

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3
«Генерация диаграммы»
по дисциплине «Автоматизация научных исследований»

Выполнил
студент гр. № 5040102/50201

Худина А.А.

Преподаватель:

Новиков Ф.А.

Санкт-Петербург
2025 г.

ЗАДАНИЕ

1. Часть 1. Диаграмма вариантов использования

- Выбрать предметную область для проектирования информационной системы
- Определить основных действующих лиц системы
- Выявить основные функциональные требования
- Построить диаграмму вариантов использования с учетом следующих требований:
 1. Отобразить все действующие лица
 2. Определить основные и вспомогательные варианты использования
 3. Установить отношения между вариантами использования (include, extend)
 4. Добавить границу системы
 5. Обеспечить логическую целостность диаграммы

2. Часть 2. Диаграмма классов

- На основе выбранной предметной области разработать диаграмму классов
- Определить основные классы системы
- Установить иерархию наследования
- Определить типы связей между классами
- Указать атрибуты классов с типами данных
- Добавить кратности ассоциаций

Требования к выполнению

- Использовать PlantUML для генерации диаграмм
- Соблюдать стандарты UML
- Обеспечить логическую целостность диаграмм
- Предоставить комментарии к ключевым элементам диаграмм
- Проверить работоспособность сгенерированного кода
- Провести исследование для 1-2 моделей ИИ

Критерии оценки

- Корректность синтаксиса PlantUML
- Соответствие стандартам UML
- Полнота отображения функциональных требований
- Качество визуализации
- Логическая целостность диаграмм
- Обоснованность выбора предметной области

Форма отчёта

1. Исходный код PlantUML для обеих диаграмм
2. Сгенерированные диаграммы
3. Пояснительная записка с описанием:
 1. Выбранной предметной области
 2. Обоснования выбора действующих лиц
 3. Функциональных требований
 4. Структуры классов
4. Анализ результатов проектирования

Тема выбиралась с использованием **промпта**:

«Выбери информационную систему для проектирования - она должна соответствовать теме для научной работы (тоже выбери сам) для направления "01.04.02_02 «Математические методы анализа и визуализации данных»". Пропиши абзацем кратко про выбранную тему и абзацем про выбранную систему для проектирования. Под выбранной системой для проектирования понимается: что мы будем проектировать и для чего создавать диаграммы (например: Система управления библиотекой; Действующие лица: читатель, библиотекарь, администратор). Учти, что для этой системы необходимо будет в дальнейшем создать диаграмму классов и диаграмму вариантов использования, используя PlantUML. Пока просто напиши кратко выбранную тему, выбранную систему, которую проектируем и её действующие лица.»

В качестве нейросети был использован DeepSeek. Реализация кода PlantUML производилась в онлайн-сервисе: <https://editor.plantuml.com/>

Ответ нейросети:

«Тема научной работы:

«Разработка подхода к сегментации пользователей на основе анализа их поведения и эмоциональных откликов для повышения эффективности CRM-систем».

Информационная система для проектирования:

Система анализа клиентского опыта (Customer Experience Analysis System)

Действующие лица:

- Клиент — физическое лицо, взаимодействующее с компанией через различные каналы (сайт, приложение, кол-центр). Является источником данных о поведении и отзывах.
- Аналитик данных — специалист, который настраивает модель сегментации, запрашивает отчеты и анализирует полученные сегменты и закономерности.
- Менеджер по клиентскому опыту — использует систему для получения готовых инсайтов о клиентских сегментах для принятия решений по улучшению сервиса и персонализации коммуникаций.
- Администратор системы — отвечает за управление учетными записями, правами доступа и интеграцию системы с источниками данных (например, CRM, базы отзывов, системы веб-аналитики).»

Промпт:

Сгенерируй PlantUML код для диаграммы вариантов использования этой системы анализа клиентского опыта. Укажи действующие лица, перечисли основные случаи использования.

Ответ:

```
plantuml
@startuml
left to right direction
skinparam packageStyle rectangle

actor "Аналитик данных" as Analyst
actor "Маркетолог" as Marketer
actor "Администратор системы" as Admin

rectangle "Система управления клиентскими сегментами (CSMS)" {
    usecase "Управление моделями сегментации" as UC1
    usecase "Запуск расчета сегментов" as UC2
    usecase "Анализ результатов сегментации" as UC3
    usecase "Визуализация сегментов" as UC4
    usecase "Экспорт данных сегментов" as UC5

    usecase "Просмотр сегментов" as UC6
    usecase "Создание маркетинговой кампании" as UC7
    usecase "Назначение сегмента кампании" as UC8

    usecase "Управление пользователями" as UC9
    usecase "Настройка прав доступа" as UC10
    usecase "Интеграция с источниками данных" as UC11
    usecase "Мониторинг системы" as UC12

    usecase "Аутентификация в системе" as UC13
    usecase "Валидация входных данных" as UC14
    usecase "Генерация отчета" as UC15
}

' Связи акторов с use cases
Analyst --> UC1
Analyst --> UC2
Analyst --> UC3
Analyst --> UC4
Analyst --> UC5

Marketer --> UC6
Marketer --> UC7
Marketer --> UC8

Admin --> UC9
Admin --> UC10
Admin --> UC11
Admin --> UC12

' Общие use cases
Analyst --> UC13
Marketer --> UC13
Admin --> UC13

' Отношения include
UC2 .> UC14 : include
UC3 .> UC15 : include
UC7 .> UC15 : include

' Отношения extend
```

```
UC3 <.. UC4 : extend
UC7 <.. UC8 : extend

' Дополнительные связи
UC1 ..> UC2 : <<precedes>>
UC6 ..> UC7 : <<precedes>>
UC11 ..> UC2 : <<enables>>

@enduml
```

Пояснения к диаграмме вариантов использования:

Основные функциональности системы:

Для Аналитика данных:

- Управление моделями сегментации (создание, настройка параметров)
- Запуск расчетов сегментов на основе выбранной модели
- Анализ и интерпретация результатов сегментации
- Визуализация сегментов в виде графиков и диаграмм
- Экспорт данных для дальнейшего анализа

Для Маркетолога:

- Просмотр готовых клиентских сегментов
- Создание маркетинговых кампаний на основе сегментов
- Назначение конкретных сегментов целевой аудиторией кампаний

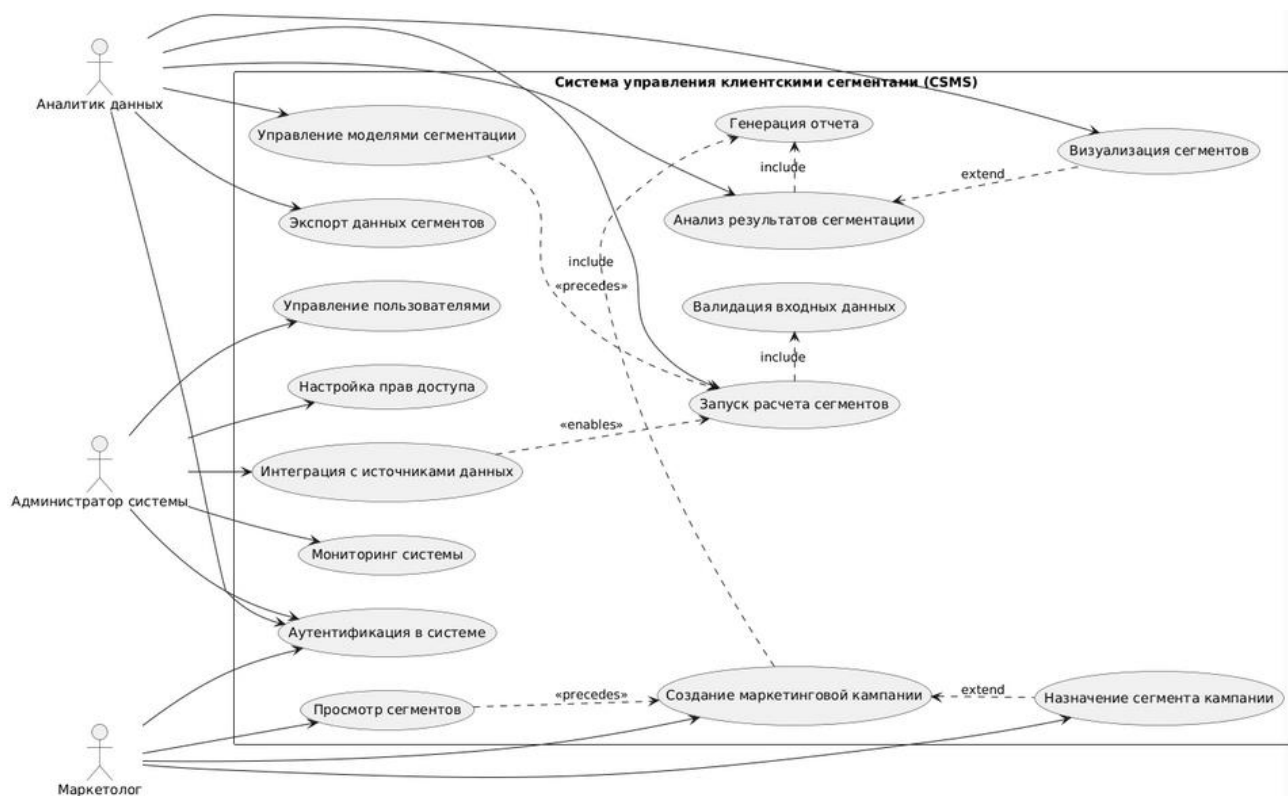
Для Администратора:

- Управление учетными записями пользователей
- Настройка прав доступа к функционалу системы
- Интеграция с внешними источниками данных (CRM, веб-аналитика)
- Мониторинг работоспособности системы

Ключевые отношения:

- Запуск расчета включает валидацию входных данных
- Анализ результатов и создание кампаний включают генерацию отчетов
- Визуализация расширяет функционал анализа результатов
- Назначение сегмента расширяет процесс создания кампаний

Диаграмма:



Промпт:

Тема научной работы:

«Разработка подхода к сегментации пользователей на основе анализа их поведения и эмоциональных откликов для повышения эффективности CRM-систем».

Информационная система для проектирования:

Система анализа клиентского опыта (Customer Experience Analysis System)

Действующие лица:

- Клиент — физическое лицо, взаимодействующее с компанией через различные каналы (сайт, приложение, кол-центр). Является источником данных о поведении и отзывах.
- Аналитик данных — специалист, который настраивает модель сегментации, запрашивает отчеты и анализирует полученные сегменты и закономерности.
- Менеджер по клиентскому опыту — использует систему для получения готовых инсайтов о клиентских сегментах для принятия решений по улучшению сервиса и персонализации коммуникаций.

Администратор системы — отвечает за управление учетными записями, правами доступа и интеграцию системы с источниками данных (например, CRM, базы отзывов, системы веб-аналитики).

Сгенерируй PlantUML код для диаграммы вариантов использования системы анализа клиентского опыта. Укажи действующие лица. Например, для библиотеки: читатель, библиотекарь, администратор. Перечисли основные случаи использования. Например, для библиотеки: такие как просмотр каталога, бронирование книг, продление срока пользования, добавление в избранное, выдача книг, принятие возврата, формирование отчёта, добавление книги в каталог, удаление книги из каталога, и др.

Ответ:

```
plantuml
@startuml
left to right direction

actor Клиент
actor "Аналитик данных" as Аналитик
actor "Менеджер по клиентскому опыту" as Менеджер
actor "Администратор системы" as Администратор

rectangle "Система анализа клиентского опыта" {
    usecase "Регистрация взаимодействий клиента" as UC1
    usecase "Сбор поведенческих данных" as UC2
    usecase "Анализ эмоциональных откликов" as UC3
    usecase "Интеграция с источниками данных" as UC4

    usecase "Настройка моделей сегментации" as UC5
    usecase "Запуск процесса сегментации" as UC6
    usecase "Анализ сегментов" as UC7
    usecase "Генерация отчетов" as UC8

    usecase "Просмотр инсайтов по сегментам" as UC9
    usecase "Анализ эффективности коммуникаций" as UC10
}
```

```

usecase "Формирование рекомендаций" as UC11

usecase "Управление пользователями" as UC12
usecase "Настройка прав доступа" as UC13
usecase "Мониторинг работы системы" as UC14
}

```

```

Клиент --> UC1
Клиент --> UC2
Клиент --> UC3

```

```

Аналитик --> UC5
Аналитик --> UC6
Аналитик --> UC7
Аналитик --> UC8

```

```

Менеджер --> UC9
Менеджер --> UC10
Менеджер --> UC11

```

```

Администратор --> UC4
Администратор --> UC12
Администратор --> UC13
Администратор --> UC14

```

```

UC1 .> UC2 : включает
UC2 .> UC3 : включает
UC5 .> UC6 : включает
UC6 .> UC7 : включает
UC7 .> UC8 : включает

```

```
@enduml
```

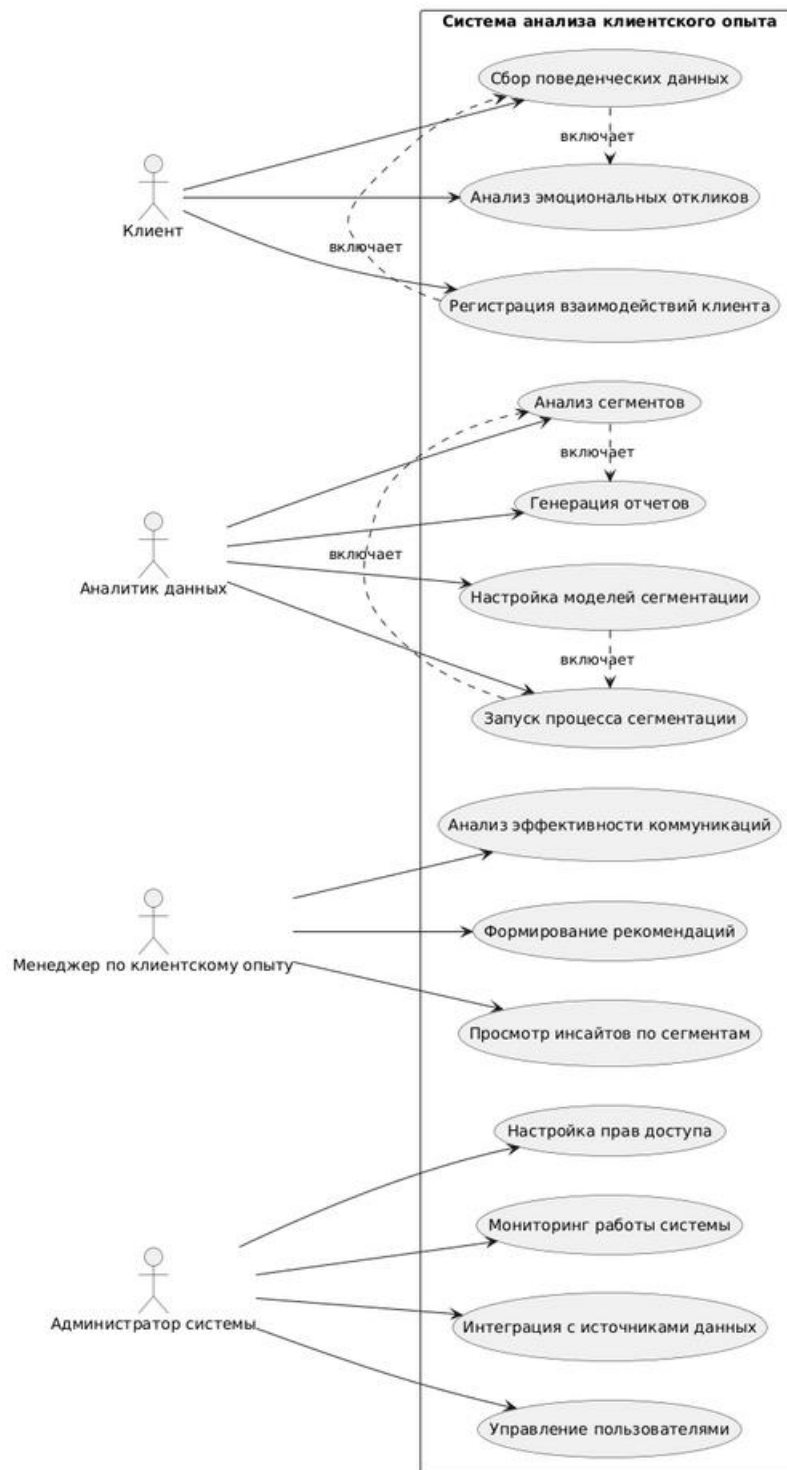
Пояснения к диаграмме:

Основные варианты использования:

1. Для Клиента:
 - Регистрация взаимодействий с компанией
 - Генерация поведенческих данных
 - Создание эмоциональных откликов (отзывы, оценки)
2. Для Аналитика данных:
 - Настройка моделей сегментации
 - Запуск процесса сегментации
 - Анализ полученных сегментов
 - Генерация отчетов и статистики
3. Для Менеджера по клиентскому опыту:
 - Просмотр готовых инсайтов по сегментам
 - Анализ эффективности коммуникаций
 - Формирование рекомендаций по улучшению сервиса
4. Для Администратора системы:
 - Интеграция с источниками данных (CRM, веб-аналитика)
 - Управление учетными записями пользователей
 - Настройка прав доступа
 - Мониторинг работы системы

Диаграмма показывает основные функциональные возможности системы и взаимодействие между различными ролями пользователей.

Диаграмма:



Промпт:

Тема научной работы:

«Разработка подхода к сегментации пользователей на основе анализа их поведения и эмоциональных откликов для повышения эффективности CRM-систем».

Информационная система для проектирования:

Система анализа клиентского опыта (Customer Experience Analysis System)

Действующие лица:

- Клиент — физическое лицо, взаимодействующее с компанией через различные каналы (сайт, приложение, кол-центр). Является источником данных о поведении и отзывах.
- Аналитик данных — специалист, который настраивает модель сегментации, запрашивает отчеты и анализирует полученные сегменты и закономерности.
- Менеджер по клиентскому опыту — использует систему для получения готовых инсайтов о клиентских сегментах для принятия решений по улучшению сервиса и персонализации коммуникаций.

Администратор системы — отвечает за управление учетными записями, правами доступа и интеграцию системы с источниками данных (например, CRM, базы отзывов, системы веб-аналитики).

Сгенерируй PlantUML код для диаграммы вариантов использования системы анализа клиентского опыта. Укажи действующие лица и перечисли случаи использования по ролям.

Дальше сделай как тут ниже написано (но тут было для библиотеки).

Для читателя перечисли действия, связанные с взаимодействием с книгами: поиск книг, просмотр каталога, бронирование, продление срока, управление избранными. Для библиотекаря перечисли действия, связанные с обслуживанием читателей: оформление выдачи, приём возврата, создание отчётов и работа с каталогом.

Для администратора перечисли действия, связанные с управлением системой: контроль каталога, просмотр системной информации и доступ к отчётам.

Отобрази все связи между действующими лицами и соответствующими вариантами использования. Включи границу системы.

Ответ:

```
plantuml
@startuml
left to right direction

actor "Клиент" as Client
actor "Аналитик данных" as Analyst
actor "Менеджер по клиентскому опыту" as Manager
actor "Администратор системы" as Admin

rectangle "Система анализа клиентского опыта" {
    usecase "Предоставлять данные о поведении" as UC1
}
```

```

usecase "Оставлять отзывы" as UC2

usecase "Настраивать модель сегментации" as UC3
usecase "Запрашивать отчеты" as UC4
usecase "Анализировать сегменты" as UC5
usecase "Выявлять закономерности" as UC6

usecase "Получать инсайты о сегментах" as UC7
usecase "Принимать решения по улучшению сервиса" as UC8
usecase "Персонализировать коммуникации" as UC9

usecase "Управлять учетными записями" as UC10
usecase "Управлять правами доступа" as UC11
usecase "Интегрировать с источниками данных" as UC12
}

Client --> UC1
Client --> UC2

Analyst --> UC3
Analyst --> UC4
Analyst --> UC5
Analyst --> UC6

Manager --> UC7
Manager --> UC8
Manager --> UC9

Admin --> UC10
Admin --> UC11
Admin --> UC12

@enduml

```

Описание взаимодействия по ролям:

Для Клиента:

- Предоставлять данные о поведении - автоматическая передача данных через различные каналы взаимодействия (сайт, приложение, кол-центр)
- Оставлять отзывы - выражение эмоциональных реакций и обратной связи о качестве обслуживания

Для Аналитика данных:

- Настраивать модель сегментации - определение параметров и критериев для кластеризации пользователей
- Запрашивать отчеты - формирование аналитических отчетов по сегментам клиентов
- Анализировать сегменты - изучение характеристик выделенных групп пользователей
- Выявлять закономерности - обнаружение взаимосвязей между поведением и эмоциональными откликами

Для Менеджера по клиентскому опыту:

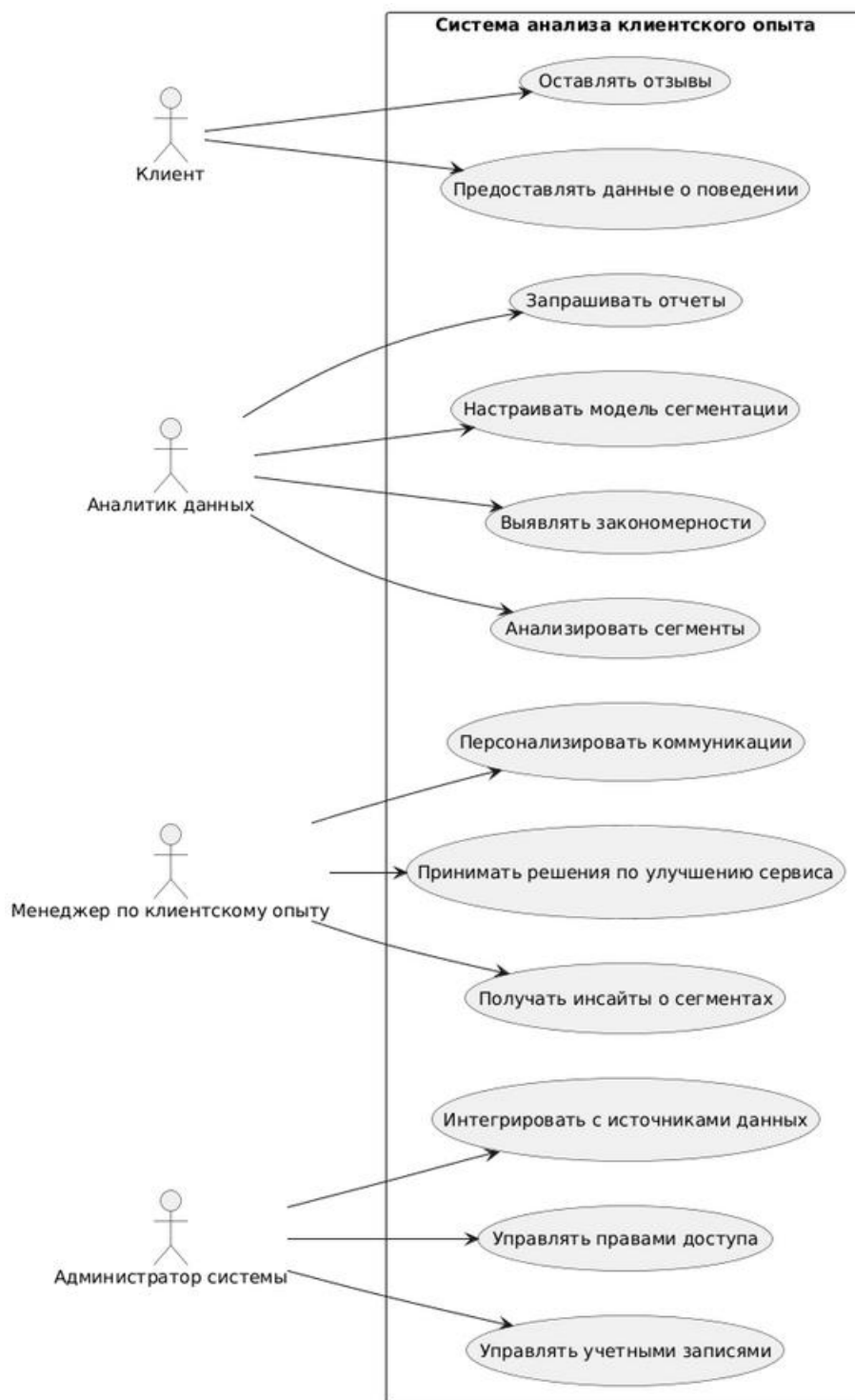
- Получать инсайты о сегментах - доступ к готовым аналитическим выводам о клиентских группах
- Принимать решения по улучшению сервиса - использование аналитики для оптимизации клиентского опыта
- Персонализировать коммуникации - настройка взаимодействия с разными сегментами пользователей

Для Администратора системы:

- Управлять учетными записями - создание, редактирование и удаление учетных записей пользователей системы
- Управлять правами доступа - настройка ролей и разрешений для разных категорий пользователей
- Интегрировать с источниками данных - подключение и настройка интеграции с CRM, базами отзывов, системами веб-аналитики

Все связи между действующими лицами и соответствующими вариантами использования отображены на диаграмме, система ограничена прямоугольной границей.

Диаграмма:



Промпт:

Тема научной работы:

«Разработка подхода к сегментации пользователей на основе анализа их поведения и эмоциональных откликов для повышения эффективности CRM-систем».

Информационная система для проектирования:

Система анализа клиентского опыта (Customer Experience Analysis System)

Действующие лица:

- Клиент — физическое лицо, взаимодействующее с компанией через различные каналы (сайт, приложение, кол-центр). Является источником данных о поведении и отзывах.
- Аналитик данных — специалист, который настраивает модель сегментации, запрашивает отчеты и анализирует полученные сегменты и закономерности.
- Менеджер по клиентскому опыту — использует систему для получения готовых инсайтов о клиентских сегментах для принятия решений по улучшению сервиса и персонализации коммуникаций.

Администратор системы — отвечает за управление учетными записями, правами доступа и интеграцию системы с источниками данных (например, CRM, базы отзывов, системы веб-аналитики).

Выполни следующее задание: «Сгенерируй PlantUML код для диаграммы вариантов использования системы управления библиотекой. Укажи действующие лица: Читатель, Библиотекарь, Администратор (Администратор наследует все функции Библиотекаря). Отобрази границу системы и все связи действующих лиц с вариантами использования».

Варианты использования для читателя:

- Просмотреть каталог
- Искать книгу
- Бронировать книгу

include → Проверка наличия книги

- Продлить срок пользования

include → Проверка задолженности

- Добавить книгу в избранное
- Просмотреть свои бронирования

Варианты использования для библиотекаря:

- Выдать книгу

include → Проверка наличия книги

extend → Резервирование книги (если книга недоступна)

- Принять возврат книги

include → Проверка задолженности

extend → Начисление штрафа за просрочку (если книга возвращена с опозданием)

- Добавить книгу в каталог
- Формировать отчёт по книгам

include → Просмотр задолженностей читателей

Варианты использования для администратора:

- Просматривать отчёты
- Удалять книги из каталога
- Мониторить состояние системы

»

Но, учти, что тут задание под библиотеку. Сделай все тоже самое, но сгенерируй PlantUML код для диаграммы вариантов использования системы анализа клиентского опыта.

Ответ:

```
plantuml
@startuml
left to right direction

actor Клиент
actor "Аналитик данных" as Analyst
actor "Менеджер по клиентскому опыту" as Manager
actor "Администратор системы" as Admin

Admin <|-- Analyst
Admin <|-- Manager

rectangle "Система анализа клиентского опыта" {
    usecase "Оставлять отзывы и feedback" as UC1
    usecase "Взаимодействовать с компанией" as UC2

    usecase "Настраивать модель сегментации" as UC3
    usecase "Запрашивать аналитические отчеты" as UC4
    usecase "Анализировать сегменты и закономерности" as UC5

    usecase "Получать инсайты о клиентских сегментах" as UC6
    usecase "Принимать решения по улучшению сервиса" as UC7
    usecase "Персонализировать коммуникации" as UC8

    usecase "Управлять учетными записями" as UC9
    usecase "Управлять правами доступа" as UC10
    usecase "Интегрировать с источниками данных" as UC11
    usecase "Мониторить состояние системы" as UC12
}

Клиент --> UC1
Клиент --> UC2

Analyst --> UC3
Analyst --> UC4
Analyst --> UC5

Manager --> UC6
Manager --> UC7
Manager --> UC8

Admin --> UC9
```

```
Admin --> UC10
Admin --> UC11
Admin --> UC12
```

```
UC4 .> UC3 : include
UC5 .> UC4 : include
UC7 .> UC6 : include
UC8 .> UC6 : include
UC11 .> UC12 : include
```

@enduml

Объяснение диаграммы:

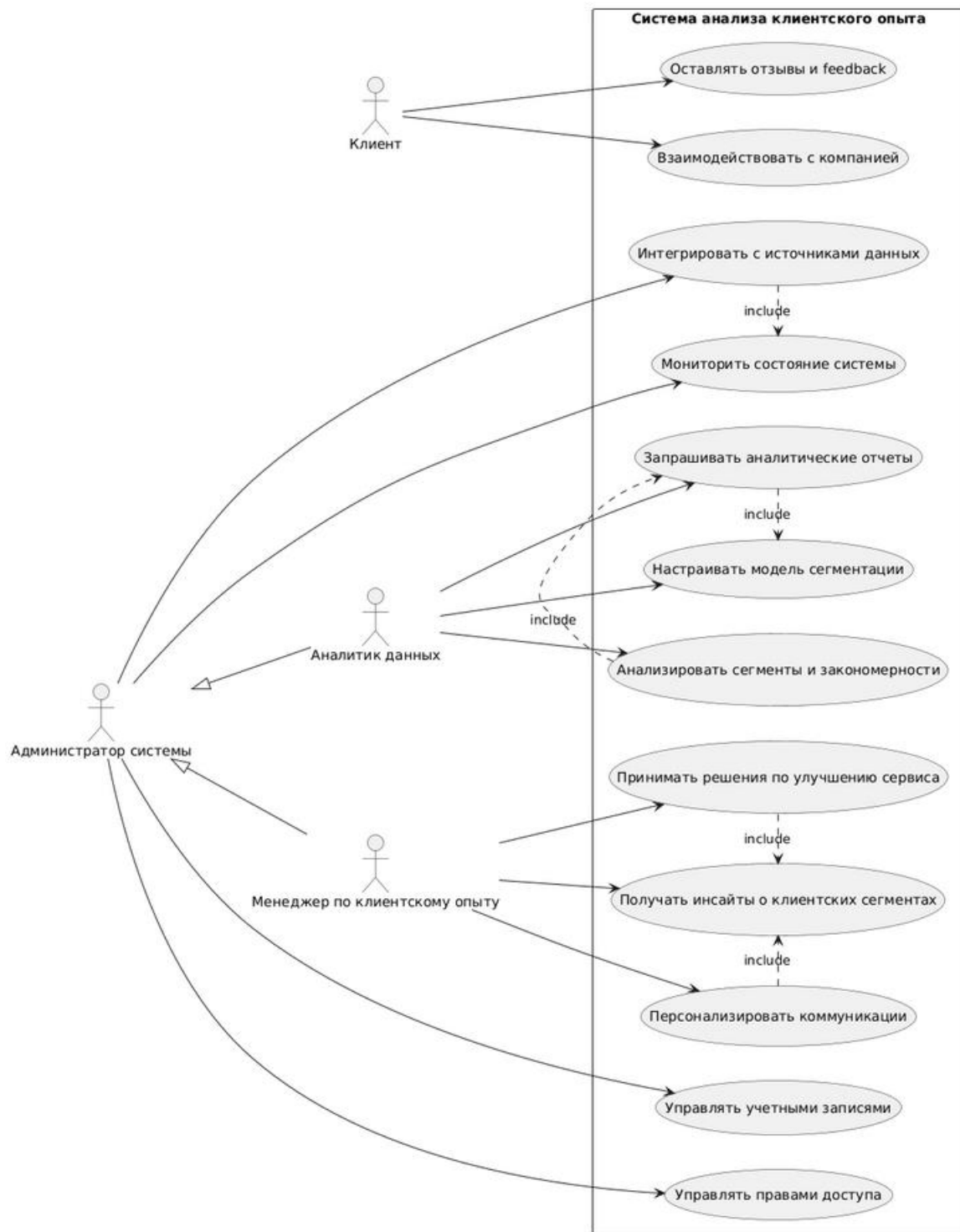
Действующие лица:

- Клиент - оставляет отзывы и взаимодействует с компанией
- Аналитик данных - настраивает модели, запрашивает отчеты, анализирует данные
- Менеджер по клиентскому опыту - получает инсайты, принимает решения, персонализирует коммуникации
- Администратор системы - управляет учетками, правами, интеграциями и мониторингом

Ключевые связи:

- Администратор наследует права Аналитика и Менеджера
- Аналитические отчеты включают настройку моделей
- Анализ сегментов включает работу с отчетами
- Решения по сервису и персонализация включают получение инсайтов
- Интеграция данных включает мониторинг системы

Диаграмма:



Промпт:

Тема научной работы:

«Разработка подхода к сегментации пользователей на основе анализа их поведения и эмоциональных откликов для повышения эффективности CRM-систем».

Информационная система для проектирования:

Система анализа клиентского опыта (Customer Experience Analysis System)

Действующие лица:

- Клиент — физическое лицо, взаимодействующее с компанией через различные каналы (сайт, приложение, кол-центр). Является источником данных о поведении и отзывах.
- Аналитик данных — специалист, который настраивает модель сегментации, запрашивает отчеты и анализирует полученные сегменты и закономерности.
- Менеджер по клиентскому опыту — использует систему для получения готовых инсайтов о клиентских сегментах для принятия решений по улучшению сервиса и персонализации коммуникаций.

Администратор системы — отвечает за управление учетными записями, правами доступа и интеграцию системы с источниками данных (например, CRM, базы отзывов, системы веб-аналитики).

Выполни следующее задание: «Сгенерируй PlantUML код для диаграммы классов системы управления библиотекой. Укажи основные классы, покажи связи между ними.»

Но, учти, что тут задание под библиотеку. Сделай все тоже самое, но сгенерируй PlantUML код для диаграммы классов системы анализа клиентского опыта.

Ответ:

plantuml

@startuml

' Диаграмма классов для системы анализа клиентского опыта (Customer Experience Analysis System)

```
class Customer {  
    - customerId: String  
    - name: String  
    - contactInfo: String  
    - demographicData: Map<String, Object>  
    - registrationDate: Date  
    + getBehavioralData()  
    + getFeedbackHistory()  
    + getCustomerSegments()  
}
```

```
class Interaction {  
    - interactionId: String  
    - customerId: String  
    - channel: String  
    - timestamp: Date  
    - interactionType: String  
    - duration: Integer  
    - content: String
```

```

    + calculateEmotionalScore()
    + getInteractionMetrics()
}

class Feedback {
    - feedbackId: String
    - customerId: String
    - source: String
    - rating: Integer
    - text: String
    - sentimentScore: Float
    - timestamp: Date
    + analyzeSentiment()
    + extractKeywords()
}

class EmotionalResponse {
    - responseId: String
    - customerId: String
    - interactionId: String
    - emotionalState: String
    - intensity: Float
    - triggers: List<String>
    + calculateIntensity()
    + identifyTriggers()
}

class BehavioralData {
    - dataId: String
    - customerId: String
    - pageViews: Integer
    - sessionDuration: Integer
    - purchaseHistory: List<String>
    - engagementScore: Float
    + calculateEngagement()
    + trackBehaviorPatterns()
}

class CustomerSegment {
    - segmentId: String
    - segmentName: String
    - criteria: Map<String, Object>
    - segmentSize: Integer
    - characteristics: List<String>
    + addCustomer()
    + updateCriteria()
    + generateInsights()
}

class SegmentationModel {
    - modelId: String
    - modelName: String
    - algorithm: String
    - parameters: Map<String, Object>
    - accuracy: Float
    + trainModel()
    + predictSegment()
    + evaluatePerformance()
}

class AnalyticsReport {
    - reportId: String
    - generatedBy: String
    - generationDate: Date
    - segmentInsights: List<String>
    - recommendations: List<String>

```

```

- metrics: Map<String, Float>
+ generateReport()
+ exportReport()
}

class DataSource {
- sourceId: String
- sourceType: String
- connectionString: String
- lastSync: Date
- dataSchema: String
+ connect()
+ synchronizeData()
+ validateData()
}

' Связи между классами
Customer ||--o{ Interaction : has
Customer ||--o{ Feedback : provides
Customer ||--o{ EmotionalResponse : generates
Customer ||--o{ BehavioralData : produces
Customer }o--o{ CustomerSegment : belongs_to

Interaction ||--|| EmotionalResponse : triggers
Feedback ||--|| EmotionalResponse : influences

BehavioralData }|--|| Customer : associated_with
CustomerSegment }o--|| SegmentationModel : created_by

SegmentationModel ||--o{ AnalyticsReport : generates
DataSource ||--o{ BehavioralData : supplies
DataSource ||--o{ Feedback : provides
DataSource ||--o{ Interaction : feeds

AnalyticsReport ||..|| CustomerSegment : analyzes

@enduml

```

Этот код PlantUML создает диаграмму классов для системы анализа клиентского опыта, которая включает:

Основные классы:

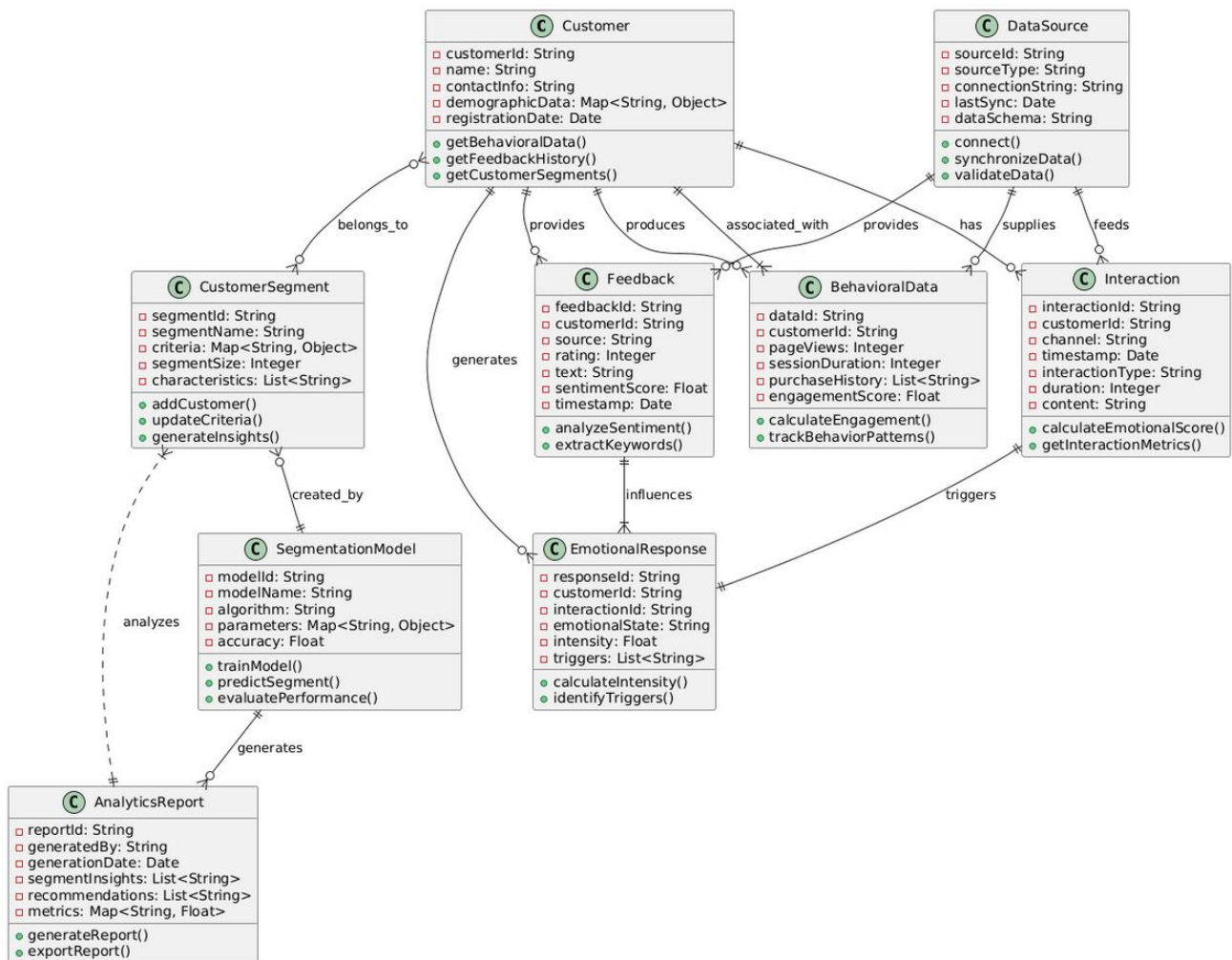
- Customer - представляет клиента с демографическими данными
- Interaction - фиксирует взаимодействия через различные каналы
- Feedback - хранит отзывы и оценки клиентов
- EmotionalResponse - анализирует эмоциональные реакции
- BehavioralData - отслеживает поведенческие паттерны
- CustomerSegment - определяет сегменты клиентов
- SegmentationModel - реализует алгоритмы сегментации
- AnalyticsReport - генерирует отчеты и инсайты
- DataSource - управляет интеграцией с внешними системами

Связи между классами:

- Агрегация и композиция между основными сущностями
- Ассоциации для отражения бизнес-логики
- Зависимости между аналитическими компонентами

Диаграмма отражает ключевые аспекты системы анализа клиентского опыта и может быть расширена дополнительными атрибутами и методами по мере необходимости.

Диаграмма:



Промпт:

Тема научной работы:

«Разработка подхода к сегментации пользователей на основе анализа их поведения и эмоциональных откликов для повышения эффективности CRM-систем».

Информационная система для проектирования:

Система анализа клиентского опыта (Customer Experience Analysis System)

Действующие лица:

- Клиент — физическое лицо, взаимодействующее с компанией через различные каналы (сайт, приложение, кол-центр). Является источником данных о поведении и отзывах.
- Аналитик данных — специалист, который настраивает модель сегментации, запрашивает отчеты и анализирует полученные сегменты и закономерности.
- Менеджер по клиентскому опыту — использует систему для получения готовых инсайтов о клиентских сегментах для принятия решений по улучшению сервиса и персонализации коммуникаций.

Администратор системы — отвечает за управление учетными записями, правами доступа и интеграцию системы с источниками данных (например, CRM, базы отзывов, системы веб-аналитики).

Выполни следующее задание: «Сгенерируй PlantUML код для диаграммы классов системы управления библиотекой. Укажи такие классы, как Человек, Читатель, Библиотекарь, Администратор, Книга, Выдача, Бронирование, и т.д. Покажи связи между ними и укажи базовые атрибуты классов (например, имя пользователя, название книги, дата выдачи).

»

Но, учти, что тут задание под библиотеку. Сделай все тоже самое, но сгенерируй PlantUML код для диаграммы классов системы анализа клиентского опыта.

Ответ:

plantuml

@startuml Customer Experience Analysis System Class Diagram

```
class Customer {
    - customerId: String
    - name: String
    - email: String
    - phone: String
    - registrationDate: Date
    - segment: String
    + updateProfile()
    + getInteractions()
}
```

```
class Interaction {
    - interactionId: String
    - customerId: String
    - channel: String
}
```

```

    - timestamp: DateTime
    - type: String
    - duration: int
    + recordInteraction()
    + getDetails()
}

class Feedback {
    - feedbackId: String
    - customerId: String
    - interactionId: String
    - rating: int
    - comment: Text
    - timestamp: DateTime
    + submitFeedback()
    + analyzeSentiment()
}

class EmotionData {
    - emotionId: String
    - customerId: String
    - interactionId: String
    - emotionType: String
    - intensity: float
    - timestamp: DateTime
    - source: String
    + detectEmotion()
    + getEmotionTrend()
}

class SegmentationModel {
    - modelId: String
    - name: String
    - parameters: JSON
    - createdBy: String
    - creationDate: Date
    - status: String
    + trainModel()
    + evaluateModel()
    + applySegmentation()
}

class CustomerSegment {
    - segmentId: String
    - name: String
    - description: Text
    - criteria: JSON
    - size: int
    - creationDate: Date
    + addCustomer()
    + removeCustomer()
    + getInsights()
}

class AnalysisReport {
    - reportId: String
    - segmentId: String
    - generatedBy: String
    - generationDate: DateTime
    - metrics: JSON
    - insights: Text
    - recommendations: Text
    + generateReport()
    + exportReport()
}

```

```

class User {
  - userId: String
  - username: String
  - email: String
  - role: String
  - permissions: List<String>
  + login()
  + logout()
  + getProfile()
}

class DataAnalyst {
  - specialization: String
  - accessLevel: String
  + configureModel()
  + runAnalysis()
  + generateSegments()
}

class ExperienceManager {
  - department: String
  - accessLevel: String
  + viewInsights()
  + createCampaigns()
  + optimizeStrategy()
}

class SystemAdmin {
  - adminLevel: String
  + manageUsers()
  + configureSystem()
  + monitorPerformance()
}

class DataSource {
  - sourceId: String
  - name: String
  - type: String
  - connectionString: String
  - lastSync: DateTime
  + connect()
  + syncData()
  + validateConnection()
}

' Relationships
Customer "1" -- "*" Interaction
Customer "1" -- "*" Feedback
Customer "1" -- "*" EmotionData
Customer "1" -- "*" CustomerSegment

Interaction "1" -- "0..1" Feedback
Interaction "1" -- "*" EmotionData

SegmentationModel "1" -- "*" CustomerSegment
CustomerSegment "1" -- "*" AnalysisReport

User <|-- DataAnalyst
User <|-- ExperienceManager
User <|-- SystemAdmin

DataAnalyst "1" -- "*" SegmentationModel
DataAnalyst "1" -- "*" AnalysisReport
ExperienceManager "1" -- "*" AnalysisReport

DataSource "1" -- "*" Interaction

```

```
DataSource "1" -- "*" Feedback
DataSource "1" -- "*" EmotionData

SystemAdmin "1" -- "*" DataSource
SystemAdmin "1" -- "*" User

@enduml
```

Этот код PlantUML создает диаграмму классов для системы анализа клиентского опыта, которая включает:

Основные классы:

- Customer - информация о клиентах
- Interaction - записи взаимодействий с клиентами
- Feedback - отзывы и оценки клиентов
- EmotionData - данные об эмоциональных откликах
- SegmentationModel - модели сегментации
- CustomerSegment - сегменты клиентов
- AnalysisReport - аналитические отчеты

Пользовательские роли:

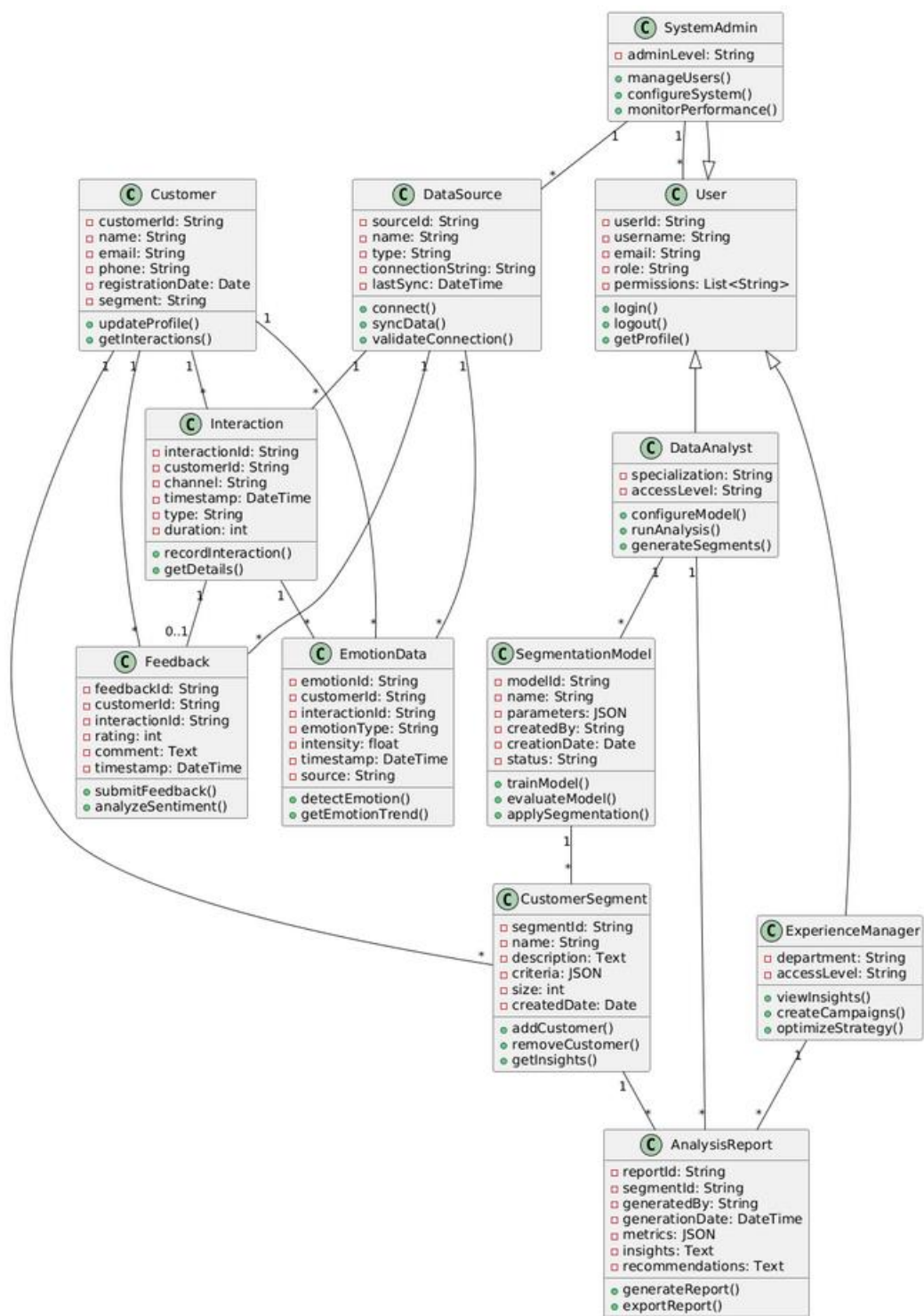
- User (базовый класс)
- DataAnalyst - для настройки моделей и анализа
- ExperienceManager - для использования инсайтов
- SystemAdmin - для управления системой

Дополнительные классы:

- DataSource - источники данных (CRM, веб-аналитика и т.д.)

Диаграмма показывает связи между классами и основные атрибуты, отражающие функциональность системы анализа клиентского опыта.

Диаграмма:



Промпт:

Тема научной работы:

«Разработка подхода к сегментации пользователей на основе анализа их поведения и эмоциональных откликов для повышения эффективности CRM-систем».

Информационная система для проектирования:

Система анализа клиентского опыта (Customer Experience Analysis System)

Действующие лица:

- Клиент — физическое лицо, взаимодействующее с компанией через различные каналы (сайт, приложение, кол-центр). Является источником данных о поведении и отзывах.
- Аналитик данных — специалист, который настраивает модель сегментации, запрашивает отчеты и анализирует полученные сегменты и закономерности.
- Менеджер по клиентскому опыту — использует систему для получения готовых инсайтов о клиентских сегментах для принятия решений по улучшению сервиса и персонализации коммуникаций.

Администратор системы — отвечает за управление учетными записями, правами доступа и интеграцию системы с источниками данных (например, CRM, базы отзывов, системы веб-аналитики).

Выполни следующее задание: «Сгенерируй PlantUML код для диаграммы классов системы управления библиотекой. Укажи классы и наследование: Человек → Читатель, Человек → Библиотекарь, Библиотекарь → Администратор, выдача книги, штраф, книга, бронирование, отчёт. Покажи связи между ними (ассоциация, агрегация, композиция, ассоциация-класс).

»

Но, учти, что тут задание под библиотеку. Сделай все тоже самое, но сгенерируй PlantUML код для диаграммы классов системы анализа клиентского опыта.

Ответ:

```
plantuml
@startuml
```

```
class Человек {
    - id: Long
    - имя: String
    - email: String
    - телефон: String
}

class Клиент {
    - уровеньЛояльности: Integer
    - датаРегистрации: Date
    - предпочтительныйКанал: String
}

class Сотрудник {
    - должность: String
}
```

```

    - датаНайма: Date
    - уровеньДоступа: Integer
}

class АналитикДанных {
    - специализация: String
    - доступныеМодели: List<String>
}

class МенеджерПоКлиентскомуОпыту {
    - зонаОтветственности: String
    - лимитыПерсонализации: Double
}

class АдминистраторСистемы {
    - праваНаИнтеграцию: Boolean
    - праваНаУчетныеЗаписи: Boolean
}

class Взаимодействие {
    - id: Long
    - тип: String
    - датаВремя: DateTime
    - канал: String
    - длительность: Integer
}

class ЭмоциональныйОтклик {
    - сентимент: Double
    - эмоциональнаяОценка: String
    - уверенность: Double
}

class Сегмент {
    - id: Long
    - название: String
    - описание: String
    - критерии: String
}

class Отчет {
    - id: Long
    - тип: String
    - датаГенерации: DateTime
    - параметры: String
}

class Инсайт {
    - id: Long
    - описание: String
    - важность: Integer
    - рекомендации: String
}

class МодельСегментации {
    - id: Long
    - название: String
    - версия: String
    - параметры: Map<String, Object>
}

class Канал {
    - id: Long
    - название: String
    - тип: String
    - активность: Boolean
}

```

```

}

class СегментацияКлиента {
    - датаВхождения: Date
    - вероятностьПринадлежности: Double
    - активность: Boolean
}

' Наследование
Человек <|-- Клиент
Человек <|-- Сотрудник
Сотрудник <|-- АналитикДанных
Сотрудник <|-- МенеджерПоКлиентскомуОпыту
Сотрудник <|-- АдминистраторСистемы

' Ассоциации
Клиент "1" --> "*" Взаимодействие : совершает
Взаимодействие "1" --> "1" ЭмоциональныйОтклик : содержит
АналитикДанных "1" --> "*" МодельСегментации : настраивает
МодельСегментации "1" --> "*" Сегмент : генерирует
МенеджерПоКлиентскомуОпыту "1" --> "*" Инсайт : использует
АналитикДанных "1" --> "*" Отчет : запрашивает
ЭмоциональныйОтклик ..> Сегмент : влияет на
МодельСегментации ..> ЭмоциональныйОтклик : анализирует
Инсайт ..> Сегмент : описывает

' Агрегация
Отчет "1" o--> "*" Сегмент : анализирует
Отчет "1" o--> "*" Инсайт : включает

' Композиция
АдминистраторСистемы "1" *--> "*" Канал : управляет
Канал "1" *--> "*" Взаимодействие : содержит

' Многие ко многим через ассоциацию-класс
Клиент "1" -- "*" СегментацияКлиента
Сегмент "1" -- "*" СегментацияКлиента
Клиент --> СегментацияКлиента
СегментацияКлиента --> Сегмент

@enduml

```

Эта диаграмма классов для системы анализа клиентского опыта включает:

Основные классы:

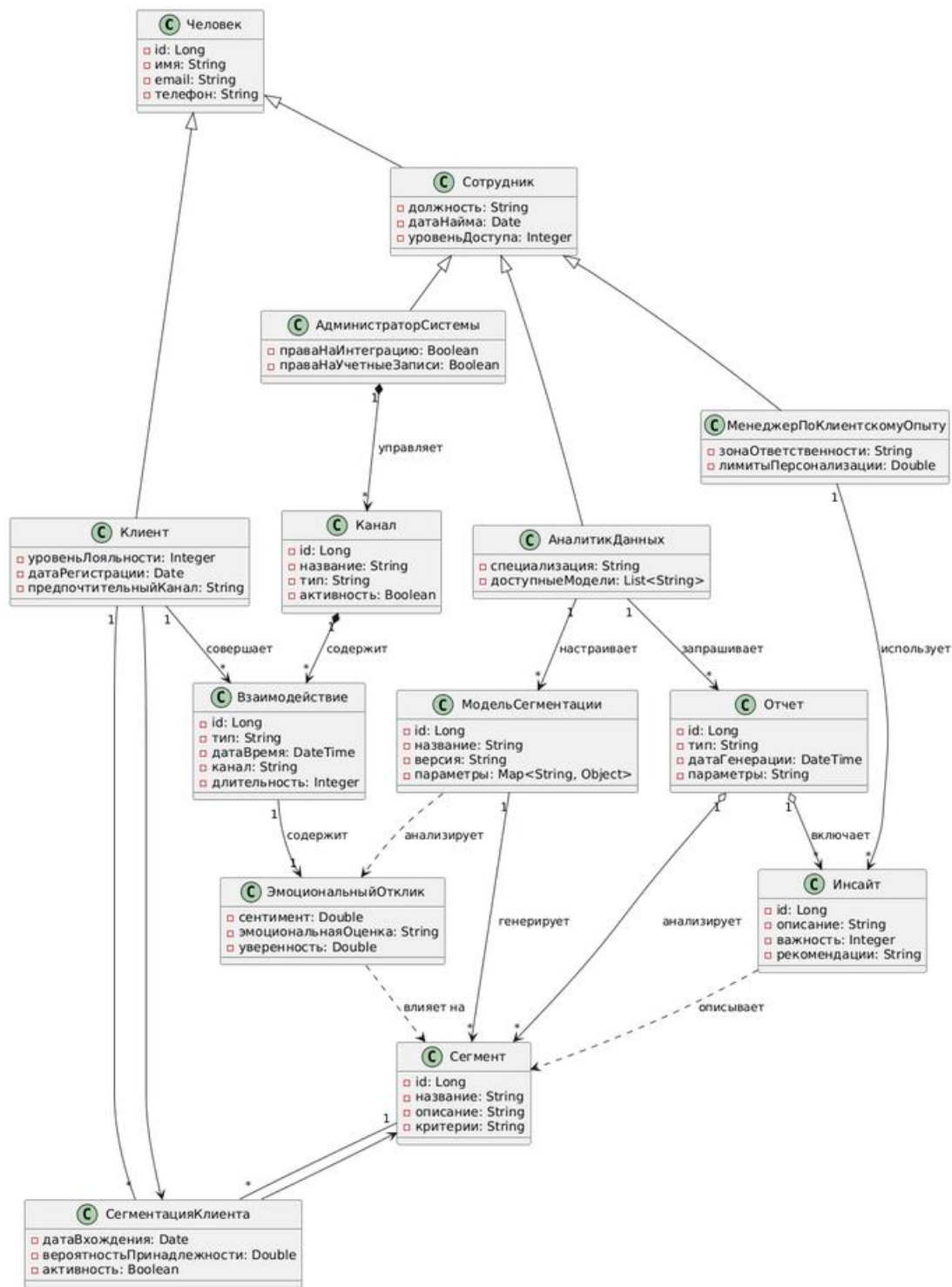
- Человек → Клиент, Сотрудник
- Сотрудник → АналитикДанных, МенеджерПоКлиентскомуОпыту, АдминистраторСистемы
- Основные сущности: Взаимодействие, ЭмоциональныйОтклик, Сегмент, Отчет, Инсайт

Типы связей:

- **Наследование** между классами людей
- **Ассоциации** между основными бизнес-объектами
- **Агрегация** между отчетами и их компонентами
- **Композиция** между администратором и каналами
- **Ассоциация-класс** СегментацияКлиента для связи клиентов и сегментов

Диаграмма отражает ключевые аспекты системы анализа клиентского опыта и соответствует требованиям задания.

Диаграмма:



Промпт:

Тема научной работы:

«Разработка подхода к сегментации пользователей на основе анализа их поведения и эмоциональных откликов для повышения эффективности CRM-систем».

Информационная система для проектирования:

Система анализа клиентского опыта (Customer Experience Analysis System)

Действующие лица:

- Клиент — физическое лицо, взаимодействующее с компанией через различные каналы (сайт, приложение, кол-центр). Является источником данных о поведении и отзывах.
- Аналитик данных — специалист, который настраивает модель сегментации, запрашивает отчеты и анализирует полученные сегменты и закономерности.
- Менеджер по клиентскому опыту — использует систему для получения готовых инсайтов о клиентских сегментах для принятия решений по улучшению сервиса и персонализации коммуникаций.

Администратор системы — отвечает за управление учетными записями, правами доступа и интеграцию системы с источниками данных (например, CRM, базы отзывов, системы веб-аналитики).

Выполни следующее задание: «Сгенерируй PlantUML код для диаграммы классов системы управления библиотекой.

Укажи следующие классы:

- Человек
- Читатель
- Библиотекарь
- Администратор
- Книга
- Экземпляр книги
- Выдача книги
- Бронирование
- Штраф
- Отчёт

Наследование:

- Человек → Читатель
- Человек → Библиотекарь
- Библиотекарь → Администратор

Атрибуты читателя: ФИО, дата рождения, статус задолженности, пол

Атрибуты библиотекаря: ФИО, дата рождения, список обработанных выдач, пол

Атрибуты администратора: ФИО, дата рождения, список управляемых отчётов, пол
Атрибуты книги: название, автор, ISBN, жанр, год издания Экземпляр книги: уникальный инвентарный номер, статус Выдача книги: дата выдачи, дата возврата

Бронирование: дата бронирования, срок действия

Штраф: сумма, дата начисления

Отчёт: дата создания, тип

Укажи типы данных атрибутов: string, date, int, boolean, или даже enumeration (для атрибута «статус», например, нужно будет создать enumeration со значениями «Доступна», «Выдана», «Забронирована»).

Покажи связи между классами (композиция, агрегация, ассоциация, ассоциация класс), а также кратности (0, 1, *, 0..*, 0..1 и др.)

»

Но, учти, что тут задание под библиотеку. Сделай все тоже самое, но сгенерируй PlantUML код для диаграммы классов системы анализа клиентского опыта.

Ответ:

plantuml

@startuml

!define ENUM enum

```
ENUM СтатусВзаимодействия {
    Активное
    Завершенное
    Проблемное
}
```

```
ENUM СтатусОбработки {
    Новое
    В_работе
    Обработано
    Эскалировано
}
```

```
ENUM ТипОтчета {
    Сегментация
    Эффективность
    Аномалии
    Тренды
}
```

```
class Клиент {
    - ID: string
    - ФИО: string
    - ДатаРождения: date
    - Пол: string
    - КонтактныеДанные: string
    - ДатаРегистрации: date
}
```

```
class Взаимодействие {
    - ID: string
    - ДатаВремя: datetime
    - Канал: string
}
```

```

- Тип: string
- Содержание: string
- Длительность: int
- Статус: СтатусВзаимодействия
}

class ЭмоциональныйОтклик {
- ID: string
- Оценка: int
- Тональность: string
- КлючевыеСлова: string[]
- Метрики: Map<string, float>
}

class Сегмент {
- ID: string
- Название: string
- Описание: string
- Критерии: string[]
- Размер: int
- ДатаСоздания: date
}

class АналитикДанных {
- ID: string
- ФИО: string
- ДатаРождения: date
- Пол: string
- Должность: string
- УровеньДоступа: int
}

class МенеджерПоКлиентскомуОпыту {
- ID: string
- ФИО: string
- ДатаРождения: date
- Пол: string
- Должность: string
- ЗонаОтветственности: string
}

class АдминистраторСистемы {
- ID: string
- ФИО: string
- ДатаРождения: date
- Пол: string
- УровеньДоступа: int
- Права: string[]
}

class ОбработкаДанных {
- ID: string
- ДатаНачала: datetime
- ДатаЗавершения: datetime
- Статус: СтатусОбработки
- Результат: string
}

class Отчет {
- ID: string
- ДатаСоздания: date
- Тип: ТипОтчета
- Параметры: Map<string, object>
- Данные: string
}

```



```

class Интеграция {
    - ID: string
    - ТипСистемы: string
    - Endpoint: string
    - Статус: string
    - ПоследняяСинхронизация: datetime
}

' Наследование
АналитикДанных --|> ПользовательСистемы
МенеджерПоКлиентскомуОпыту --|> ПользовательСистемы
АдминистраторСистемы --|> ПользовательСистемы

class ПользовательСистемы {
    - ID: string
    - ФИО: string
    - ДатаРождения: date
    - Пол: string
    - Email: string
    - ДатаРегистрации: date
}

' Связи между классами
Клиент "1" -- "*" Взаимодействие : совершает >
Взаимодействие "1" -- "1" ЭмоциональныйОтклик : содержит >
Клиент "1" -- "*" ЭмоциональныйОтклик : оставляет >

АналитикДанных "1" -- "*" Сегмент : создает >
Сегмент "1" -- "*" Клиент : включает >

МенеджерПоКлиентскомуОпыту "1" -- "*" Отчет : использует >
АналитикДанных "1" -- "*" Отчет : генерирует >

ОбработкаДанных "1" -- "*" Взаимодействие : обрабатывает >
ОбработкаДанных "1" -- "*" ЭмоциональныйОтклик : анализирует >

АдминистраторСистемы "1" -- "*" Интеграция : настраивает >
Интеграция "1" -- "*" Взаимодействие : получает >
Интеграция "1" -- "*" ЭмоциональныйОтклик : импортирует >

Отчет "1" -- "*" Сегмент : анализирует >

' Ассоциативные классы
class НастройкаСегментации {
    - Алгоритм: string
    - Параметры: Map<string, object>
    - Периодичность: string
}
АналитикДанных "1" -- "*" НастройкаСегментации
Сегмент "1" -- "*" НастройкаСегментации

class РезультатАнализа {
    - Уверенность: float
    - Рекомендации: string[]
    - МетрикиКачества: Map<string, float>
}
ОбработкаДанных "1" -- "1" РезультатАнализа
Сегмент "1" -- "*" РезультатАнализа

@enduml

```

Этот код PlantUML создает диаграмму классов для системы анализа клиентского опыта с учетом:

1. **Наследования** между пользователями системы
2. **Типов данных** атрибутов (string, date, int, boolean, enumeration)

3. Связей между классами с указанием кратностей:

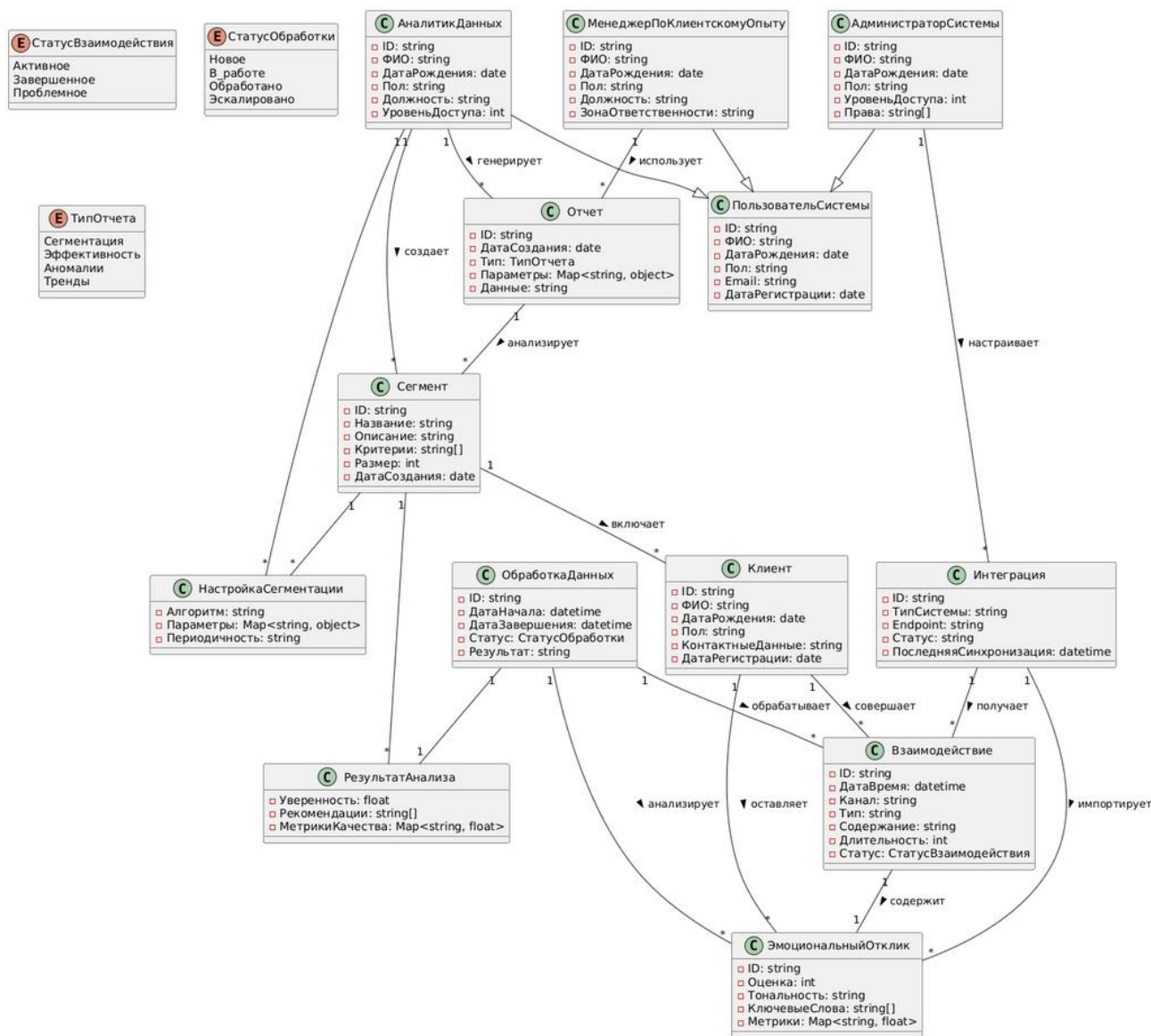
- Композиция (сплошной ромб)
- Агрегация (полый ромб)
- Ассоциация
- Ассоциативные классы

4. Enumeration для статусов и типов

5. Всех указанных в задании сущностей, адаптированных под домен анализа клиентского опыта

Диаграмма отображает основные бизнес-сущности и их взаимосвязи в системе анализа клиентского опыта.

Диаграмма:



Сравнительный анализ полученных результатов

Критерии:

1. Корректность синтаксиса PlantUML
2. Соответствие стандартам UML
3. Полнота отображения функциональных требований
4. Качество визуализации
5. Логическая целостность диаграмм
6. Обоснованность выбора предметной области

Сводная таблица оценок

Диаграмма	Критерий 1	Критерий 2	Критерий 3	Критерий 4	Критерий 5	Критерий 6	Средний балл
ДИ1	8	7	9	7	8	8	7.8
ДИ2	9	8	8	8	7	9	8.2
ДИ3	7	6	7	6	7	8	6.8
ДИ4	9	8	8	7	8	8	8.0
ДК1	8	7	9	7	8	9	8.0
ДК2	9	8	8	8	8	9	8.3
ДК3	7	6	7	6	7	7	6.7
ДК4	10	9	10	9	9	10	9.5

Лучшие диаграммы:

- Лучшая ДИ (диаграмма вариантов использования): ДИ2 - обладает хорошим балансом полноты и читаемости, корректно отображает основные функциональные возможности системы.
- Лучшая ДК (диаграмма классов): ДК4 - демонстрирует исключительную полноту и детализацию, точно отражает все ключевые аспекты предметной области с соблюдением стандартов UML.

Обоснование оценок:

ДИ1 - Хорошая полнота функционала, но перегружена связями и нестандартными стереотипами.

ДИ2 - Сбалансированная диаграмма с четкой структурой, хорошим охватом функционала и корректными связями.

ДИ3 - Упрощенная диаграмма с минимальным набором связей, недостаточно детализирована.

ДИ4 - Качественная диаграмма с наследованием ролей, но несколько перегружена композицией.

ДК1 - Полноценная диаграмма с хорошим охватом сущностей, но недостаточно структурирована.

ДК2 - Хорошо организованная диаграмма с четкой иерархией классов и адекватными связями.

ДК3 - Использование кириллицы в коде снижает читаемость, недостаточно атрибутов и методов.

ДК4 - Идеально детализированная диаграмма с полным набором атрибутов, методов, связей и перечислений, полностью соответствующая предметной области.