

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

## РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практическим работам №5-8**

по дисциплине «Системная и программная инженерия»

**Выполнили:**

|  |  |
| --- | --- |
| Студенты группы ИКБО-15-22 | Кудинов А.В. Оганнисян Г.А.  Шаралапов Д.А. |
| Принял преподаватель | Запорожских А.И. |

Москва 2025 г.

**СОДЕРЖАНИЕ**

[Практическая работа №5 3](#_Toc197760783)

[Практическая работа №6 5](#_Toc197760784)

[Практическая работа №7 6](#_Toc197760785)

[Практическая работа №8 8](#_Toc197760786)

[Вывод 13](#_Toc197760787)

# Практическая работа №5

На рисунках 1–2 представлены диаграммы классов и объектов системы.

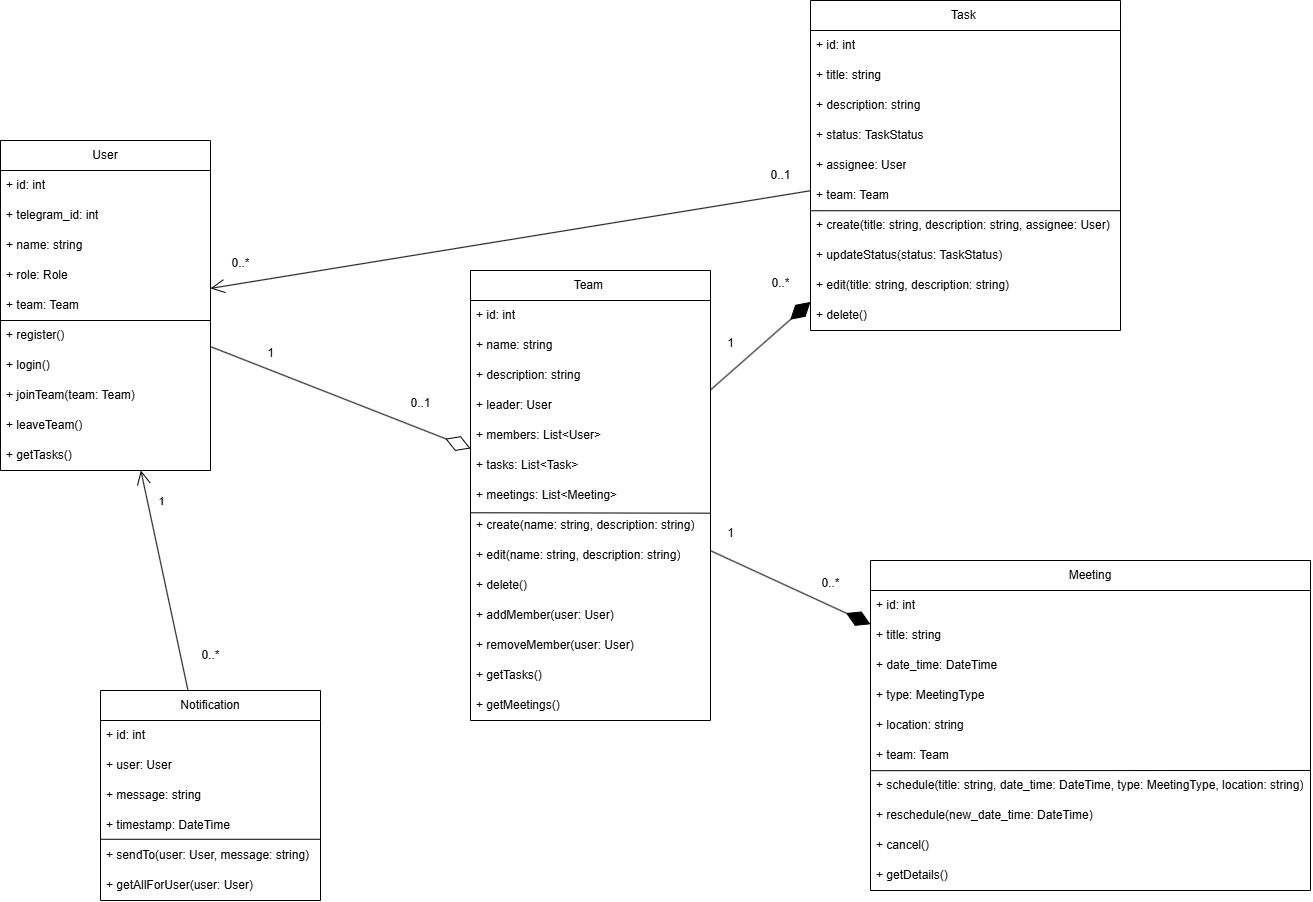


Рисунок 1 – Диаграмма классов

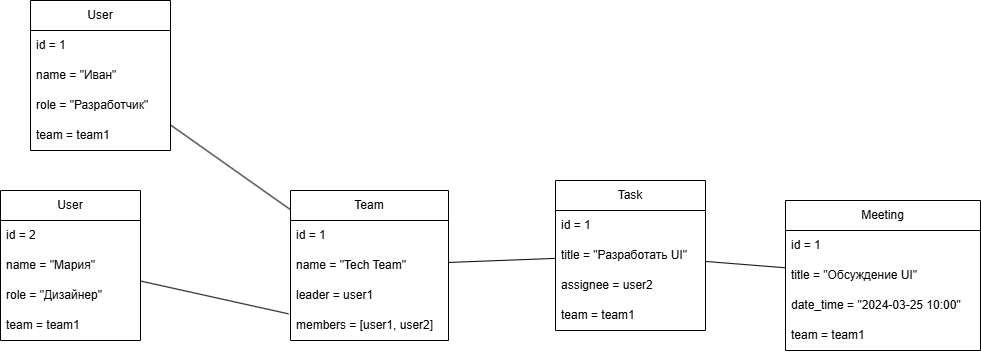


Рисунок 2 – Диаграмма объектов

На рисунках 3 – 5 представлены диаграммы в нотации IDEF0.

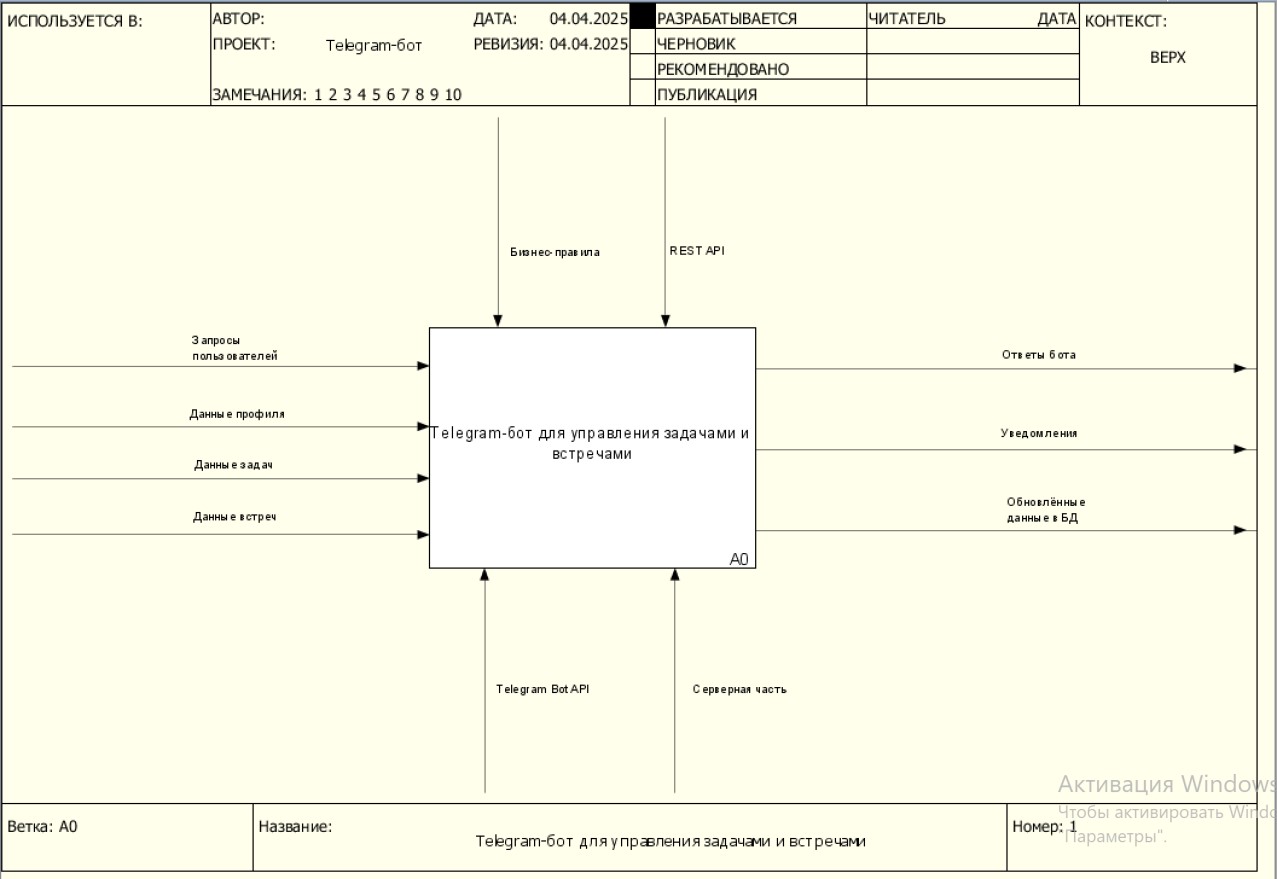


Рисунок 3 – Процесс «Telegram-бот для управления задачами и встречами»

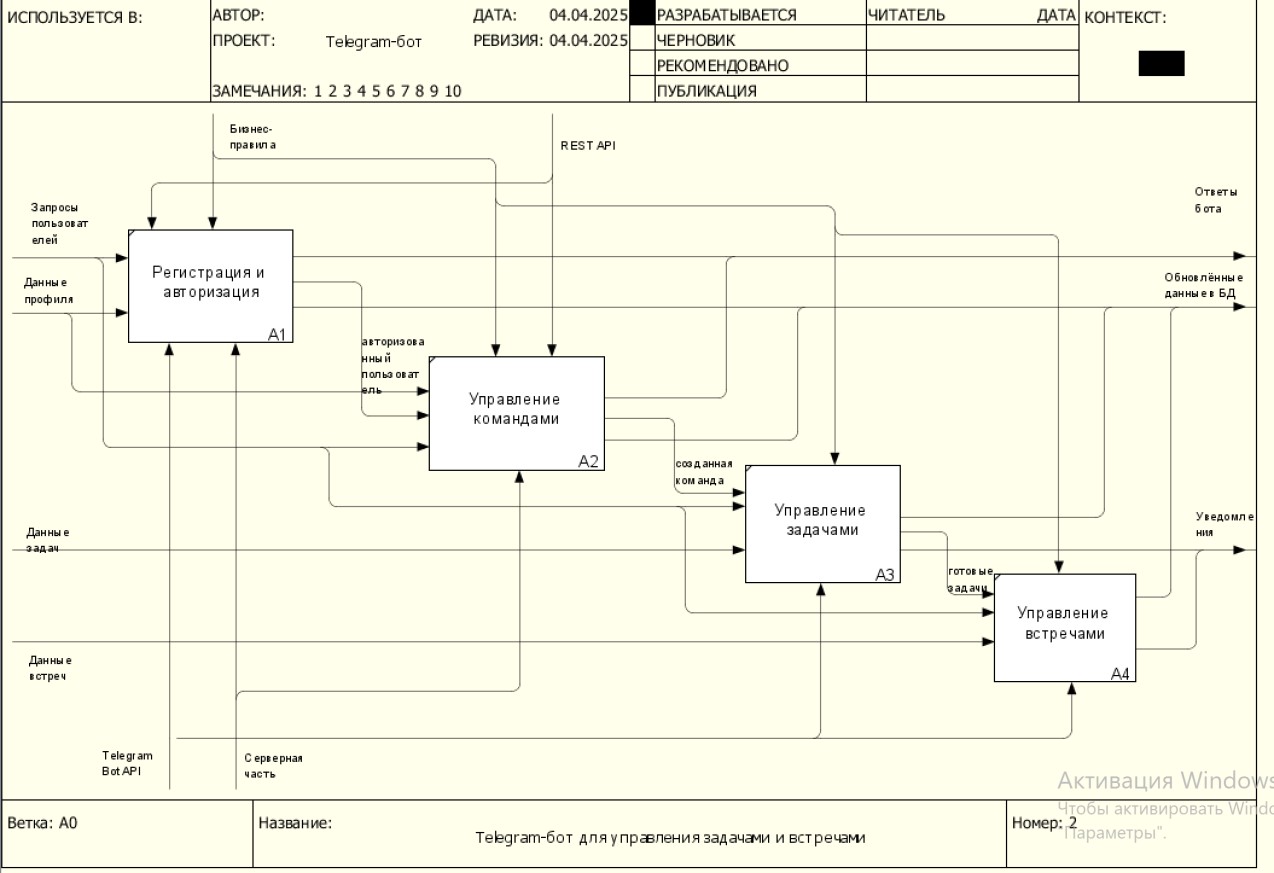


Рисунок 4 – Декомпозиция процесса «Telegram-бот для управления задачами и встречами»

# Практическая работа №6

На рисунках 6–8 представлены диаграммы в нотации DFD и модель базы данных.

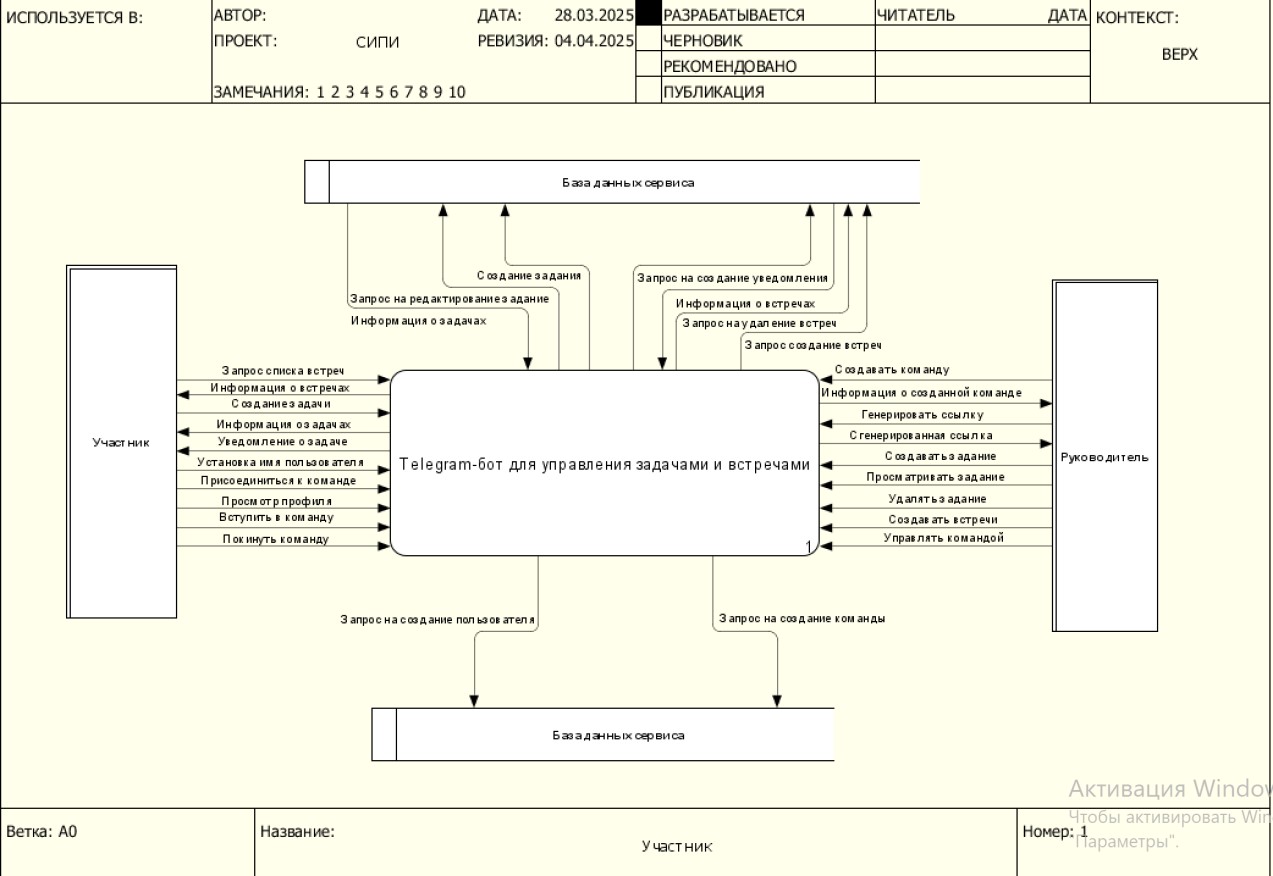


Рисунок 5 – Диаграмма в нотации DFD

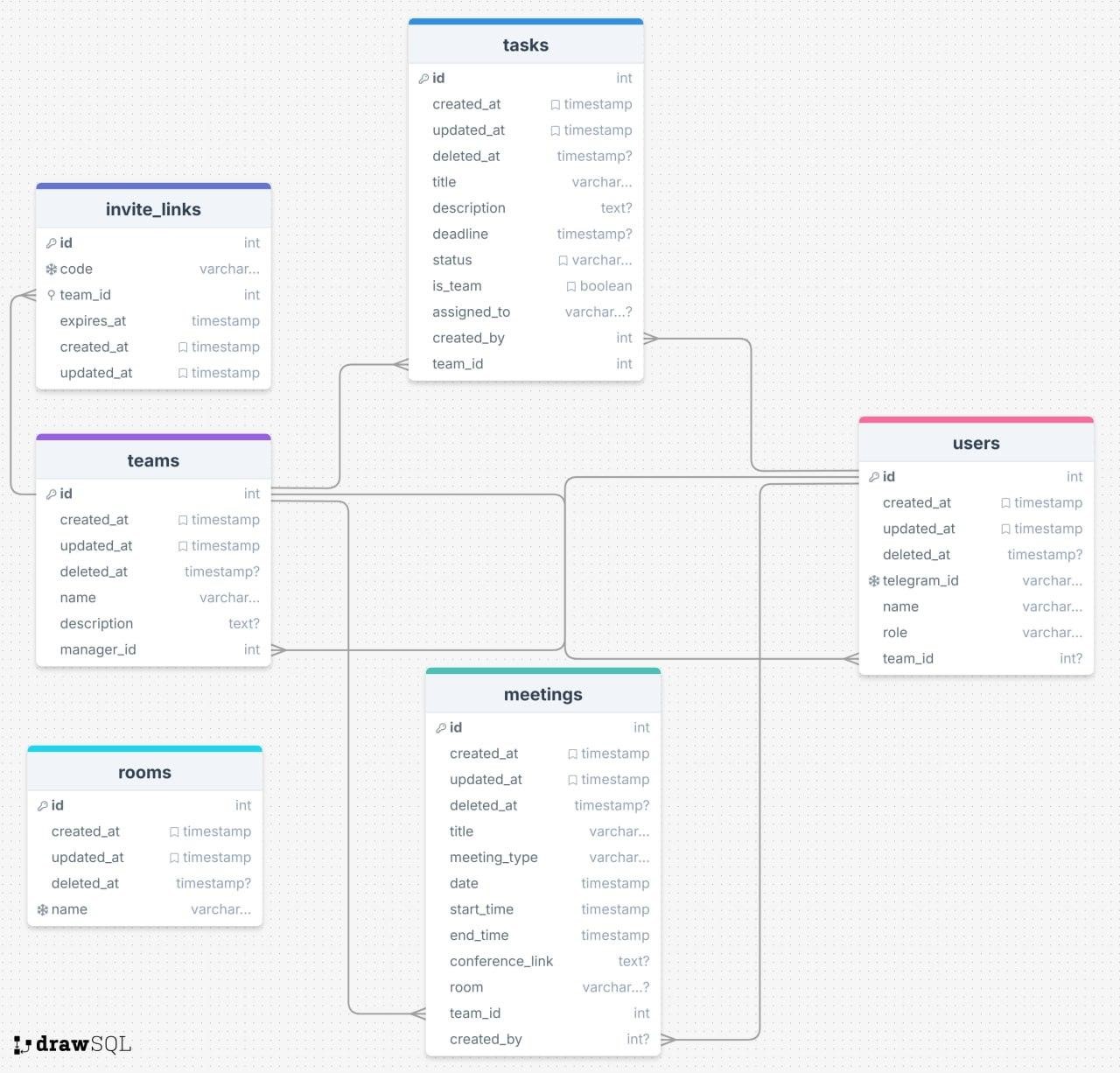


Рисунок 6 –Модель базы данных

# Практическая работа №7

Трехуровневая архитектура — это широко применяемая архитектура программного обеспечения, в которой приложения разделены на три логических и физических уровня: уровень представления (пользовательский интерфейс), уровень приложения, на котором осуществляется обработка данных, и уровень данных, предназначенный для хранения и управления данными, относящимися к приложению.

Основное преимущество трехуровневой архитектуры заключается в том, что поскольку каждый уровень имеет собственную инфраструктуру, разработкой каждого уровня может заниматься отдельная команда разработчиков. Кроме того, каждый уровень можно обновлять и масштабировать по мере необходимости, не затрагивая другие уровни. На рисунке 7 представлена архитектура приложения.

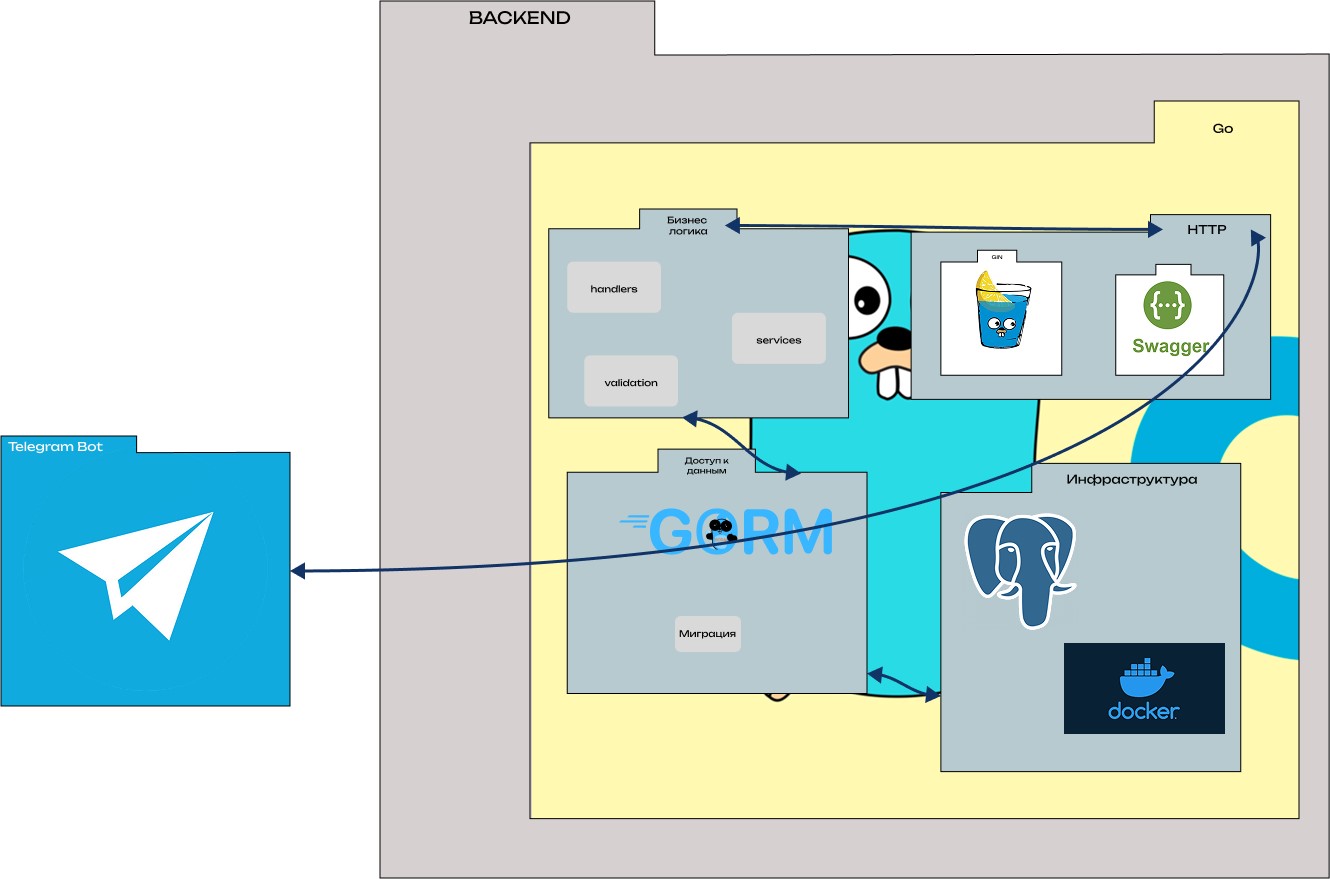


Рисунок 7 – Архитектура приложения

1. Уровень представления (Клиент) – Telegram Bot API

Telegram Bot API – используется для взаимодействия с пользователями через Telegram. Бот обрабатывает команды, кнопки и текстовые сообщения, валидирует ввод (например, проверяет корректность даты для встреч) и отправляет уведомления.

1. Уровень приложений (Сервер приложений) – Golang(Gin)

Golang в сочетании с фреймворком Gin представляет собой мощное решение для backend-разработки Telegram-бота.

1. Уровень данных (Сервер БД) – PostgreSQL

PostgreSQL: Выбор PostgreSQL обусловлен его надежностью, масштабируемостью и поддержкой разнообразных типов данных.

Таблица 3 – Матрица требований

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Требование** | **Суть** | **Автор** | **Ссылки** | **Критерий проверки** |
| 1 | Уровень представлений | | |  |  |
| 1.1 | Регистрация пользователя | “При первом запуске бот запрашивает имя и должность " | Николай Суворов | https://habr. com/ru/artic les/549408/ | Регистрация нового пользователя |
| 1.2 | Авторизация по Telegram ID | “Используется Telegram  ID для повторного  входа ” | Чибиток Д.С. | https://habr. com/ru/com pany/infopu lse/blog/346  318/ | При повторном  запуске бот  узнаёт пользователя и показывает его команды |
| 1.3 | Интерфейс управления командами | Меню для создания, редактирования, удаления команд | Чибиток Д.С. | https://habr. com/ru/com panies/first/ articles/497  342/ | Интерфейс отображает список команд, кнопки «Создать»,  «Удалить» |
| 1.4 | Отправка уведомлений | Бот асинхронно отправляет  напоминания и статусы | Жаворонк ов М.А. | https://habr. com/ru/com  panies/first/ articles/497  342/ | При изменении задачи пользователь получает уведомление |
| 2 | Уровень приложений | | |  |  |
| 2.1 | Производитель  ность при одновременной нагрузке | Система должна обеспечивать отклик не более 500мс при 1000 одновременных | Жаворонк ов М.А. | https://devel oper.mozilla .org/ru/docs  /Web/Perfor | Стресс- тестирование серверного программного |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | пользовательских запросах | |  | mance/Fund amentals | обеспечения |
| 2.2 | Надежность | Система должна  гарантировать круглогодичную доступность системы не менее 99% (простой не более 55 часов) | | Чибиток Д.С. | https://habr. com/ru/com panies/yoo money/articl es/591803/ | Мониторинг времени работы системы |
| 2.3 | Масштабируем  ость | Система должна быть готова к вертикальному и горизонтальному масштабированию в  случае увеличения  нагрузки | | Чибиток Д.С. | https://habr. com/ru/artic les/415773/ | Планы масштабирован ия аппаратного обеспечения |
| 3 | Уровень данных | |  | | | |
| 3.1 | Telegram ID в базе | Telegram используется уникальный идентификатор | ID как | Алексей Яковенко | https://habr. com/ru/com panies/amve ra/articles/8 48644/ / | В таблице users Telegram ID — уникальный ключ |
| 3.2 | Шифрование  данных во  время передачи | Система должна использовать протокол HTTPS для передачи  данных между  клиентом и сервером | | Чибиток Д.С. | https://proto n.me/blog/h  ttps | Использование  протокола  HTTPS |

# Практическая работа №8

Обоснованный выбор ГОСТа проекта:

Для оформления ТЗ был выбран ГОСТ 34.602-2020, поскольку он наиболее полно и точно позволяет описать требования к разрабатываемому технологическому и информационному комплексу.

ТЗ по выбранному ГОСТу:

1. Общие сведения:

1.1. Полное наименование системы и ее условное обозначение:

* «Телеграм-бот для управления задачами и встречами"», условное обозначение: «TaskBoard».

1.2. Наименование разработчиков системы и реквизиты заказчика:

* Заказчик – кафедра МОСИТ;
* Разработчики – студенты группы ИКБО-15-22 Кудинов А.В.,

Оганисян Г.А, Шаралапов Д.А.

1.3. Основания для разработки АС:

* Практическая работа по дисциплине «Системная и программная инженерия»

1.4. Плановые сроки начала и окончания работы по созданию системы:

* Сбор и анализ требований – 01 февраля 2025 г. - 28 февраля 2025 г.
* Проектирование архитектуры (REST API, БД, Telegram‑бот). – 01 марта 2025 г. – 15 марта 2025 г.
* Разработка и модульное тестирование. – 15 марта 2025 г. – 20 апреля 2025 г.
* Интеграция, развёртывание и приёмочное тестирование – 21 апреля 2025 г. – 30 апреля 2025 г.

1.5. Источник финансирования работ по созданию АС:

* Собственные средства разработчиков.

1.6. Порядок оформления и предъявления заказчику результатов работ по созданию системы:

* К результатам труда разработчика относится:
  + 1. Оригинальное аппаратное обеспечение;
    2. Оригинальное программное обеспечение;
    3. Уникальные структуры данных;
    4. Типовые проектные решения и особенности построения распределенной системы;
    5. Проектная и рабочая документация.
* Результаты передаются заказчику частями по завершении каждой стадии работы по созданию системы.

1. Назначение и цели создания (развития) системы:

2.1. Назначение системы:

* Обеспечить инструмент для управления задачами внутри команды и организации встреч с помощью Telegram-бота;

2.2. Цели создания системы:

* Автоматизация процесса распределения задач;
* Упрощённое взаимодействие между руководителем и участниками через интерфейс Telegram.

1. Характеристика объектов автоматизации:

3.1. Краткие сведения об объекте автоматизации:

* Решаемые проблемы: децентрализация задач, отсутствие контроля за встречами, путаница в планировании.

3.2. Сведения об условиях эксплуатации объекта автоматизации:

* Telegram-бот работает на сервере, использует API Telegram; - Пользователи взаимодействуют через Telegram-приложение.

1. Требования к системе:

4.1. Требования к системе в целом:

* Взаимодействие с пользователем — Telegram-интерфейс;
* В качестве протокола взаимодействия между компонентами Системы на транспортно-сетевом уровне необходимо использовать протокол TCP/IP;
* Для организации информационного обмена между компонентами Системы должны использоваться специальные протоколы прикладного уровня, такие как HTTP и его расширение HTTPS.
* Для организации доступа пользователей к приложению должен использоваться протокол презентационного уровня HTTP и его расширение HTTPS.

4.2. Требования к численности и квалификации персонала системы и режиму его работы:

* Руководитель: Может создавать до 3 команд; Управляет задачами и встречами; Назначает участникам задания; Приглашает участников через ссылки
* Участник: Может состоять только в одной команде; Получает задачи, отмечает их выполнение; Отправляет отчёты.
* Ограничения: В одной команде — не более 5 участников.

4.3. Требования к надежности и безопасности:

* Конфиденциальность аккаунтов пользователе.

4.3.1. Требования по защите информации от несанкционированного доступа:

* Требования не предъявлены.

4.3.2. Требования по сохранности информации при авариях:

* Требования не предъявлены

4.4. Требования к видам обеспечения:

4.4.1. Требования к информационному обеспечению:

* В качестве входной информации выступают: данные пользователя.
* В качестве выходной информации выступают: информация о задачах и встречах.

4.4.2. Требования к программному обеспечению:

* На сервере должна быть установлена ОС Microsoft Windows не ниже 10 версии;
* На сервере должна быть установлена СУБД PostgreSQL.
* Наличие Telegram.

4.4.3. Требования к техническому обеспечению:

* Для функционирования ИС необходимо: локальная вычислительная сеть на основе протокола TCP/IP с пропускной способностью 10/100 Мбит/с.
* Сервер должен удовлетворять следующим минимальным требованиям:
  + - 1. процессор Intel Xeon или аналогичный;
      2. 4 GB и более оперативной памяти;
      3. 5 GB – жесткий диск
* Требования, предъявляемые к конфигурации клиентских станций:
  + - 1. процессор, с тактовой частотой не менее 3000 MHz;
      2. 4 GB оперативной памяти;
      3. Пользователь может использовать мобильное устройство на базе iOS, Android.

5. Состав и содержание работ по созданию системы:

- По окончании работ должен быть предоставлен отчет о проделанных работах и успешной работоспособности системы.

1. Порядок контроля и приемки системы:
   1. Приемка этапа заключается в рассмотрении и оценке проведенного объема работ и предъявленной технической документации в соответствии с требованиями настоящего технического задания.
   2. Ответственность за организацию и проведение приемки системы должен нести заказчик. Приемка системы должна производиться по завершению приемки всех задач системы. При этом необходимо предоставить обеспечение материальной частью (технические средства), проектной документацией и специально выделенным персоналом.
   3. Заказчик должен предъявлять систему ведомственной приемочной комиссии, при этом он обязан обеспечить нормальные условия работы данной комиссии в соответствии с принятой программой приемки.
   4. Завершающим этапом при приемке системы должно быть составление акта приемки.
2. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие:
   1. Для обеспечения готовности объекта к вводу системы в действие провести комплекс мероприятий:

- Приобрести компоненты технического и программного обеспечения, заключить договора на их лицензионное использование; - Завершить работы по установке технических средств.

1. Требования к документированию: отчетные материалы должны включать в себя скриншоты интерфейса системы, листинги наиболее важных элементов кода, а также доступные функции для каждой роли, участвующей в системе.
2. Источники разработки: не представлены.

# Вывод

В результате выполнения практических работ были построены диаграмма классов, диаграмма объектов, диаграмма основного процесса в нотации IDEF0, диаграмма в нотации DFD, построена нормализованная схема базы данных, описана архитектура системы и обоснован выбор определенных программных решений, построена архитектурная диаграмма разработки, а также составлено подробное ТЗ по выбранному ГОСТу.