

# Software Architecture Document

## (Описание Архитектуры)

### 1. Introduction (Введение)

*[Введение представляет собой обзор на весь документ в целом и включает в себя следующие разделы - назначение, область применения, определения и аббревиатуры, ссылки и обзор.]*

#### 1.1 Purpose

*[Данный документ описывает архитектуру приложения как набор точек зрения на неё - use case view, logical view, process view, deployment view, implementation view (может быть стоит перевести названия?). Взаимодействие элементов разных точек зрения представлено в виде UML-диаграмм.]*

*[Укажите назначение данного документа.]*

Этот документ описывает архитектуру системы ИСУВИ как набор точек зрения:

- **Use Case View:** Основные сценарии использования.
- **Logical View:** Структура приложения и взаимодействие его компонентов.
- **Process View:** Взаимодействующие процессы и их временная привязка.
- **Deployment View:** Конфигурация и развертывание системы.
- **Implementation View:** Технические аспекты реализации.

Документ предназначен для разработчиков, тестировщиков, проектировщиков и стейкхолдеров, чтобы обеспечить единое понимание архитектуры и ключевых решений.

#### 1.2 Scope (Область применения)

*[Приведите краткое описание области применения данного документа, к какому(им) проекту(ам) он относится, кем будет использоваться и т.д.]*

Система ИСУВИ предназначена для автоматизации управления временными изменениями и мониторинга катаклизмов в мультивселенной.

- Пользователи: сотрудники УВИ (Агенты, Администраторы, Менеджеры, Варианты и др.).
- Область применения: мультивселенная с множественными временными линиями.
- Применение: решение задач управления временными аномалиями, контроля нагрузки станка и сбора ресурсов с самоуничтожающихся таймлайнов.

#### 1.3 Definitions, Acronyms and Abbreviations (Определения и аббревиатуры)

*[Укажите значение терминов и аббревиатур, которые употребляются в данном документе. Возможно указание ссылки на Глоссарий проекта.]*

Описаны в  Gloss .

#### 1.4 References (Ссылки)

*[Перечислите списком названия документов, на которые ссылаетесь в данном, укажите их источники.]*

- T3
- SRS
- UC
- Gloss

## 1.5 Overview (Обзор документа)

*[Приведите краткое описание остальных разделов документа.]*

| Секция |  |
|--------|--|
| 2      | Архитектурное представление. Включает типы диаграмм для отображения разных аспектов системы. |
| 3      | Архитектурные цели и ограничения.  |
| 4      | Представление сценариев использования.   |
| 5      | Логическое представление компонентов и взаимодействий.                                       |
| 6      | Представление процессов и их временной привязки.   |
| 7      | Конфигурация и развертывание.  |
| 8      | Описание реализованной системы.  |
| 9      | Характеристики производительности.   |
| 10     | Удовлетворение требованиям качества.   |

## 2. Architectural Representation (Представление архитектуры)

*[Данный раздел описывает в общем архитектуру системы. Укажите, какие типы диаграмм необходимы для описания разных точек зрения. Рекомендуется воспользоваться следующей таблицей:]*

| Diagram\View     | Use Case View               | Logical View                 | Implementation view | Process view** | Deployment View |
|------------------|-----------------------------|------------------------------|---------------------|----------------|-----------------|
| Use Case Diagram | +                           | -                            | -                   |                | -               |
| Class Diagram    | +(Взаимодействие сущностей) | +(Описание основных классов) | +(Полное описание)  |                | -               |

|                       |                             |  |  |  |   |
|-----------------------|-----------------------------|--|--|--|---|
|                       |                             | и интерфейсов их взаимодействия)   | классов с указанием их методов/полей, указать типы связей между классами)                          |  |   |
| Activity Diagram      | +<br>(Абстрактное описание) | +<br>(Более подробное описание, уровни взаимодействия должны совпадать с диаграммой пакетов) | +<br>(Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов). |  | - |
| State Machine Diagram | +<br>(Абстрактное описание) | +<br>(Более подробное описание, уровни взаимодействия должны совпадать с диаграммой пакетов) | +<br>(Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов). |  | - |
| Sequence Diagram      | +<br>(Абстрактное описание) | +<br>(Более подробное описание, уровни взаимодействия должны совпадать с диаграммой пакетов) | +<br>(Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов). |  | - |
| Cooperative Diagram   | +<br>(Абстрактное описание) | +<br>(Более подробное описание, уровни взаимодействия должны совпадать с диаграммой пакетов) | +<br>(Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов). |  | - |

|                    |   |   |  |   |   |
|--------------------|---|---|--|---|---|
| Package Diagram    | - | + | -  |   | -   |
| Data Base Diagram  | - | - | +(Полная ER модель базы данных + её даталогическая модель) |   | -   |
| Deployment Diagram | - | - | -  |   | +(Подробная диаграмма развертывания с указанием характеристик машин и интерфейсов взаимодействия) |
| Timeline diagram   |   |   |  | + |   |

*\*Activity, Sequence, Cooperative и State Machine диаграммы составляются на основе одного прецедента (каждый тип диаграмм - на основе своего).*

*\*\*Всё представление описывается только в случае, если в системе есть процессы, жестко привязанные к определенным моментам времени (пример - наступление нового месяца, времени суток и т.д.)*

### 3. Architectural Goals and Constraints (Цели и ограничения архитектуры)

*[Перечислите здесь все архитектурно-значимые факторы - важные прецеденты, специфичные требования к работе системы и т.д.]*

## 3.1 Цели архитектуры

Архитектура системы ИСУВИ направлена на достижение следующих целей:

1. **Масштабируемость:** Обеспечить возможность добавления новых функциональных модулей без значительного изменения существующего кода.
2. **Надежность:** Система должна стабильно функционировать при высоких нагрузках и в случае сбоя обеспечивать быстрое восстановление.
3. **Производительность:** Минимизировать время отклика системы ( $\leq 3$  секунд на запрос).
4. **Модульность:** Логическое разделение системы на модули (Frontend, Backend, Database) для упрощения разработки и поддержки.
5. **Безопасность:** Использовать шифрование данных, защиту от SQL-инъекций и строгую авторизацию для всех пользователей.

## 3.2 Ограничения архитектуры

**Технологические:**

- Фронтенд: React.
- Бэкенд: Java Spring Framework.
- База данных: PostgreSQL.
- Сервер: Helios (используется Docker для развертывания).

**Ресурсные:**

- Разработка ведется командой из 3 человек.
- Отсутствие выделенного бюджета.
- Ограниченные вычислительные мощности (сервер Helios).

**Производительность:**

- Максимальное количество одновременно активных пользователей: 100.
- Время отклика системы:  $\leq 3$  секунд.

**Функциональные:**

- Система должна поддерживать:
  - Управление катаклизмами.
  - Контроль нагрузки на станок.
  - Назначение и выполнение заданий.

**Временные:**

- Срок разработки: 6 месяцев с началом тестирования на 5-м месяце.

## 3.3 Архитектурно значимые прецеденты по ролям

**Вариант:**

- Создание катаклизма.
- Просмотр активных катаклизмов.

#### **Менеджер**

- Назначение задания
- Отмена или перераспределение задания

#### **Агент**

- Выполнение задания
- Отчет о выполнении задания

#### **Администратор**

- Создание и управление учетными записями
- Удаление пользователей и их данных

#### **Управляющий станком**

- Мониторинг нагрузки на станок
- Настройка параметров работы станка

### **4. Use-Case View**

*[Данный раздел содержит описание основных сценариев использования системы разными типами пользователей. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]*

## 4.1 Use Case Diagram

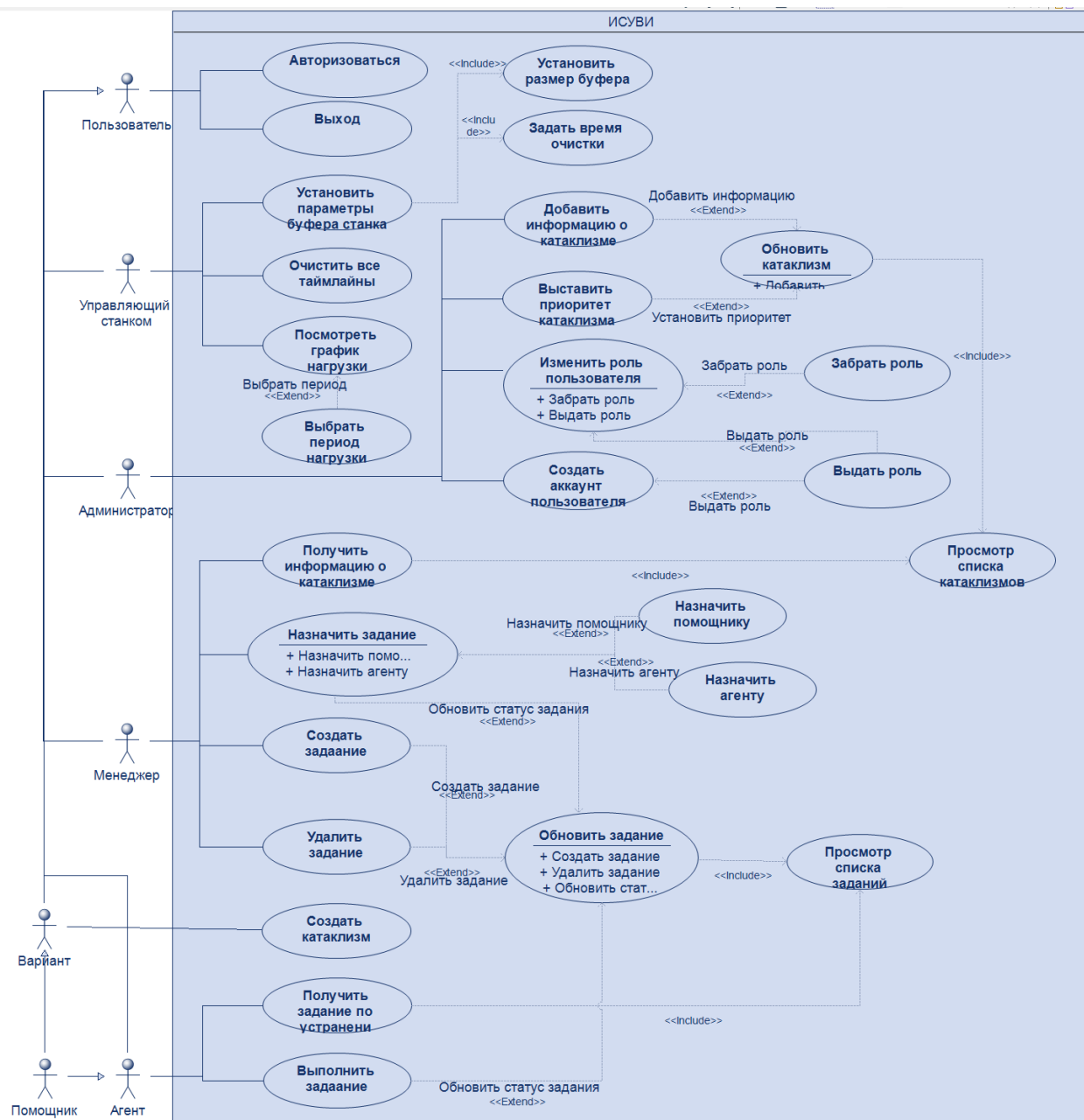


Рисунок 4.1.1 – UC

## 4.2 Class Diagram (Создание катаклизма)

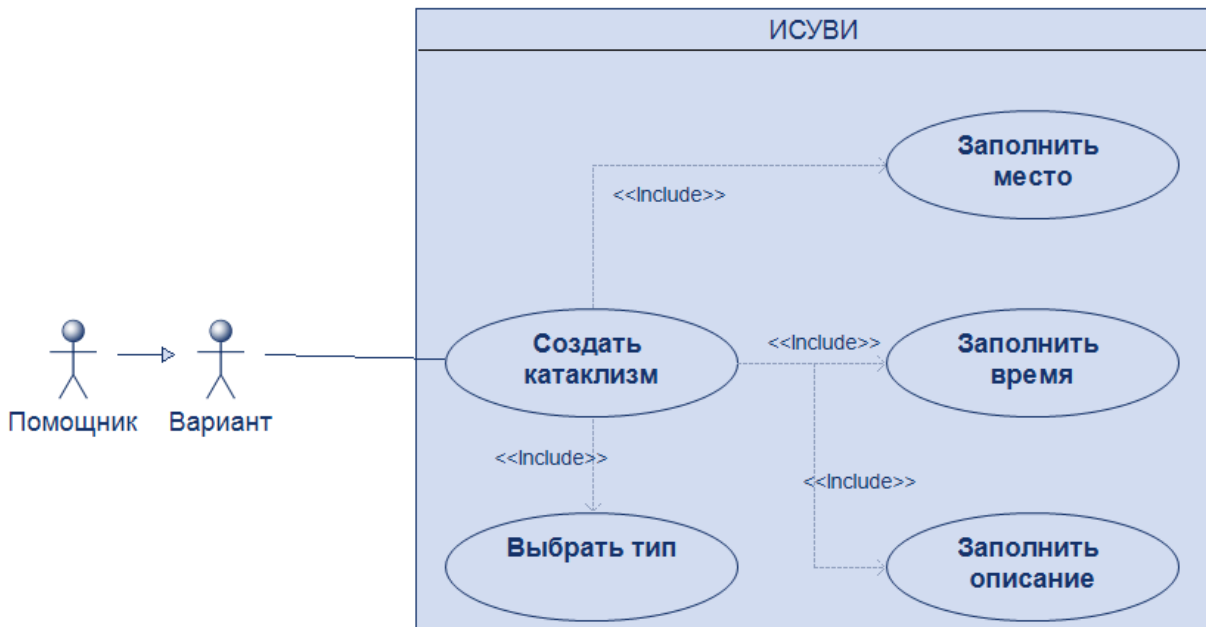


Рисунок 4.2.1 – UC Создание катаклизма

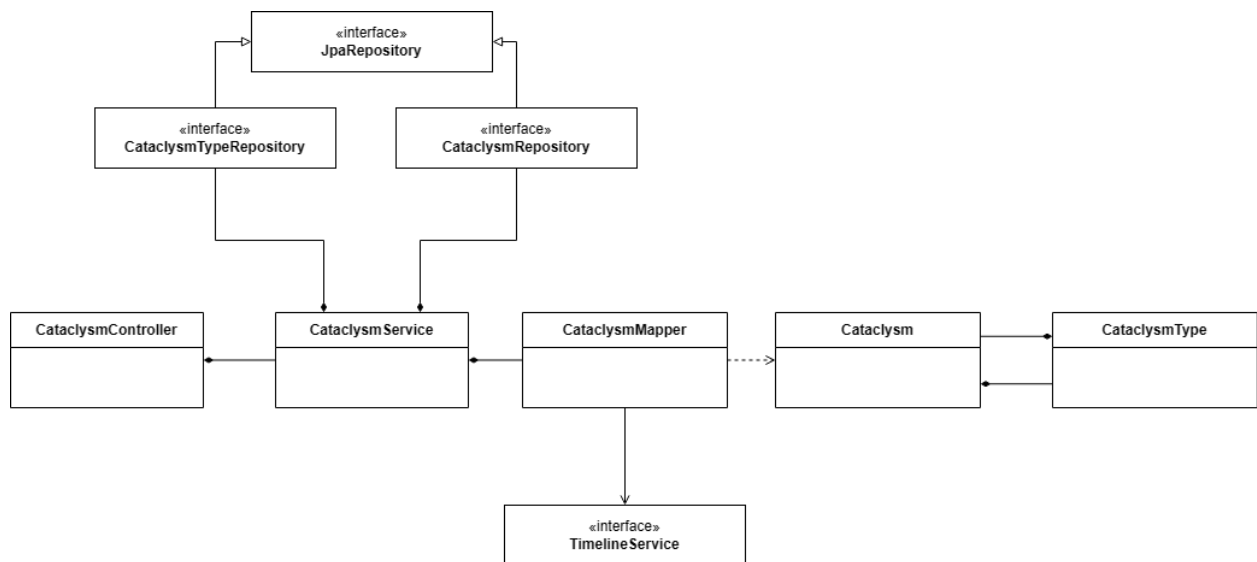


Рисунок 4.2.2 – Class diagram (UC Создание катаклизма)



#### 4.3 Activity Diagram (Обновление информации о катаклизме)



Рисунок 4.3.1 – UC Обновление информации о катаклизме

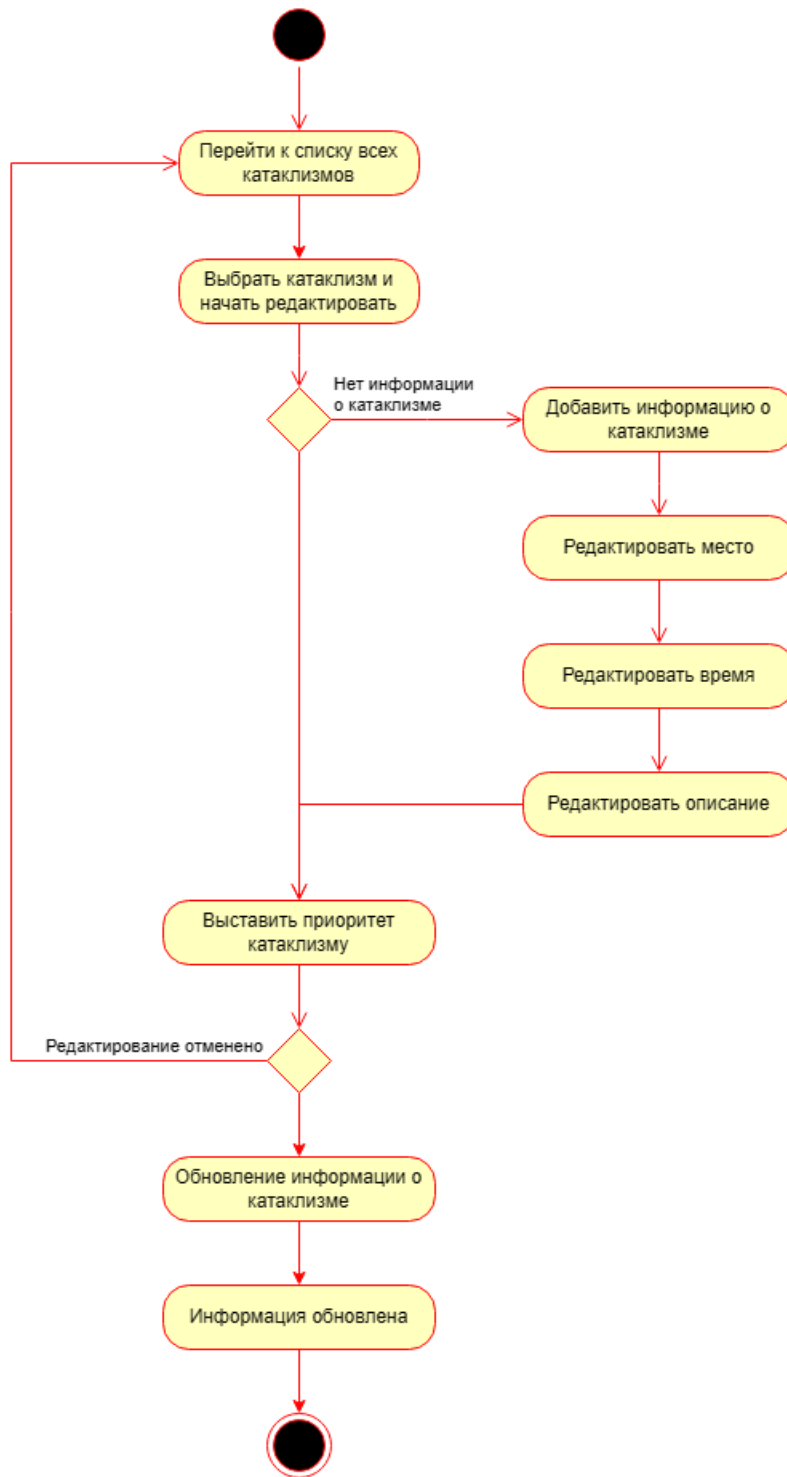


Рисунок 4.3.2 – Activity Diagram (UC Обновление информации о катаклизме)

#### 4.4 State Machine Diagram (Выполнить задание)

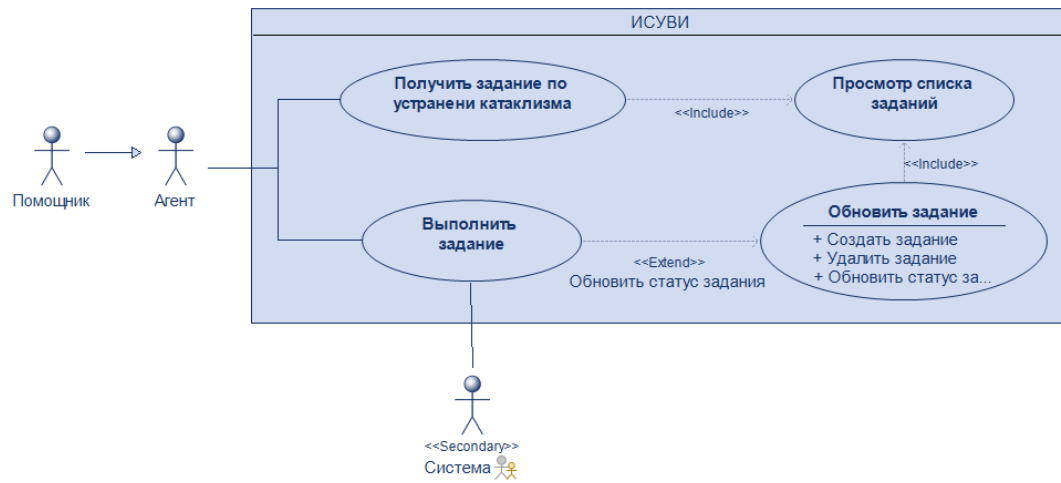


Рисунок 4.4.1 – UC Выполнить задание

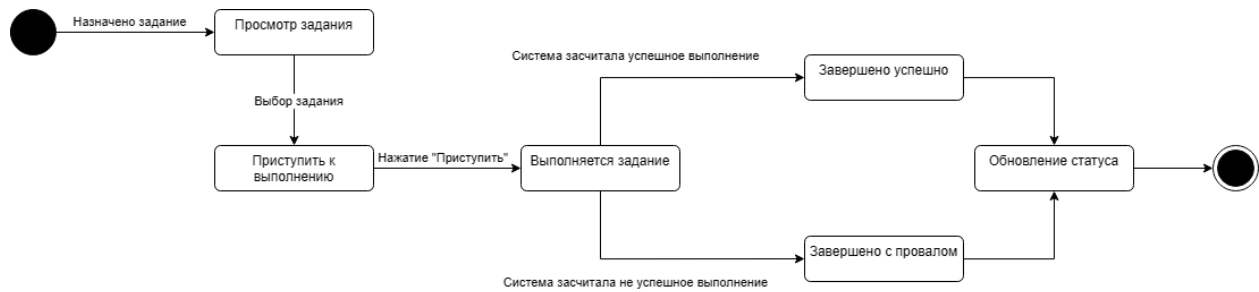


Рисунок 4.4.2 – State Machine Diagram (UC Выполнить задание)

#### 4.5 Sequence Diagram (Создать задание)

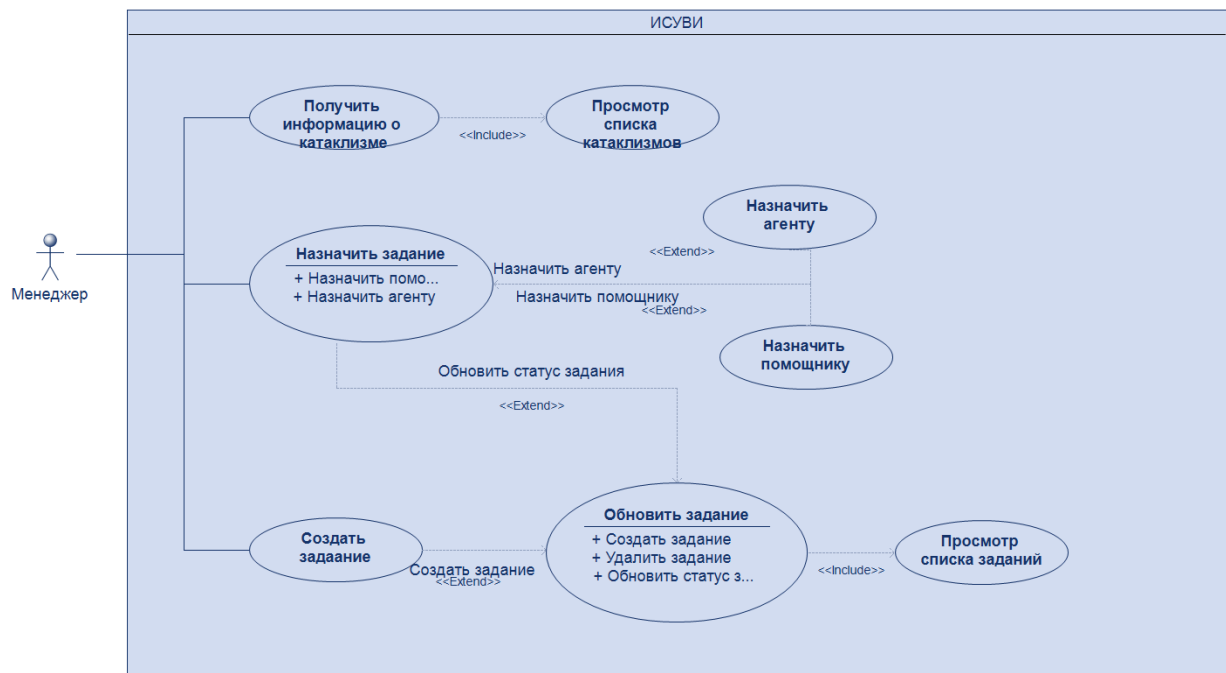


Рисунок 4.5.1 – UC Создать задание



Рисунок 4.5.2 – Sequence Diagram (UC Создать задание)

#### 4.6 Cooperative Diagram (Выполнить задание + Создать задание)

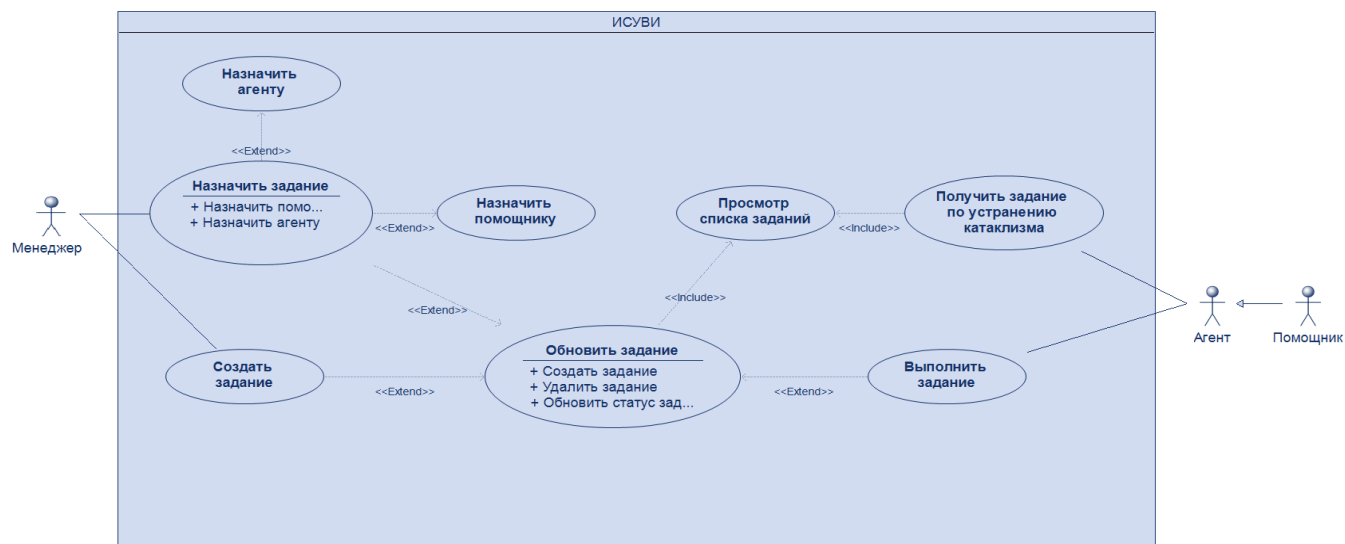


Рисунок 4.6.1 – UC Выполнить задание + Создать задание

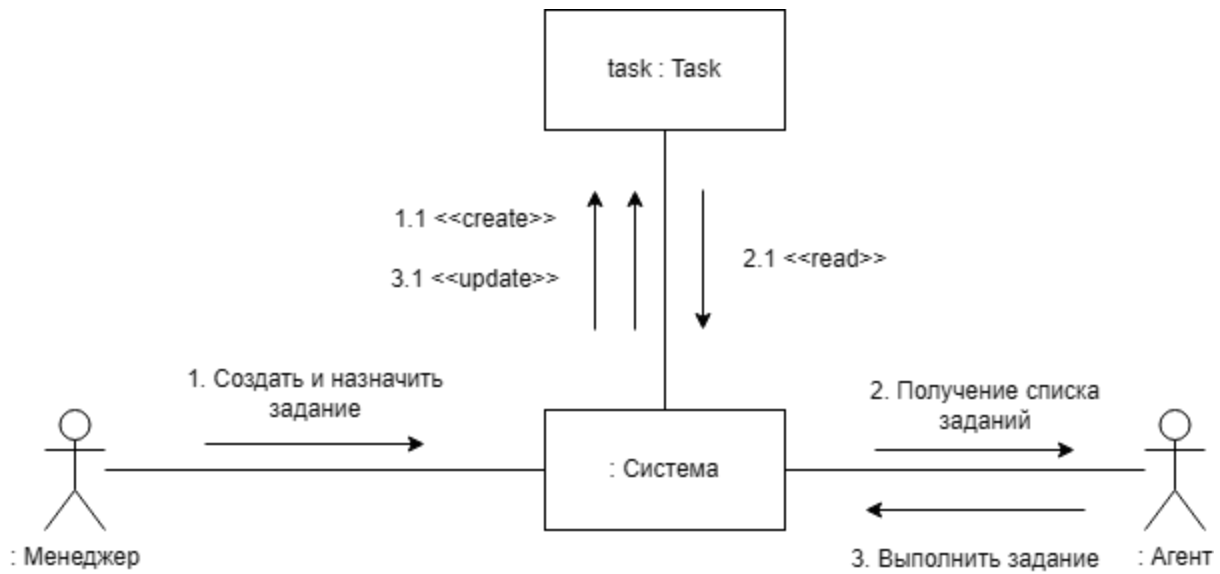


Рисунок 4.6.2 – Cooperative Diagram (UC Выполнить задание)

## 5. Logical View

*[Данный раздел содержит описание слоев, на которые делится приложение, а также интерфейсов их взаимодействия. Приведите описание каждого из слоев, как они связаны между собой, их назначение. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]*

### 5.1 Class Diagram (Выполнить задание)



Рисунок 5.1.1 – UC Выполнить задание



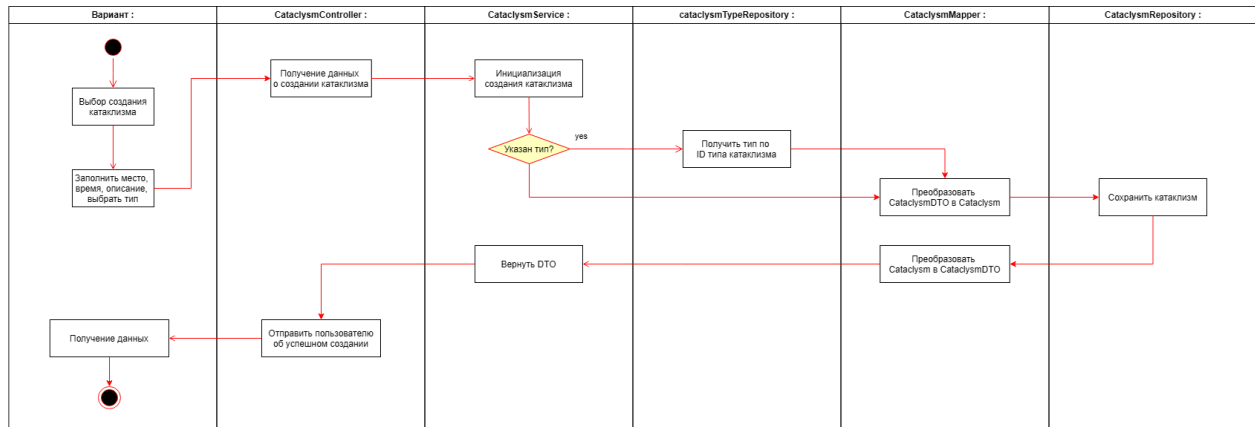


Рисунок 5.2.2 – Activity Diagram (UC Создание катаклизма)

### 5.3 State Machine Diagram (Создать задание)

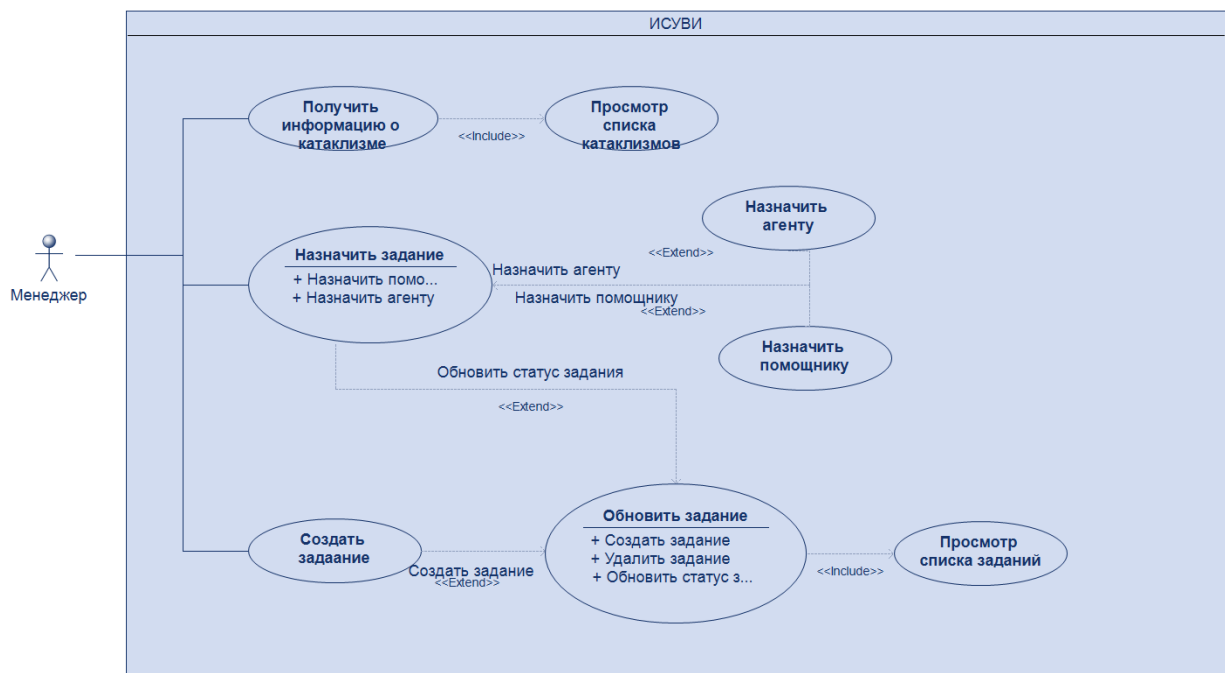


Рисунок 5.3.1 – UC Создать задание



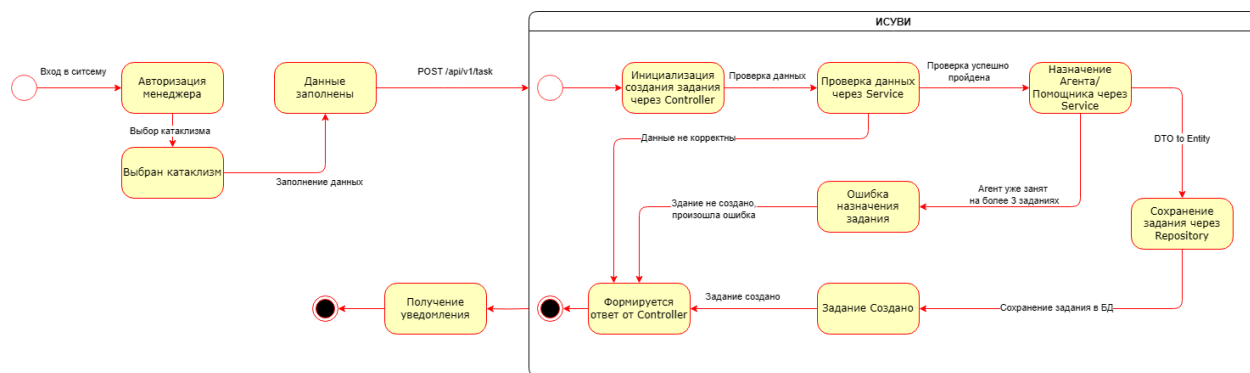


Рисунок 5.3.2 – State Machine Diagram (UC Создать задание)

#### 5.4 Sequence Diagram (Обновление информации о катаклизме)



Рисунок 5.4.1 – UC Обновление информации о катаклизме

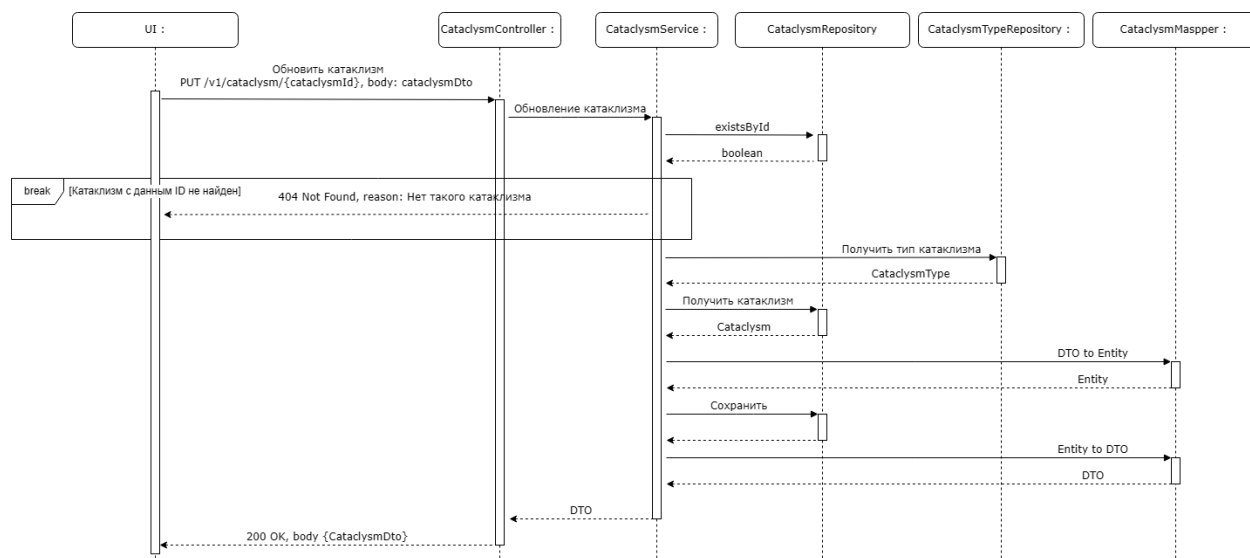


Рисунок 5.4.2 – Sequence Diagram (UC Обновление информации о катаклизме)

## 5.5 Cooperative Diagram (Обновление информации о катаклизме)

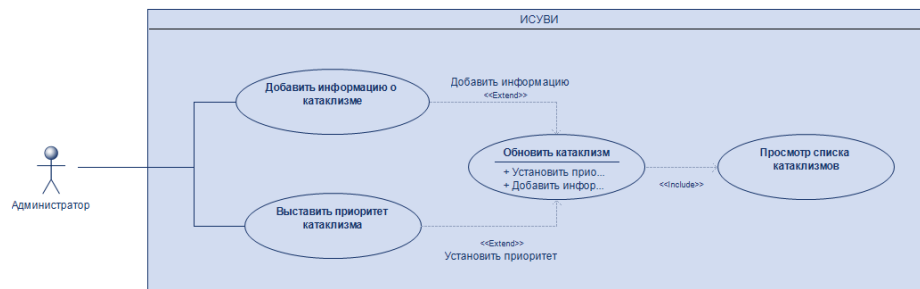


Рисунок 5.5.1 – UC Обновление информации о катаклизме

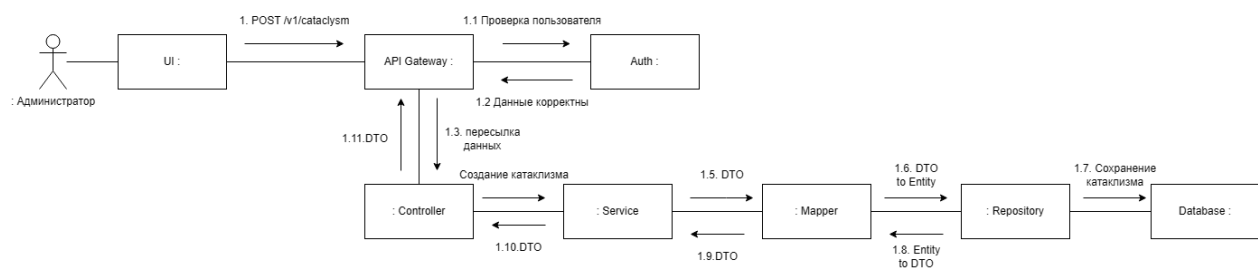


Рисунок 5.5.2 – Cooperative Diagram (UC Обновление информации о катаклизме)

5.6 Package Diagram

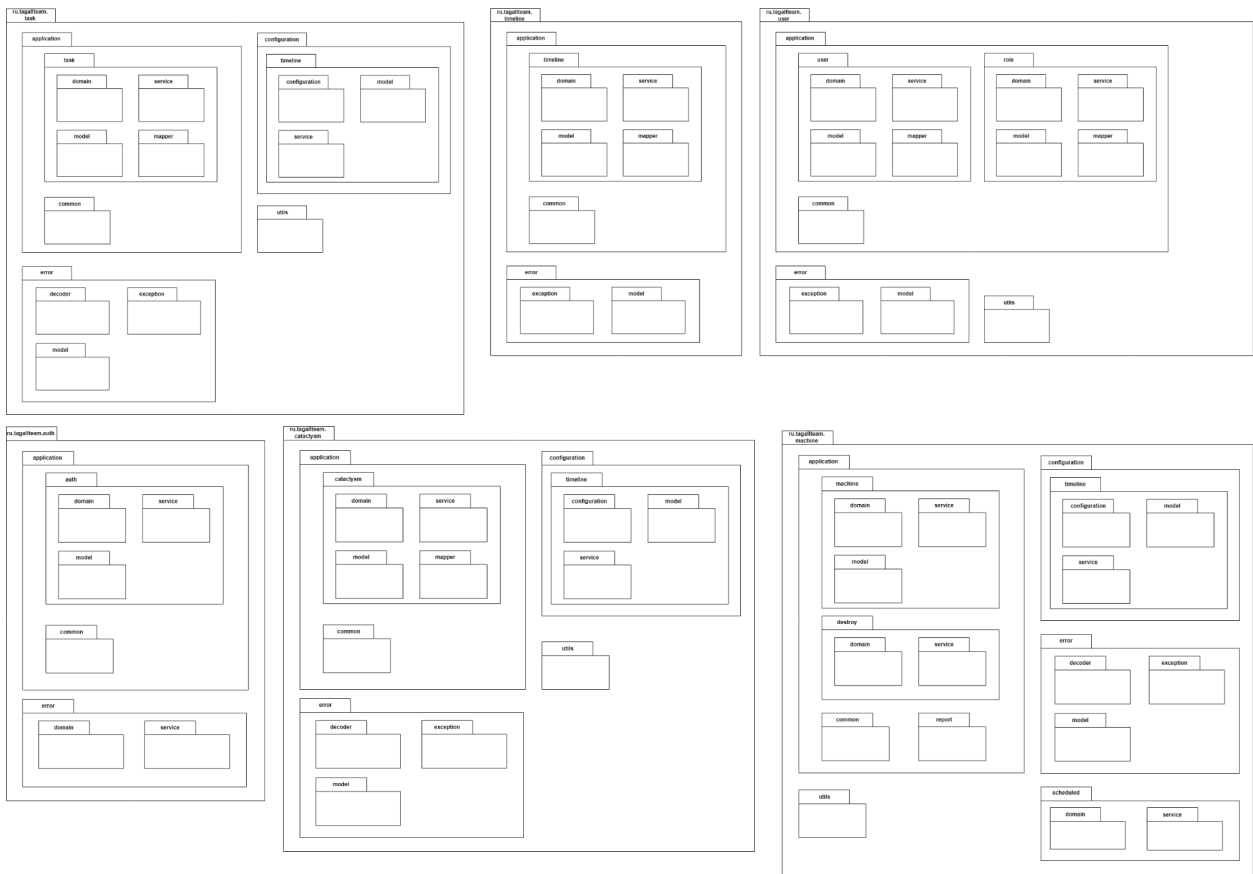


Рисунок 5.6 – Package Diagrams

6. Process View

[Данный раздел описывает систему как совокупность взаимодействующих процессов, привязанных к определенным моментам времени. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]

Неприменимо, так как в системе отсутствуют процессы, жестко привязанные к времени.

7. Deployment View

[Данный раздел содержит описание конфигурации файлов, из которых состоит система, мест их расположения и описание взаимодействия их друг с другом. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]

| Resource<br>(Ресурс) | Quantity<br>(Количество) | Name and Type<br>(Название и тип)         |
|----------------------|--------------------------|---|
| Процессор            | 16 ядер                  | Intel(R) Xeon(R) CPU E5-2643 0 @ 3.30GHz. |

|                    |          |   |
|--------------------|----------|---|
| Оперативная память | 128 ГБ   | DDR4, рабочая частота $\geq 2400$ МГц.            |
| Накопитель         | 6 дисков | 559 ГБ x 2 SSD, 5.5 ТБ x 4 HDD.                   |
| Сетевой интерфейс  | 1        | Intel(R) Ethernet Controller X710 для 10GbE SFP+. |

Будет использовать три аккаунта разработчиков для запуска 7 микросервисов на Helios.  
Распределение микросервисов по аккаунтам:

- 1 аккаунт: timeline-service, auth-service, task-service
- 2 аккаунт: user-service, cataclysm-service, machine-service
- 3 аккаунт: gateway-service

Ниже представлена диаграмма разворачивания системы на сервер Helios. Развертывание будет происходить с помощью bash скрипта по протоколу SSH с компьютера разработчиков.

### 7.1 Deployment diagram

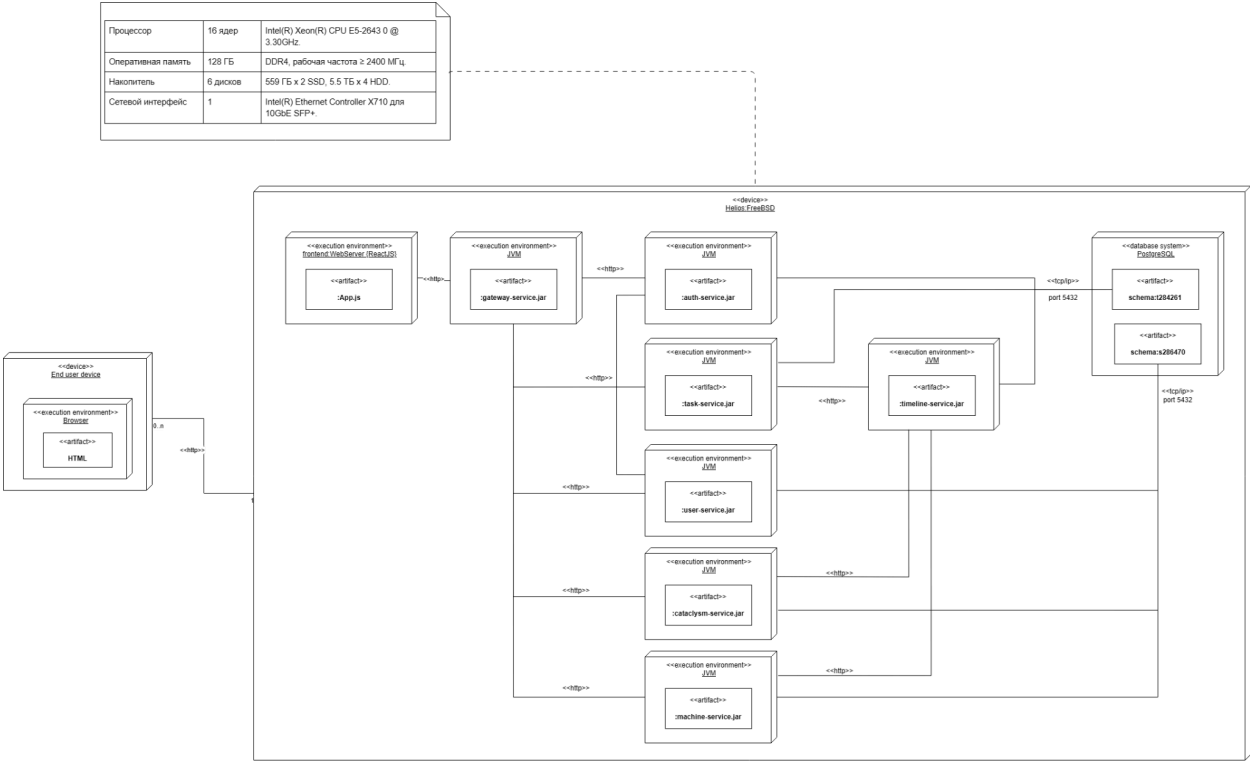


Рисунок 7.1.1 – Диаграмма разворачивания системы

### 8. Implementation View

[Данный раздел содержит описание системы в уже реализованном виде. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]

## 8.1 Class Diagram (Создать аккаунт пользователя с ролью)

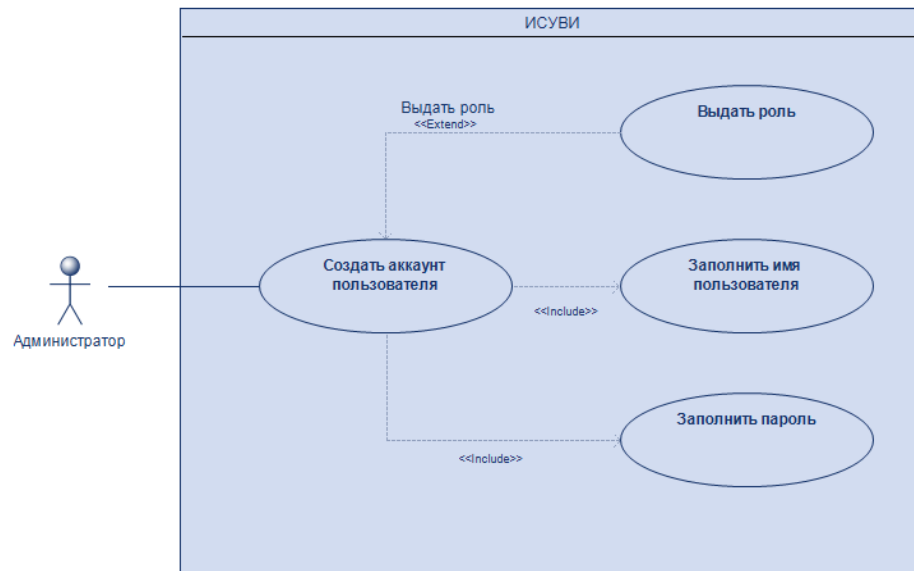


Рисунок 8.1.1 – UC Создать аккаунт пользователя с ролью

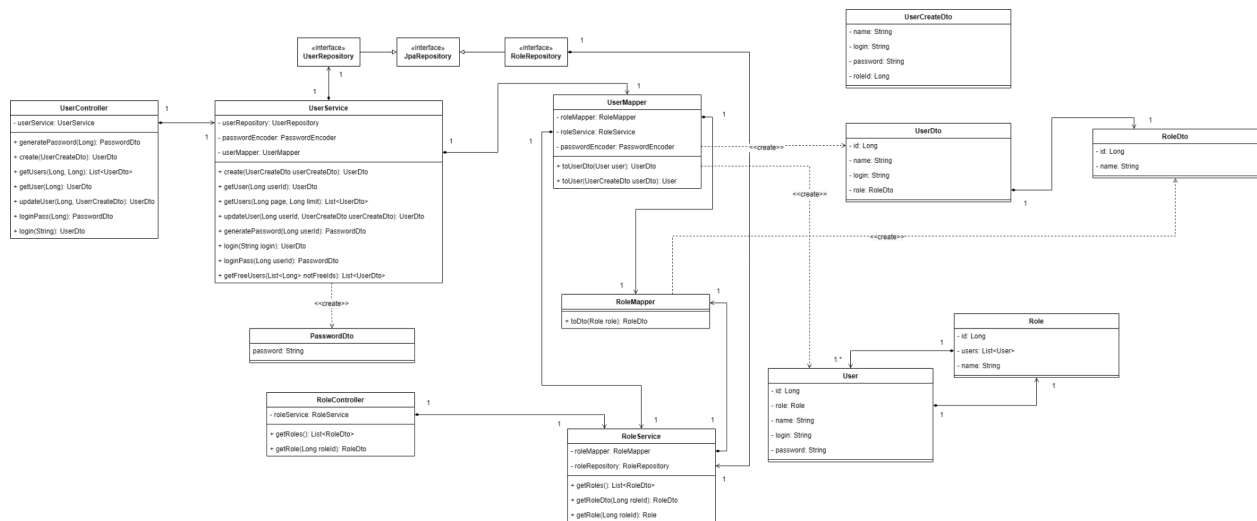


Рисунок 8.1.2 – Class Diagram (UC Создать аккаунт пользователя с ролью)

## 8.2 Activity Diagram (Выполнить задание)

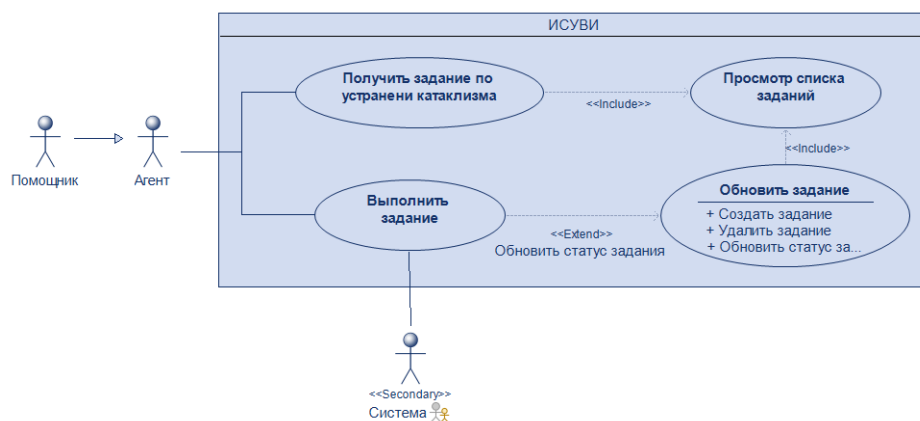


Рисунок 8.2.1 – УС Выполнить задание

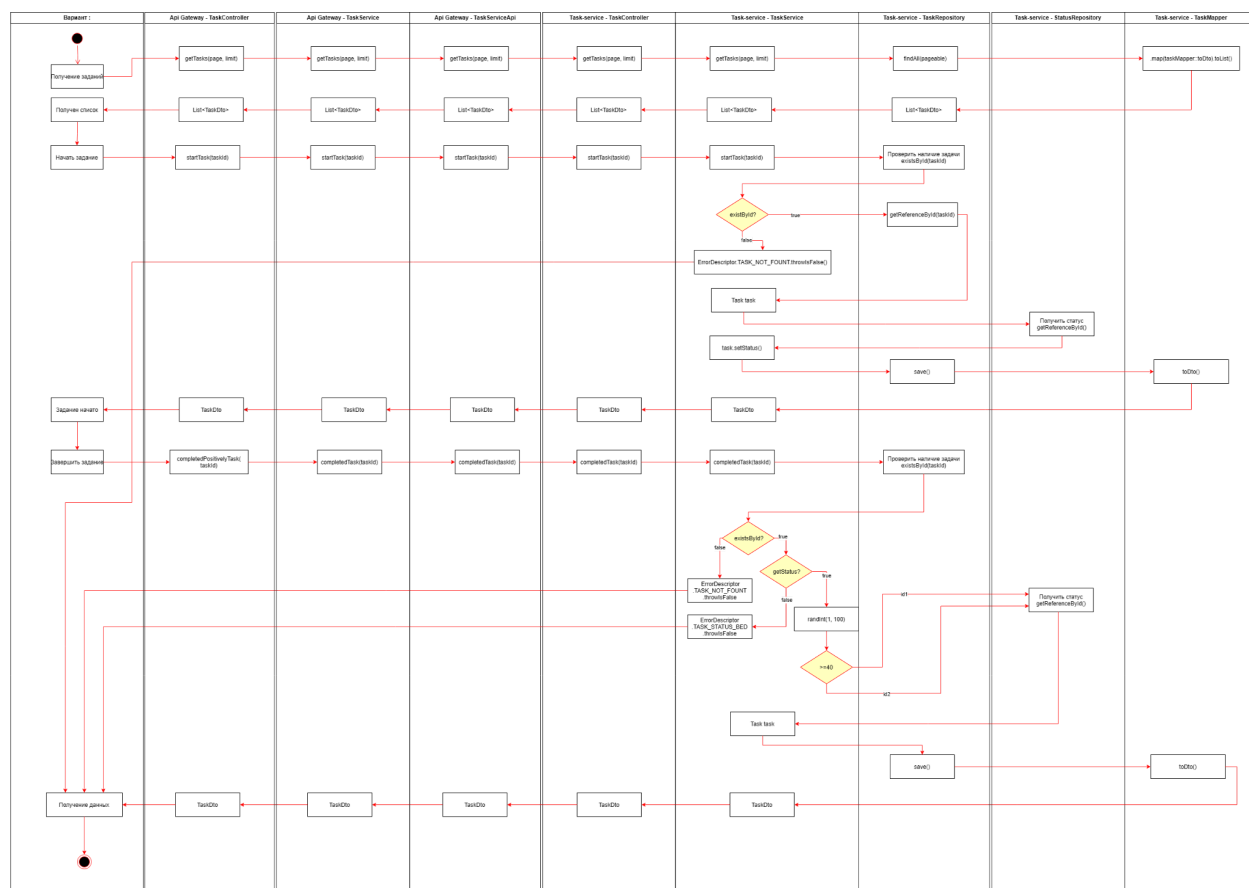


Рисунок 8.2.2 – Activity Diagram (UC Выполнить задание)

### д8.3 State Machine Diagram (Обновление информации о катаклизме)

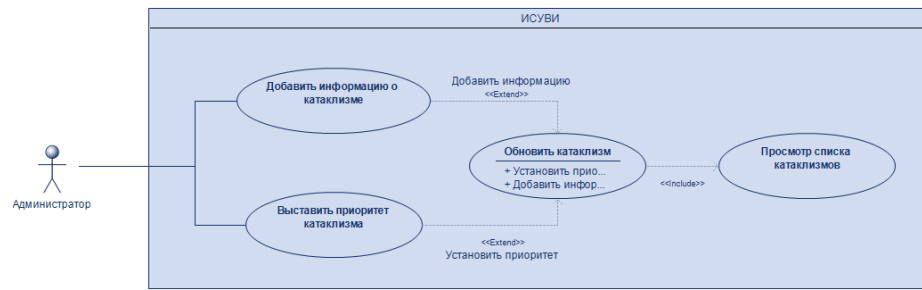


Рисунок 8.3.1 – UC Обновление информации о катаклизме

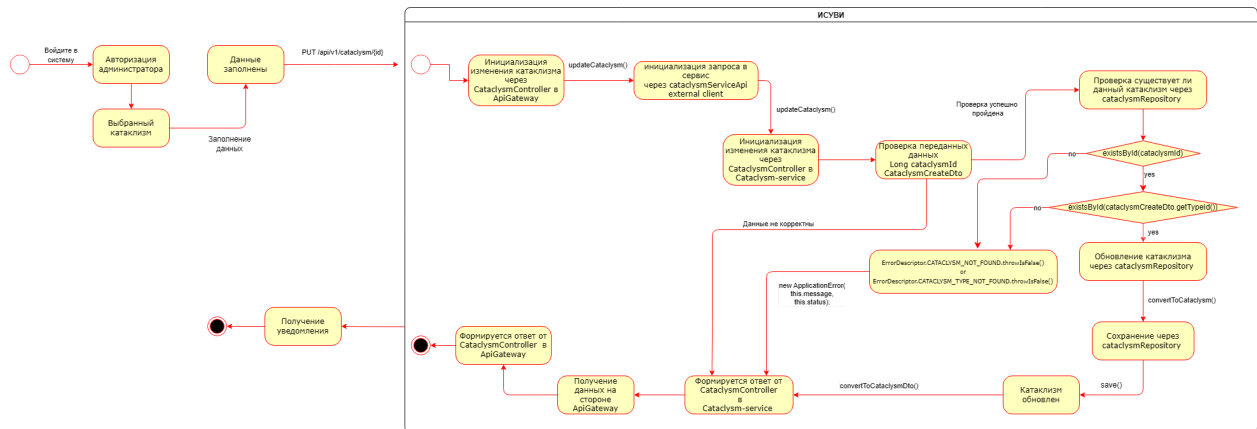


Рисунок 8.3.2 – State Machine Diagram (UC Обновление информации о катаклизме)

### 8.4 Sequence Diagram (Установить параметры буфера станка)



Рисунок 8.4.1 – UC Установить параметры буфера станка

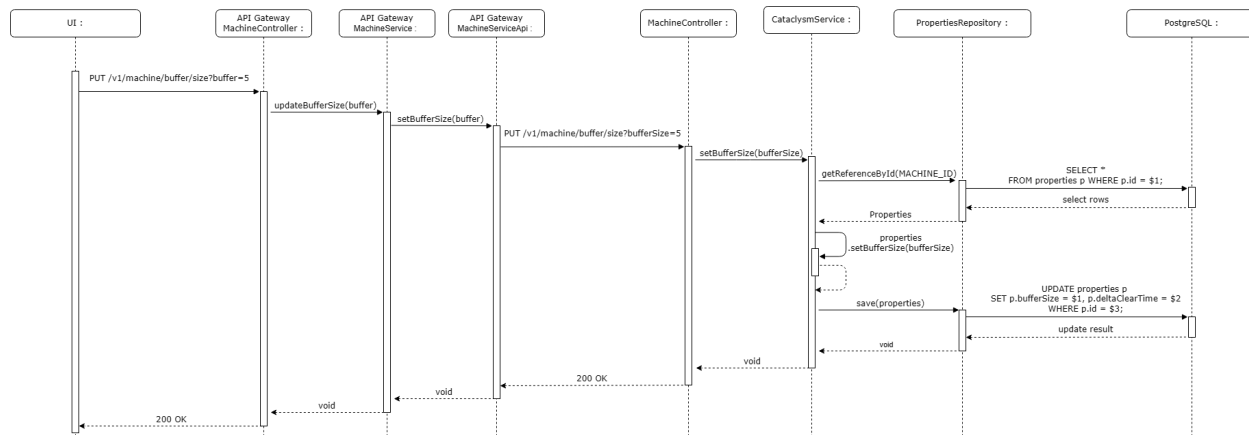


Рисунок 8.4.2 – Sequence Diagram (UC Установить параметры буфера станка)

## 8.5 Cooperative Diagram (Создать задание)

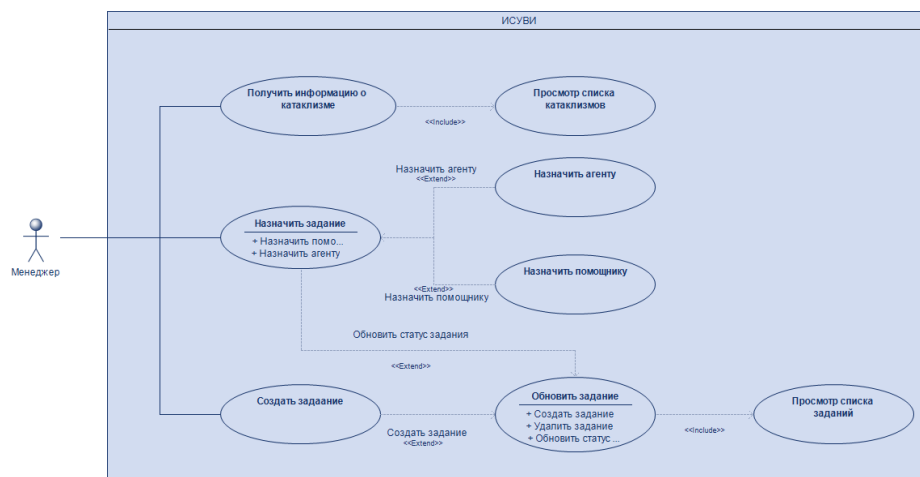


Рисунок 8.5.1 – UC Создать задание

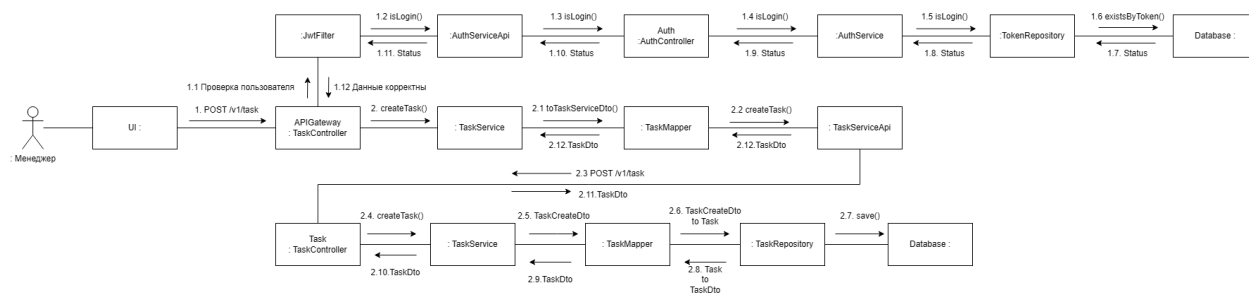


Рисунок 8.5.2 – Cooperative Diagram (UC Создать задание)



8.6 Data Base Diagram



Рисунок 8.6.1 - Инфологическая модель



## 9. Size and Performance (Производительность)

*Данный раздел описывает основные характеристики измерения производительности системы и их границы, которые могут оказать влияние на архитектуру.]*

- На основе сериала, вычислено, что в системе работает около 100 человек одновременно и система должна выдерживать ~100 rps, т. к. каждый пользователь генерирует 1 запрос в секунду.
- Ограничение по времени ответа не более 3 секунд. [SRS 3.4.1](#)

## 10. Quality (Качество)

*Данный раздел описывает, каким образом архитектура системы удовлетворяет её показателям качества - масштабируемости, надежности, мобильности, безопасности и т.д.]*

- Максимальное время возможности восстановления системы не должно превышать 24 часа.
- Система должна демонстрировать уровень доступности, при котором допустимое время простоя:
  - в день — не более 12 часов;
  - в месяц — не более 24 часов.
- Система должна хранить пароли в базе данных PostgreSQL в зашифрованном виде.