НИУ ИТМО

Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе №2

по дисциплине Бизнес-логика программных систем

|  |  |
| --- | --- |
| Студент группы № P33151 | Шипулин Павел Андреевич |
| Преподаватель | Кривоносов Егор Дмитриевич |

Санкт-Петербург

2024

# Задание (вариант 1111)

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 декларативное управление транзакциями.
4. В качестве менеджера транзакций необходимо использовать Java EE JTA, предварительно преобразовав приложение в war, развёртываемый на сервере приложений WildFly.

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 + JAAS. Информацию об учётных записах пользователей необходимо сохранять в файле XML, 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.

Содержание отчёта:

1. Текст задания.
2. Модель потока управления для автоматизируемого бизнес-процесса.
3. Спецификация пользовательских привилегий и ролей, реализованных в приложении.
4. UML-диаграммы классов и пакетов разработанного приложения.
5. Спецификация REST API для всех публичных интерфейсов разработанного приложения.
6. Исходный код системы или ссылка на репозиторий с исходным кодом.
7. Выводы по работе.

# Заметки

## Согласование прецендентов (для транзакций)

1. Адекватно настроить транзакцию для обращения к банку. Несмотря на то, что сервер приложения обращается к банку по публичной ссылке – защищать этот процесс не нужно, так как банк является условностью.

## Согласование ролей

+ сделать

? может быть

?? сомнительно

{ Комментарий преподавателя }

1. Публично (не требует токена):
   1. + Поиск курсов по имени.
   2. + Зарегистрироваться.
2. Клиент (требует токен):
   1. + Выйти из системы.
   2. Записаться на курс (оплата через систему).
   3. +?? Настройка профиля [ информация о профиле может лежать кусками в разных таблицах – банковская карта, ФИО, дата рождения… ] { Необязательно }
3. Админ (требует токен):
   1. + Выйти из системы.
   2. Запрос в банк для оплаты курса клиентом. (транзакция)
   3. + CRU, блокировать курс
   4. + (Раз-)блокировать клиентов

# Выполнение