

**Федеральное государственное автономное образовательное  
учреждение  
высшего образования  
«Национальный исследовательский университет ИТМО»  
Факультет Программной Инженерии и Компьютерной Техники**



**Лабораторная работа №4  
по дисциплине  
Бизнес-логика программных систем**

Выполнили Студенты группы Р3312  
**Кобелев Р.П. Балин А.А.**  
к. т. н. Преподаватель:  
**Бобрусь А.В.**

г. Санкт-Петербург  
2025г.

# Содержание

<b>1</b>	<b>Задание</b>	<b>2</b>
<b>2</b>	<b>Модель потока управления для автоматизируемого бизнес-процесса со всеми внесёнными изменениями</b>	<b>2</b>
2.1	Отправка ответа на задание . . . . .	2
2.2	Проверка JWT токена . . . . .	3
2.3	Подача заявки на получение сертификата . . . . .	3
2.4	Поступление на курс . . . . .	4
2.5	Получение списка действующих заявок на получение сертификата . . . . .	5
2.6	Вход в аккаунт . . . . .	5
2.7	Выставление оценки на ответ студента . . . . .	6
2.8	Учитель принимает решение о выдаче сертификата студенту . . . . .	6
2.9	Просмотр статуса по заявке на получение сертификата . . . . .	7
2.10	Просмотр всех отправленных ответов на задание . . . . .	7
2.11	Просмотр информации по заданию . . . . .	8
2.12	Просмотр всех неоценённых заданий . . . . .	9
2.13	Уведомления пользователям и учителям . . . . .	10
2.13.1	Проверка активности пользователя . . . . .	11
<b>3</b>	<b>Блок-схема архитектуры приложения с указанием ”точек интеграции” ВРМ-фреймвока с остальными подсистемами.</b>	<b>12</b>
<b>4</b>	<b>Исходный код</b>	<b>13</b>
<b>5</b>	<b>Вывод</b>	<b>13</b>

# 1 Задание

Переработать программу, созданную в результате выполнения лабораторной работы #3, следующим образом:

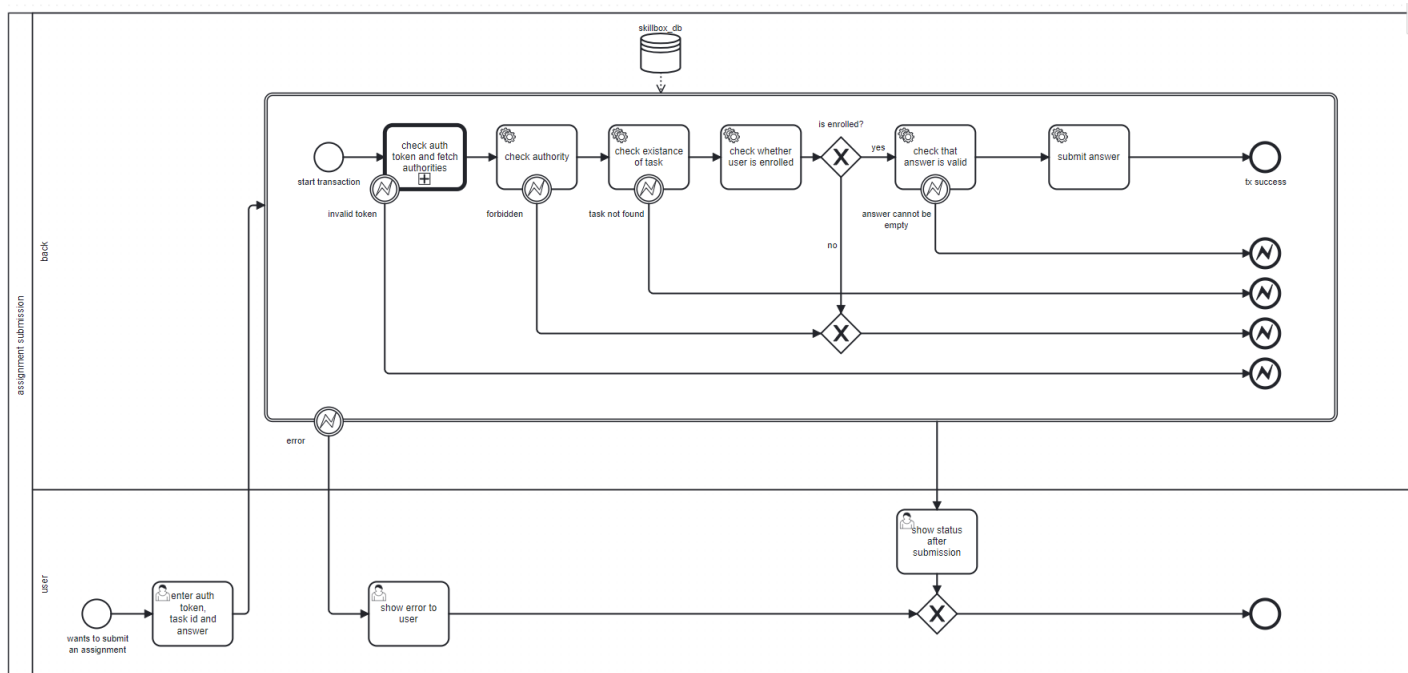
1. Для управления бизнес-процессом использовать BPM-движок Camunda.
2. Заменить всю "статическую" бизнес-логику на "динамическую" на базе BPMS. Весь бизнес-процесс, реализованный в ходе выполнения предыдущих лабораторных работ (включая разграничение доступа по ролям, управление транзакциями, асинхронную обработку и периодические задачи), должен быть сохранён!.
3. BPM-движок должен быть встроен в веб-приложение (embedded mode).
4. Для описания бизнес-процесса необходимо использовать онлайн-сервис BPMN.io.
5. Пользовательский интерфейс приложения должен быть сгенерирован с помощью генератора форм Camunda.
6. Итоговая сборка должно быть развёрнута на сервере helios под управление сервера приложений Apache Tomcat.

## Правила выполнения работы:

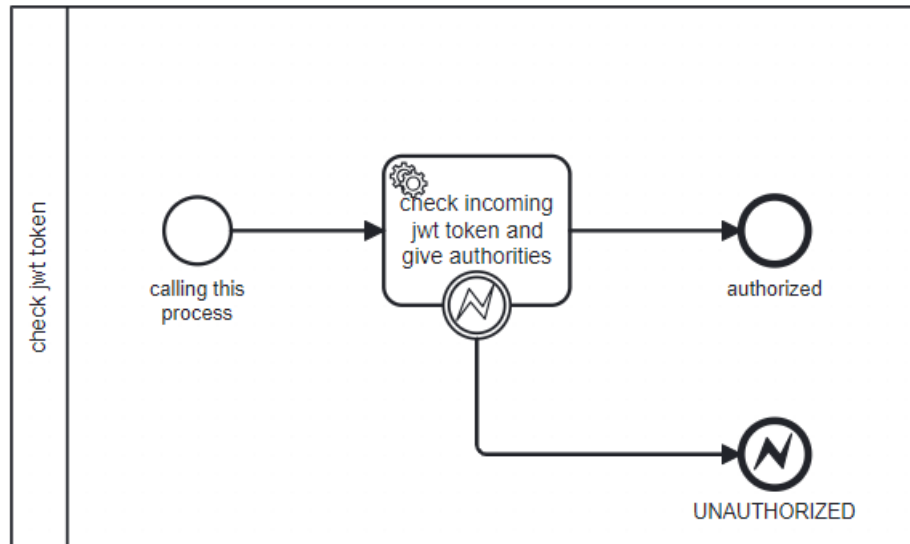
1. Описание бизнес-процесса необходимо реализовать на языке BPMN 2.0.
2. Необходимо интегрировать в состав процесса, управляемого BPMS, всё, что в принципе возможно в него интегрировать. Если какой-то из компонентов архитектуры приложения (например, асинхронный обмен сообщениями с помощью JMS) не поддерживается, необходимо использовать для интеграции с этой подсистемой соответствующие API и адаптеры.
3. Распределённую обработку задач и распределённые транзакции на BPM-движок переносить не требуется.

# 2 Модель потока управления для автоматизируемого бизнес-процесса со всеми внесёнными изменениями

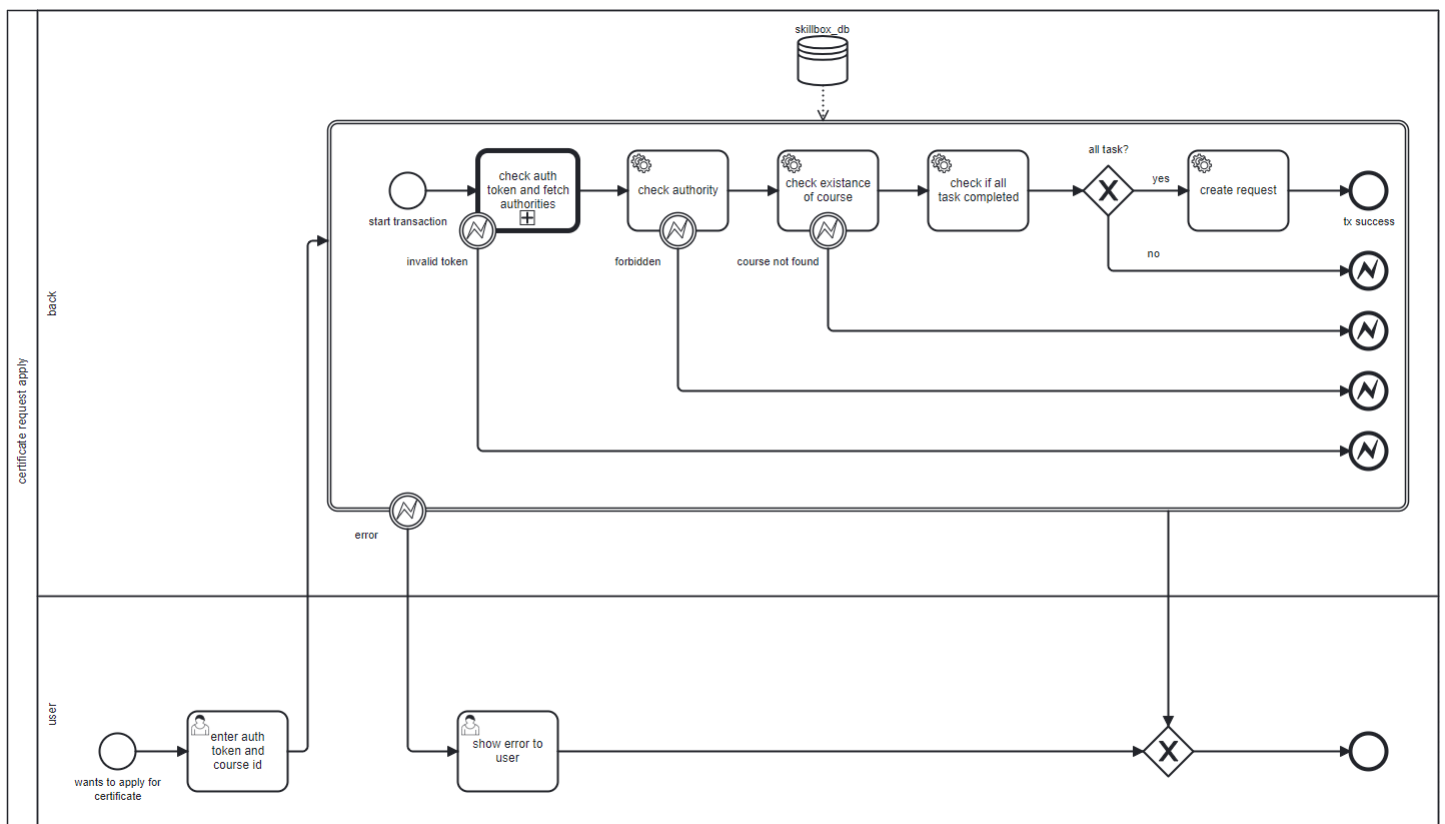
## 2.1 Отправка ответа на задание



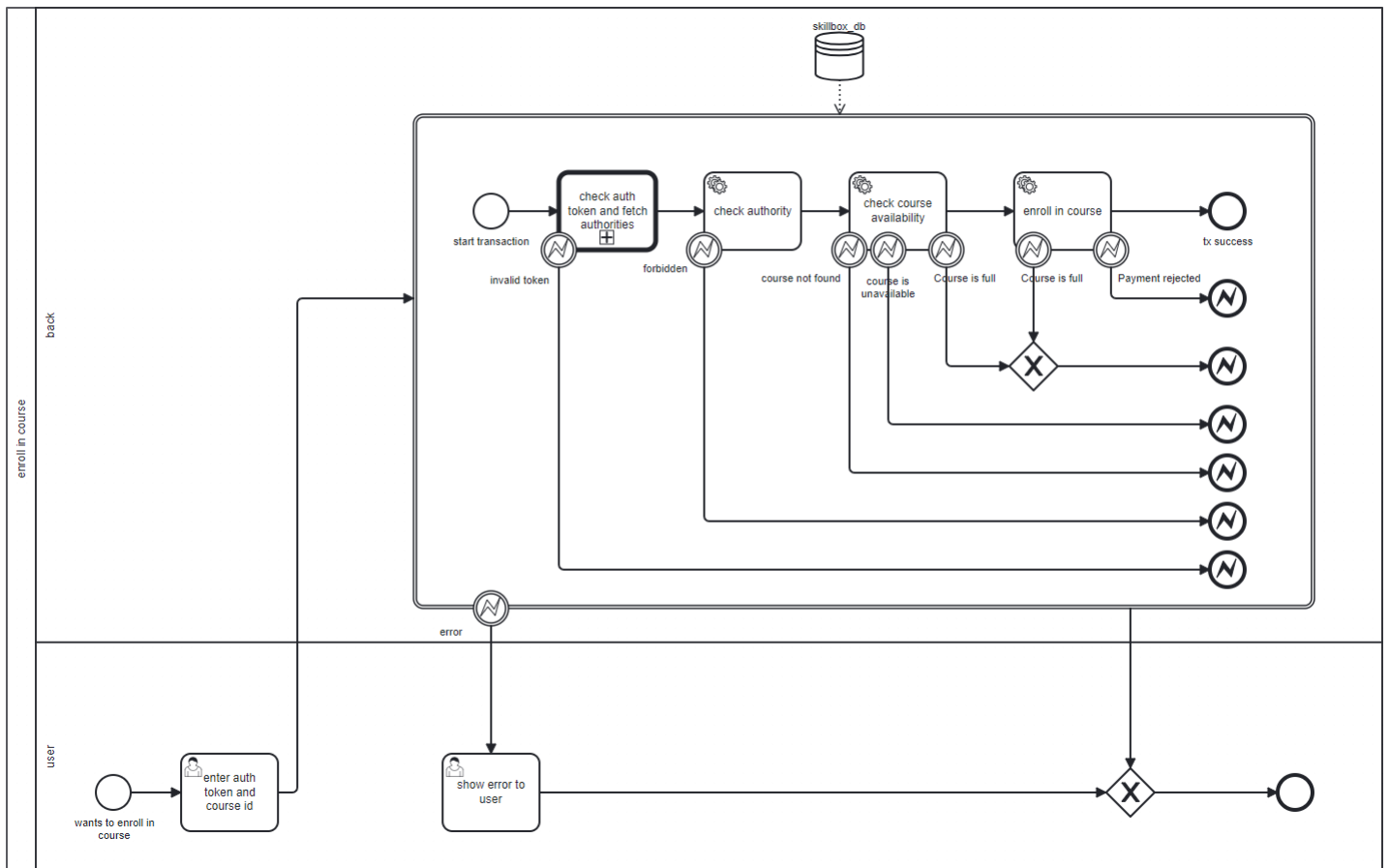
## 2.2 Проверка JWT токена



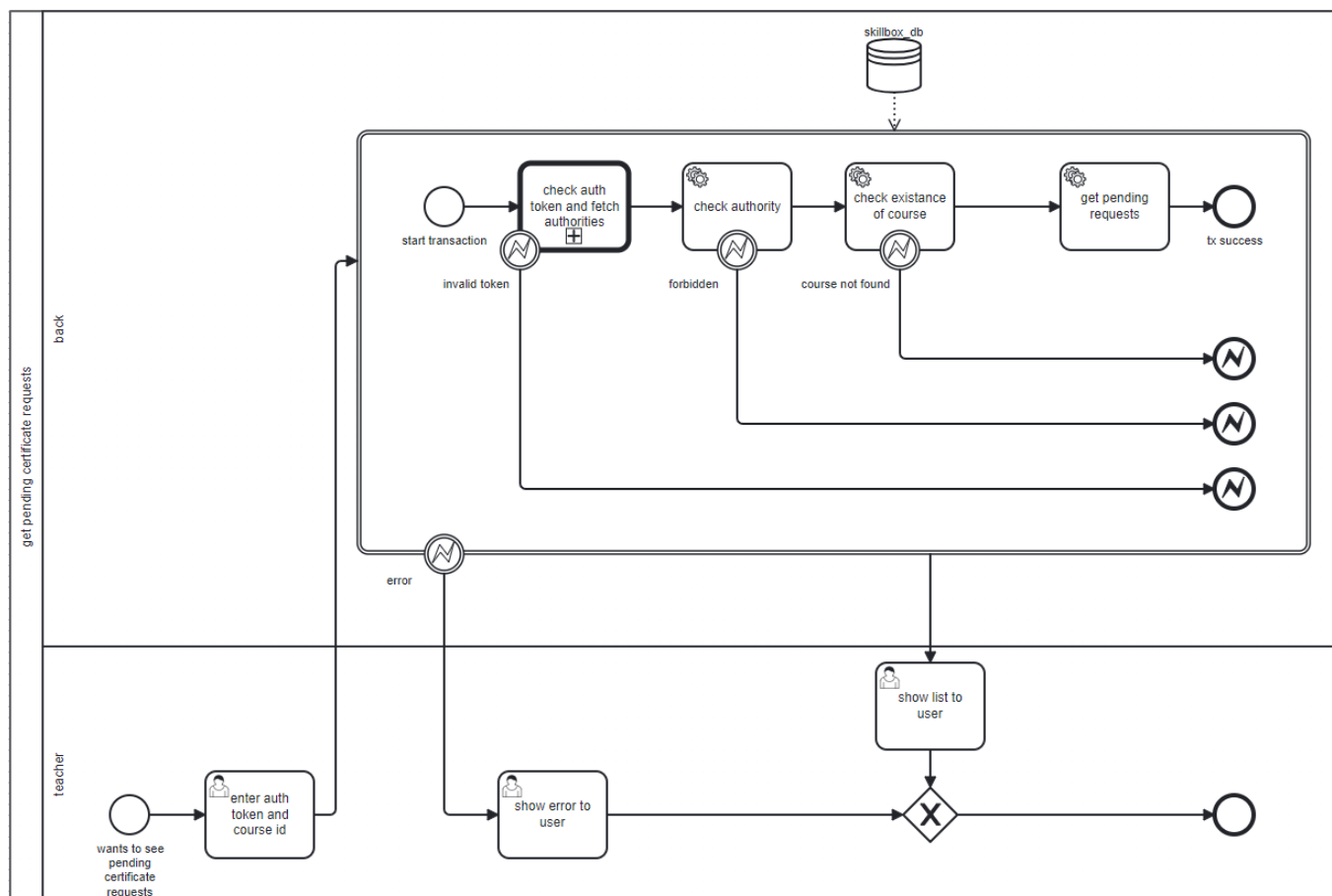
## 2.3 Подача заявки на получение сертификата



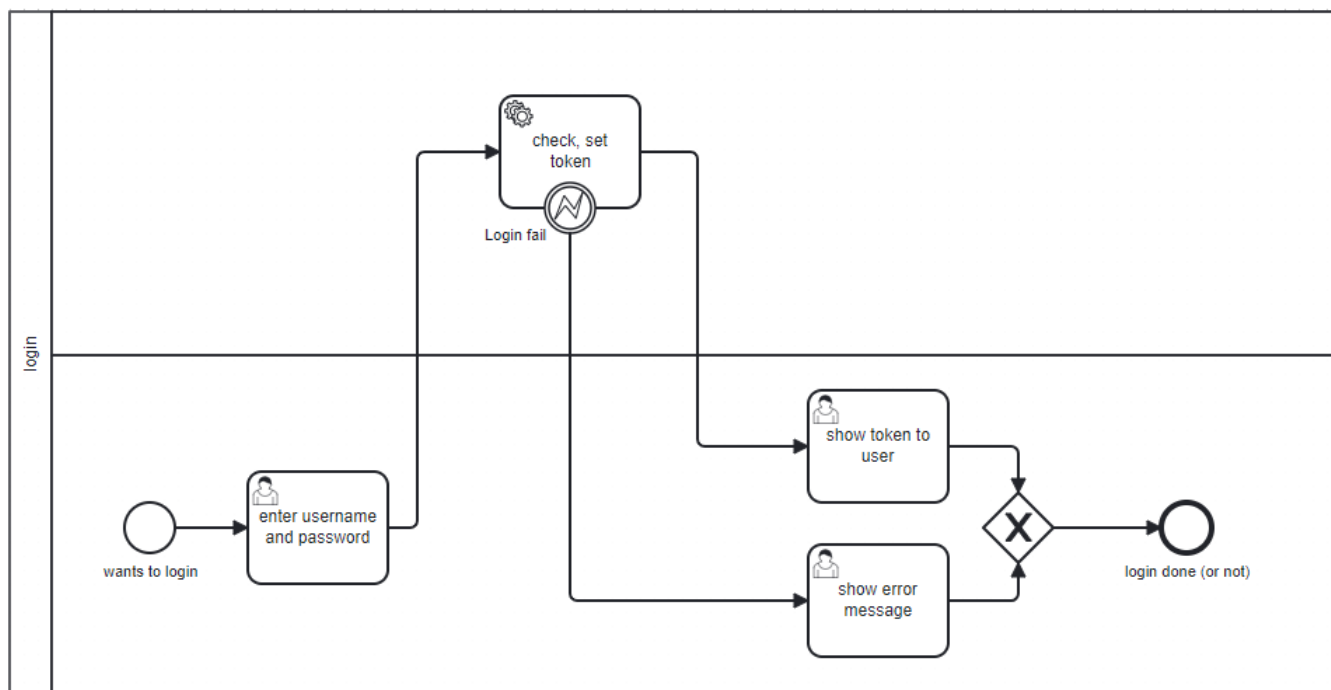
## 2.4 Поступление на курс



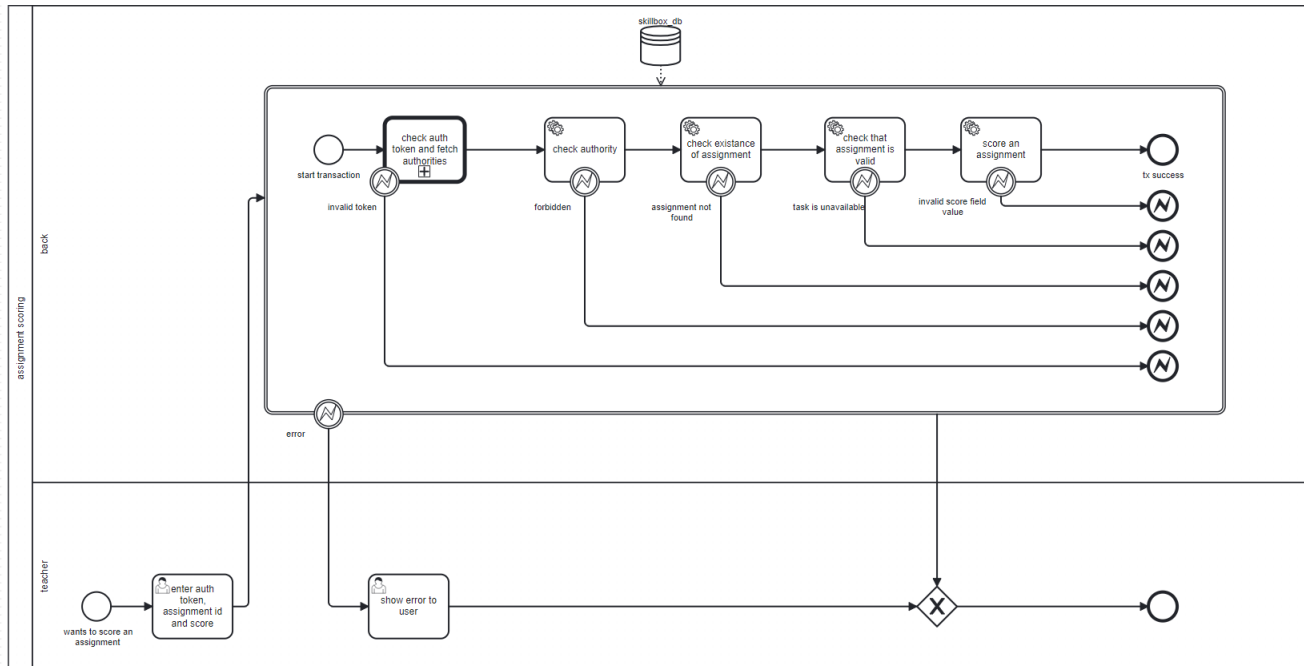
## 2.5 Получение списка действующих заявок на получение сертификата



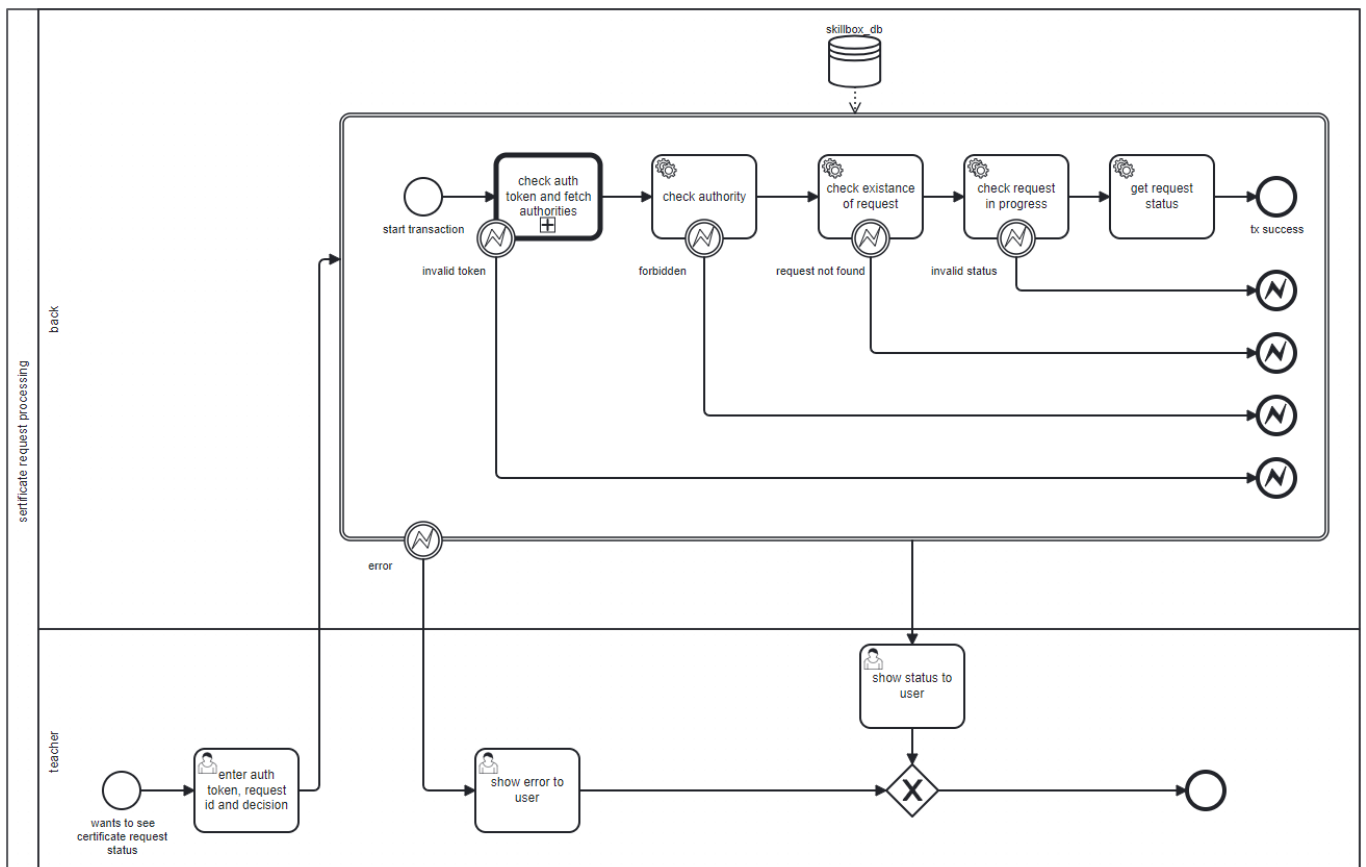
## 2.6 Вход в аккаунт



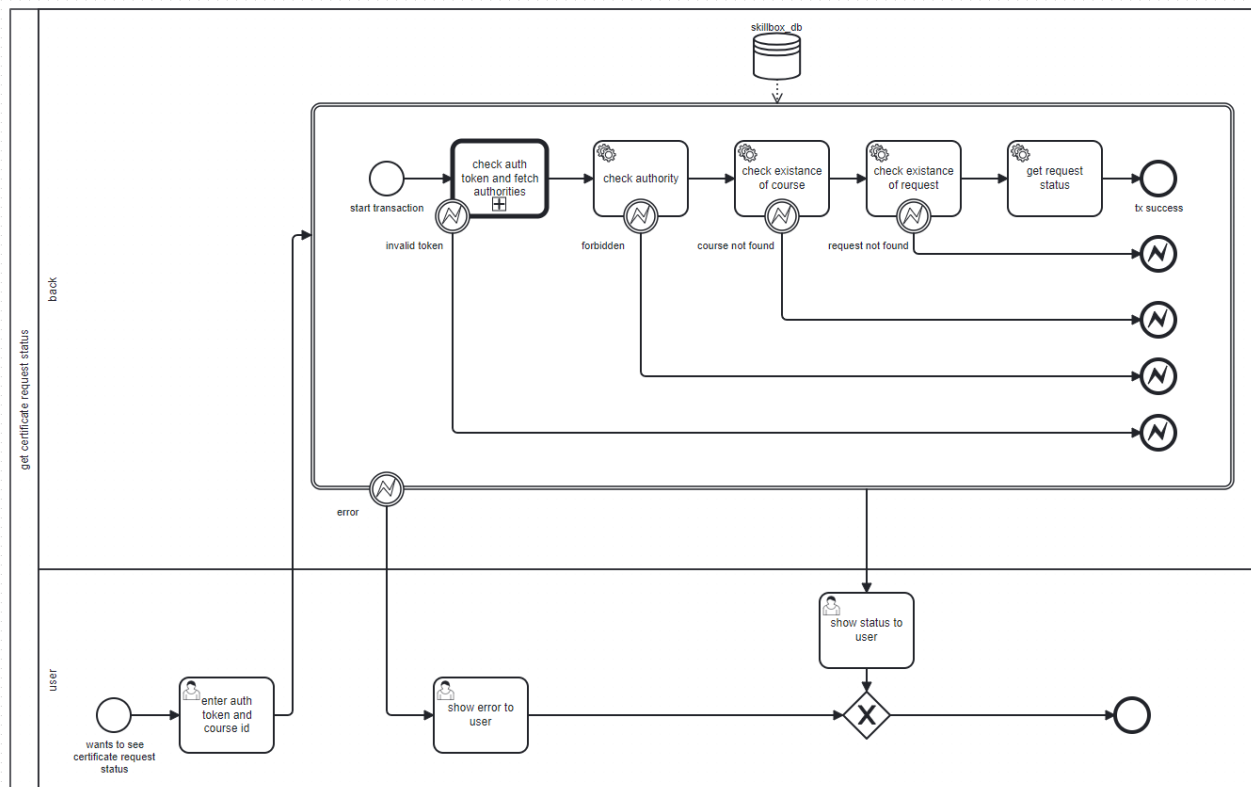
## 2.7 Выставление оценки на ответ студента



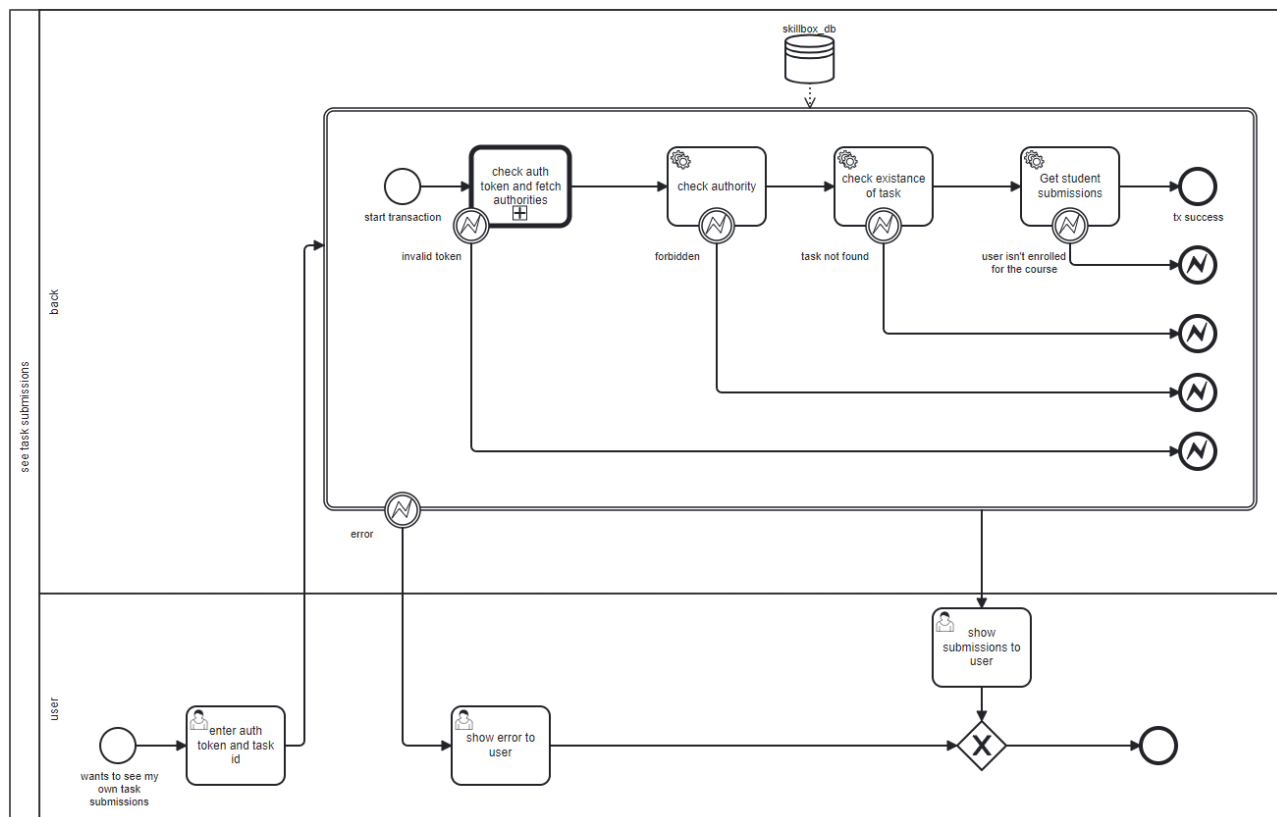
## 2.8 Учитель принимает решение о выдаче сертификата студенту



## 2.9 Просмотр статуса по заявке на получение сертификата

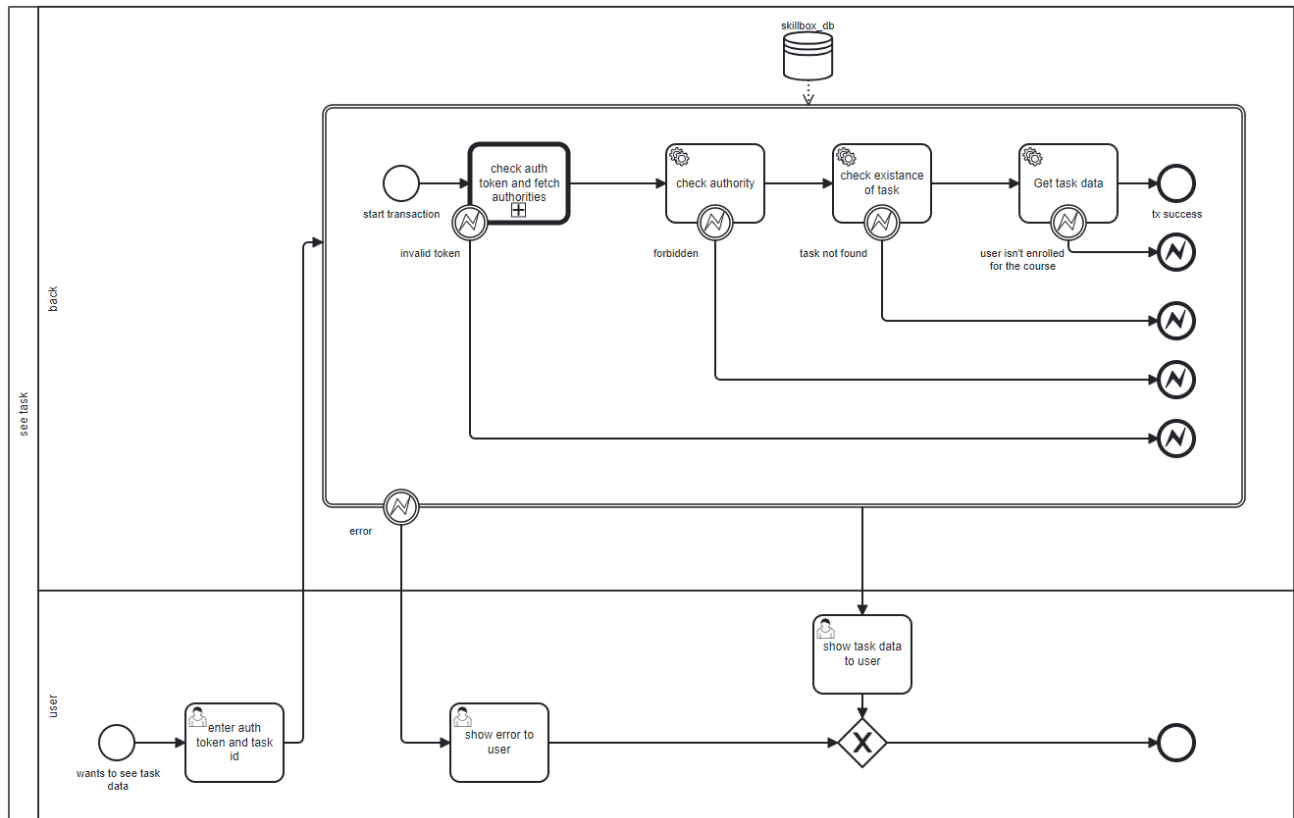


## 2.10 Просмотр всех отправленных ответов на задание

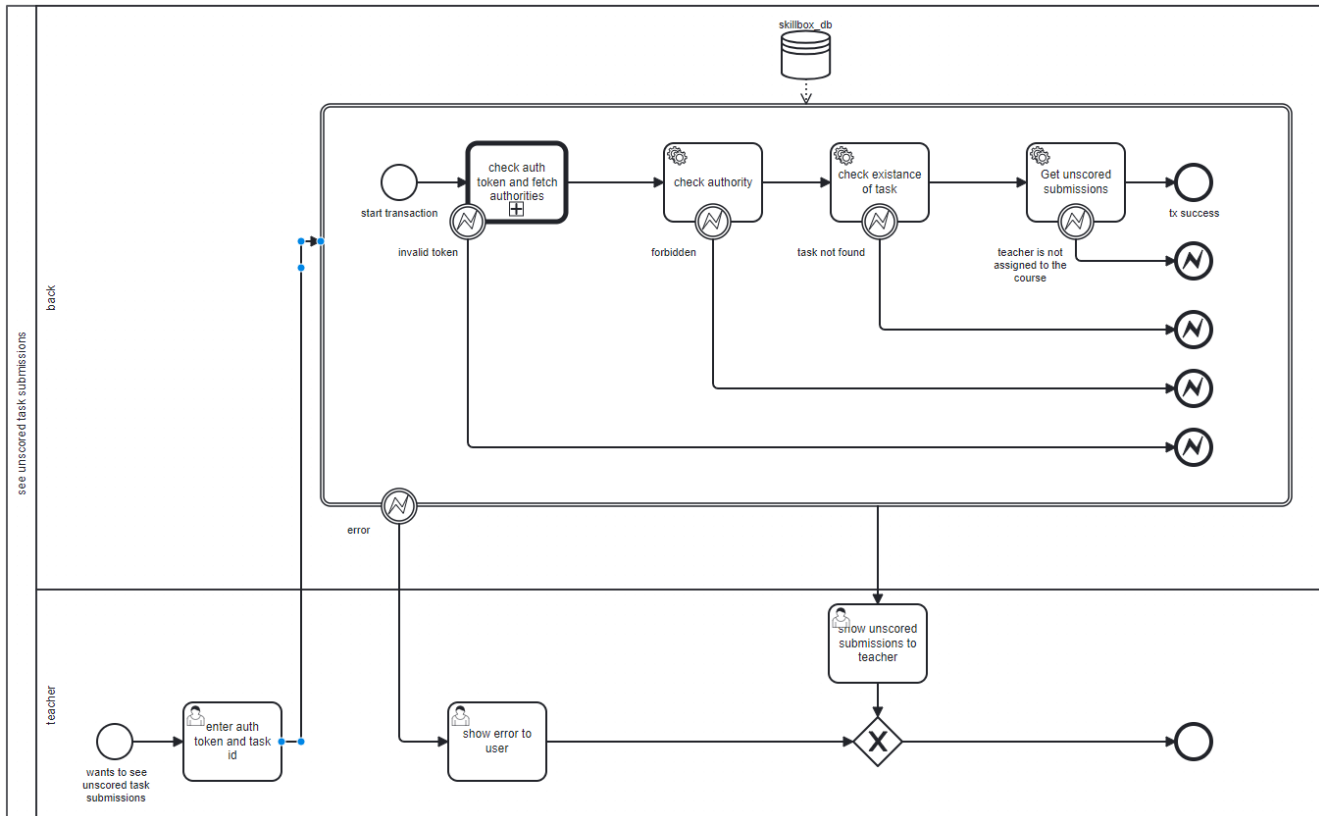




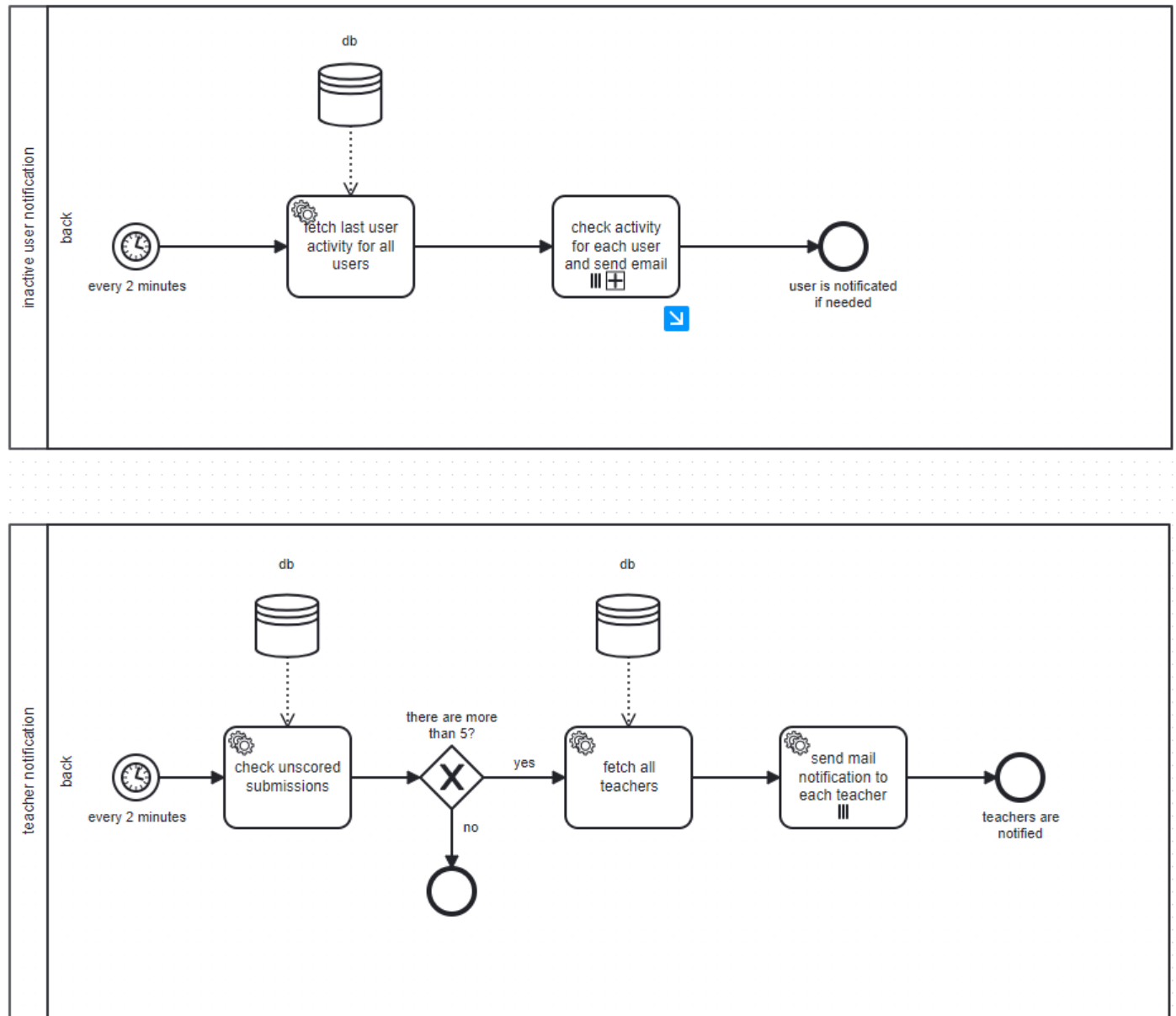
## 2.11 Просмотр информации по заданию



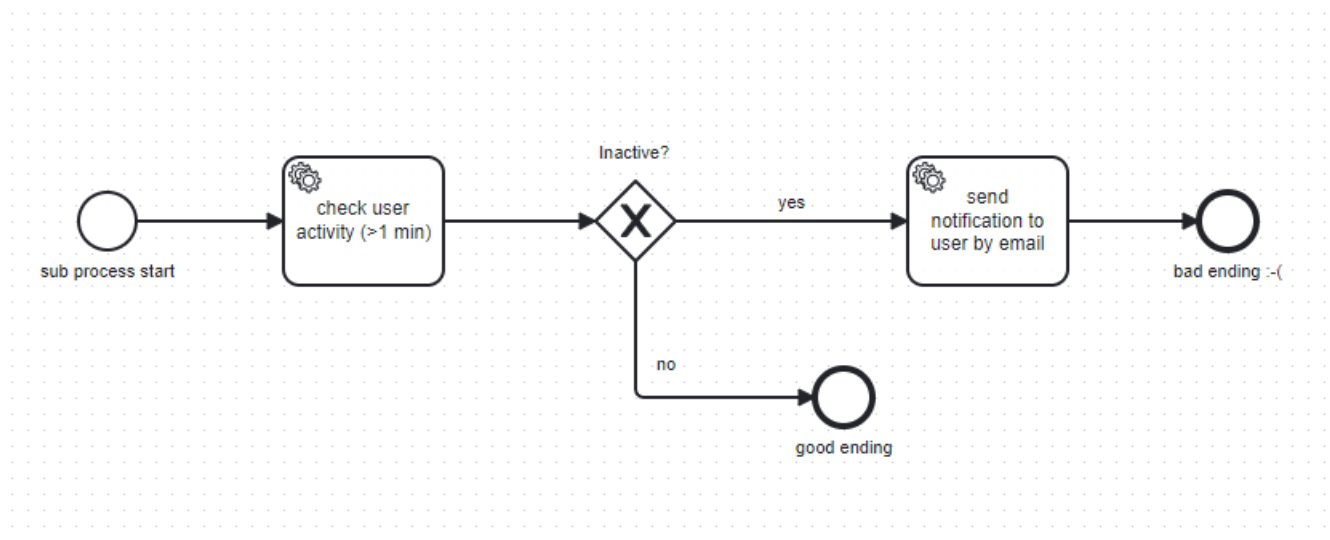
## 2.12 Просмотр всех неоценённых заданий



## 2.13 Уведомления пользователям и учителям



### 2.13.1 Проверка активности пользователя



### 3 Блок-схема архитектуры приложения с указанием ”точек интеграции” ВРМ-фреймвока с остальными подсистемами.

TaskServiceBpms
+String name: "taskServ"
+checkViewTaskAuthority(DelegateExecution) +checkGradeTaskAuthority(DelegateExecution) +checkSubmitTaskAuthority(DelegateExecution) +isExistingTask(Long) +isExistingSubmission(Long) +isValidSubmission(Long, Long) +validateAnswer(String) +getStudentSubmissions(Long, Long) : String +getTaskById(Long, Long) : String +getUnscoredSubmissions(Long, Long) : String +isEnrolled(Long, Long) : boolean +submitTask(Long, Long, String) : String +scoreSubmission(Long, Long, Integer)

CourseServiceBpms
+String name: "courseServ"
+checkBaseCourseAuthority(DelegateExecution) +checkCourseAvailability(Long) +enrollInCourse(Long, Long) +processPayment(User, Course) : PaymentHistory +disenrollFromCourse(Long, Long) +refundPayment(PaymentHistory)

CertificateServiceBpms
+String name: "certServ"
+isExistingCourse(Long) +isExistingRequest(Long, Long) +isExistingRequest(Long) +isRequestInProgress(Long) +checkAuthority(DelegateExecution) +checkTeacherAuthority(DelegateExecution) +areAllTaskDone(Long, Long) : boolean +createCertificateRequest(Long, Long) +getPendingRequests(Long) : String +checkRequestStatus(Long, Long) : String +requestCertificateGeneration(Long, Long) +processCertificateRequest(Boolean, Long) : String

AuthServiceBpms
+String name: "authServ"
+auth(String, String) : String

JwtTokenValidationDelegate
+String name: "jwtTokenValidationDelegate"
+execute(DelegateExecution)

## 4 Исходный код

Исходный код лежит по ссылке: <https://github.com/Romariok/BLPS>

## 5 Вывод

В ходе выполнения лабораторной работы было проведено внедрение BPM-движка Camunda. Для этого "статическая" бизнес-логика была заменена на "динамическую" на базе BPMS.

Приложение протестировано на сервере helios под управлением сервера приложений Apache Tomcat.