога	Запропоновано	Середня	Середня	Середня
REQ009	Спілкування у приватному чаті	Можливість користувачів обмінюватися приватними повідомленнями у веб-застосунку для конфіденційної комунікації	Функціональна вимога	Затверджено	Висока	Висока	Низька

Ідентифікатор	Стисла назва вимоги	Повна назва вимоги	Тип	Статус	Складність	Пріоритетність	Ризикованість
REQ010	Персоналізовані рекомендації	Індивідуалізовані рекомендації щодо книг на основі їхніх вподобань	Функціональна вимога	Затверджено	Висока	Висока	Середня
REQ011	Відправлення форми зворотного зв'язку	Відправлення повідомлення адміністраторам чи розробникам веб-застосунку для висловлення своїх пропозицій, зауважень або скарг	Функціональна вимога	Запропоновано	Середня	Середня	Середня
REQ012	Отримання додаткового функціоналу	Отримувати доступ до додаткового функціоналу веб-застосунку	Функціональна вимога	Затверджено	Середня	Середня	Середня
REQ013	Перегляд облікового запису	Перегляд свого особистого облікового запису для перевірки та редагування особистої інформації	Функціональна вимога	Підтверджено	Низька	Низька	Середня
REQ014	Перегляд та зміна налаштувань	Переглядати та змінювати налаштування веб-застосунку	Функціональна вимога	Запропоновано	Низька	Низька	Середня
REQ015	Зміна налаштувань у приватному чаті	Можливість користувача змінювати налаштування приватного чату	Функціональна вимога	Затверджено	Низька	Низька	Низька
REQ016	Забезпечення підтримки різних мов	Працювати на різних мовах для забезпечення зручного користування різноманітною аудиторією	Функціональна вимога	Затверджено	Середня	Середня	Середня
REQ017	Відстеження та оновленнями сервісу	Забезпечення можливості відстеження стану та оновлення сервісу для підтримки його безперервної роботи	Функціональна вимога	Підтверджено	Середня	Середня	Середня
REQ018	Перегляд детальної інформації про книгу	Надання користувачам можливості перегляду розширеної та докладної інформації про кожну книгу	Функціональна вимога	Запропоновано	Низька	Низька	Середня

Ідентифікатор	Стисла назва вимоги	Повна назва вимоги	Тип	Статус	Складність	Пріоритетність	Ризикованість
REQ019	Відповідь на запит користувача	Забезпечення можливості системи відповідати на запити користувачів шляхом надання корисної та зрозумілої інформації	Функціональна вимога	Затверджено	Середня	Середня	Низька
REQ020	Написання відповідей на запити користувачів	Забезпечення можливості системи відповідей на запити користувачів, враховуючи їхні запити	Функціональна вимога	Затверджено	Середня	Середня	Середня
REQ021	Введення даних в платіжній системі	Реалізація можливості користувачів вводити свої платіжні дані	Функціональна вимога	Запропоновано	Середня	Середня	Низька
REQ022	Потреба в додатковому функціоналі	Визначення необхідності та потреби у впровадженні додаткового функціоналу	Функціональна вимога	Затверджено	Середня	Середня	Середня
NFU001	Внесення даних в БД	Внесення даних в БД не більше 1 сек.	Нефункціональна вимога	Затверджено	Середня	Середня	Середня
NFU002	Безпечність та захищеність	Безпечність веб-сервісу та захищеність особистих даних користувачів	Нефункціональна вимога	Запропоновано	Висока	Висока	Середня
NFU003	Інтерфейс	Інтерфейс має бути інтуїтивно зрозумілий та легкий у користуванні	Нефункціональна вимога	Затверджено	Середня	Середня	Середня
NFU004	Масштабованість	Масштабованість для підтримки великої кількості відвідувань	Нефункціональна вимога	Затверджено	Висока	Висока	Середня
NFU005	Надійність	Надійність системи та робота без перебоїв	Нефункціональна вимога	Затверджено	Висока	Висока	Низька
NFU006	Формат зображень	Вхідні зображення формату .jpg, .png	Нефункціональна вимога	Затверджено	Низька	Низька	Середня

Звіти Enterprise Architect - Scenario Builder

TESTING
 Standard System test. Перевірка можливості реєстрації з новими валідними даними Description: 1. Неавторизований користувач переходить до форми реєстрації. 2. Неавторизований користувач вводить унікальний email. 3. Неавторизований користувач вводить валідний пароль. 4. Неавторизований користувач підтверджує пароль. 5. Неавторизований користувач натискає кнопку "Зареєструватися". Input: Унікальний email і пароль нового користувача у відповідну форму у якому використовуються лише латинські літери. Acceptance Criteria: Користувач повинен бути успішно зареєстрований у системі. Results: Новий користувач зареєстрований у системі. [Last run Not Run at Not run yet!.]

Рис. Б.1. – Zvim Scenario Builder – Реєстрація

TESTING
 Standard System test. Перевірка можливості входу в обліковий запис з валідними даними Description: 1. Неавторизований користувач переходить на форму входу в обліковий запис. 2. Неавторизований користувач вводить свій валідний пароль 3. Неавторизований користувач вводить свій email. 4. Неавторизований користувач натискає кнопку "Увійти". Input: Валідний email і пароль користувача у відповідну форму у якому використовуються лише латинські літери. Acceptance Criteria: Користувач повинен успішно увійти в обліковий запис. Results: Повідомлення про успішний вхід в обліковий запис. [Last run Not Run at Not run yet!.]

Рис. Б.2. – Zvim Scenario Builder – Вхід в обліковий запис

 Standard System test. Перевірка можливості відправлення форми зворотного зв'язку з валідними даними Description: 1. Користувач переходить на форму зворотного зв'язку. 2. Користувач вводить свій валідний email. 3. Користувач вводить повідомлення. 4. Користувач натискає кнопку "Відправити". Input: Email і повідомлення користувача у відповідну форму. Acceptance Criteria: Повідомлення користувача повинно бути успішно відправлене. Results: Повідомлення успішно надіслане. [Last run Not Run at Not run yet!.]
--

Рис. Б.3. – Zvim Scenario Builder – Форма зворотного зв'язку

TESTING
 Standard System test. Перевірка можливості користувача змінювати налаштування приватного чату. Description: 1. Авторизований користувач переходить до налаштувань приватного чату. 2. Авторизований користувач обирає опцію зміни фону чату та встановлює новий фон. 3. Авторизований користувач виконує опцію очищення чату. 4. Авторизований користувач обирає опцію видалення чату. Input: Новий фон чату, опція очищення чату, опція видалення чату. Acceptance Criteria: Фон чату успішно змінено на вказаний новий фон. Чат успішно очищено, всі повідомлення видалені. Чат успішно видалено, користувач повернутий на головний екран або на сторінку з чатами. Results: Зміни налаштувань чату були успішно виконані, і вони відображаються правильно на інтерфейсі користувача. [Last run Not Run at Not run yet!.]

Рис. Б.4. – Zvim Scenario Builder – Зміна налаштувань

TESTING
 Standard System test. Перевіряємо можливість користувача спілкуватися з іншим користувачем у приватному чаті. Description: 1. Перехід до переліку чатів на вкладці "Чати". 2. Авторизований користувач вибирає іншого користувача для спілкування у приватному чаті. 3. Авторизований користувач вводить текст повідомлення. 4. Авторизований користувач надсилає повідомлення. Input: Текст повідомлення, користувач, з яким потрібно спілкуватися. Acceptance Criteria: Повідомлення успішно надіслано та відображено у приватному чаті. Results: Повідомлення успішно надіслано та відображено у приватному чаті. [Last run Not Run at Not run yet!.]

Рис. Б.5. – Zvim Scenario Builder – Приватний чат

TESTING
 Standard System test. Перевіряємо можливість авторизованого користувача здійснити пошук книги за назвою. Description: 1. Перехід авторизованого користувача на вкладку Пошуку. 2. Авторизований користувач вводить назву книги в поле пошуку. 3. Авторизований користувач натискає кнопку "Пошук". Input: Назва книги, яку шукає користувач у відповідну пошукову форму. Acceptance Criteria: Система повинна показати результати пошуку, які відповідають введеній назві книги. Results: Результати пошуку відображаються коректно. [Last run Not Run at Not run yet!.]

Рис. Б.6. – Zvim Scenario Builder – Пошук книги

TESTING
 Standard System test. Перевірка можливості користувача написати цитату або рецензію до обраної книги. Description: 1. Авторизований користувач відкриває сторінку обраної книги. 2. Авторизований користувач обирає розділ для написання цитати або рецензії. 3. Авторизований користувач вводить текст цитати або рецензії. 4. Авторизований користувач зберігає написане. Input: Обрана книга, текст цитати або рецензії. Acceptance Criteria: Цитата або рецензія успішно зберігається та відображається на сторінці книги. Results: Цитата або рецензія успішно збережена та відображена на сторінці книги. [Last run Not Run at Not run yet!.]

Рис. Б.7. – Zvim Scenario Builder – Написання цитати/рецензії

TESTING
 Standard System test. Перевірка можливості користувача оплатити додатковий функціонал для доступу до розширених можливостей системи. Description: 1. Авторизований користувач переходить до розділу "Оплата додаткового функціоналу" в особистому кабінеті. 2. Авторизований користувач коректно вводить дані своєї картки для оплати. 3. Авторизований користувач підтверджує оплату. Input: Вибраний додатковий функціонал, дані картки користувача для оплати. Acceptance Criteria: Обрані додаткові можливості успішно активовані після оплати. Results: Додатковий функціонал успішно оплачений та активований для авторизованого користувача. [Last run Not Run at Not run yet!.]

Рис. Б.8. – Zvim Scenario Builder – Оплата дод. функціоналу

TESTING
 Standard System test. Перевіряємо можливість користувача переглядати свій обліковий запис та налаштовувати особисті дані. Description: 1. Авторизований користувач входить в особистий кабінет. 2. Авторизований користувач переходить до розділу "Мій профіль" або "Налаштування". Input: Обраний розділ веб-сайту. Acceptance Criteria: Користувач може переглянути та змінити особисті дані та налаштування облікового запису. Results: Обліковий запис та налаштування користувача успішно відображені та доступні для зміни. [Last run Not Run at Not run yet!.]

Рис. Б.9. – Zvim Scenario Builder – Перегляд облікового запису, налаштувань

TESTING
 Standard System test. Перевірка можливості користувача обрати бажаний формат читання для книги. Description: 1. Авторизований користувач відкриває сторінку обраної книги. 2. Авторизований користувач обирає бажаний формат читання (E-Book, Audio-Book). 3. Авторизований користувач переходить до інтерфейсу формату, який він обрав. Input: Обрана книга, обраний формат читання (E-Book, Audio-Book). Acceptance Criteria: Обраний формат читання успішно зберігається та доступний користувачеві під час читання/прослуховування книги. Results: Обраний формат читання успішно збережений та доступний користувачеві. [Last run Not Run at Not run yet!.]

Рис. Б.10. – Zvim Scenario Builder – Вибір формату читання

TESTING
 Standard System test. Перевіряємо можливість авторизованого користувача оцінити книгу. Description: 1. Авторизований користувач обирає книгу. 2. Перехід до вкладки детального опису книги. 3. Авторизований користувач вибирає оцінку для книги (від 1 до 5). 4. Авторизований користувач підтверджує свій вибір. Input: Обрана користувачем книга та оцінка. Acceptance Criteria: Оцінка книги користувачем зберігається у системі. Results: Оцінка книги успішно збережена. [Last run Not Run at Not run yet!.]

Рис. Б.11. – 3vim Scenario Builder – Оцінка книги

TESTING
 Standard System test. Перевіряємо можливість користувача додати іншого користувача у список друзів. Description: 1. Авторизований користувач переходить до профілю іншого користувача. 2. Авторизований користувач натискає кнопку "Додати до друзів". Input: Обраний профіль користувача, якого потрібно додати у друзі. Acceptance Criteria: Користувач успішно додається до списку друзів. Results: Користувач успішно доданий до списку друзів. [Last run Not Run at Not run yet!.]

Рис. Б.12. – 3vim Scenario Builder – Додавання в друзі

TESTING
 Standard System test. Перевіряємо можливість авторизованого користувача додати книгу до свого списку бажань. Description: 1. Перехід до вкладки з переліком книг, або до вкладки детального опису книги. 2. Авторизований користувач натискає кнопку "Додати до списку". Input: Обрана книга авторизованого користувача. Acceptance Criteria: Книга успішно додається до списку бажань авторизованого користувача. Results: Книга успішно додана до списку бажань. [Last run Not Run at Not run yet!.]

Рис. Б.13. – 3vim Scenario Builder – Додавання книги до списку

TESTING
 Standard System test. Перевіряємо можливість авторизованого користувача створювати персональні списки книг. Description: 1. Вибір авторизованим користувачем у боковому меню функції створення нового списку, або вибір цієї функції на вкладці "Мої книги". 2. Введення назви для списку книг. 3. Вибір кольору списку, у якому він буде відображатися. 4. Авторизований користувач натискає кнопку "Створити список". Input: Дані для створення списку. Acceptance Criteria: Список книг створений та збережений у особистому кабінеті користувача. Results: Список книг успішно створений та збережений. [Last run Not Run at Not run yet!.]

Рис. Б.14. – 3vim Scenario Builder – Створення списку

Фрагмент коду

Фрагменти форм з index.php:

```

<form id="contact_form" name="contact_form" enctype="multipart/form-data" class="modal-
content" method="POST" action="zapyt.php" data-name="Email Form" redirect="/main" data-
redirect="/main">

    <div class="form-group">
        <input class="text-field w-input" maxlength="256" name="email_user"
placeholder="Your email" type="text" id="email_user" data-name="email" data-ms-
member="email" required/>
    </div>
    <div class="form-group">
        <input class="text-field-2 w-input" maxlength="256" name="pass_user"
placeholder="Your password" type="password" id="pass_user" data-name="pass" data-ms-
member="signup-date" required/>
    </div>
    <div class="form-group">
        <input class="text-field-2-copy w-input" maxlength="256"
name="repass_user" placeholder="Repeat your password" type="password" id="repass_user"
data-name="repass" data-ms-member="signup-date" required/>
    </div>
    <button id="signBtn" type="submit" data-wait="Please wait..."
class="submit-button w-button">
        <b> <a href="main.php">SIGN UP </a></b>
    </button>
    <p class="p2_seventh_page">
        By continuing, you accept <strong class="bold-text">Readify &#x27;s
Terms of Service</strong>
        and confirm that you have read our <strong>Privacy Policy</strong>
        .
    </p>
    <input type="submit" data-wait="Please wait..." class="submit-button-copy
w-button" value="or continue as Liza Kulishova"/>
    <p class="p3_seventh_page">Already Sign up?</p>
    <a href="main.php" class="log_in">
        <strong>Log In!</strong>
    </a>
</form>

<div id="id01" class="email">
    <span onclick="document.getElementById('id01').style.display='none'" class="close"
title="Закреть Email">&times;</span>
    <form id="email_form" enctype="multipart/email-data" class="email-content"
method="POST" action="form.php">

```

```

        <div class="container">
        <h1 style="margin-left: 15px" name=reg class="tip">SEND US A TIP</h1>

        <div class="container-xxl">
        <div class="row row-cols-2">
        <div class="email-group">
            <input type="text" id="email_form" class="email_form" placeholder="Your
Email" name="email_form" required></div>
        <div class="email-group">
            <input type="text" id="mess_form" class="mess_form" placeholder="Your
Message" name="mess_form" required></div>

            <button id="formBtn" type="submit" data-wait="Please wait..." class="frmbutton
w-button">
                <b>SEND</b>
            </button>
        </div>
    </div>
</form>

```

Main.js:

```

$(function () {
    $('#contact_form').submit(function (event) {
        event.preventDefault();
        var formValid = true;
        $('#contact_form input').each(function () {
            var formGroup = $(this).parents('.form-group');
            if (this.checkValidity()) {
                formGroup.addClass('has-success').removeClass('has-error');
            } else {
                formGroup.addClass('has-error').removeClass('has-success');
                formValid = false;
            }
        });
        if (formValid) {
            var email_user = $("#email_user").val();
            var pass_user = $("#pass_user").val();
            var repass_user = $("#repass_user").val();
            var formData = new FormData();
            formData.append('email_user', email_user);
            formData.append('pass_user', pass_user);
            formData.append('repass_user', repass_user);
            $.ajax({
                type: "POST",
                url: "zapyt.php",
                data: formData,
                contentType: false,
                processData: false,
            });
        }
    });
});

```

```

cache: false,
success: function (data) {
    try {
        var $data = JSON.parse(data);
        if ($data.result == "success") {
            // Виконуємо перехід на сторінку main.php
            window.location.href = "main.php";
        } else {
            // Обробляємо помилки, якщо такі є
            $('#error').text('Якщо Ви не хочете вводити дані, зателефонуйте нам!');
            if ($data.email_user) {
                $('#email_user').parents('.form-group').addClass('has-
error').removeClass('has-success');
                $('#error').text('Ви не ввели електронну пошту');
            }
            if ($data.pass_user) {
                $('#pass_user').parents('.form-group').addClass('has-
error').removeClass('has-success');
                $('#error').text('Ви не ввели пароль');
            }
            if ($data.repass_user) {
                $('#repass_user').parents('.form-group').addClass('has-
error').removeClass('has-success');
                $('#error').text('Ви не ввели повторення паролю');
            }
        }
    } catch (e) {
        $('#error').text('Виникла помилка при обробці даних. Спробуйте
зателефонувати нам.');
```

Form.js:

```

$(function () {
    $('#email_form').submit(function (event) {
        event.preventDefault();
        var emailValid = true;
        $('#email_form input').each(function () {
            var emailGroup = $(this).parents('.email-group');
```

```

    if (this.checkValidity()) {
        emailGroup.addClass('has-success').removeClass('has-error');
    } else {
        emailGroup.addClass('has-error').removeClass('has-success');
        emailValid = false;
    }
});
if (emailValid) {
    var email_form = $("#email_form").val();
    var mess_form = $("#mess_form").val();
    var emailData = new FormData();
    emailData.append('email_form', email_form);
    emailData.append('mess_form', mess_form);
    $.ajax({
        type: "POST",
        url: "form.php",
        data: emailData,
        contentType: false,
        processData: false,
        cache: false,
        success: function (data) {
            try {
                var $data = JSON.parse(data);
                if ($data.result === "success") {
                    // Виконуємо перехід на сторінку main.php
                    window.location.href = "index.php";
                } else {
                    // Обробляємо помилки, якщо такі є
                    $('#error').text('Якщо Ви не хочете вводити дані, зателефонуйте нам!');
                    if ($data.email_form) {
                        $('#email_form').parents('.email-group').addClass('has-
error').removeClass('has-success');
                        $('#error').text('Ви не ввели електронну пошту');
                    }
                    if ($data.mess_form) {
                        $('#mess_form').parents('.email-group').addClass('has-
error').removeClass('has-success');
                        $('#error').text('Ви не ввели повідомлення');
                    }
                }
            } catch (e) {
                $('#error').text('Виникла помилка при обробці даних. Спробуйте
зателефонувати нам.');
```

```

    }
  });
}
});
});

```

Zapyt.php:

```

<?php
$data['result'] = 'error';
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $data['result'] = 'success';
    $email_user = $_POST['email_user'] ?? "";
    $pass_user = $_POST['pass_user'] ?? "";
    $repass_user = $_POST['repass_user'] ?? "";
    // Check if the form fields are received properly
    var_dump($email_user, $pass_user, $repass_user);
    if (empty($email_user)) {
        $data['result'] = 'error';
        $data['email_user'] = 'Field email_user is empty.';
    }
    if (empty($pass_user)) {
        $data['result'] = 'error';
        $data['pass_user'] = 'Field pass_user is empty.';
    }
    if (empty($repass_user)) {
        $data['result'] = 'error';
        $data['repass_user'] = 'Field repass_user is empty.';
    }
}
if ($data['result'] == 'success') {
    include("con_bd.php");
    $stmt = mysqli_prepare($link, "INSERT INTO user(email_user, pass_user, repass_user)
VALUES (?, ?, ?)");
    mysqli_stmt_bind_param($stmt, 'sss', $email_user, $pass_user, $repass_user);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_close($stmt);
    mysqli_close($link);
}
echo json_encode($data);
?>

```

Form.php:

```

<?php
$data['result'] = 'error';
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $data['result'] = 'success';

```

```

$email_form = $_POST['email_form'] ?? "";
$mess_form = $_POST['mess_form'] ?? "";
// Check if the form fields are received properly
var_dump($email_form, $mess_form);
if (empty($email_form)) {
    $data['result'] = 'error';
    $data['email_form'] = 'Field email_form is empty.';
}
if (empty($mess_form)) {
    $data['result'] = 'error';
    $data['mess_form'] = 'Field mess_form is empty.';
}
}
if ($data['result'] == 'success') {
    include("con_bd.php");
    $stmt = mysqli_prepare($link, "INSERT INTO form(email_form, mess_form) VALUES (?,
?");
    mysqli_stmt_bind_param($stmt, 'ss', $email_form, $mess_form);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_close($stmt);
    mysqli_close($link);
}
echo json_encode($data);
?>

```

Con_bd.php:

```

<?php
    /*Змінні для з'єднання з БД */
    $hostname = "localhost";
    $username = "root";
    $passwordbd = "root";
    $dbName = "readify";
    /* з'єднатися*/
    $link = mysqli_connect($hostname,$username,$passwordbd,$dbName);
    if (!$link) {
        echo "Помилка: Неможливо встановити з'єднання з MySQL." . PHP_EOL;
        echo "Код помилки error: " . mysqli_connect_errno() . PHP_EOL;
        echo "Текст помилки error: " . mysqli_connect_error() . PHP_EOL;
        exit;
    }
    $link = mysqli_connect($hostname, $username, $passwordbd, $dbName);
?>

```

КОРОТКИЙ ЗВІТ ПОДІБНОСТІ



Ім'я користувача: Інформаційних систем в економіці Шкуратовська Те...	ID перевірки: 1016321906
Дата перевірки: 05.06.2024 08:16:00 EEST	Тип перевірки: Doc vs Internet + Library
Дата звіту: 05.06.2024 13:27:40 EEST	ID користувача: 100005745

Назва документа: Кулішова_Є_IH_401

Кількість сторінок: 126 Кількість слів: 17586 Кількість символів: 139095 Розмір файлу: 20.00 MB ID файлу: 1016116644

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

4.83%
Схожість

Найбільша схожість: 1.31% з джерелом з Бібліотеки (ID файлу: 1007991670)

3.04% Джерела з Інтернету	281	Сторінка 128
4.55% Джерела з Бібліотеки	506	Сторінка 131

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 37 сторінок

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗВИТКУ ТА ПІДТРИМКИ ЧИТАЦЬКОЇ СПІЛЬНОТИ

В ході ознайомлення з предметною областю інформаційної системи (ІС) веб-застосунку для підтримки та розвитку читачької спільноти було виявлено значну актуальність теми. Ознайомившись з різними типами веб-сервісів пов'язаних з книгами та читанням таких як Goodreads, Rork, LibraryThing, Google Play Books, Bookmate, проаналізувавши переваги та недоліки функціоналу та зручності кожного, було прийнято розробити ІС, яка б поєднувала в собі можливості доступу до електронних та аудіокниг, обговорення літературних творів у спільнотах та чатах, а також створення особистих списків читання.

Визначено функціональну модель задачі, характеристику самої задачі, вихідну та вхідну інформацію, а також розглянуто математичне забезпечення та алгоритм функціонування системи. Обрано функціональний – методології IDEF0, DFD та IDEF3 (діаграми контекстна та декомпозиції), процесний - методологія BPMN (діаграми взаємодії, процесів, діалогів та хореографічна) та об'єктний підходи (діаграми прецедентів та послідовності мовою UML) до проектування веб-застосунку з метою забезпечення комплексного та систематичного підходу до розробки. Було розроблено рішення для ІС за різними видами забезпечення.

Для реалізації ідеї було використано різноманітні програмні середовища, які надали зручний та ефективний інструментарій для різних етапів роботи. Figma була потрібна для дизайну та прототипування веб-застосунку. Bizagi був використаний для процесного моделювання, включаючи діаграми взаємодії та процесів. Lucidchart використовувався для процесного моделювання, зокрема для створення діаграм діалогів. Drawio допоміг при виконанні хореографічного моделювання та створенні алгоритмів розв'язання задачі. У Visual Paradigm було створено діаграми для об'єктно-орієнтованого моделювання, включаючи діаграми використання, класів, послідовностей та діяльності. EdrawMax був потрібний для об'єктно-орієнтованого моделювання, зокрема для створення діаграм компонентів та розгортання. Creately було використано для створення діаграм станів, допомагаючи визначити стани та переходи між ними.

Підсумовуючи, можна визначити, що розроблені моделі ІС веб-застосунку відповідають вимогам предметної області. Використання сучасних інформаційних технологій та програмних середовищ сприяло створенню функціонального та ефективного інструменту для підтримки та розвитку читачької спільноти.

Список використаних джерел

1. Автоматизоване проектування інформаційних систем: навч. посіб. / О.О. Денісова. – К.: КНЕУ, 2011. – 412 с.
2. Проектування баз і сховищ даних: Навч.-метод. посіб. для самост. вивч. дисц. / Ситник Н.В., Краснюк М.Т. — К.: КНЕУ, 2005. — 264 с.
3. Проектування інформаційних систем: през. курсу / Козак І.А.: КНЕУ
4. Інформаційні технології віртуальних організацій: Навч.-метод. посіб. для самост. вивч. дисц. / Козак І.А., Мелашенко О.Б. – К.: КНЕУ, 2005. – 154 с.
5. Forbes. Українці читають все більше: частина людей, які читають книги щодня, зросла вдвічі [Електронний ресурс] / Дослідження УІК. – 2023. – Режим доступу до ресурсу: <http://surl.li/sohmn>
6. Документація для роботи з Figma [Електронний ресурс] – Режим доступу до ресурсу: <https://help.figma.com/hc/en-us/categories/360002042553-Figma-design>
7. Документація для роботи з UML діаграмами [Електронний ресурс] – Режим доступу до ресурсу: <https://www.uml-diagrams.org/>

8. Документація для роботи з Bizagi [Електронний ресурс] – Режим доступу до ресурсу: <https://help.bizagi.com/process-modeler/en/>

9. Документація для роботи з Visual Paradigm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visual-paradigm.com/support/documents/vpuserguide.jsp>

Науковий керівник: Козак І. А., к.е.н., доцент.