

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Бизнес логика программных систем»

Лабораторная работа №2

Вариант 1108

Студент

Патутин Владимир

Иннокентьев Артем

Р33101

Преподаватель

Цопа Е. А.

Санкт-Петербург, 2022 г.

Задание лабораторной работы

Доработать приложение из лабораторной работы #1, реализовав в нём управление транзакциями и разграничение доступа к операциям бизнес-логики в соответствии с заданной политикой доступа.

Управление транзакциями необходимо реализовать следующим образом:

1. Переработать согласованные с преподавателем прецеденты (или по согласованию с ним разработать новые), объединив взаимозависимые операции в рамках транзакций.
2. Управление транзакциями необходимо реализовать с помощью Spring JTA.
3. В реализованных (или модифицированных) прецедентах необходимо использовать программное управление транзакциями.
4. В качестве менеджера транзакций необходимо использовать Atomikos.

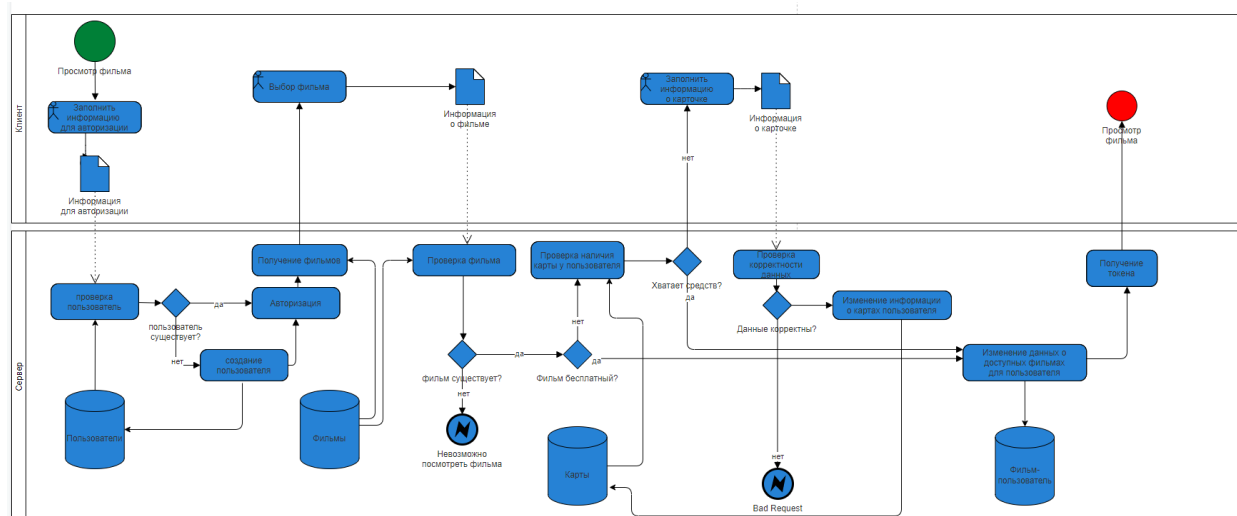
Разграничение доступа к операциям необходимо реализовать следующим образом:

1. Разработать, специфицировать и согласовать с преподавателем набор привилегий, в соответствии с которыми будет разграничиваться доступ к операциям.
2. Специфицировать и согласовать с преподавателем набор ролей, осуществляющих доступ к операциям бизнес-логики приложения.
3. Реализовать разработанную модель разграничений доступа к операциям бизнес-логики на базе Spring Security. Информацию об учётных записях пользователей необходимо сохранять в реляционную базу данных, для аутентификации использовать JWT.

Правила выполнения работы:

1. Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.
2. Доработанное приложение необходимо развернуть на сервере **helios**.

Модель бизнес-процесса «Редактирование и проверка статьи»

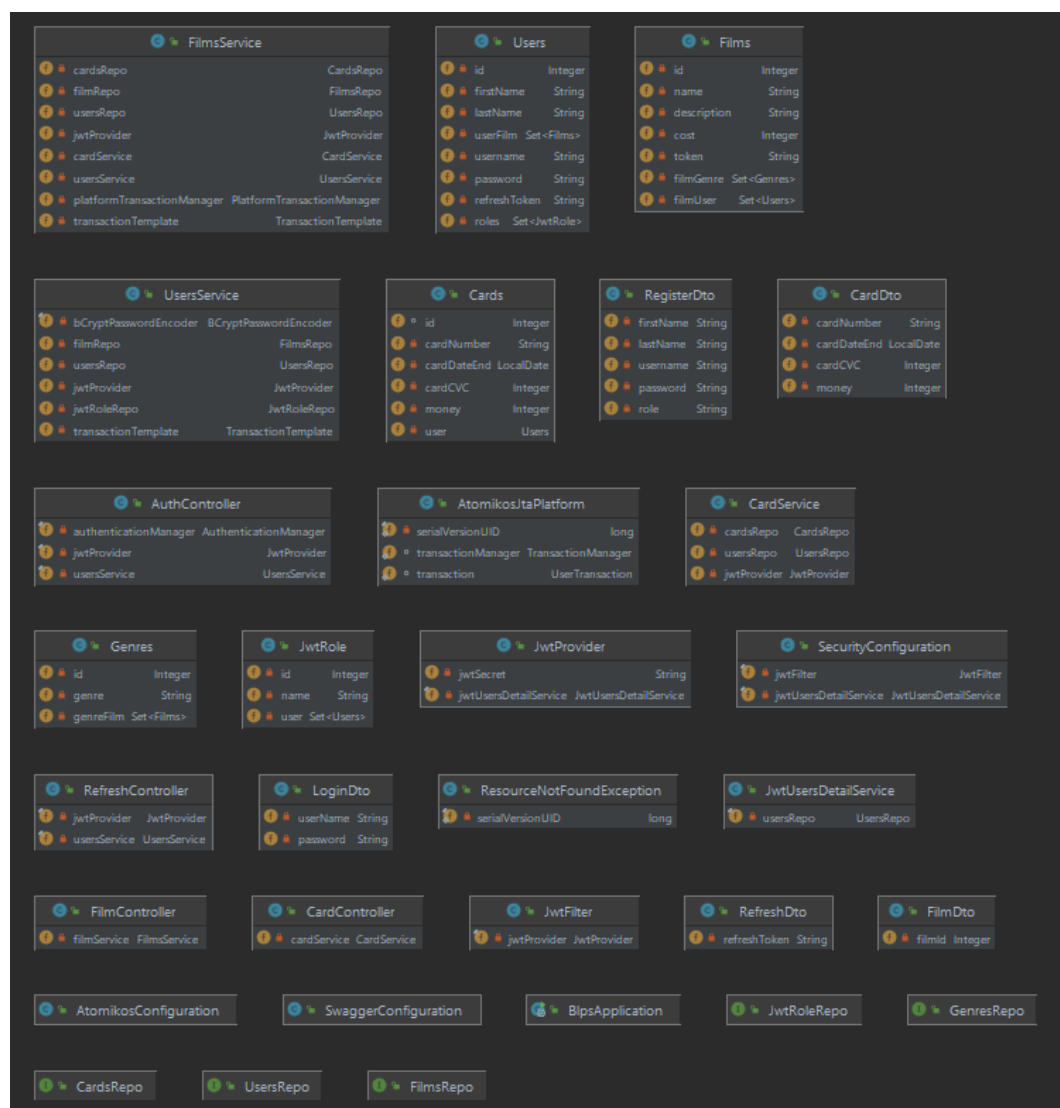


Спецификация пользовательских привилегий и ролей

В разработанном приложении реализована одна роль – пользователь.

Пользователь имеет доступ к просмотру фильмов и их покупке

UML-Диаграмма классов



Спецификация REST API разработанного приложения

POST /card/addCard add card

Parameters Try it out

Name	Description
data * required object (body)	data Example Value Model <pre>{ "cardCVC": 0, "cardDateEnd": "2022-03-24", "cardNumber": "string", "money": 0, "userId": 0 }</pre> Parameter content type application/json

POST /youtube/addUser add user

Parameters Try it out

Name	Description
data * required object (body)	data Example Value Model <pre>{ "phoneNumber": "string" }</pre> Parameter content type application/json

POST /youtube/allFilms get all films

Parameters Try it out

No parameters

POST /youtube/selectFilm get select film

Parameters Try it out

Name	Description
data * required object (body)	data Example Value Model <pre>{ "filmId": 0, "userId": 0 }</pre> Parameter content type application/json

Исходный код

<https://github.com/DeltaHeavyVIP/Business-logic-software-systems>

Вывод

Во время выполнения лабораторной работы мы изучили реализацию транзакций на уровне бизнес-логики. Ознакомились с работой менеджеров транзакций в Spring Boot приложениях.

Кроме того, мы использовали Spring Security для разграничения доступа к данным и использовали JWT для реализации авторизации и аутентификации.