Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра теорії та технології програмування

Звіт

з виробничої практики студента 3 курсу, групи ТТП-32

Ходакова Максима Олеговича

Під час проходження виробничої практики на базі факультету комп’ютерних наук та кібернетики було спроектовано та розроблено систему управління зоопарками. Система побудована на платформі Java з використанням таких технологій як: Java Spring Boot, Spring MVC, Spring Security, String Data, Thymeleaf, MariaDB, Apache Kafka та RabbitMQ.

Система побудована як багатомодульний (мікросервісний) проект з трьома окремими сервісами з асинхронною комунікацією через брокери повідомлень.

Почнемо з огляду zoo-module:

Цей модуль надає доступ для управління самими сутностями предметної області з забезпеченням безпеки.

Вигляд форми логіну

A screen shot of a login form

Description automatically generated

Валідація на існування користувача

A screenshot of a login page

Description automatically generated

Вигляд форми реєстрації

A screen shot of a computer

Description automatically generated

Валідація перед реєстрацією

A screenshot of a register

Description automatically generated

Після успішної реєстрації переходимо до сторінки логіну

A screenshot of a login page

Description automatically generated

При успішному логіні користувача зустрічає домашня сторінка з привітанням

A screenshot of a computer

Description automatically generated

Далі надається можливість управління країнами в системі разом зі статистикою по ним

A screenshot of a computer

Description automatically generated

Також надається можливість додавання та оновлення країниA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Далі користувач може створити та переглядати наявні зоопарки, створення нового зоопарку можливий за наявності країни.

A screenshot of a phone

Description automatically generated

A screenshot of a zoo

Description automatically generated

Також можемо оброблювати тварин які наявні в системі до конкретних зоопарків.

Видалення зоопарку каскадно видаляє всіх тваринок а видалення країни видаляє всі наявні в країні зоопарки.

Управління тваринами є технологічної основою проекту, створити тварину можна вручну, приклад:  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

На основній сторінці наявна статистика по тваринкам, JQuery search по назві та можливість оновити/видалити дані та оновити по країнам информацію.

A screenshot of a computer

Description automatically generated

Додавання Many - To - Many зв’язок між країною та твариною.

Оновлення основних даних по тварині.

A screenshot of a phone

Description automatically generated

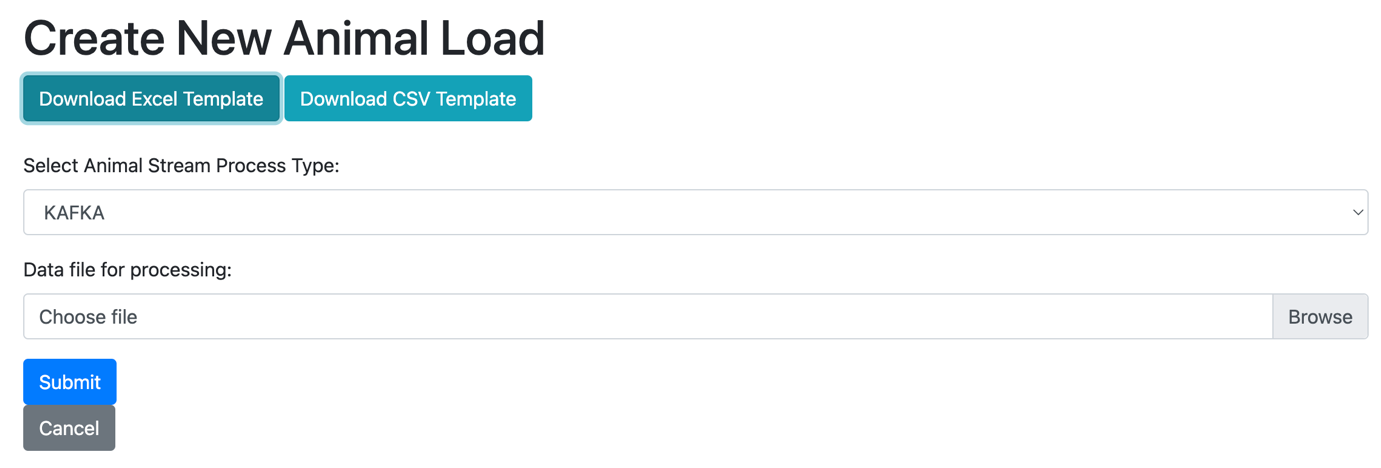
Створення тварин може бути масовим через файл, тут вступають в дію мікросервіси producer-module та consumer-module.

Тут користувач може бачити завантаженні файли з тваринами

A screenshot of a computer

Description automatically generated

Користувач може завантажити ексель темплейт файлу, обрати тип комунікацій між мікросервісами для аналізу або Kafka або RabbitMQ для комунікації, підтримують для обох Retryable патерн разом з Dead Letter Topic/Dead Letter Queue, тобто в разі помилки споживача load видаляється окремою scheduled джобою, а опрацьовується в разі успіху теж окремою крон джобою. Також опрацьовуєтьсся за допомогою окремих ендпоїнтів під Kafka та під RabbitMQ відповідно.



Після завантаження отримали нову статистику та індикатор Processed No це значить що consumer-module ще не провалідував дані з файлу

A screenshot of a stream load

Description automatically generated

A screenshot of a computer

Description automatically generated

Коли індикатор Yes це означає що система опрацювала файл даний, побачимо результати:

Тут ми бачимо данні які не пройшли валідацію та причини цього. Також надається можливість видалення з бази цієї інформації

A screenshot of a computer

Description automatically generated

Також наведемо приклади обробки помилок через DLT/DLX через відповідне API.

A screenshot of a computer

Description automatically generated

Видалили невалідний animal

A screenshot of a phone

Description automatically generated

При цьому валідна тварина без фотографії була завантажена

A screenshot of a computer

Description automatically generated

Як бачимо мікросервіси системи працюють злагоджено.

Система окрім UI інтерфейсу має REST API яке задокументоване та захищене протоколом OAuth2 задля безпеки користування.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Також наявні ендпоїнти для забезпечення безпеки за протоколом OAuth2:

A screenshot of a computer

Description automatically generated

Тепер producer-module ендпоїнти:

A screenshot of a computer

Description automatically generated

Всі ці ендпоїнти надають можливість не обмежуватися доступом до UI а розцінювати цю мікросевісну систему як сторонній сервіс для інших програм з потребую в цих даних. Також система має таку структуру реляційної БД (в даному випадку MariaDB):

A screenshot of a computer

Description automatically generated

Висновок  
 Була розроблена розподілена система управління зоопарками – тваринами – країнами з застосуванням відмово стійких, асинхронних та безпечних технологій з покриттям Unit тестами та використанням брокерів повідомлень, REST API, MVC та сучасних найкращих практик, дизайн патернів та розробки корпоративного програмного забезпечення.