МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Нижегородский государственный университет им. Н. И. Лобачевского»

(ННГУ)

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных технологий

УЧЕБНЫЙ КУРС

«Проектирование и архитектура программных систем»

для подготовки по направлению 09.03.04 «Программная инженерия»

ФУНКЦИОНАЛЬНАЯ СПЕЦИФИКАЦИЯ

«QA-система на основе ИИ»

Выполнили студенты группы 3822Б1ПР1
Ворошилов Виталий Александрович, Крылов Михаил Георгиевич,
Моисеев Артём Владимирович, Морозов Егор Алексеевич,
Рамс Сергей Никитич

Проверил к.т.н., доцент
Лебедев Илья Генадьевич

Нижний Новгород
2025

Содержание

1. История проекта 3
2. Цели дизайна 4
2.1. Требования пользователя 4
2.2. Системные требования 4
2.3. Сценарии использования 4
3. Исключенные возможности и неподдерживаемые сценарии 8
4. Предположения и зависимости 10
5. Проект решения 12
5.1. Концептуальный проект 12
5.2. Логический проект 12
5.3. Физический проект 13
6. Требования к инсталляции и деинсталляции 15

1. История проекта

- Октябрь 2025
  - Утверждена концепция и рамки проекта
  - Подготовлены данные для обучения модели
  - Определена архитектура системы
  - Выбран датасет для разработки модели
- Ноябрь 2025
  - Разработан контроллер вычислительного узла
  - Разработан первый прототип серверной части
- Декабрь 2025
  - Разработан веб-интерфейс
  - Реализован Telegram-бот
  - Финализирована разработка серверной части
  - Организован процесс развертывания

2. Цели дизайна

2.1. Требования пользователя

- Доступ к платформе через веб-интерфейс или Telegram-бот
- Организация взаимодействия с QA-системой в формате чата
- Возможность задавать неограниченное количество вопросов к присланному тексту
- Хранение истории для быстрого поиска ответов на уже заданные вопросы
- Требования относительно точности модели, выражающиеся в конкретных метриках: Exact Match (EM) ≈ 0.4, F1 Score ≈ 0.8

2.2. Системные требования

- Серверная часть
  - не менее 4 ГБ дискового пространства
  - не менее 4 ГБ оперативной памяти
  - стабильное интернет-соединение
- Вычислительный узел
  - GPU с не менее 8 ГБ видеопамяти
  - не менее 30 ГБ дискового пространства
  - стабильное интернет-соединение

2.3. Сценарии использования

Рисунок 2.3.1 – Сценарии использования (диаграмма Use Case)

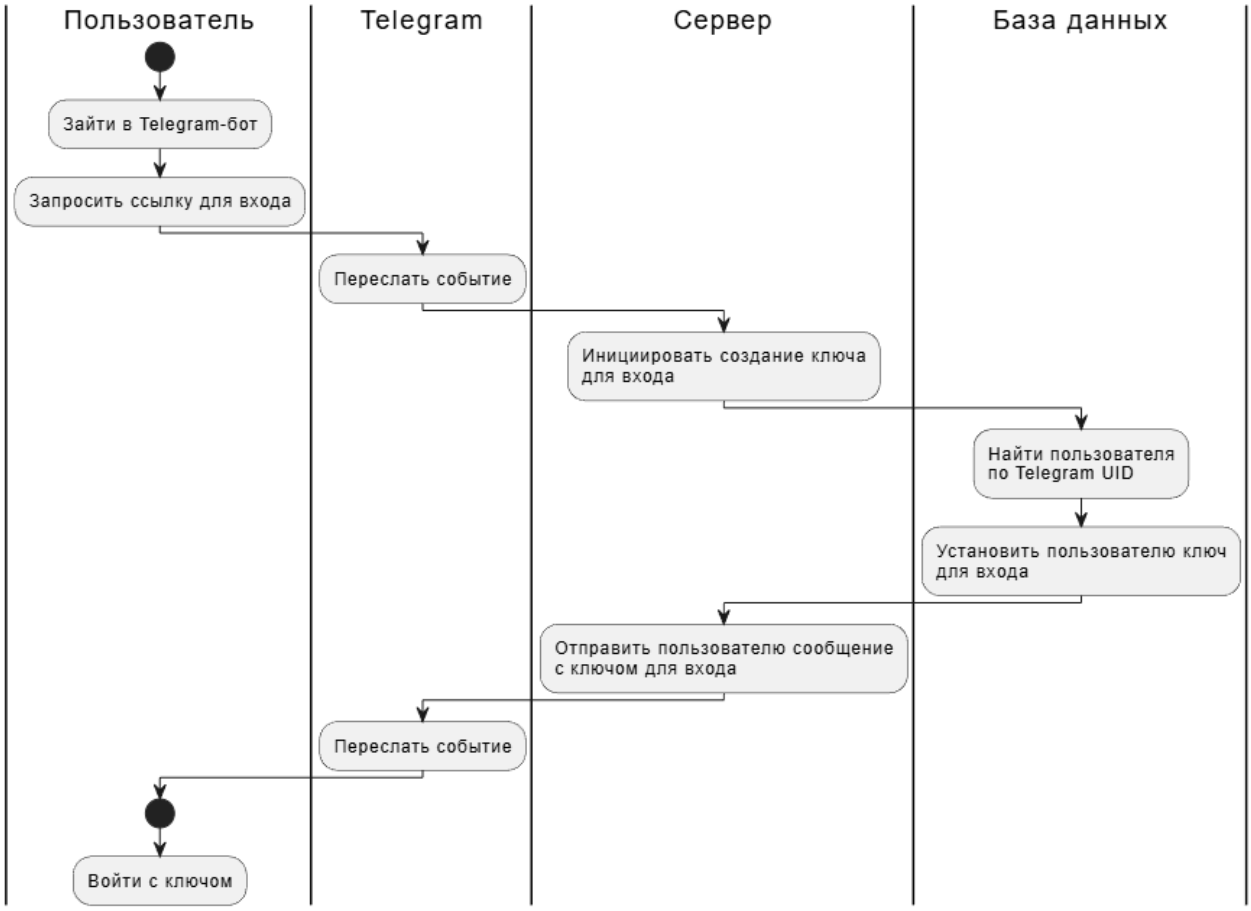

Рисунок 2.3.2 – Flow авторизации через Telegram-бот

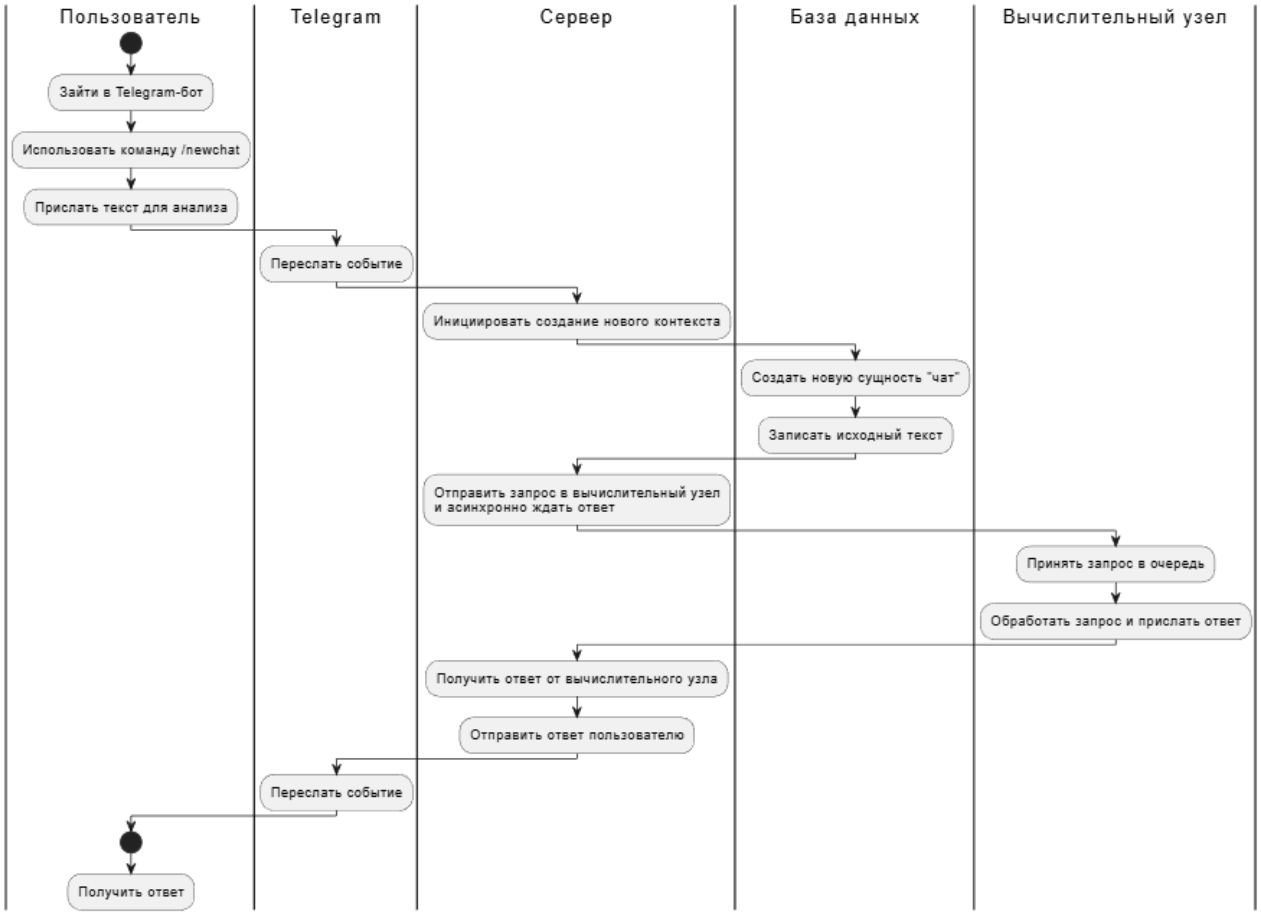

Рисунок 2.3.3 – Flow создания контекста и запроса через Telegram-бот

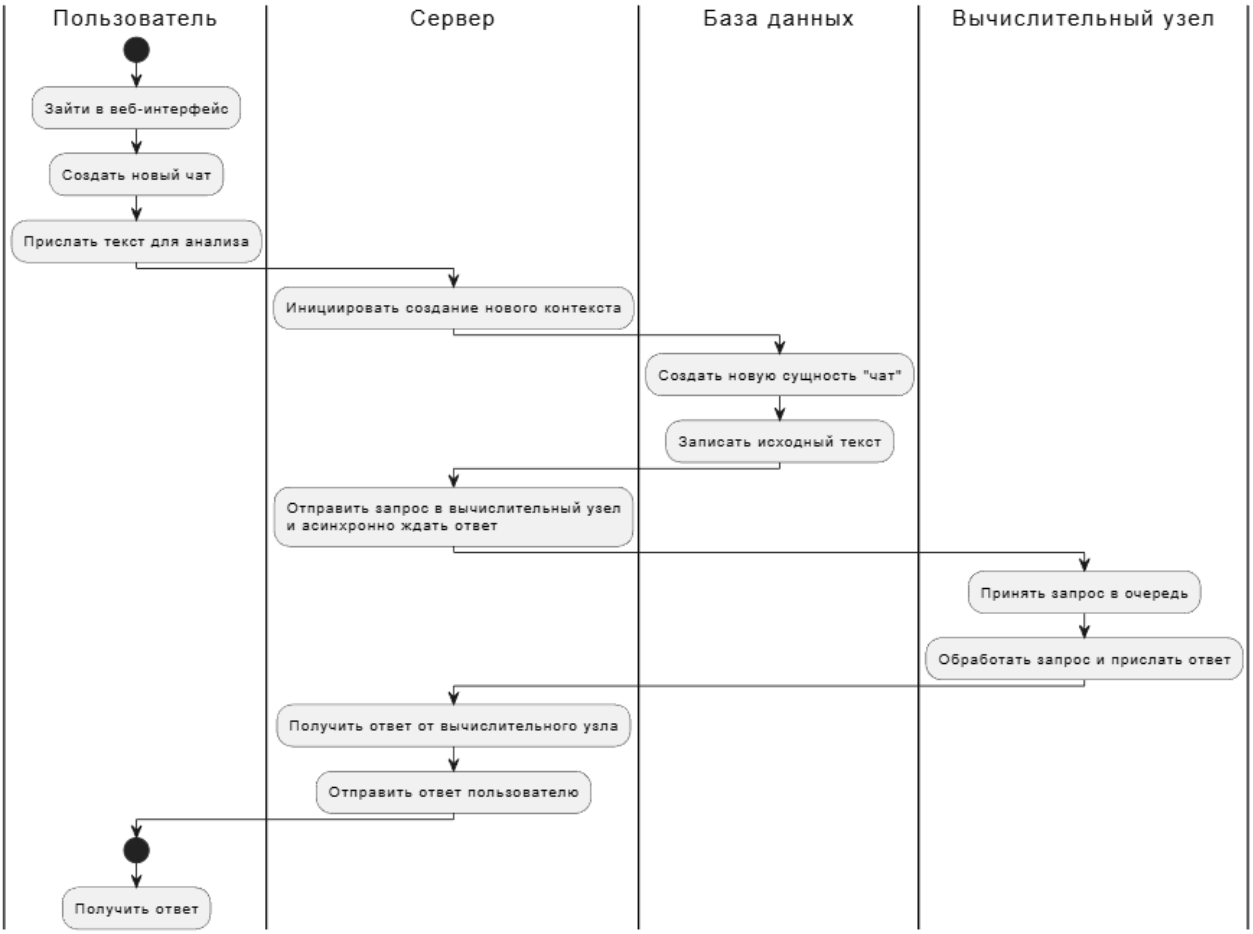

Рисунок 2.3.4 – Flow создания контекста и запроса через веб-интерфейс

3. Исключенные возможности и неподдерживаемые сценарии

В текущем релизе проекта не найдут отражение возможности:

1. Обработка изображений / OCR (распознавание текста с картинок)
   - Описание: приём и обработка изображений (фотографий/сканов) с последующим распознаванием текста и ответом на вопросы по распознанному содержимому.
   - Причина исключения: внедрение OCR требует дополнительной модели/сервиса, увеличивает сложность пайплайна и существенно увеличивает объём работ и требования к вычислительной инфраструктуре.
   - Что потребуется: модуль OCR (Tesseract, коммерческие API или нейросетевые решения), пайплайн предобработки изображений, тестовые наборы с метками, интеграция с текущим API; оценка качества OCR и добавление этапа валидации.
   - Приоритет: средний/низкий (может быть включено в релиз +2 при наличии ресурсов).
2. Голосовой ввод / ASR (speech-to-text)
   - Описание: приём голосовых сообщений и преобразование их в текст для последующей обработки моделью QA.
   - Причина исключения: требуется интеграция ASR, дополнительные сервисы/лицензии, тестирование качества распознавания для рус. и англ.
   - Что потребуется: ASR-провайдер (open-source/коммерческий), обработка шумов, UX для голосового взаимодействия, дополнительные тесты.
   - Приоритет: низкий/по запросу.
3. Мультиязычность за пределами русского и английского
   - Описание: поддержка языков, кроме RU/EN (например, украинский, немецкий и т.д.).
   - Причина исключения: модель и датасеты настроены на RU/EN; расширение требует новых наборов данных и дообучения/замены модели.
   - Что потребуется: датасеты для целевых языков, дообучение модели, тестирование качества, локализация интерфейсов.
   - Приоритет: средний (при наличии запроса со стороны заказчика).
4. Загрузка и обработка больших файлов / пакетный режим
   - Описание: загрузка больших многомегабайтных документов, пакетная обработка множества запросов к большому количеству документов за один раз.
   - Причина исключения: ограничение по объёму контекста и простота первого релиза; большие файлы требуют разбивки на чанки, индексирования и других оптимизаций.

- Что потребуется: механизм chunking / индексирования (document store, vector DB), фоновые задачи для пакетной обработки, менеджер очередей (Celery / RQ / Kafka).
- Приоритет: средний/высокий для корпоративных сценариев, но не для MVP.

5. Расширенная аналитика качества ответов / пользовательская оценка (лайк/дизлайк с учётом PII)
   - Описание: хранение оценок пользователей для обучения/улучшения модели, сбор и обработка фидбека.
   - Причина исключения: требует дополнительных интерфейсов, хранения пользовательских данных и механизма обработки фидбека.
   - Что потребуется: UI для оценки, метрики хранения, pipeline для анализа фидбека, соглашение о хранении данных.
   - Приоритет: средний (полезно для доработки модели после релиза).
6. Развёртывание модели полностью на стороне клиента (offline/local)
   - Описание: возможность запускать модель локально на машине пользователя без сервера.
   - Причина исключения: даже облегчённые модели требуют слишком много ресурсов; контекст и безопасность упрощаются при серверном подходе.
   - Что потребуется: разработка лёгких on-device моделей, упаковка и инструкции по установке, поддержка разнообразных платформ.
   - Приоритет: низкий/по запросу.
7. Расширенные объяснения (explainability) ответов и provenance (подробные ссылки на источник с точной разбивкой)
   - Описание: автоматическая генерация подробных объяснений, почему модель ответила именно так, и указание точных источников/параграфов.
   - Причина исключения: это требует дополнительной логики извлечения/ранжирования и доказуемой обработки источников; выходит за рамки базового функционала.
   - Что потребуется: retrieval-augmented подход с явным индексом, механизмы оценивания достоверности, UI для отображения provenance.
   - Приоритет: средний.

4. Предположения и зависимости

- Предположения:

1. Доступность вычислительных ресурсов (GPU)
    - Предположение: для этапа fine-tuning и тестирования будет доступен GPU-сервер с минимум 8 GB VRAM.
    - Последствия при нарушении: потребуются более лёгкие подходы (transfer learning без fine-tuning) или аренда облачных ресурсов — увеличатся сроки и стоимость.
    - Рекомендация: подтвердить наличие GPU на старте; если нет — бюджетировать облачные ноды (AWS/GCP/Yandex/etc).
2. Языковой диапазон — русский и английский
    - Предположение: пользователи формулируют вопросы на RU/EN; других языков нет в MVP.
    - Последствия: если появится необходимость других языков — нужно переобучение/поменять модель.
    - Рекомендация: зафиксировать в ТЗ поддержку только RU/EN.
3. Ожидаемая нагрузка (QPS и поведение пользователей)
    - Предположение: средняя активность — 5–10 запросов в день на одного пользователя; пиковые нагрузки не превышают проектных значений.
    - Последствия: при резком росте пользователей потребуется масштабирование и доработка архитектуры.
    - Рекомендация: провести нагрузочные тесты перед продакшеном.
4. Контент вводится вручную (вставка текста), без сложных документов
    - Предположение: пользователь вставляет текст в форму (plain text). Загрузка сложных форматов / больших файлов не предполагается.
    - Последствия: если пользователи начнут отправлять большие файлы, потребуется механизм разбивки и хранение.
    - Рекомендация: добавить в интерфейс валидацию размера контекста и подсказку о лимите.
5. Минимальные требования к времени отклика:
    - Предположение: целевой отклик не превышает 10 секунд для одного запроса.
    - Последствия: при несоблюдении — ухудшение UX, потребуется оптимизация модели/инфраструктуры.
    - Рекомендация: мониторинг p50/p95 и бюджетирование на оптимизацию (кеширование, batching).
6. Логирование без хранения PII
    - Предположение: логи не будут содержать персональные данные, или будут анонимизированы.
    - Последствия: хранение PII требует юридических согласований и мер защиты.
    - Рекомендация: настроить механизмы анонимизации и документировать политику хранения.

- Зависимости:
  1. Сторонние библиотеки и фреймворки
     - Зависит от: PyTorch, Transformers (Hugging Face), FastAPI, Aiogram, Vue.js и т. д.
     - Риски: несовместимость версий, критические уязвимости.
     - Меры по снижению рисков: фиксировать версии в requirements, тестировать обновления в staging.
  2. Telegram API / сторонние мессенджеры
     - Зависит от: работоспособности Telegram API (webhook/long-polling).
     - Риски: изменения API, rate limits.
     - Меры по снижению рисков: использовать retry/backoff, документировать лимиты, предусмотреть альтернативу (web).
  3. Инфраструктура/облако/хостинг
     - Зависит от: доступности выбранного хостинга/облачных ресурсов (вирт. машины, GPU).
     - Риски: сбои, превышение бюджета.
     - Меры по снижению рисков: резервные планы, мониторинг затрат, ограничение автоскейлинга без явного разрешения.
  4. Наличие обучающих/валидационных датасетов
     - Зависит от: качества и полноты датасетов (например, SberQUAD или аналоги) для валидации метрик EM/F1.
     - Риски: недостаток примеров по предметной области приведёт к плохой точности.
     - Меры по снижению рисков: заранее подготовить и аннотировать тестовую выборку; предусмотреть ручную разметку.
  5. Правовые и регуляторные требования
     - Зависит от: требований к хранению данных, авторским правам на контент, локальных законов.
     - Риски: необходимость внесения изменений в политику хранения и обработку данных.
     - Меры по снижению рисков: проконсультироваться с юристом и задокументировать политику конфиденциальности.
  6. Командные ресурсы и кадры
     - Зависит от: наличия назначенных членов команды (1 руководитель, 1 веб-разработчик, 2 ML, 1 backend).
     - Риски: отвлечение сотрудников, болезни, увольнения.
     - Меры по снижению рисков: минимальные резервы, документирование знаний (README, runbooks), перекрытие ролей.

5. Проект решения

5.1. Концептуальный проект

Пользователь вводит текст и задает вопросы к нему:

- через веб-интерфейс - с помощью текстовых полей
- через Telegram-бот - через общение с ботом

Система анализирует текст при помощи нейросети и выдает точный ответ, основанный на предоставленном тексте.

Пример:

- Текст: Столица России – Москва.
- Вопрос: Столица РФ?
- Ответ: Москва.

Роли:

- Пользователь: отправляет текст и вопрос
- Сервер: сохраняет запрос для истории и пересылает его в вычислительный узел
- Вычислительный узел: нейросеть T5 находит ответ

5.2. Логический проект

В серверной части применяются элементы DDD-подхода: логика внутри компонентов разделена на функциональные элементы, взаимодействующие друг с другом через явно определенные интерфейсы.

Для разработки клиентской части использовался фреймворк Vue.js, применяющий MVVM подход.

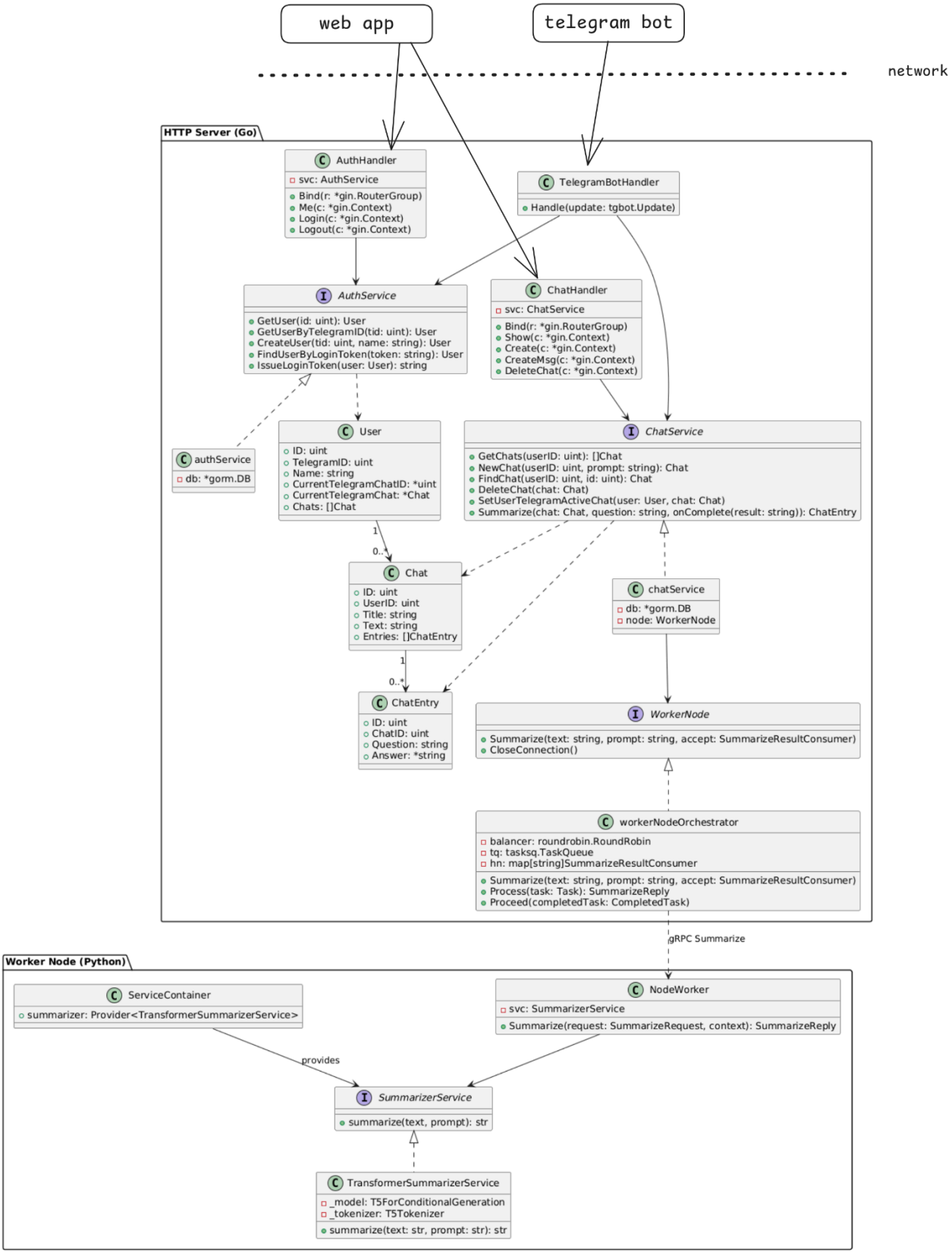

Рисунок 5.2.1 – Диаграмма классов

5.3. Физический проект

Платформа реализуется как приложение с трехзвенной клиент-серверной архитектурой:

- Звено 1: представление данных (веб-интерфейс, Telegram-бот)
- Звено 2: промежуточное звено (серверная часть)
- Звено 3: управление ресурсами (база данных, вычислительный узел)

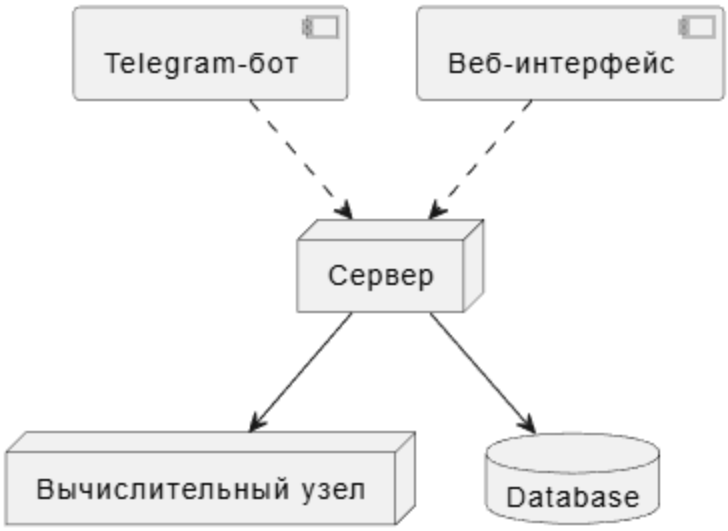

Рисунок 5.3.1 – Диаграмма компонентов

6. Требования к инсталляции и деинсталляции

- Серверная часть: серверная часть упакована в Docker-контейнер, для инсталляции требуется лишь Docker
- Вычислительный узел: контроллер вычислительного узла упакован в Docker-контейнер, для инсталляции требуется лишь Docker
- Клиентская часть:
  - Веб-клиент приложения предназначен для работы в веб-браузере, инсталляция не требуется, что повышает гибкость, удобство использования и доступность
  - Для доступа к Telegram-боту необходим клиент мессенджера Telegram