**Software Architecture Document**

**(Описание Архитектуры)**

| **Дата** | **Авторы** | **Изменение** |
| --- | --- | --- |
| 08.11.2022 | Авраменко А.Д.  Давыдов И.Д.  Шишкин Н.Д. | Initial version |
| 22.11.2022 | Авраменко А.Д.  Давыдов И.Д.  Шишкин Н.Д. | Доделан - logical  Сделан - implementation |
| 29.11.2022 | Авраменко А.Д.  Давыдов И.Д.  Шишкин Н.Д. | Изменено Logical:  Class, StateMachine  Implementation:  Class, Activity, StateMachine, DB |

**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. Взаимодействие элементов разных точек зрения представлено в виде UML-диаграмм.]*

*[Укажите назначение данного документа****.****]*

В этом документе представлен обзор архитектуры системы с использованием ряда различных представлений для отображения различных аспектов системы с точки зрения разных пользователей, таких как конечные пользователи, разработчики или руководители проекта. Документ предназначен для фиксации и распространения важных архитектурных решений, которые были приняты в системе.

**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.]*

Данный документ относится к проекту по созданию системы для автоматизации процесса полного цикла проведения модификации, будет использоваться заказчиком и командой разработки.

**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.]*

1. Glossary

**1.4 References (Ссылки)**

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

1) Джим Арлоу, Айла Нейштадт. UML 2 и унифицированный процесс.

2) Лукьяненко С. В. “Геном”.

3) Vision

4) Требования к системе

5) Прецеденты использования

**1.5 Overview (Обзор документа)**

*[Приведите краткое описание остальных разделов документа.]*

Данный Software Architecture Document содержит следующие разделы:

**Architectural Representation** -

описывает в общем архитектуру системы с указанием, какие типы диаграмм необходимы для описания разных точек зрения.

**Architectural Goals and Constraints** -

описывает все архитектурно-значимые факторы - важные прецеденты, специфичные требования к работе системы и т.д.

**Use-Case View** -

содержит описание основных сценариев использования системы разными типами пользователей.

**Logical View** -

содержит описание слоев, на которые делится приложение, а также интерфейсов их взаимодействия.

**Process View** -

описывает систему как совокупность взаимодействующих процессов, привязанных к определенным моментам времени.

**Deployment View** -

содержит описание конфигурации файлов, из которых состоит система, мест их расположения и описание взаимодействия их друг с другом.

**Implementation View** -

содержит описание системы в уже реализованном виде.

**Size and Performance** -

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

**Quality** -

описывает, каким образом архитектура системы удовлетворяет показателям качества - масштабируемости, надежности, мобильности, безопасности и т.д.

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

Разработка серверной и клиентской частей ведется параллельно, взаимодействуют они через REST API. Серверная часть следует шаблону MVC.

## **Используемые технологии**

IDE и редакторы:

* Frontend: VS code
* Backend: IntelliJ IDEA

Языки:

* Frontend: TypeScript, HTML, CSS
* Backend: Java 8, SQL

Фреймворки:

**Angular**

Компонентный фреймворк, используется нами для создания одностраничной фронтенд части.

**Spring Boot**

Предоставляет полезные функции для создания и запуска веб-приложений, в частности веб-службу REST.

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

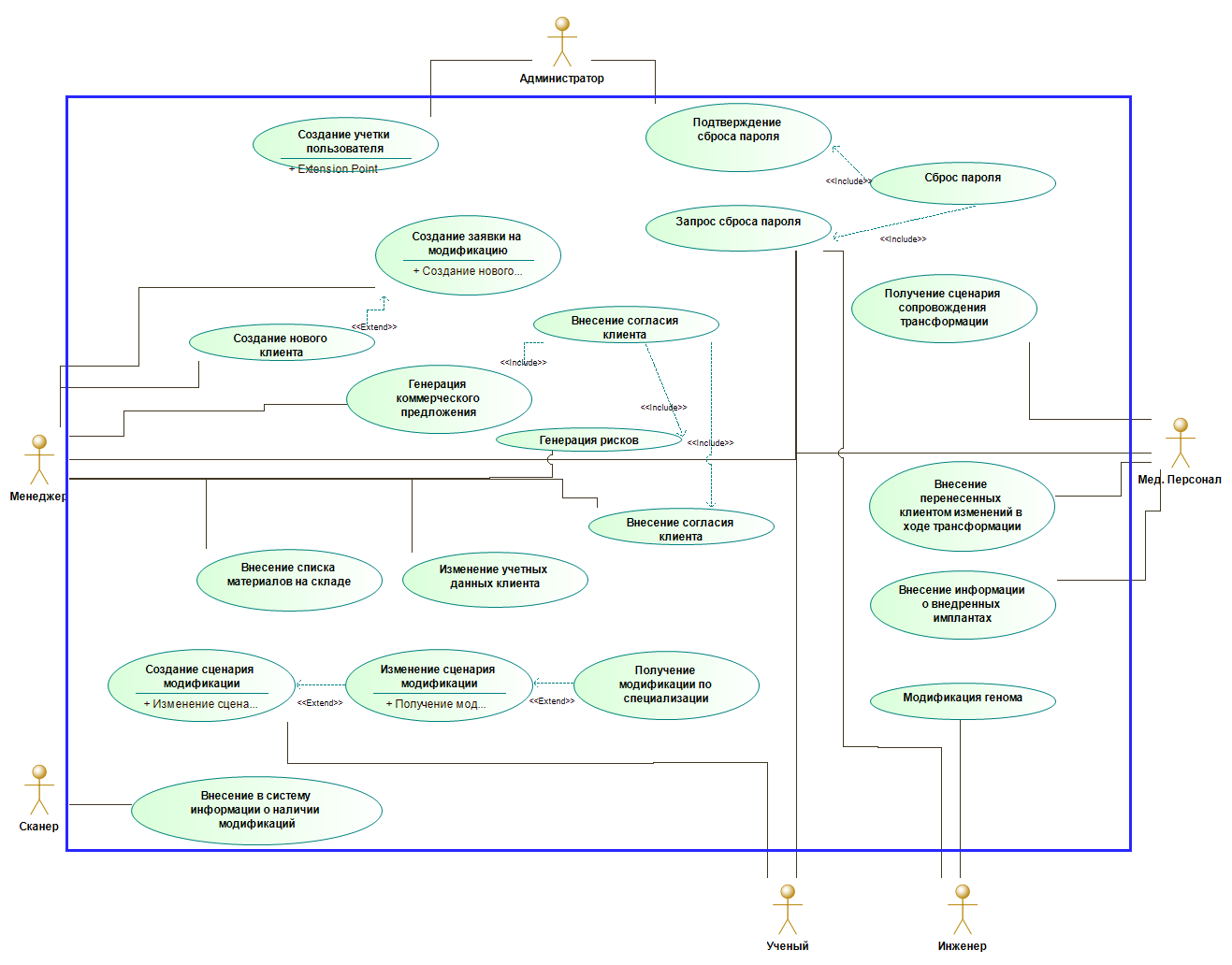
*[Перечислите здесь все архитектурно-значимые факторы - важные прецеденты, специфичные требования к работе системы и т.д.]*

1. В качестве языка программирования серверной части используется Java c фреймворком Spring Boot.
2. Все требования к производительности и нагрузке, как указано в документах Vision и SRS, должны приниматься во внимание при разработке архитектуры.
3. Система разграничивает функциональность Менеджера, Медика, Администратора, Инженера и Ученого.
4. Система должна использовать СУБД PostgreSQL для хранения данных.
5. Коммуникация между сервером и клиентской частью должна быть организована с применением принципов REST на базе протоколов HTTP/HTTPS.

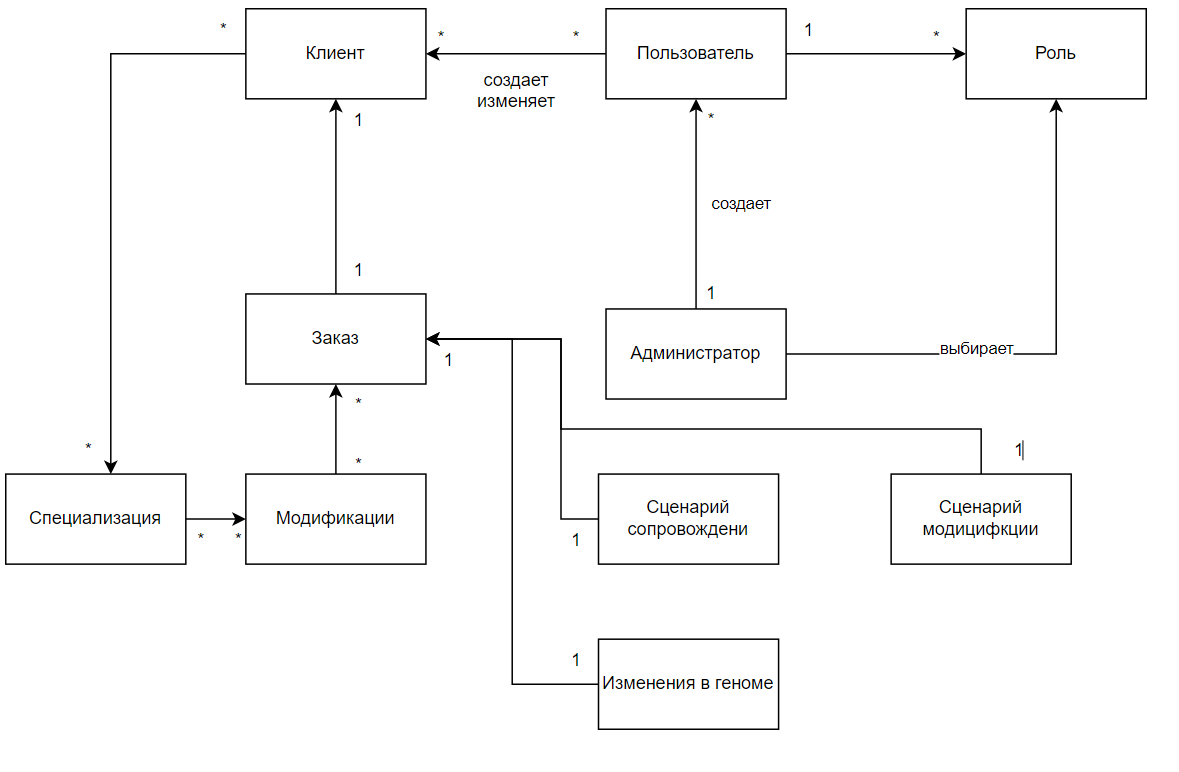
**4.** **Use-Case View**

*[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**

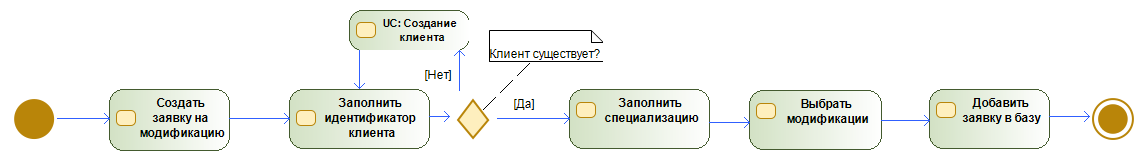
****

**Class diagram 1**

****

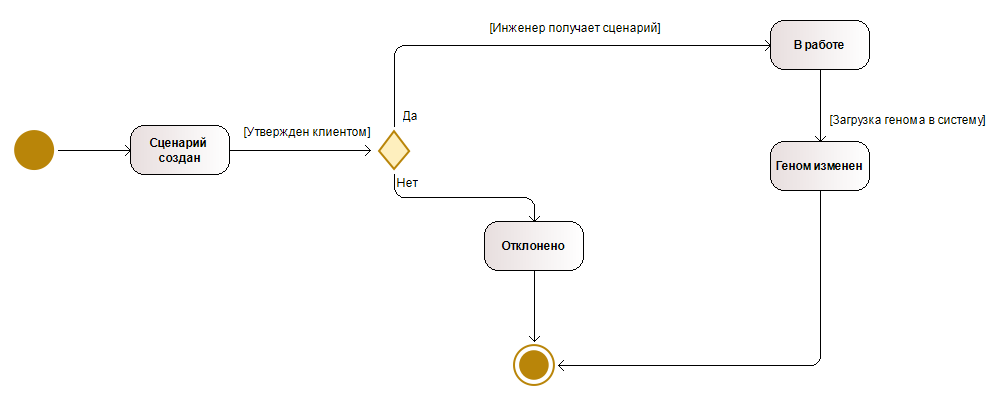
**Activity diagram 1**

Прецедент: Создание заявки на модификацию

****

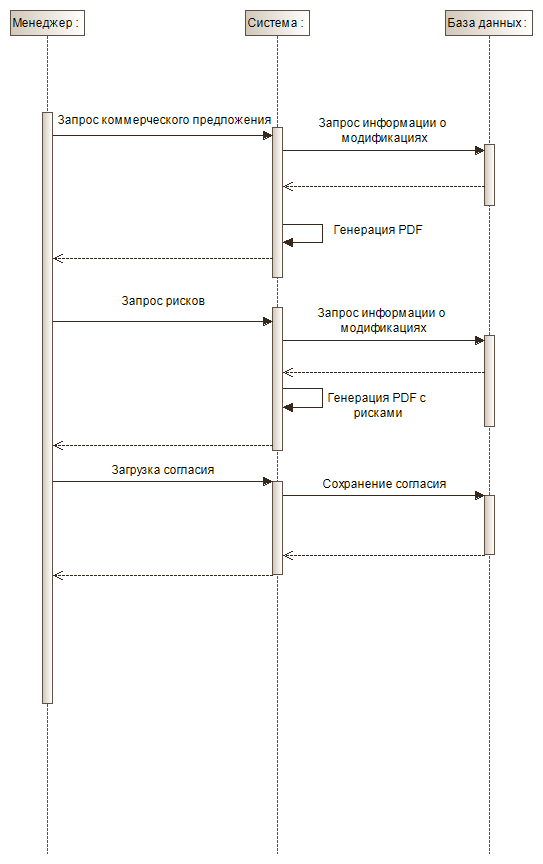
**State Machine Diagram**

Прецедент: Модификация генома

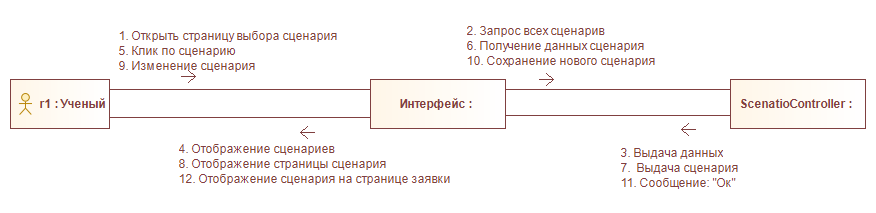


***Sequence Diagram 1***

Прецедент: Внесение согласия клиента

Предусловие: заявка на модификацию создана в системе 

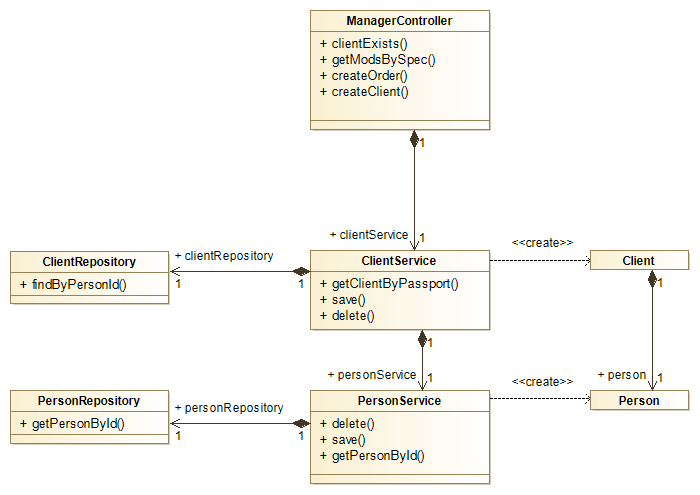
***Cooperative Diagram 1***

Прецедент: модификация существующего сценария****

**5.** **Logical View**

***Class Diagram 2***

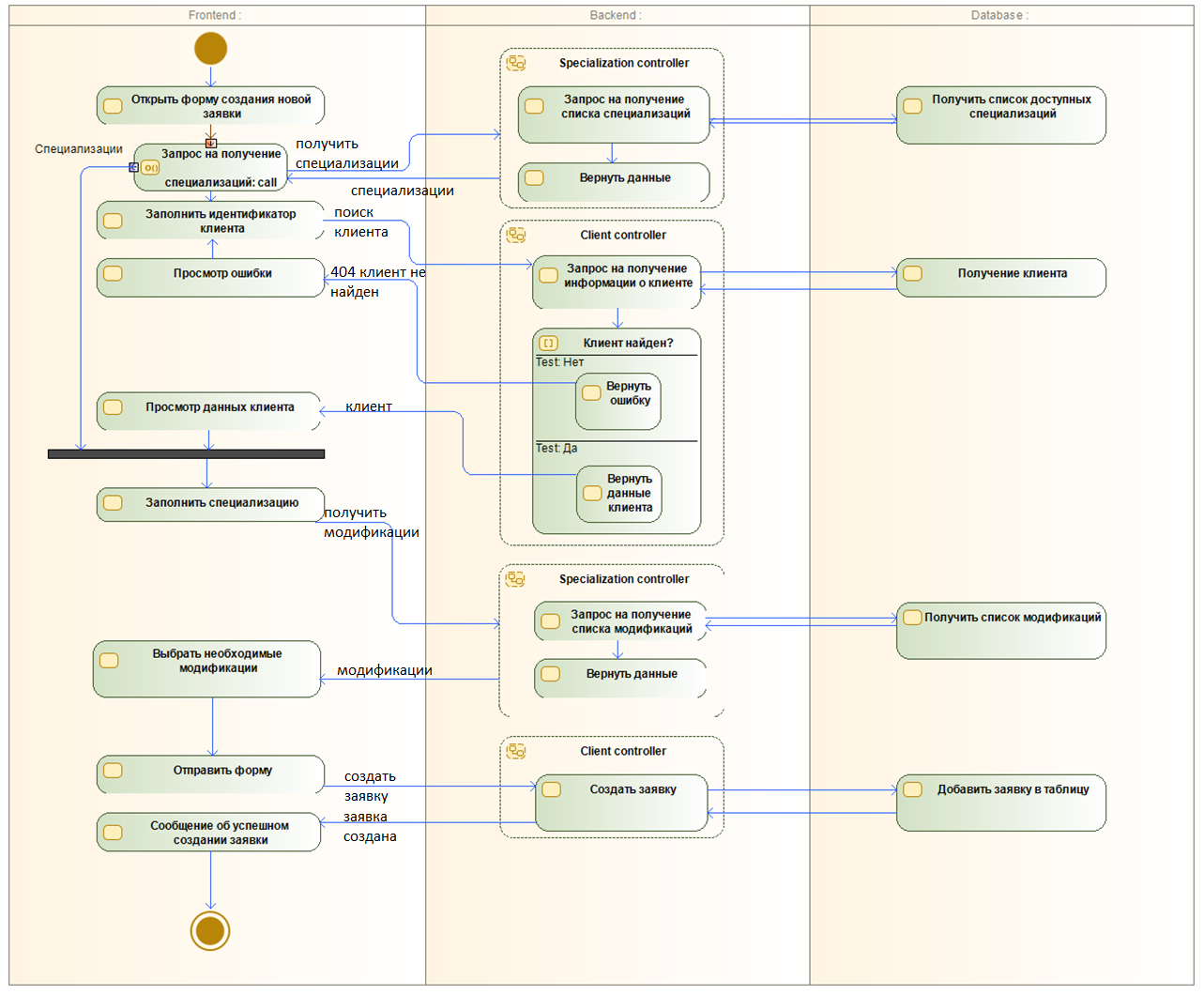
Прецедент: создание клиента

******

***Activity Diagram 2***

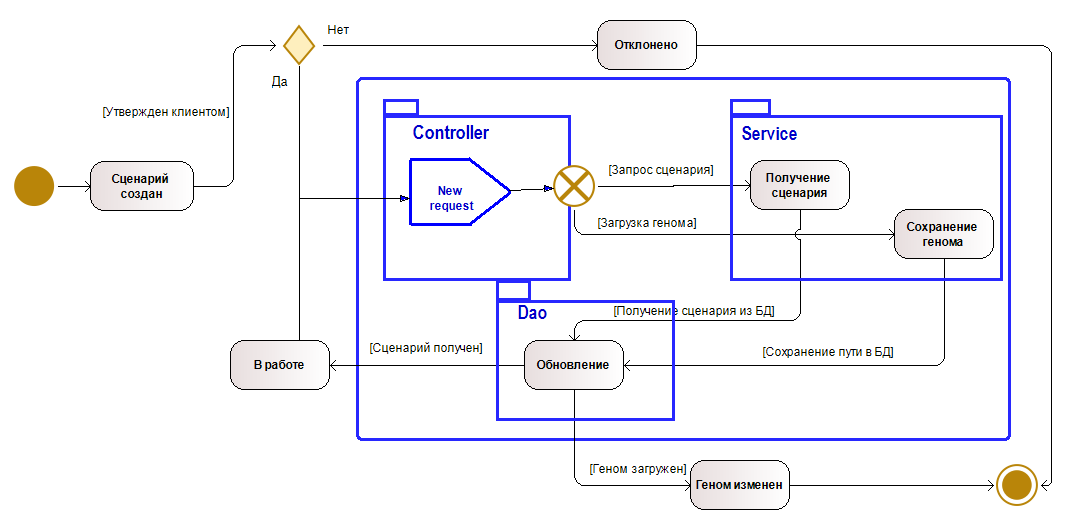
Прецедент: Создание заявки на модификацию

Предусловие: Менеджер вошел в систему



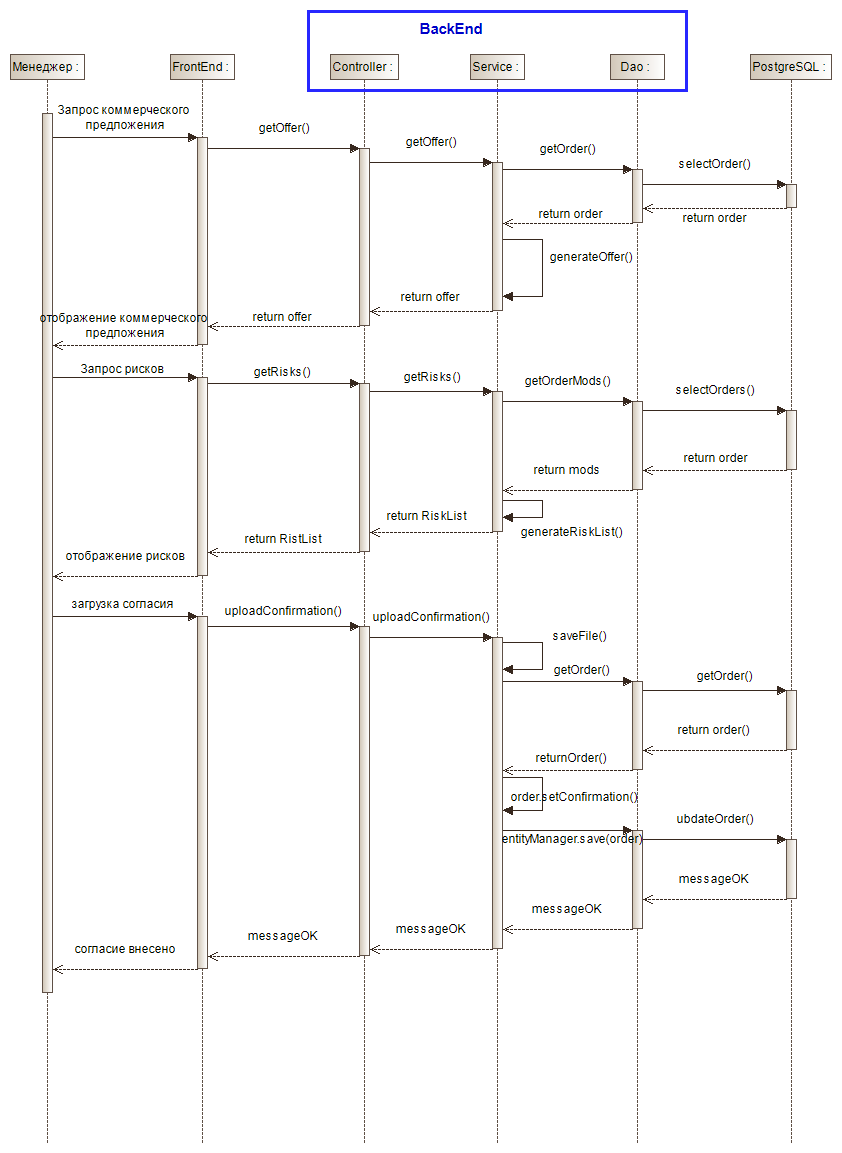
**State Machine Diagram**

Прецедент: Модификация генома

******

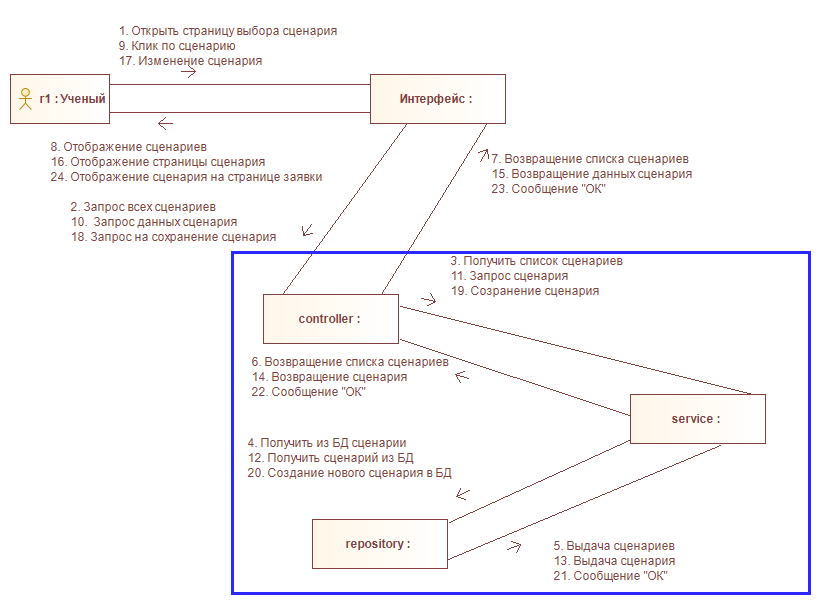
***Sequence Diagram 2***

Прецедент: Внесение согласия клиента

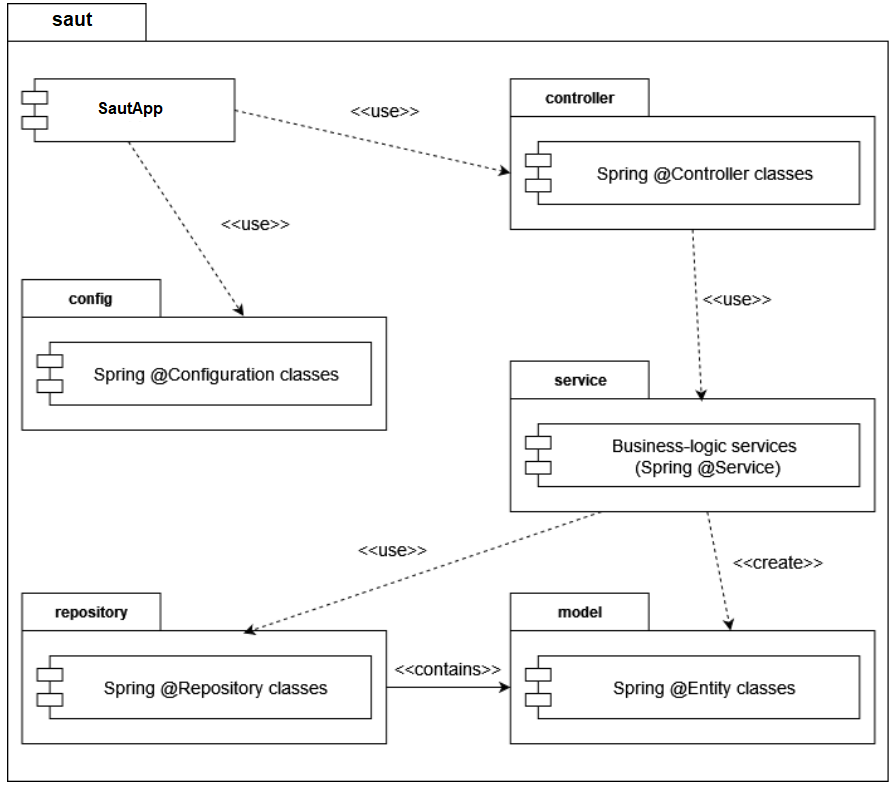


***Cooperative Diagram 2***

Прецедент: модификация существующего сценария



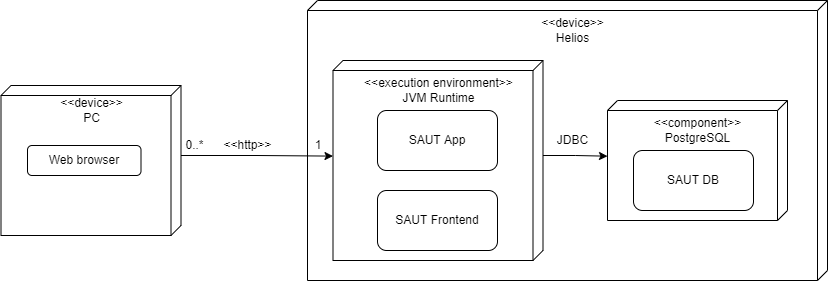
***Package Diagram***



**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.]*

**7.** **Deployment View**

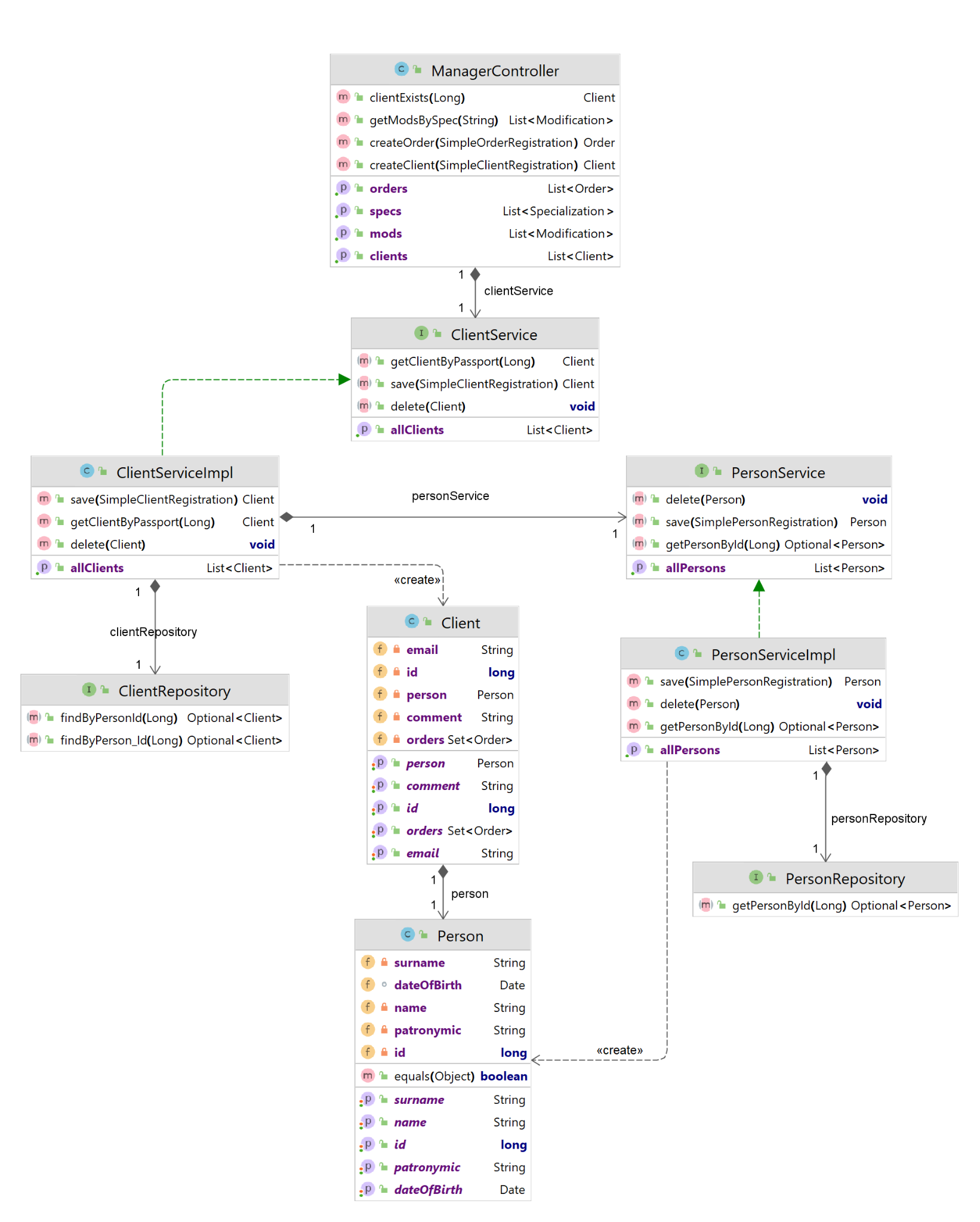


**8.** **Implementation View**

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

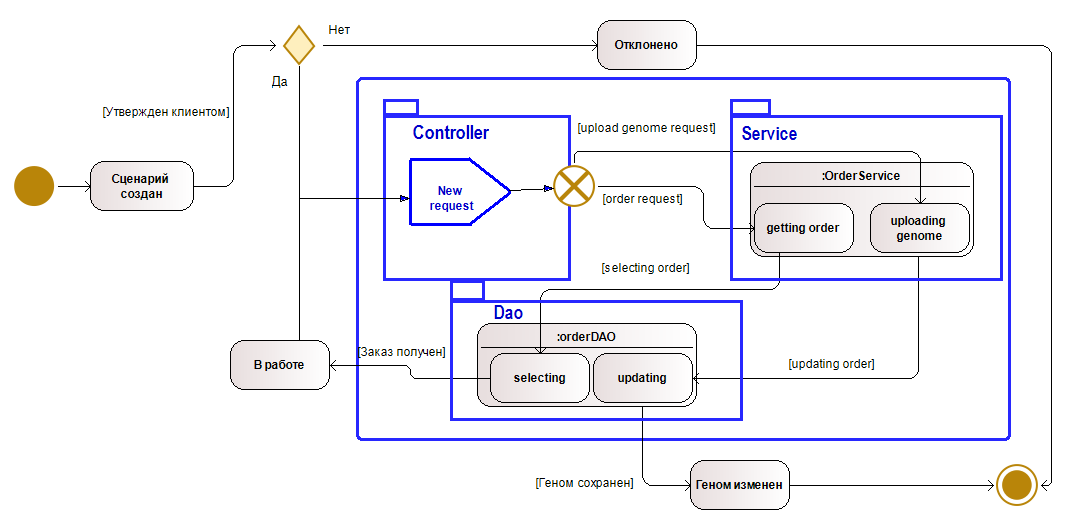
**Class Diagram 3**

*Прецедент - Создание клиента*

**

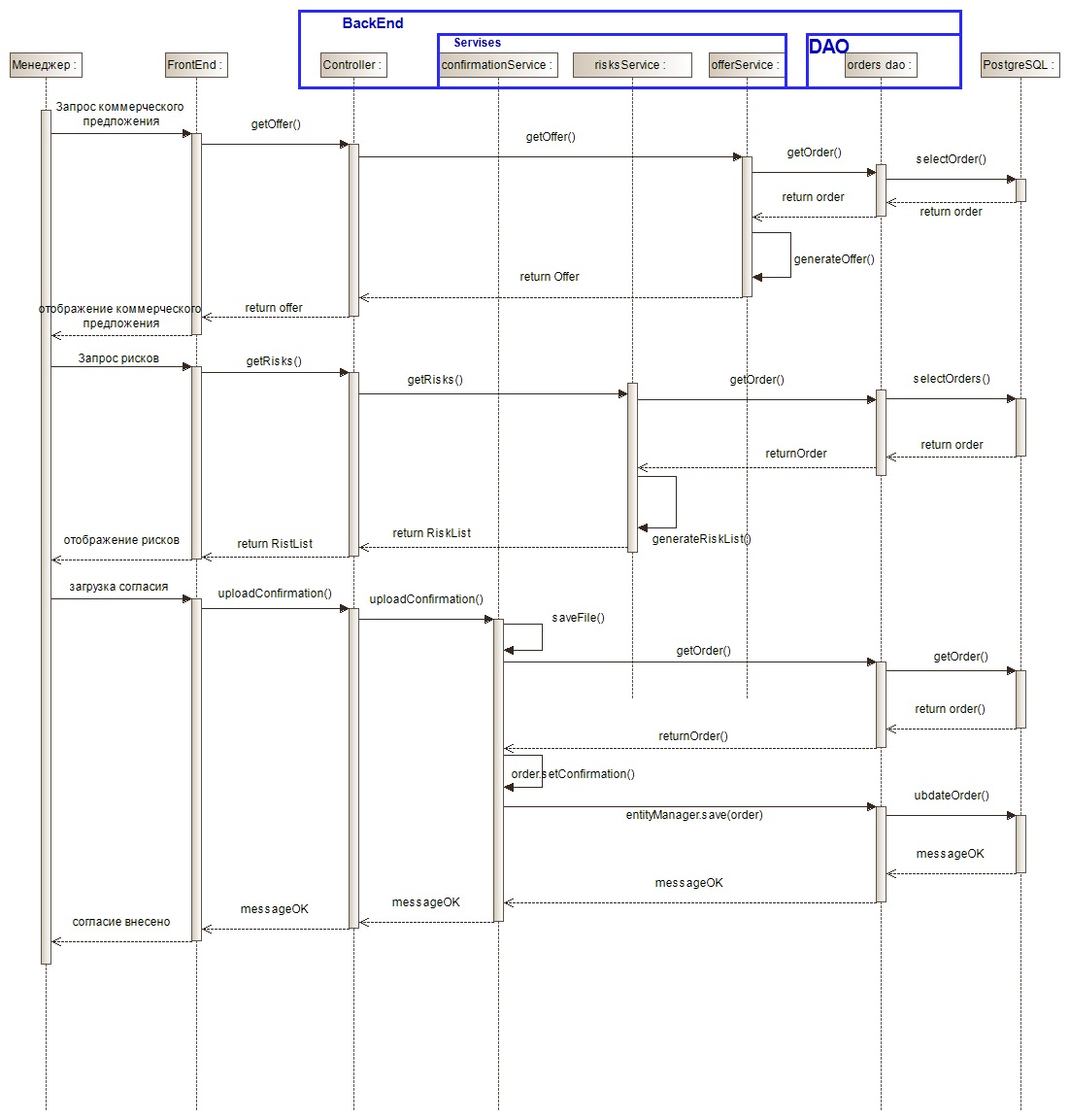
**State Machine Diagram**

Прецедент: Модификация генома

******

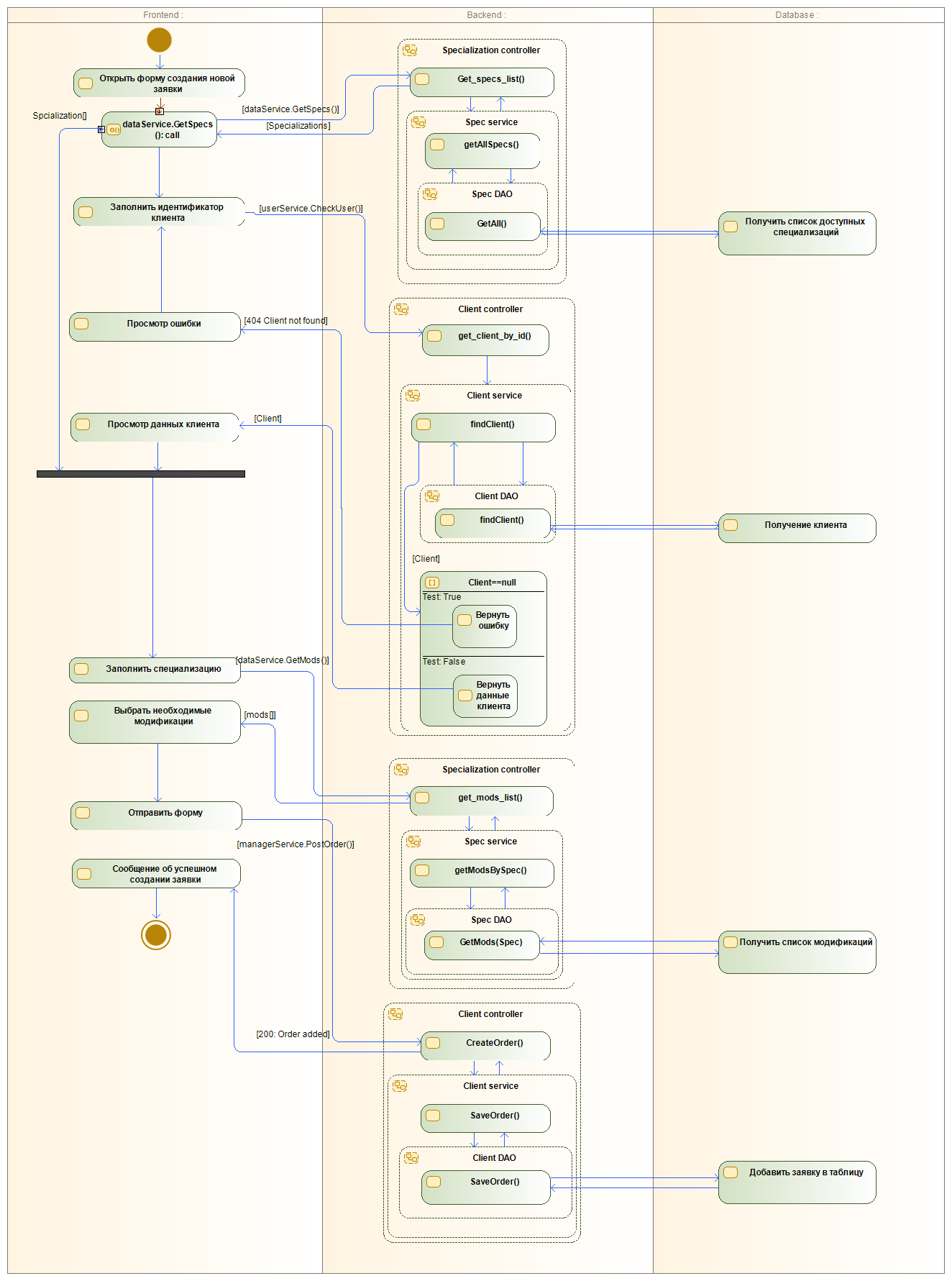
***Sequence Diagram 2***

Прецедент: Внесение согласия клиента

****

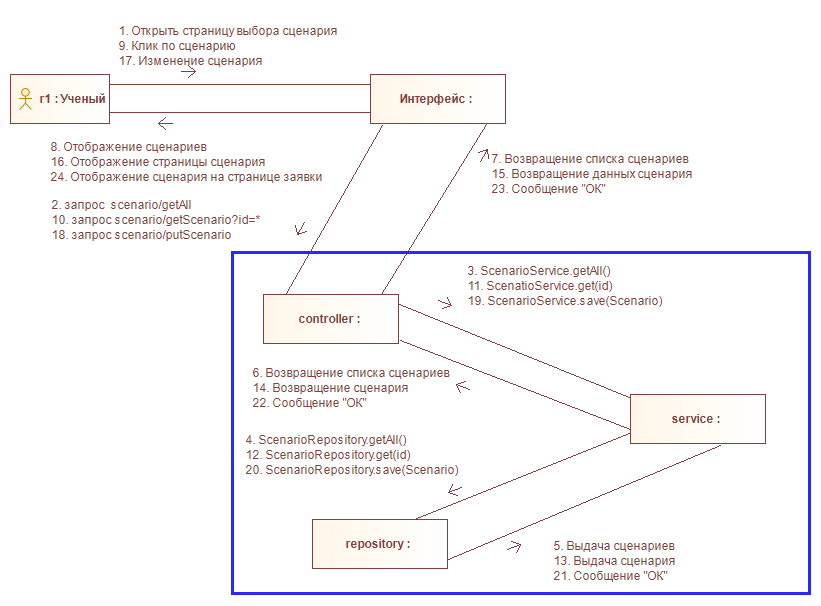
***Activity Diagram 2***

Прецедент: Создание заявки на модификацию

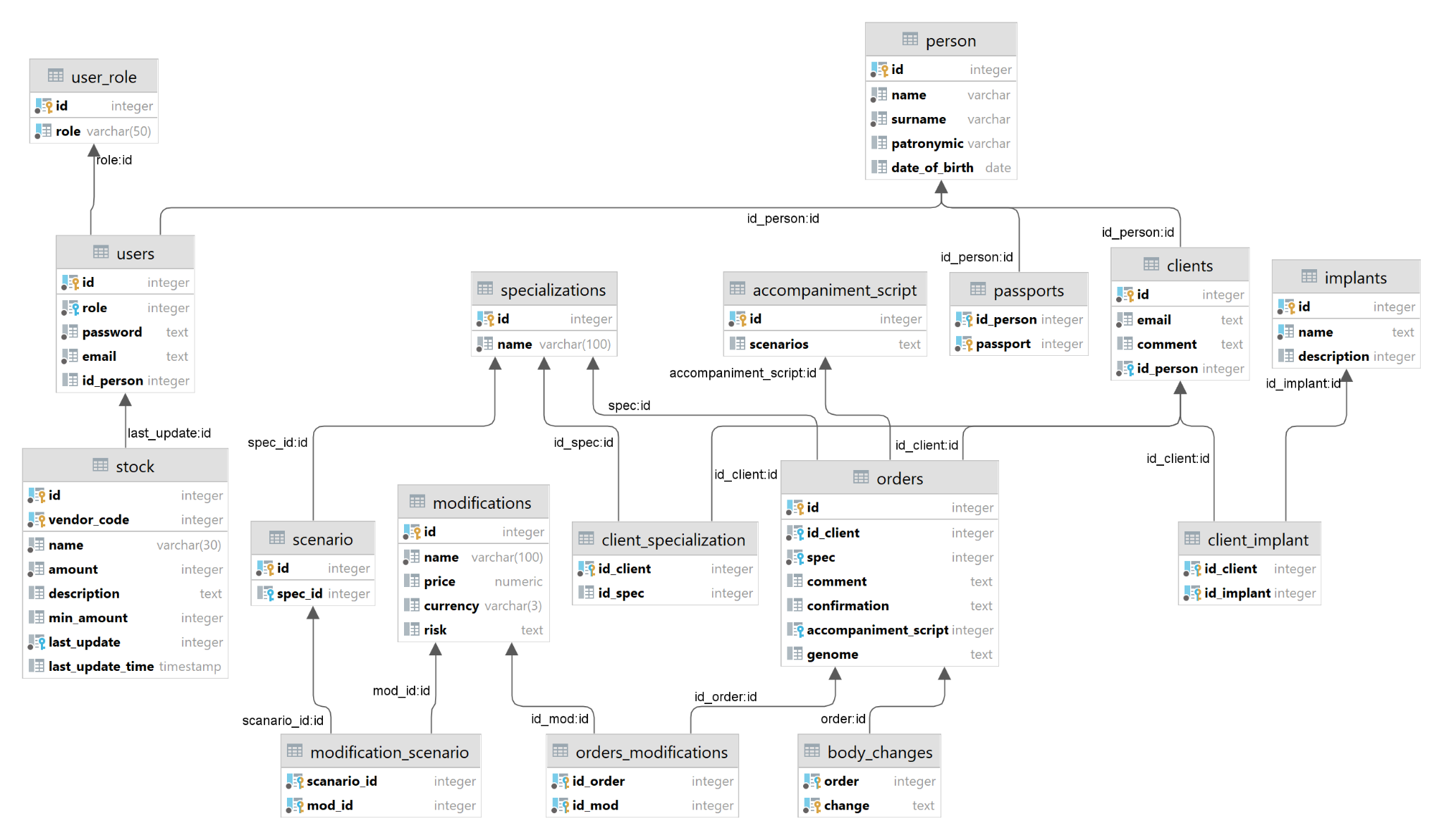
****

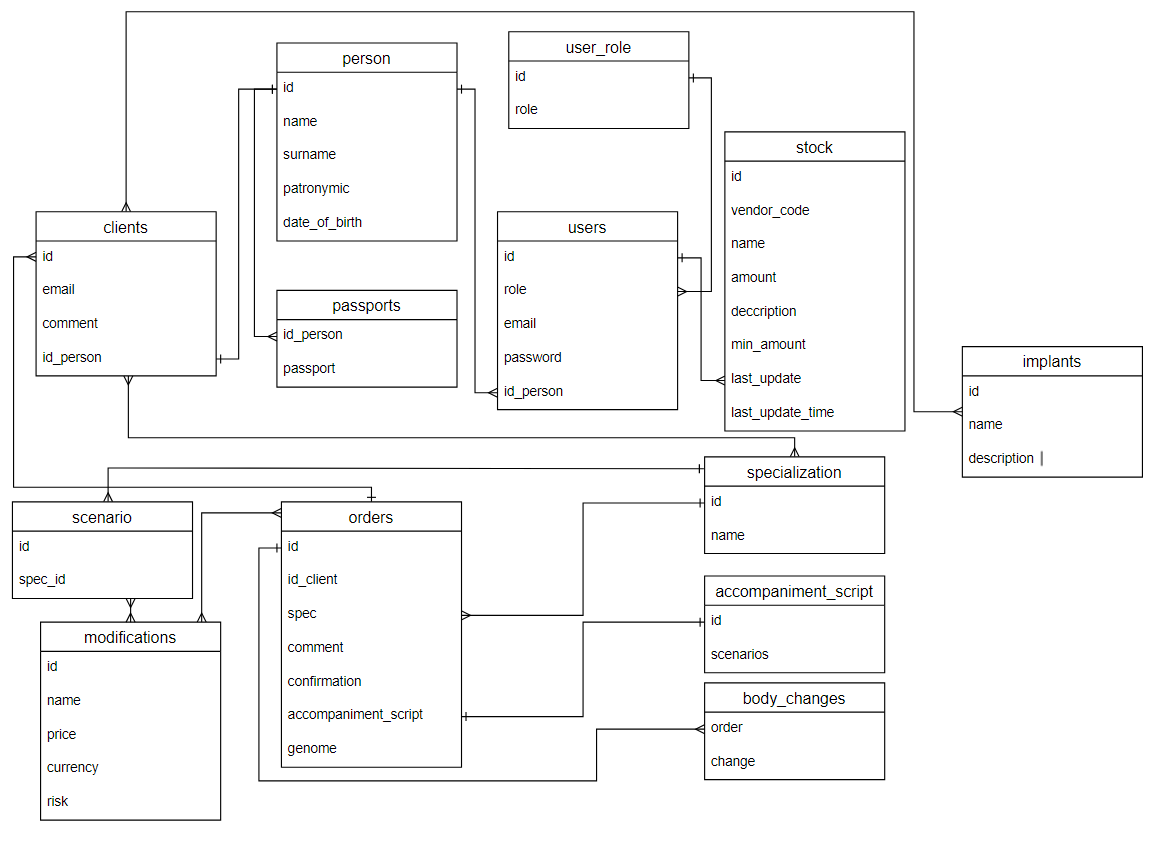
***Cooperative Diagram 2***

Прецедент: модификация существующего сценария

****

**DataBase Diagram**

****

****

**Описание сущностей**

1. Person - таблица людей информация о которых хранится в система
   1. id - уникальный идентификатор человека
   2. name, surname, patronymic - ФИО
   3. date\_of\_birth - дата рождения
2. passports - таблица, хранящая информацию о паспортах людей
   1. id\_person - уникальный идентификатор человека
   2. passport - номер документа
3. users - таблица, хранящая информацию о пользователях системы, к ним относятся люди напрямую взаимодействующие с системой
   1. id - уникальный идентификатор пользователя
   2. role - роль пользователя в системе
   3. email - корпоративный емэйл пользователя
   4. password - пароль пользователя, хранится в зашифрованном виде
   5. id\_person - уникальный идентификатор человека
4. user\_roles - таблица ролей пользователей в системе
   1. id - уникальный идентификатор роли
   2. role - название роли в системе
5. clients - таблица клиентов организации в которой работает система
   1. id - уникальный идентификатор клиента
   2. email - личный емэйл клиента
   3. comment - поле для ввода комментарий о клиента
   4. id\_person - уникальный идентификатор человека
6. orders - таблица заказов
   1. id - уникальный идентификатор заказа
   2. id\_client - уникальный идентификатор клиента
   3. spec - уникальный идентификатор специализации
   4. comment - поле для ввода комментарий о клиента
   5. confirmation - путь к файлу согласия клиента
   6. accompaniment\_script - уникальный идентификатор сценария сопровождения
   7. genome - путь к файлу согласия клиента
7. specialization - таблица специализаций
   1. id - уникальный идентификатор специализации
   2. name - название специализации
8. accompaniment\_script - таблица сценариев сопровождения
   1. id - уникальный идентификатор сценария
   2. scenarios - сценарий сопровождения в свободной форме
9. implants - таблица содержащая информацию о имплантах клиентов
   1. id - уникальный идентификатор импланта
   2. name - название импланта
   3. description - описание импланта
10. body\_changes - таблица содержащая информацию о изменениях произошедших в период окукливания клиента
    1. order - уникальный идентификатор заказа
    2. change - произошедшее изменение
11. scenario - таблица сценариев модификации
    1. id - уникальный идентификатор сценария
    2. spec\_id - уникальный идентификатор специализации
12. modification - таблица модификаций
    1. id - уникальный идентификатор модификации
    2. name - название модификации
    3. price - стоимость модификации
    4. currency - валюта в которой указана стоимость
    5. risk - список рисков, которые могут быть вызваны модификацией
13. stock - таблица по содержимому склада
    1. id - уникальный идентификатор товара
    2. vendor\_code - артикул
    3. name - наименование товара
    4. amount - количество товара на складе
    5. description - описание
    6. min\_amount - минимально необходимое количество
    7. last\_update - уникальный идентификатор пользователя последнего изменившего запись
    8. last\_update\_time - время последнего изменения записи

**9.** **Size and Performance (Производительность)**

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

* Система должна обеспечивать одновременную работу 250 уникальных пользователей
* Обработка пользовательского запроса должна занимать не более 5 секунд
* Система должна обрабатывать не менее 5 запросов в секунду

Для повышения производительности системы возможно применение сторонних средств фильтрации входящего трафика для снижения количества запросов, разработка собственных решений не подразумевается. Для обеспечения минимального времени обработки используется балансировка запросов. Для обеспечения балансировки, в пиковое время использования системы поднимаются несколько реплик приложения и запросы распределяются между ними.

**10.** **Quality (Качество)**

1. Система использует открытую платформу Spring Boot, зарекомендовавшую себя в качестве надежного инструмента построения серверных приложений.
2. Система поддерживает механизм транзакций. Ошибки в работе с одним клиентом не влияют на работу других.
3. Система поддерживает аутентификацию по логину и паролю для обеспечения безопасности используя открытую платформу Spring Security.
4. Минимизировано число запросов клиента к серверу. Используется REST API.
5. Минимизировано число запросов сервера к базе данных.