

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования «Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа №3
по дисциплине «Бизнес логика программных систем»

Группа: Р3312

Выполнили: Балин А. А., Кобелев Р. П.

Проверил: Бобрусь А. В.

Оглавление

Задание	3
Исходный код	4
Модель потока управления:	5
UML-диаграммы разработанных классов	6
Диаграмма развёртывания	7
Спецификация REST API для всех публичных интерфейсов разработанного приложения	8
Вывод	9

Задание

Доработать приложение из лабораторной работы #2, реализовав в нём асинхронное выполнение задач с распределением бизнес-логики между несколькими вычислительными узлами и выполнением периодических операций с использованием планировщика задач, а также интеграцию с внешней информационной системой.

Требования к реализации асинхронной обработки:

Перед выполнением работы необходимо согласовать с преподавателем набор прецедентов, в реализации которых целесообразно использование асинхронного распределённого выполнения задач. Если таких прецедентов использования в имеющейся бизнес-процесса нет, нужно согласовать реализацию новых прецедентов, доработав таким образом модель бизнес-процесса из лабораторной работы #1.

Асинхронное выполнение задач должно использовать модель доставки "очередь сообщений".

В качестве провайдера сервиса асинхронного обмена сообщениями необходимо использовать очередь сообщений на базе Apache ActiveMQ.

Для отправки сообщений необходимо использовать протокол OpenWire с использованием библиотеки Java OpenWire Transport.

Для получения сообщений необходимо использовать слушателя сообщений JMS на базе Spring Boot (@JmsListener).

Требования к реализации распределённой обработки:

Обработка сообщений должна осуществляться на двух независимых друг от друга узлах сервера приложений.

Если логика сценария распределённой обработки предполагает транзакционность выполняемых операций, они должны быть включены в состав распределённой транзакции.

Требования к реализации запуска периодических задач по расписанию:

Согласовать с преподавателем прецедент или прецеденты, в рамках которых выглядит целесообразным использовать планировщик задач. Если такие прецеденты отсутствуют -- согласовать с преподавателем новые и добавить их в модель автоматизируемого бизнес-процесса.

Реализовать утверждённые прецеденты с использованием планировщика задач Spring (@Scheduled).

Требования к интеграции с внешней Корпоративной Информационной Системой (EIS):

Корпоративная Информационная Система, с которой производится интеграция, а также её функциональные возможности выбираются на усмотрение преподавателя и согласуются с ним.

Взаимодействие с внешней Корпоративной Информационной Системой должно быть реализовано с помощью технологии JCA (Jakarta Connectors).

Правила выполнения работы:

Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.

Доработанное приложение необходимо либо развернуть на сервере helios, либо продемонстрировать его работоспособность на собственной инфраструктуре обучающегося.

Исходный код

<https://github.com/Romariok/BLPS>

Модель потока управления:

Можно посмотреть на сайте <https://demo.bpmn.io/>, загрузив файл docs/diagram3.bpmn с репозитория.

UML-диаграммы разработанных классов

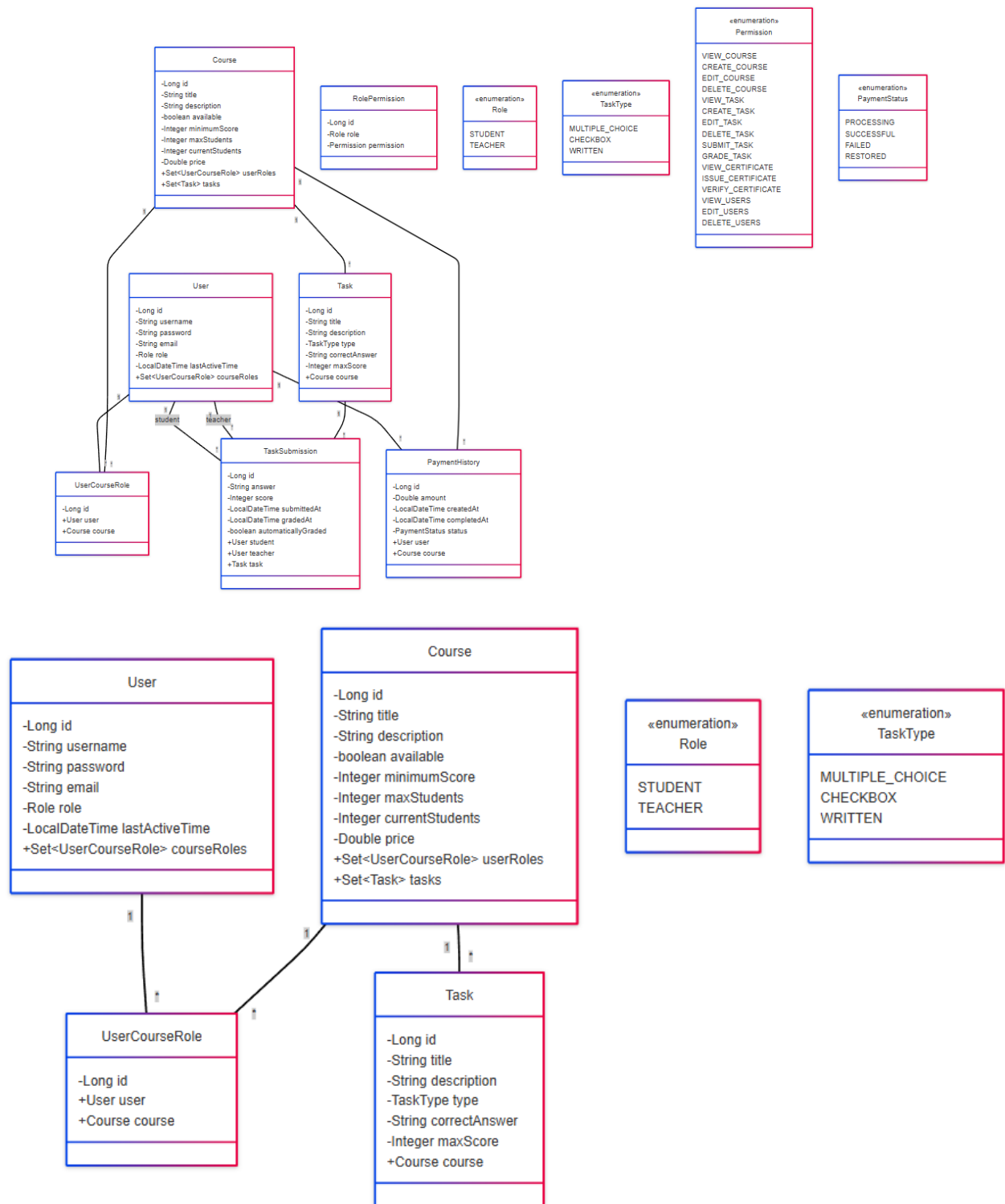
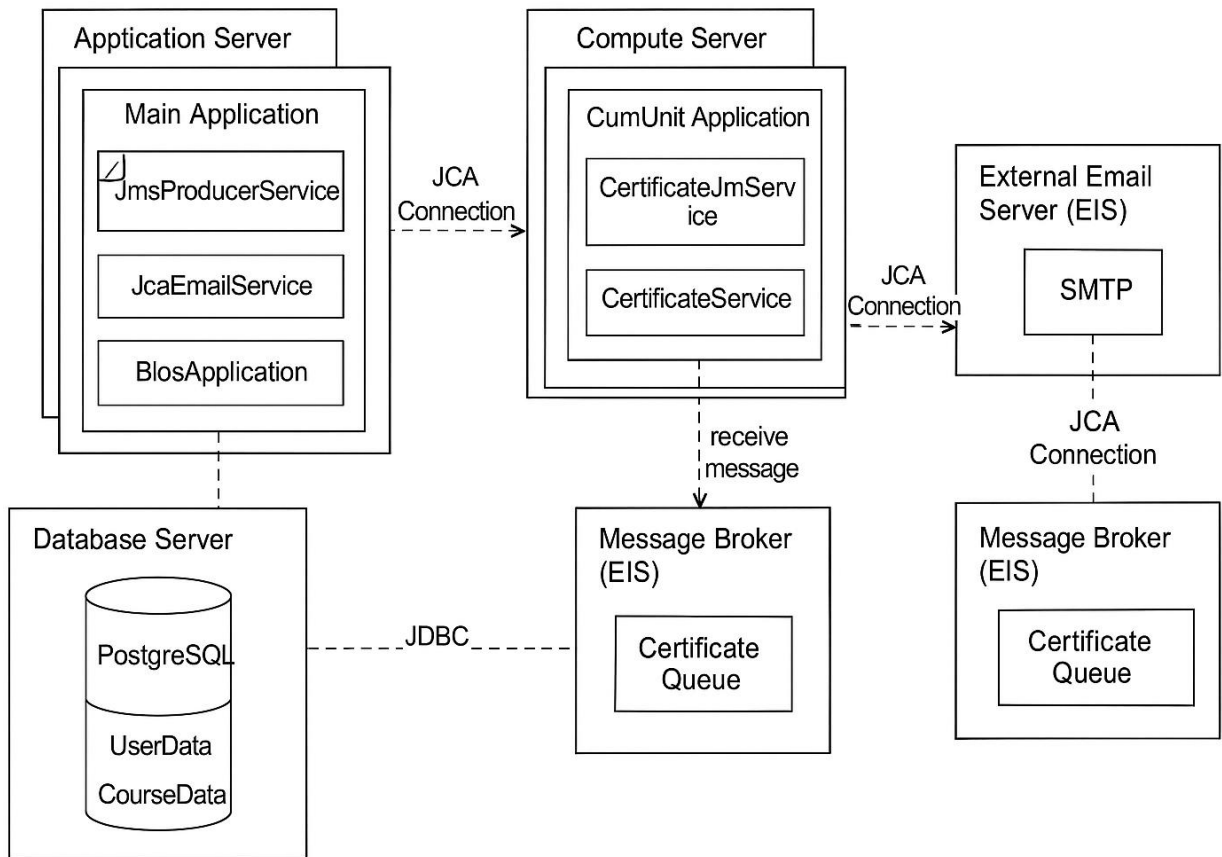


Диаграмма развёртывания



Спецификация REST API для всех публичных интерфейсов разработанного приложения

Ничего нового с прошлой ЛР.

Вывод

В ходе выполнения познакомились с JCA, потрогали JMS, отправляли приколы по почте.