Университет ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники

# Лабораторная работа №3

по дисциплине

# “Бизнес-логика программных систем”

Вариант: 6062

Группа: Р33112

Выполнили: Атикеев Роман,

Кульбако Артемий Юрьевич

Преподаватель: Каюков Иван Алексеевич

Санкт-Петербург

2021 год

## 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.

**Требования к реализации асинхронной обработки:**

1. Перед выполнением работы неободимо 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.
2. Асинхронное выполнение задач должно использовать модель доставки "очередь сообщений".
3. В качестве провайдера сервиса асинхронного обмена сообщениями необходимо использовать очередь сообщений на базе RabbitMQ.
4. Для отправки сообщений необходимо использовать протокол MQTT. Библиотеку для реализации отправки сообщений можно взять любую на выбор студента.
5. Для получения сообщений необходимо использовать слушателя сообщений JMS на базе Spring Boot (@JmsListener).

**Требования к реализации распределённой обработки:**

1. Обработка сообщений должна осуществляться на двух независимых друг от друга узлах сервера приложений.
2. Если логика сценария распределённой обработки предполагает транзакционность выполняемых операций, они должны быть включены в состав распределённой транзакции.

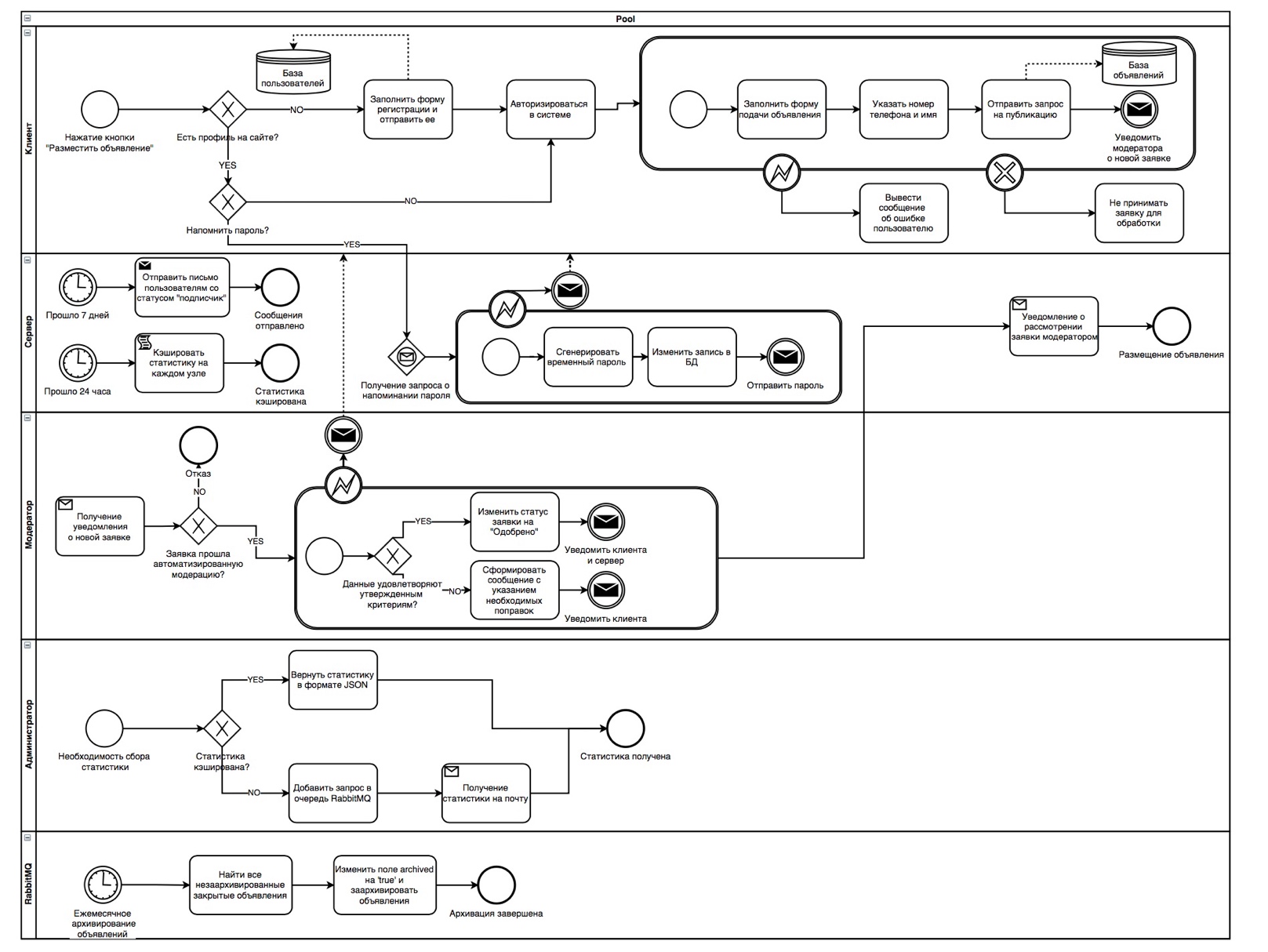
**Требования к реализации запуска периодических задач по расписанию:**

1. 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.
2. Реализовать утверждённые прецеденты с использованием планировщика задач Spring (@Scheduled).

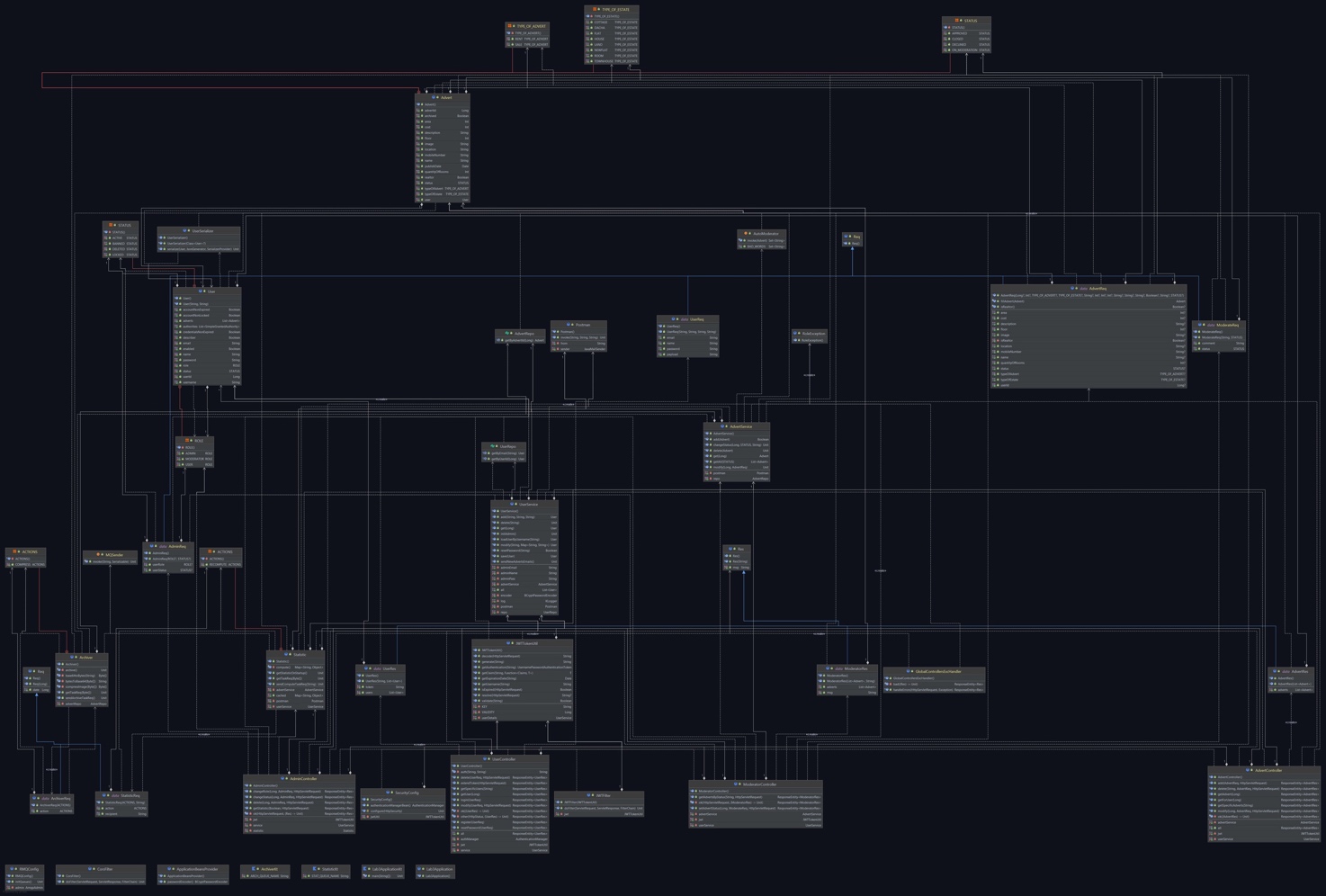
**Правила выполнения работы:**

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

## Модель потока управления



## UML-диаграмма



## REST APIИзображение выглядит как текст Автоматически созданное описаниеИзображение выглядит как текст Автоматически созданное описание

## 

## Исходный код и развертывание

Исходный код программы можно найти в репозитории на github:

https://github.com/testpassword/Business-logic-of-software-systems

## 

## Вывод

В данной лабораторной работе мы познакомились с механизмом работы асинхронного выполнения задач, а также с выполнением периодических операций использованием планировщика задач используя соответствующую аннотацию “@Scheduled”. Помимо этого в процессе выполнения данной лабораторной работы были разработаны REST API, а также была выполнена непосредственно реализация самого бизнес-процесса с помощью языка kotlin.