Високорівнева архітектура системи "БукМаркет"

1. Архітектурний підхід

Платформа "БукМаркет" базується на мікросервісній архітектурі, що дозволяє забезпечити гнучкість, масштабованість та відмовостійкість системи. **Основні технологічні рішення включають:**

- Фронтенд: Single Page Application (SPA) з використанням React.js
- Бекенд: Мікросервісна архітектура з використанням Node.js
- Хмарне розгортання: Контейнеризація з Docker і оркестрація з Kubernetes
- Бази даних: MongoDB (основна БД), Redis (кешування), ElasticSearch (повнотекстовий пошук)
- Зовнішні інтеграції: RESTful API, WebSockets для повідомлень реального часу

2. Мікросервісна архітектура

2.1. Основні мікросервіси

1) Сервіс автентифікації та авторизації

- Реєстрація та автентифікація користувачів
- Управління JWT токенами
- Двофакторна автентифікація
- Інтеграція з OAuth провайдерами

2) Сервіс управління користувачами

- Профілі користувачів
- Налаштування приватності
- Історія активності
- Управління адресами доставки

3) Сервіс каталогу книг

- Додавання та редагування оголошень
- Пошук та фільтрація книг
- Управління фотографіями
- Інтеграція з зовнішніми API для отримання метаданих книг

4) Сервіс повідомлень

- Обмін повідомленнями між користувачами
- Управління чатами
- Сповіщення про нові повідомлення
- Архівування історії спілкування

5) Сервіс транзакцій

- Створення та управління замовленнями
- Інтеграція з платіжними системами
- Управління статусами транзакцій
- Звіти та аналітика продажів

6) Сервіс доставки

- Інтеграція з логістичними API
- Розрахунок вартості доставки
- Генерація накладних
- Відстеження статусу доставки

7) Сервіс рейтингів та відгуків

- Додавання та модерація відгуків
- Обчислення рейтингів
- Аналіз відгуків
- Рекомендації на основі рейтингів

8) Сервіс сповіщень

- Email сповіщення
- Push-сповіщення для мобільних додатків
- Внутрішні сповіщення в системі
- Управління підписками на сповіщення

9) Сервіс адміністрування

- Модерація контенту
- Управління користувачами та блокування
- Аналітика та звіти
- Налаштування системних параметрів

2.2. Інфраструктурні сервіси

1) API Gateway

- Маршрутизація запитів до відповідних мікросервісів
- Аутентифікація та авторизація запитів
- Балансування навантаження

- Кешування відповідей

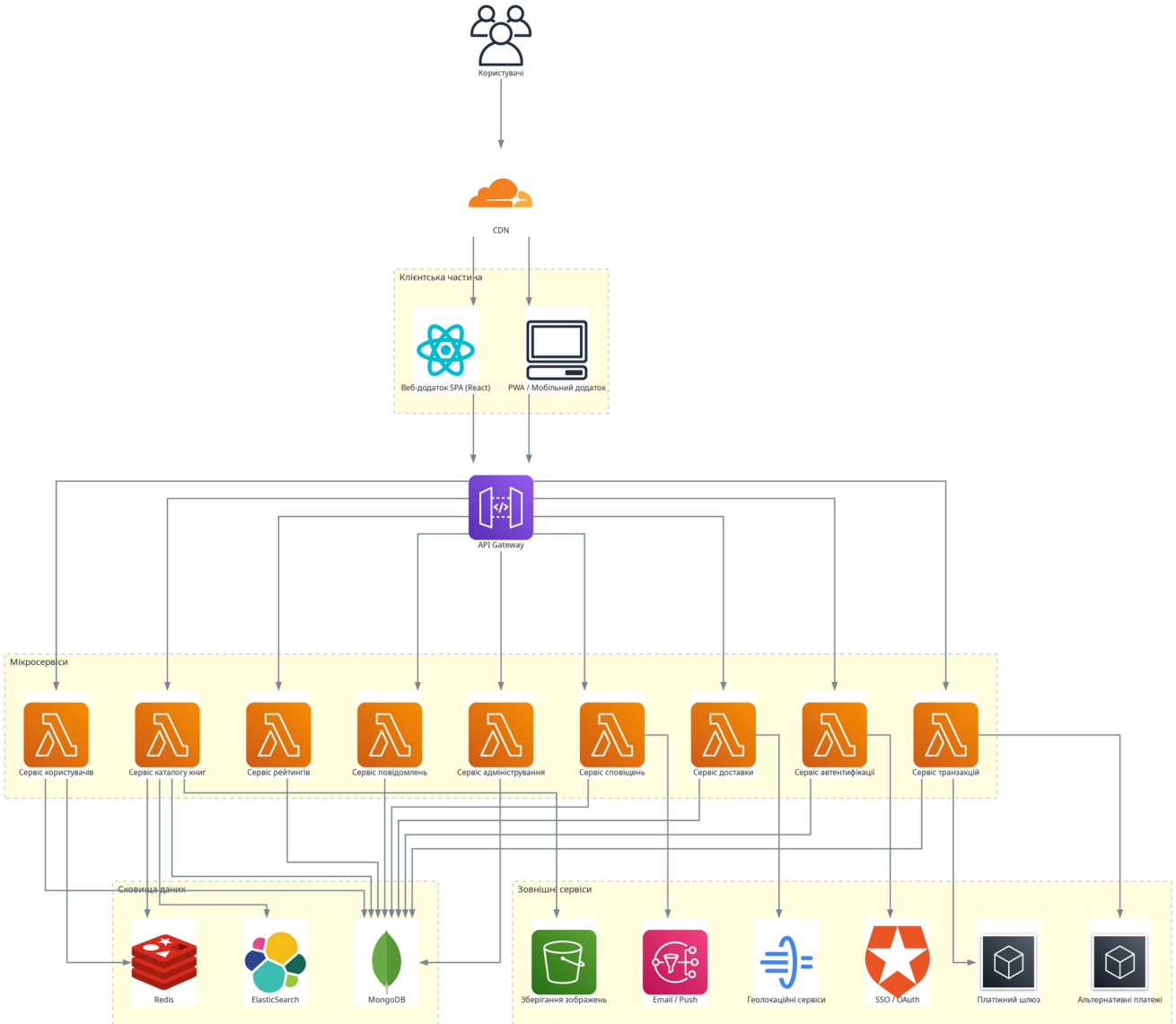
2) Сервіс логування

- Централізований збір логів з усіх сервісів
- Аналіз та моніторинг логів
- Сповіщення про помилки

3) Сервіс моніторингу

- Збір метрик продуктивності
- Візуалізація метрик
- Сповіщення при перевищенні порогових значень

3. Схема взаємодії компонентів



БукМаркет - Високорівнева архітектура

Згенерована діаграма показує взаємодію основних компонентів системи:

- Клієнтська частина (Веб-додаток SPA)
- API Gateway
- Мікросервіси (Автентифікація, Книги, Транзакції, Доставка)
- Сховища даних (MongoDB, ElasticSearch)
- Зовнішні сервіси (Зображення, Платежі, OAuth)

4. Шаблони конкурентності (concurrency patterns)

4.1. Event-Driven Architecture

Опис: Архітектура, заснована на виробництві, виявленні, споживанні та реагуванні на події.

Застосування в системі:

1. Сповіщення між мікросервісами про зміни стану (наприклад, зміна статусу замовлення)
2. Асинхронна обробка задач, які не потребують миттєвої відповіді (наприклад, відправка email)
3. Реагування на події користувача через WebSockets для оновлення інтерфейсу в реальному часі

4.2. Actor Model

Опис: Модель паралельних обчислень, де "актори" є універсальними примітивами паралельного виконання.

Застосування в системі:

1. Обробка повідомлень у чаті, де кожен чат є окремим актором
2. Обробка користувацьких сесій, де кожна сесія - окремий актор
3. Ізоляція обробки замовлень для забезпечення атомарності операцій

4.3. Producer-Consumer Pattern

Опис: Шаблон, в якому один або більше виробників (producers) створюють дані, які зберігаються в буфері, а один або більше споживачів (consumers) використовують ці дані.

Застосування в системі:

1. Обробка завантаження та оптимізації зображень книг
2. Система масових сповіщень, де сервіс створює повідомлення, а обробники відправляють їх користувачам
3. Асинхронна обробка платежів, де транзакція створюється одразу, а обробляється пізніше

4.4. Saga Pattern

Опис: Шаблон для управління розподіленими транзакціями, де кожна локальна транзакція оновлює базу даних і публікує повідомлення або подію для запуску наступної локальної транзакції в сазі.

Застосування в системі:

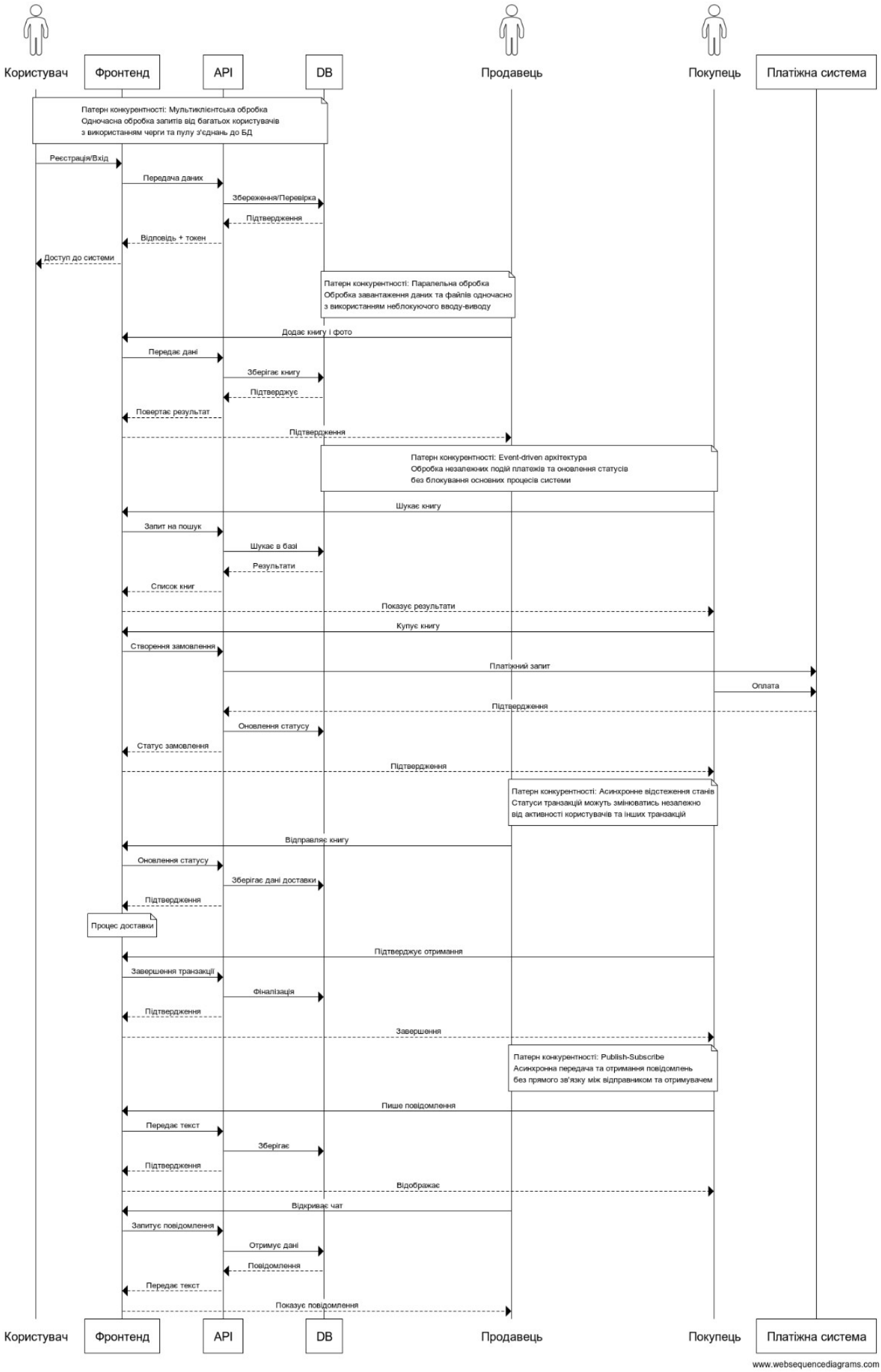
1. Процес оформлення замовлення (перевірка наявності → резервування → оплата → підтвердження)
2. Процес доставки книги з кількома етапами та можливістю скасування на кожному
3. Процес реєстрації користувача з підтвердженням email та встановленням профілю

4.5. Circuit Breaker Pattern

Опис: Шаблон, який запобігає відмові всієї системи при відмові окремих компонентів, дозволяючи швидко визначити проблему і відновити роботу після її вирішення.

Застосування в системі:

1. Захист від відмов при інтеграції з зовнішніми платіжними системами
2. Захист від перевантаження системи повідомлень при піковому навантаженні
3. Захист від каскадних відмов при проблемах з базою даних



5. Технологічний стек

5.1. Фронтенд

- Основний фреймворк: React.js
- Робота зі станами: Redux, Redux Toolkit
- Стилізація: Styled Components, Material-UI
- Маршрутизація: React Router
- Запити до API: Axios, React Query
- Валідація форм: Formik, Yup
- Тестування: Jest, React Testing Library

5.2. Бекенд

- Основна мова: Node.js з TypeScript
- Фреймворк для API: Express.js, Nest.js
- Валідація даних: Joi, class-validator
- Документація API: Swagger/OpenAPI
- Тестування: Jest, Supertest

5.3. Бази даних та сховища

- Основна БД: MongoDB
- Кешування: Redis
- Повнотекстовий пошук: ElasticSearch
- Зберігання файлів: AWS S3 або аналоги
- Управління міграціями: MongoDB Migrations

5.4. Messaging та інтеграції

- Брокер повідомлень: RabbitMQ або Kafka
- WebSockets: Socket.io
- Платіжні шлюзи: Stripe, PayPal
- Логістичні API: Нова Пошта, Укрпошта, тощо
- Геолокаційні сервіси: Google Maps API

5.5. DevOps та інфраструктура

- Контейнеризація: Docker
- Оркестрація: Kubernetes
- CI/CD: GitLab CI або GitHub Actions
- Моніторинг: Prometheus, Grafana
- Логування: ELK Stack (Elasticsearch, Logstash, Kibana)
- Хмарні провайдери: AWS, GCP або Azure

6. Стратегія масштабування

6.1. Горизонтальне масштабування

- Автоматичне масштабування контейнерів на основі навантаження
- Розподіл навантаження між екземплярами сервісів
- Кешування для зменшення навантаження на бази даних

6.2. Вертикальне масштабування

- Оптимізація запитів до бази даних
- Використання індексів та денормалізації для підвищення продуктивності
- Розподіл читання та запису для бази даних (Read Replicas)

6.3. Географічне масштабування

- Використання CDN для статичного контенту
- Геошардинг для розташування даних ближче до користувачів
- Регіональні бекапи та відновлення в разі відмови

7. Безпека та захист даних

7.1. Аутентифікація та авторизація

- Багатофакторна автентифікація
- Токени доступу з обмеженим терміном дії
- Рольова та атрибутивна система контролю доступу

7.2. Захист даних

- Шифрування даних у стані спокою та під час передачі
- Анонімізація чутливих даних для аналітики
- Аудит дій користувачів та адміністраторів

7.3. Захист від атак

- Захист від OWASP Top 10 вразливостей
- Обмеження кількості запитів (Rate Limiting)
- Моніторинг та виявлення аномалій

8. Моніторинг та спостережуваність

8.1. Збір метрик

- Технічні метрики (CPU, пам'ять, дисковий простір)
- Бізнес-метрики (кількість транзакцій, активні користувачі)

- Метрики продуктивності (час відгуку, затримки)

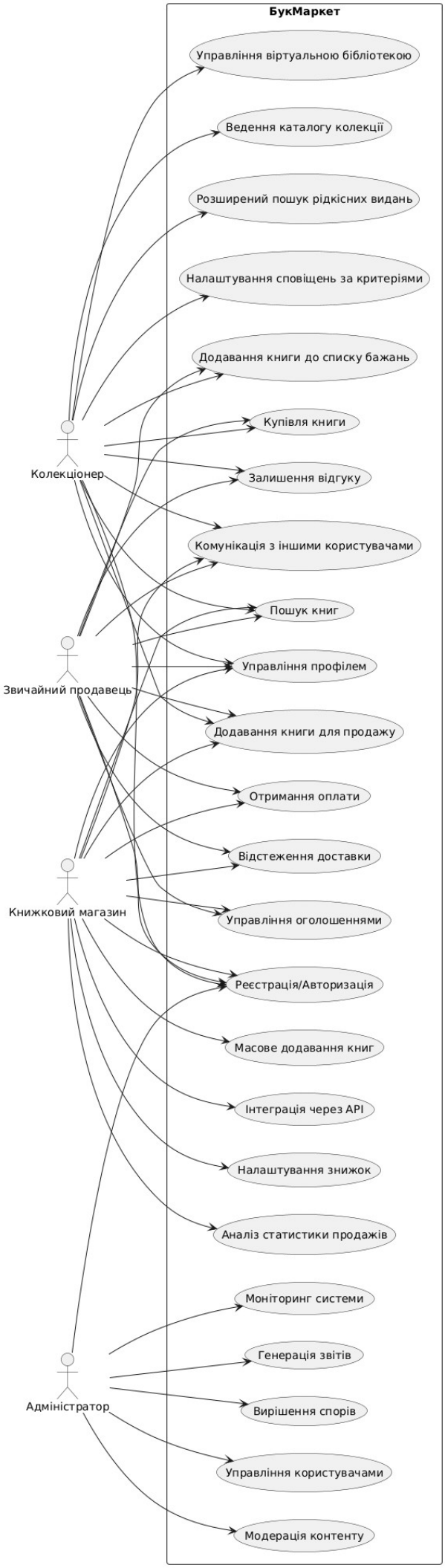
8.2. Інструменти моніторингу

- Prometheus для збору метрик
- Grafana для візуалізації
- Alertmanager для сповіщень

8.3. Централізоване логування

- Збір логів з усіх сервісів
- Аналіз логів для виявлення проблем
- Пошук та фільтрація логів

Додаток 1


БукМаркет
Управління віртуальною бібліотекою
Ведення каталогу колекції
Розширений пошук рідкісних видань
Налаштування сповіщень за критеріями
Додавання книги до списку бажань
Купівля книги
Залишення відгуку
Комунікація з іншими користувачами
Пошук книг
Управління профілем
Додавання книги для продажу
Отримання оплати
Відстеження доставки
Управління оголошеннями
Реєстрація/Авторизація
Масове додавання книг
Інтеграція через API
Налаштування знижок
Аналіз статистики продажів
Моніторинг системи
Генерація звітів
Вирішення спорів
Управління користувачами
Модерація контенту
Колекціонер
Звичайний продавець
Книжковий магазин
Адміністратор