МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

КАФЕДРА АВТОМАТИКИ ТА УПРАВЛІННЯ В ТЕХНІЧНИХ СИСТЕМАХ

КУРСОВИЙ ПРОЕКТ

з дисципліни «Сучасні методології і технології розробки ПЗ - 3»

Магазин книг

|  |  |
| --- | --- |
| **Керівник**: асистент Бабійчук А.А.  Допущено до захисту  «\_\_\_» \_\_\_\_\_\_\_\_ 2019 р  Захищено з оцінкою  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Виконав**: Тимченко О.Ю.  Студент ІІІ курсу  групи ІT-61  залікова книжка  № ІT-6124 |

Київ 2019

**Міністерство освіти та науки України**

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

Кафедра Кафедра автоматики та управління в технічних системах

Дисципліна Сучасні методології і технології розробки ПЗ - 3

Спеціальність Інженерія програмного забезпечення

Курс ІІІ Група IТ-61 Семестр 6

**ЗАВДАННЯ**

**на курсовий проект (роботу) студента**

Тимченка Олексія Юрійовича

(прізвище, ім’я, по батькові)

1. **Тема проекту** (роботи) Магазин книг

2. **Термін здачі** студентом закінченого проекту (роботи) “ 13 ” червня 2019р.

3. **Вхідні дані до проекту** (роботи) технічне завдання

4. **Зміст пояснювальної записки (перелік питань, які підлягають розробці):**

*розробка інформаційного забезпечення, розробка програмного забезпечення, розробка технічного забезпечення.*

**5. Перелік графічного (ілюстративного) матеріалу**

*Схема бази даних, Архітектура системи, Інтерфейс програми*

**6**. **Дата видачі завдання** “ 14 ” квітня 2019р.

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК**   
**виконання курсового проекту (роботи)**

**студентом**  Тимченко О. Ю.

(прізвище, ініціали)

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання | Термін  виконання | Примітки |
| 1. | Характеристика об’єкту автоматизації | 26.02 |  |
| 2. | Постанова задачі системи | 14.04 |  |
| 3. | Аналіз вимог до системи | 14.04 |  |
| 4. | Розробка інформаційного забезпечення | 24.04 |  |
| 5. | Розробка програмного забезпечення | 15.05 |  |
| 6. | Виконання опису технічного забезпечення | 22.05 |  |
| 7. | Виконання графічних документів | 29.05 |  |
| 8. | Оформлення пояснювальної записки | 05.06 |  |
| 9. | Здача курсового проекту | 13.06 |  |

**Студент** \_\_\_\_\_\_\_\_\_\_ Тимченко О. Ю. .

(підпис) (ПІБ)

**Керівник**\_\_\_\_\_\_\_\_\_\_ Бабійчук А. А. 0

(підпис) (ПІБ)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 р.

Анотація

В рамках курсової роботи було описано та розглянуто на практиці створення сучасного розподіленого додатку що побудований на базі мікросервісної архітектури. Описана актуальність теми та взаємодія окремих компонентів цієї архітектури. Також обґрунтований вибір тих чи інших технологій, структура програми. Щоб забезпечити комфортну роботу з додатком було проведено роботи по створенню зручного користувацького інтерфейсу, а також описані усі клієнтські інтерфейси. Робота була виконана в середовищі розробки IntelliJ IDEA, мовою програмування Java з використанням фреймворку Spring.

Метою даної роботи є практичне застосування вивченого матеріалу, а також створення веб-сервісу з графічним інтерфейсом «Магазин книг» що може бути корисним для широкого кола потенційних користувачів.  
 Обсяг роботи – 15 сторінок, 1 додаток.

ЗМІСТ

ВСТУП………………………………………………………………………….6

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ………………………………………………….7

1.1 Загальні відомості та характеристика об’єкту автоматизації……7

1.2 Постановка завдання…………………………………………….....7

2 ВИМОГИ ДО СИСТЕМИ…………………………………………………...8

2.1 Вимоги до розроблюваної системи……………………………......8

2.2 Вимоги до середовища виконання………………………………....8

3 ОПИС ВИДІВ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ…………………………….10

3.1 Технології та засоби розробки системи………………………….10

3.2 Інформаційне забезпечення системи…………………………..…10

4 ОПИС СИСТЕМИ………………………………………………………….12

4.1 Опис структури та призначення сервісів………………………...12

4.2 Опис функціонування системи…………………………………...13

ВИСНОВКИ…………………………………………………………………..15

ДОДАТОК А………………………………………………………………….16

Вступ

На даний момент, стає цілком очевидно що кількість та масштаб веб-сервісів невпинно зростає разом зі збільшенням інформатизації суспільства та кількості користувачів. В зв’язку с цим, розробникам необхідно будувати сервіси які мають добре масштабуватися. Для досягнення такої мети необхідно правильно обрати архітектуру для веб-додатку. На даний момент найбільш повно цим вимогам відповідає мікросервісна архітектура, вона полягає в том що повна система розбивається на набір невеликих сервісів - мікросервісів, а взаємодія між різними частинами функціональності проходить через API, тобто мікросервіси мають за комунікувати через який-небудь протокол передачі даних, наприклад HTTP чи TCP/IP.

В даній курсовій роботі описано навчального проекту під назвою магазин книг. Додаток розроблюється з ціллю ознайомлення з технологіями, архітектурами та правильнім підходом до процесу розробки сучасного програмного забезпечення. Пріоритетною задачею є набуття навичок побудови інформаційних систем на основі мікросервісної архітектури.

Для розробки обрано мову програмування Java з широким використанням можливостей фреймворку Spring. В якості провайдеру черги повідомлення було обрано RabbitMq. Розробка велась в середовищі IntelliJ IDEA. Обґрунтування вибору технологій, їх перелік і опис виконано у відповідному розділі.

.

1. Загальні Положення

1.1 Загальні відомості та характеристика об’єкту автоматизації

Дана робота виконана в рамках курсу «Сучасні методології і технології розробки ПЗ – 3» та базується на циклі лабораторних робіт виконаних під час курсу. Для побудови мікросервісної архітектури використана розглянута за курс технологія Spring Cloud.

Основним завданням даної роботи це закріплення вже набутих навичок розробки програмного забезпечення та опанування технології Spring. Особливо важливим є закріплення основних постулатів та принципів мікросервісної архітектури.

1.2 Постановка завдання

Метою роботи є створення веб-додатку для магазину книг, що дозволить переглядати наявні книги, їх жанри та клієнтів, а також модифікувати їх та додавати власні . Вся система має бути побудована на мікросервісній архітектурі.

Для досягнення мети були сформульовані такі задачі:

* аналіз існуючих рішень за мікросервісною архітектурою;
* проектування та розробка оптимальної архітектури системи;
* розробка графічного інтерфейсу користувача;
* створення, редагування та видалення об’єктів предметної області;
* логування операцій з даними;
* ознайомлення з використаними технологіями;

1. ВиМОГИ ДО СИСТЕМИ

2.1 Вимоги до розроблюваної системи

Провівши аналіз вимог що поставлені до роботи та технічного завдання можуть бути сформовані загальні вимоги.

Загальні вимоги:

* використання мікросервісної архітектури;
* використання технології Spring Cloud;
* використання платформи Rabbit MQ

Вимоги до прикладного програмного інтерфейсу, функціоналу:

* створення, редагування, видалення об’єктів предметної області;
* логування всіх запитів що змінюють об’єкти предметної області
* можливість перегляду всіх існуючих об’єктів предметної області;
* різна доступна функціональність для 2-х різних типів користувачів, а саме: адміністратор та звичайний користувач.

Вимоги до веб-додатку:

* можливість реєстрації користувача;
* валідація вводу користувача.
* заборона входу користувачам що авторизувалися у системі;
* можливість вибору ролі: адміністратора чи звичайного користувача;
* надання доступу до перегляду логів лише для користувачів з роллю адміністратор;
* інтуїтивно зрозумілий інтерфейс для легкого створення, редагування, видалення книг, їх жанрів та клієнтів усім авторизованим користувачам.
* можливість перегляду списку всіх книг , їх жанрів та клієнтів, а також перегляду деталей цих об’єктів.

2.2 Вимоги до середовища виконання

Для забезпечення роботи системи необхідно виконати мінімальні технічні вимоги: персональний комп’ютер з мінімум 8 ГБ оперативної пам’яті, наявність веб-браузера, встановлений maven, сервер баз даних MqSql та платформа Rabbit MQ.

Список програмного забезпечення та бібліотек, що потрібно встановити для коректної роботи системи:

* Java, JDK 8;
* Spring Cloud;
* RabbitMQ;
* MySql
* Maven.

1. Опис ВИДіВ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Технології та засоби розробки системи

Для побудови мікросервісної архітектури була використана технологія Spring Cloud. Усі сервіси системи, зокрема що відповідають за серверну частину, взаємодію з базою даних, зв'язок з клієнтським додатком, власне сервіс для логування об’єктів та створення графічного відображення, були створені на мові програмування Java в середовищі розробки Intellij IDEA. В якості менеджеру пакетів Java використовувався Maven. Також була використана база даних MySql.

3.2 Інформаційне забезпечення системи

В якості системи для зберігання даних була використана система MySql. Це вільна система керування реляційними базами даних, яка являється однією з найбільш популярних СУБД. Вона призначена для роботи з великим обсягом інформації. Є також дуже вживаною у випадках розробки інтернет сайтів та інформаційних систем.

В даній курсовій роботі була створена база даних, що містить наступні таблиці:

* користувачі;
* ролі користувачів;
* книги;
* жанри книг;
* клієнти;
* логи що представляють собою сутність що записує усі зміни інших об’єктів предметної області.

Для логування запитів що змінюють об’єкти предметної області використовується брокер повідомлень RabbitMQ з трьома чергами для запису: створення, редагування та видалення.  
Для додавання запитів у дві останні колекції використовується брокер повідомлень RabbitMQ з двома чергами для запису в кожну колекцію окремо.

Загальна структура бази даних відображена на рисунку 1.

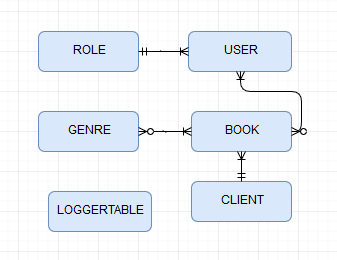


Рисунок 1. Структура бази данних

Для роботи з базою даних в проектах Spring Cloud використовується технологія Spring Data JPA, таким чином, доступ до бази даних надається через абстракцію у вигляди репозиторіїв що дозволяє працювати з об’єктами та їх колекціями замість того щоб працювати напряму з таблицями бази даних. Для зв’язку між сервісами використовувались HTTP запити і REST підхід. Також для зв’язку були використані черги повідомлень.

1. ОПИС сИСТЕМИ
   1. Опис структури та призначення сервісів

Загалом система складається з 6 мікросервісів:

1. Eureka Server;
2. Eureka Client;
3. API Service;
4. Logger;
5. MySql database;
6. RabbitMq message queue provider;

Їх структура представлена на рисунку 2.

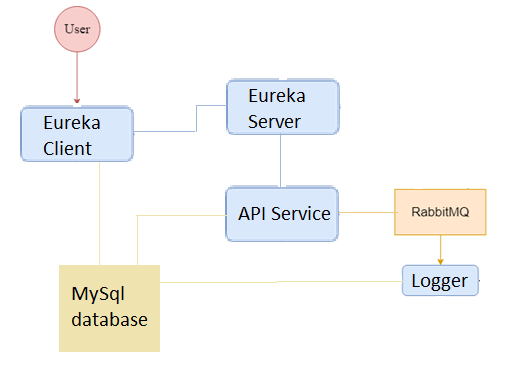


Рисунок 2. Структура мікросервісів

Усі сервіси реалізовані з використанням Java на технології Spring з сервером управління і сервером конфігурації що є частиною функціоналу який було імплементовано за допомогою Spring Cloud.

1й сервіс Eureka Server поєднує всі сервіси бекенд частини. При запуску, інші сервіси реєструються на еврика сервері і таким чином полегшує доступ для комунікації зареєстрованим сервісам між собою. Це можливо завдяки використанню абстракцій Spring Cloud FeignCients та імен сервісів. Також існує можливість використання “load balancing”, тобто розгортання декількох екземплярів клієнту з метою зменшення навантаження на кожен із них.

2й сервіс Eureka Client. Являє собою клієнтську частину, що надає доступ для взаємодії з даними користувачу. Сервіс грає роль точки входу в веб-додаток, а тому містить автентифікацію та авторизацію. Як уже було сказано вище, сервіс реєструється на Eureka Server, отримує дані з API Server за допомогою абстракцій що надаються фреймворком Spring.

3й сервіс Logger займається логуванням запитів до сервісу API Service, що були виконані через клієнтську частину. Запити на логування він отримує від брокера повідомлень RabbitMQ. Повідомлення генерується сервісом API Service, таким чином Logger виступає у ролі консьюмера повідомлень.

4й сервіс API Service. Даний сервіс являє собою АПІ яке надає доступ до бази даних шляхом виконання HTTP запитів. Він має доступ до бази даних, містить у собі всі репозиторії книг, жанрів і клієнтів та має REST-endpoints для роботи з ними. При запитах на отримання даних від повертає JSON та статус код операції. Створені, оновлені чи видалені дані разом з назвою об’єкта та його статусом відправляється до черги RabbitMQ для логування. Таким чином, даний сервіс виступає у ролі продюсера повідомлень.

4.2 Опис функціонування системи

Під час розробки, малось на увазі що користувач при роботі з кінцевим веб-додатком буде робити це наступним чином:

Після відкриття веб-сторінки інформаційної системи користувач потрапить на стартову сторінку. Тут користувачу необхідно зробити вибір: зареєструватися або увійти у систему. Якщо це перший вхід користувача, то необхідно обрати опцію реєстрації та виконати інструкції на екрані: вказати ім’я користувача, пароль та роль. Якщо користувач уже зареєстрований в системі то необхідно обрати опцію входу та, слідуючи інструкціям, вказати свій логін та пароль.

Коли процес реєстрації або автентифікації успішно завершено, користувач отримує доступ до усього функціоналу що доступний для його ролі. Звичайний користувач має можливість переглядати усі наявні книги, жанри та клієнтів, а також деталі кожного об’єкта окремо. Також існує функціонал що дозволяє створювати, редагувати та видаляти ці об’єкти. Користувачу з роллю адміністратор доступні всі ті самі сторінки що і звичайному користувачу, а також сторінка перегляду записів журналу логів.

Користувач має змогу вийти зі свого акаунту за допомогою кнопки вихід що розташована на верхній частині будь-якої сторінки. Після виходу, доступ до всього функціоналу буде втрачено до моменту повторної авторизації через форму входу.

Приклади використання програми наведені у додатку А.

ВИСНОВКИ

Курсова робота була виконана з виконанням усіх поставлених вимог що були описані в відповідному розділі даної пояснювальної записки. Вимоги до побудованої інформаційної системи були виконані у повному обсязі. Результатом став побудований на базі мікросервісної архітектури та фреймворку Spring інформаційна система «Магазин книг».

За час роботи над додатком було набуто нових та поліпшені старі практичні навички розробки сучасного програмного забезпечення. Особливо необхідно виділити отримані навички розробки сервісів що взаємодіють між собою через стандартизовані протоколи зв’язку.

Результатом є додаток що повністю відповідає поставленому завданню та готовий для поліпшення та подальшого розширення функціоналу в майбутньому.

ДОДАТОК А

Скріншоти роботи програми

