

## Бот для анализа эмоционального состояния

Участник:

ученик 10 «А» класса ГБОУ «Школа Глория»

Бердников Артем Валерьевич

Руководители:

педагог ГБОУ «Школа Глория»

Зацепина Маргарита Викторовна

педагог технопарка «РГСУ»

Ковалёв Ягуар Игоревич

# Актуальность

В современном мире человек живёт в постоянном напряжении и стрессе. Все большую популярность набирают боты, помогающие пользователям в заботе о себе. Отслеживание и запись состояния за день, а также анализ его изменения, наведёт человека на мысли: “что я сделал сегодня такого, из-за чего мое состояние хуже, чем вчера?” или “чем я занимался, что я чувствую себя лучше, чем на прошлой неделе?”. Таким образом, бот поможет выявлять закономерности вашего хорошего самочувствия и пребывать в нем чаще.

# Цель и задачи проекта

## **Цель:**

Создать удобного и простого в использовании Telegram-бота, который будет проводить ежедневные опросы, собирать данные о настроении пользователей и помогать в поддержании эмоционального равновесия.

## **Задачи:**

1. Изучить техническую литературу по теме, выбрать технологии и среду разработки и ознакомиться с основными концепциями создания Telegram-бота, его возможным функционалом, программными решениями.
2. Спроектировать чат-бота для отслеживания эмоционального состояния, в котором пользователь тратил бы наименьшее время из возможного для прохождения опросов.
3. Разработать и протестировать Telegram-бот.

# Обзор аналогов



Daylio

- + Отметка состояния и активностей за день
- + Оценка сна
- + Возможность ставить новые цели
- + Множество вариантов статистик и динамик изменения состояния

- Есть реклама
- Для полного функционала нужна премиум подписка
- Нужно устанавливать отдельное приложение

# Обзор аналогов



**Moodfit**

- + Отметка состояния
- + Отслеживание сна, питания и физической активности
- + Конфиденциальность и защита паролем
- + Персонализированные отчеты

- Без премиум подписки доступен лишь дневник настроения
- В бесплатной версии нет пароля
- Нужно устанавливать отдельное приложение
- Не поддерживается русский язык

# Выбор среды и технологий разработки

# Язык программирования - Python

- + Простота и читаемость
- + Широкое сообщество
- + Асинхронное программирование



Рисунок 2 - Python

# Библиотека для создания Telegram-бота – aiogram

- + Асинхронность
- + Простота использования
- + Полная поддержка Telegram Bot API



Рисунок 3 - aiogram



# Система управления базой данных – PostgreSQL

- +** Высокая производительность и надежность
- +** Поддержка сложных запросов
- +** Хорошая интеграция с Python

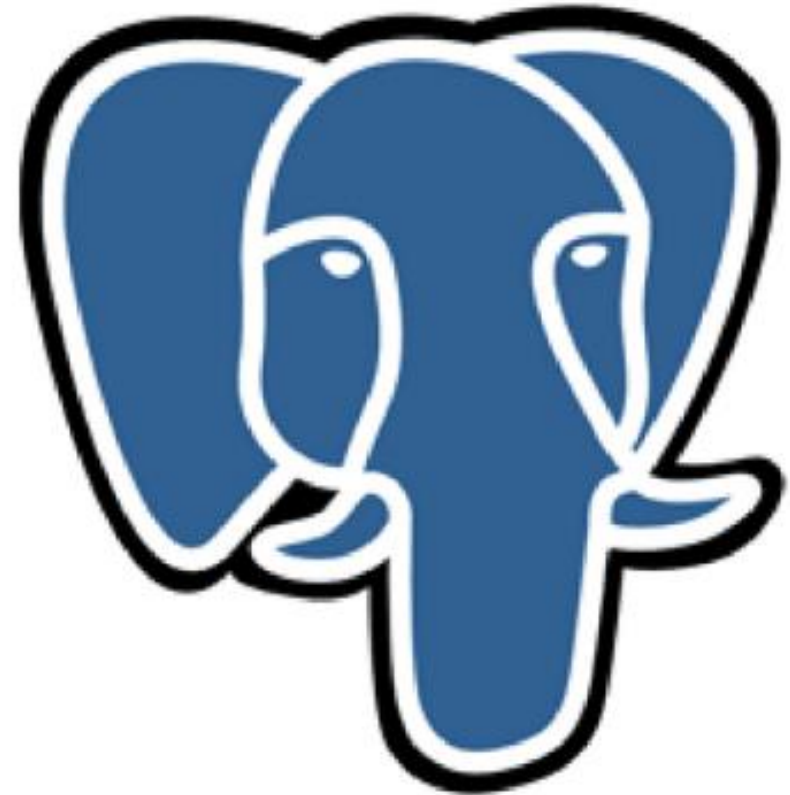


Рисунок 4 - PostgreSQL

# Платформа контейнеризации - Docker

+ Простота настройки

+ Удобство работы

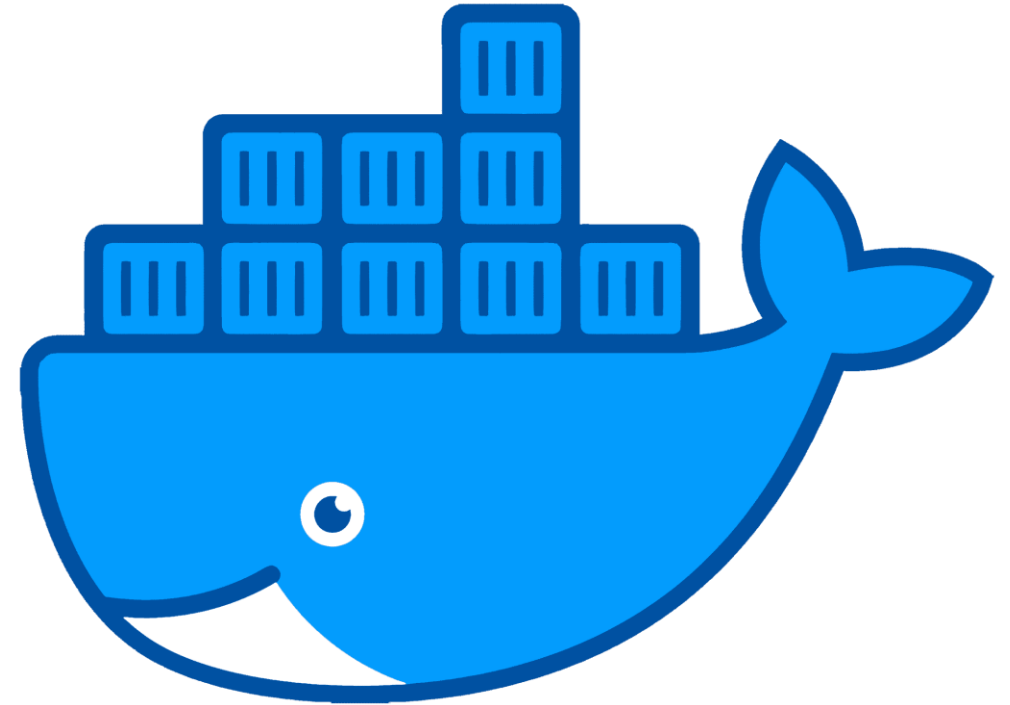


Рисунок 5 - docker

# Разработка концепта

Регистрация



Выбор времени

Прохождение  
опроса



Статистика

# Функциональное моделирование

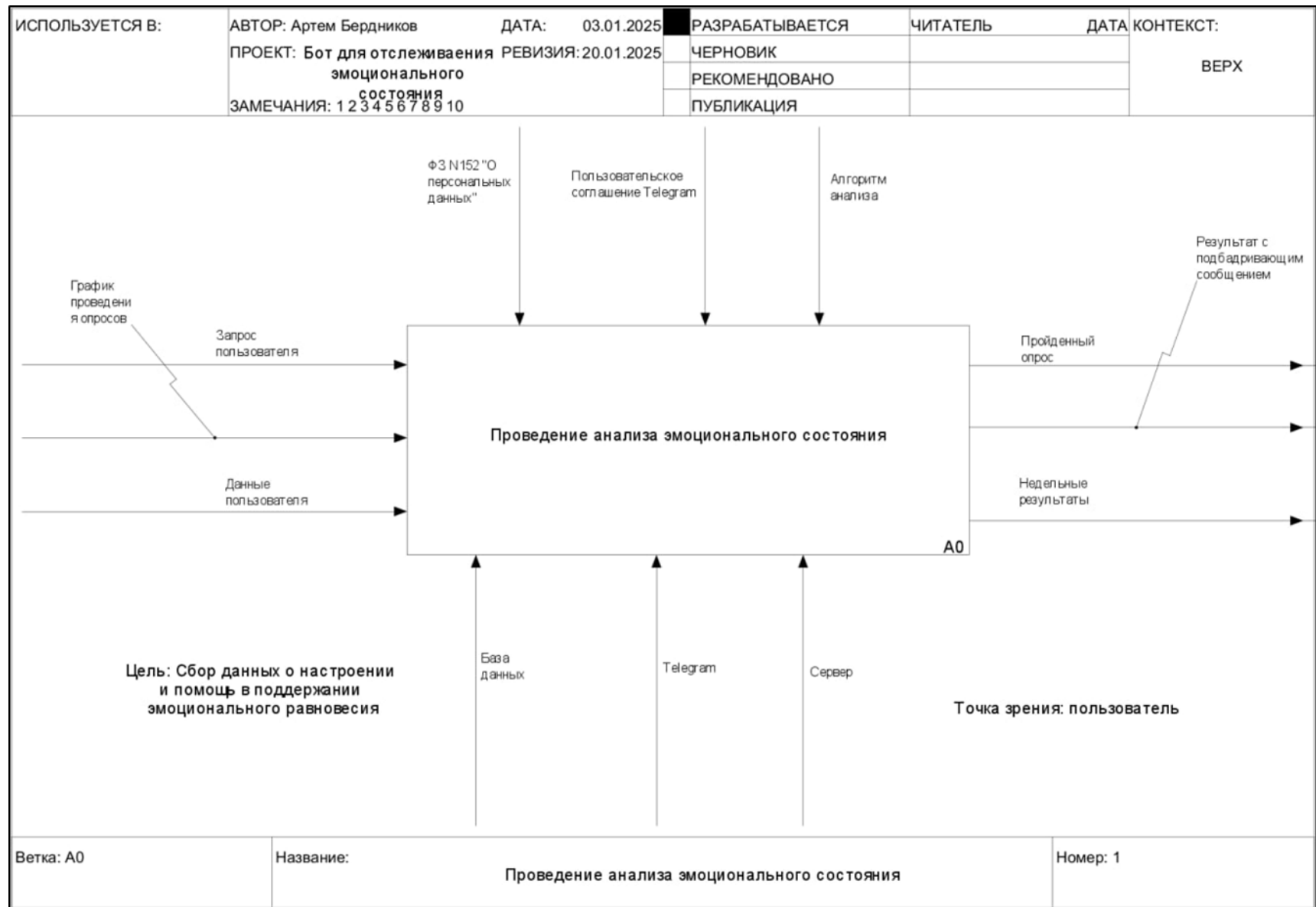


Рис. 6. Контекстная диаграмма «Проведение анализа эмоционального состояния»

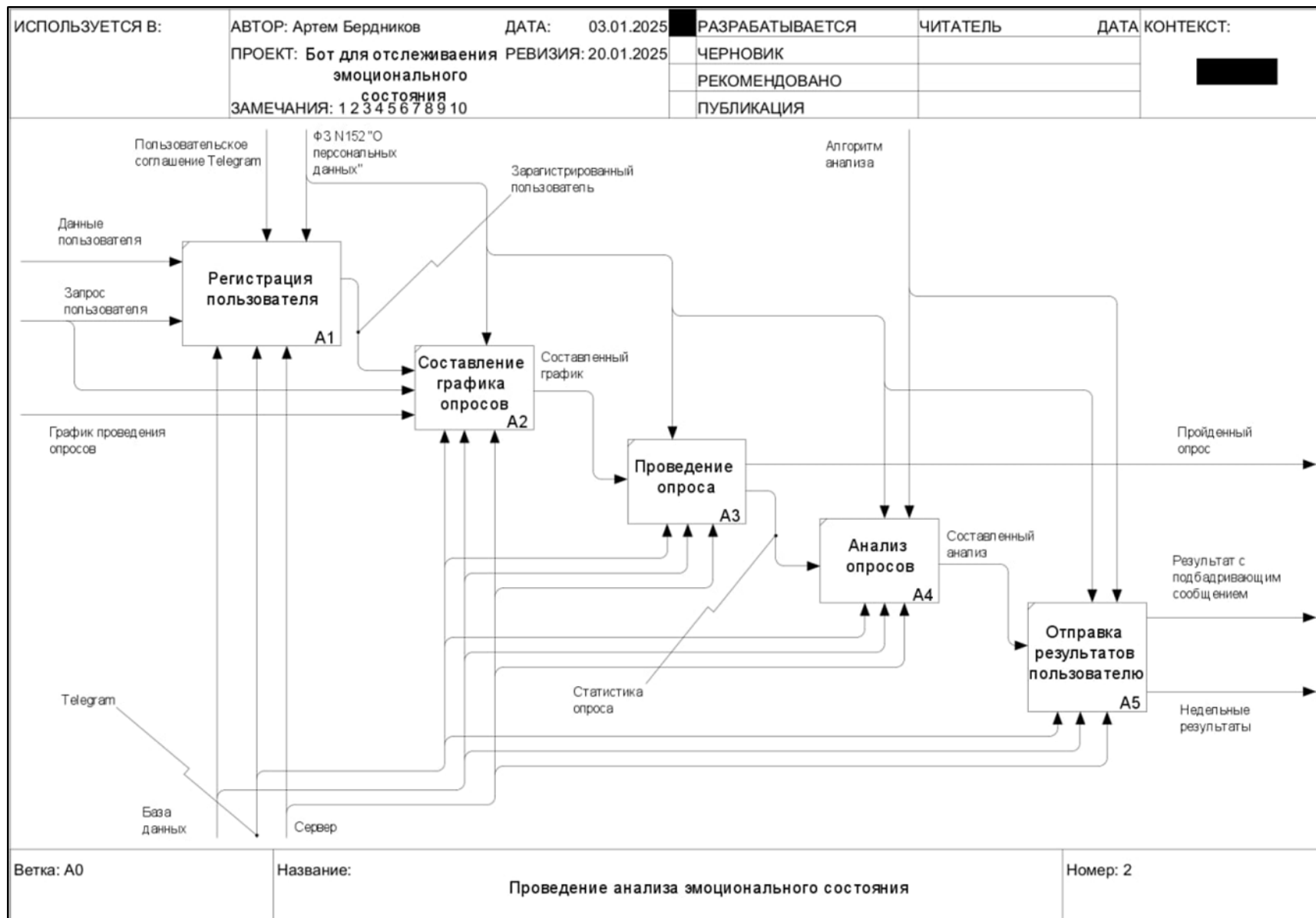


Рис. 7. Декомпозиции процесса «Проведение анализа эмоционального состояния»

# ERD-диаграммы

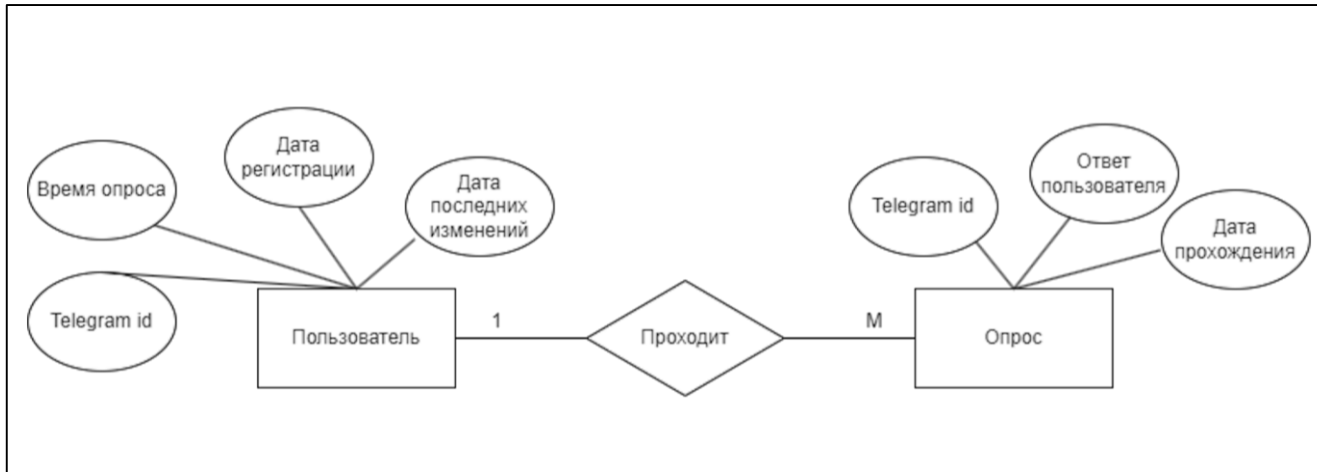


Рис. 8. Логическая ERD-диаграмма в нотации Чена

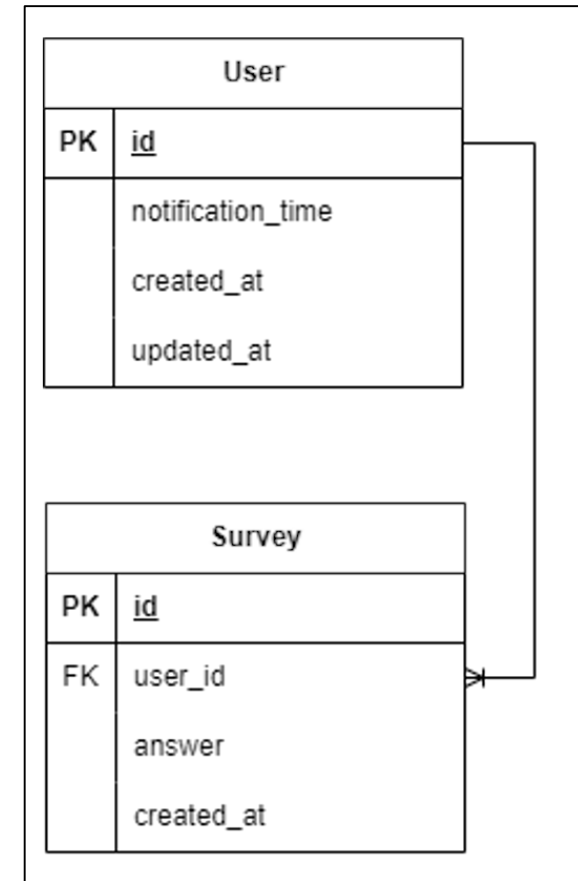


Рис. 9. Физическая ERD-диаграмма в нотации Мартина

# Статистика и метрика



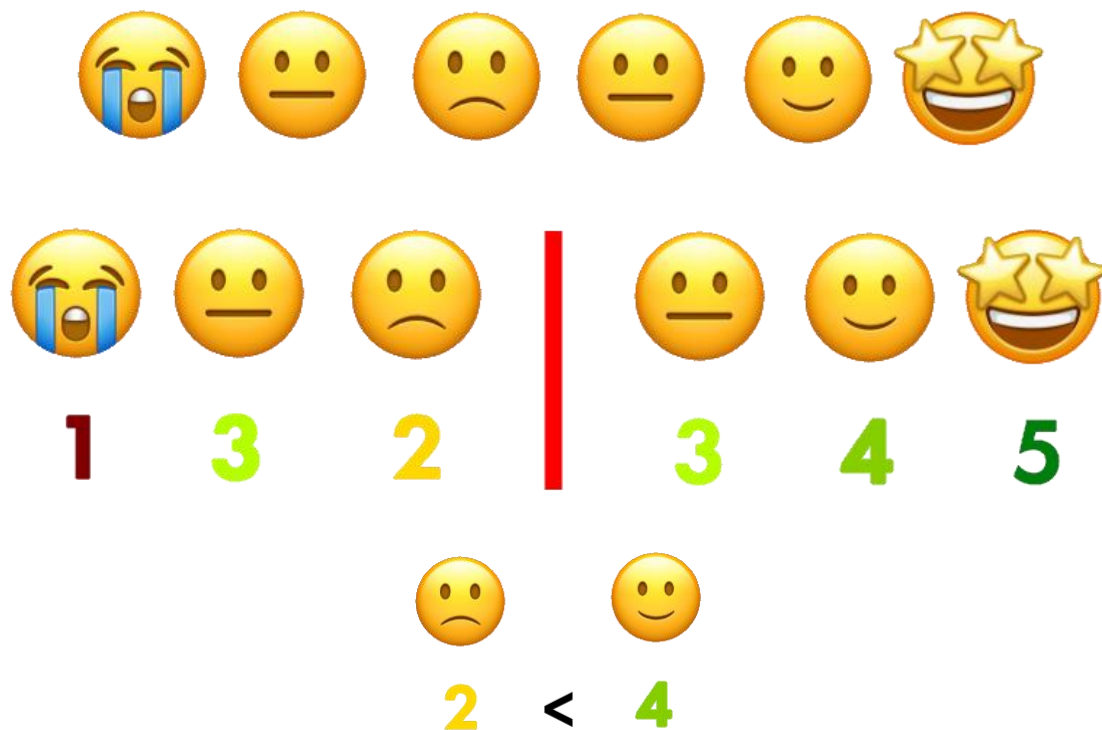
## Взвешенное среднее настроение

Среднее взвешенное — это среднее арифметическое значение, в котором учтён вес каждого из слагаемых, для которых рассчитывается это среднее значение. Другими словами, каждому исходному показателю присваивается определённый вес.

$$\bar{x} = \frac{x_1 f_1 + x_2 f_2 + \dots + x_n f_n}{f_1 + f_2 + \dots + f_n} = \frac{\sum x f}{\sum f}$$

# Тренд настроения

Тенденция изменения настроения за неделю: положительная, отрицательная или стабильная.



# Эмоциональная стабильность

Среднеквадратичное отклонение — характеристика, используемая в статистике для измерения степени изменчивости или разброса данных.

$$D(X) = M(X^2) - (M(X))^2$$

$$\sigma = \sqrt{D(X)}$$

# График



Рис. 10. График настроения за неделю

# База данных

```
CREATE_USERS_TABLE = """
CREATE TABLE IF NOT EXISTS users (
    id BIGINT PRIMARY KEY,
    notification_time TIME NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);
"""

CREATE_SURVEY_RESULTS_TABLE = """
CREATE TABLE IF NOT EXISTS survey_results (
    id SERIAL PRIMARY KEY,
    user_id BIGINT NOT NULL REFERENCES users (id),
    answer INTEGER NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);
"""

async def setup_database():
    conn = await connect(config.DATABASE_URL)
    try:
        await conn.execute(CREATE_USERS_TABLE)
        await conn.execute(CREATE_SURVEY_RESULTS_TABLE)
        logger.info("Таблицы созданы.")
    except Exception as e:
        logger.error(f"Ошибка при создании таблиц: {e}")
    finally:
        await conn.close()

async def get_database_connection():
    logger.info("Получение соединения с базой данных")
    conn = await connect(config.DATABASE_URL)
    return conn
```

Рис. 11. Создание, настройка  
и подключение базы данных

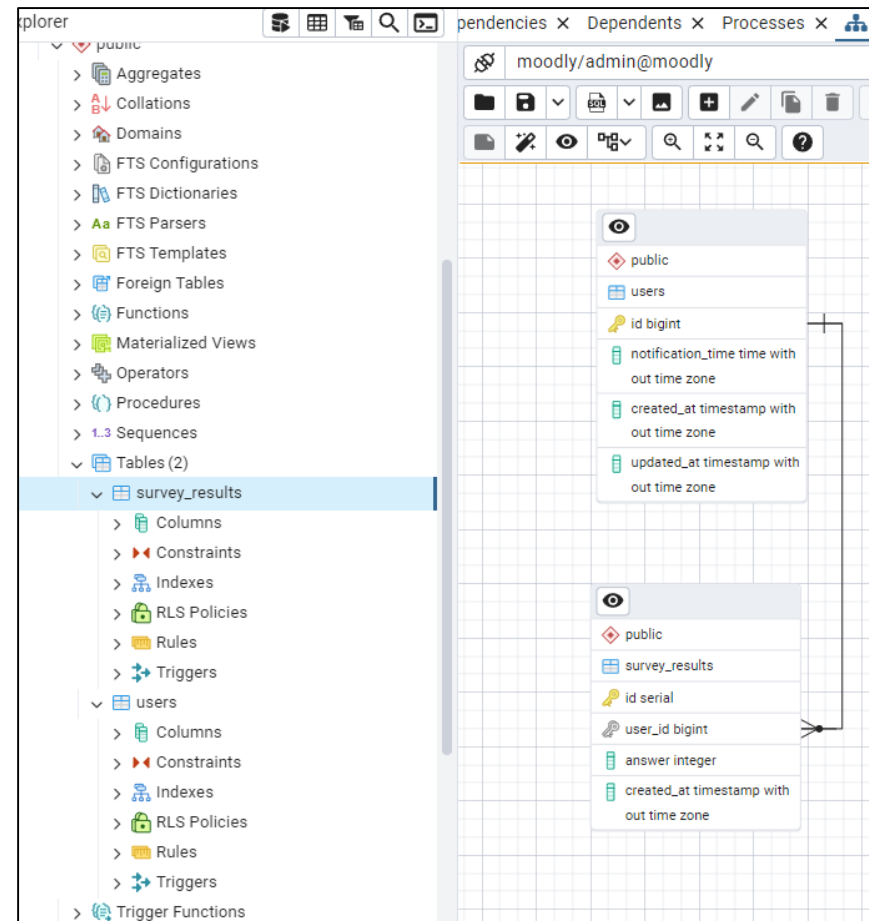


Рис. 12. pgAdmin

# Проведение опроса

```
async def survey_handler(message: types.Message, state: FSMContext):
    user_id = message.from_user.id
    logger.info(f"Запуск опроса для пользователя {user_id}")
    await message.reply(
        "Как вы себя чувствуете сегодня? Выберите смайлик:", reply_markup=keyboard
    )
    await state.set_state(SurveyState.in_progress)
    logger.info(f"Состояние установлено для пользователя {user_id}")

async def handle_survey_response(message: types.Message, state: FSMContext):
    logger.info("Обработчик handle_survey_response вызван")
    mood = None

    if message.text == BUTTON_TEXTS[1]:
        mood = 1
    elif message.text == BUTTON_TEXTS[2]:
        mood = 2
    elif message.text == BUTTON_TEXTS[3]:
        mood = 3
    elif message.text == BUTTON_TEXTS[4]:
        mood = 4
    elif message.text == BUTTON_TEXTS[5]:
        mood = 5

    logger.info(f"Определённое настроение: {mood}")
```

Рис. 13. Проведение опроса

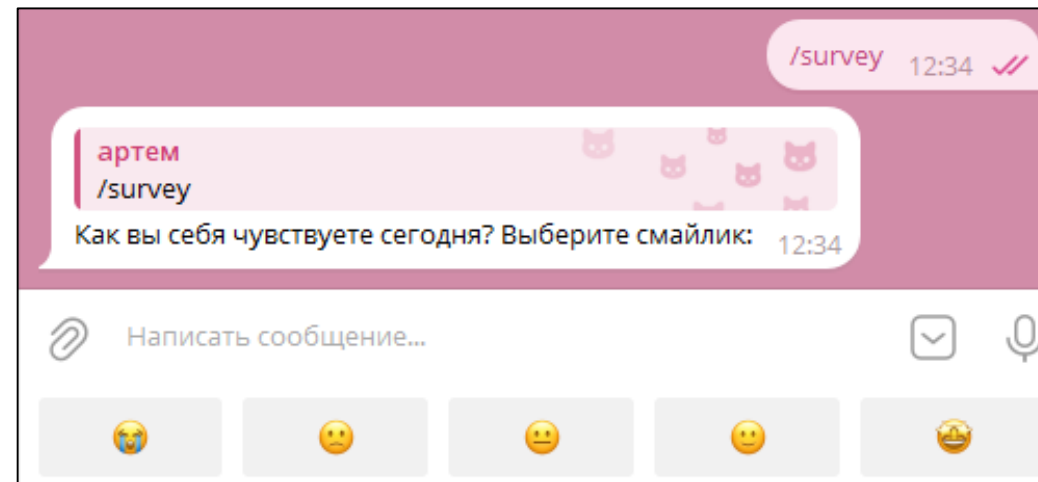


Рис. 14. Проведение опроса в чате

# Построение графика

```
async def create_mood_chart(results):
    if not results:
        return None

    dates = [row["created_at"] for row in results]
    moods = [row["answer"] for row in results]

    plt.figure(figsize=(10, 5))
    plt.plot(dates, moods, marker="o", linestyle="-", color="#FE53BB")
    plt.title("График настроения за неделю")
    plt.xlabel("Дата")
    plt.ylabel("Настроение")
    plt.yticks(range(1, 6), [BUTTON_TEXTS2[i] for i in range(1, 6)])
    plt.grid(True)
    mplcyberpunk.make_lines_glow()
    mplcyberpunk.add_gradient_fill(alpha_gradientglow=0.38)
    buf = io.BytesIO()
    plt.savefig(buf, format="png")
    buf.seek(0)
    plt.close()

    return buf
```

Рис. 15. Построение графика



Рис. 16. График в чате

# Тестирование

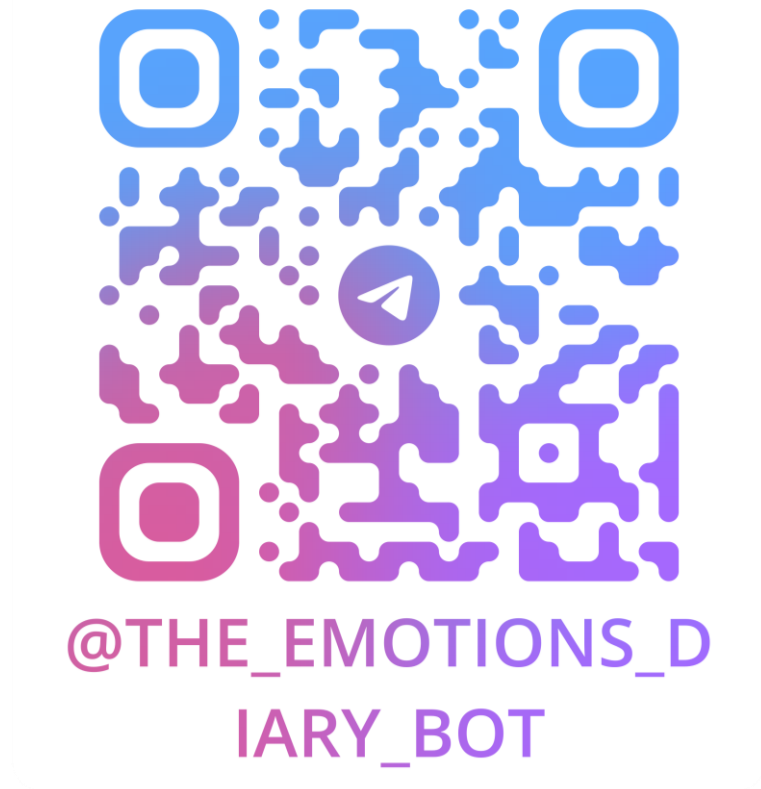


Рис. 17. QR-код Moodly



Рис. 18. Moodly



# Тестирование

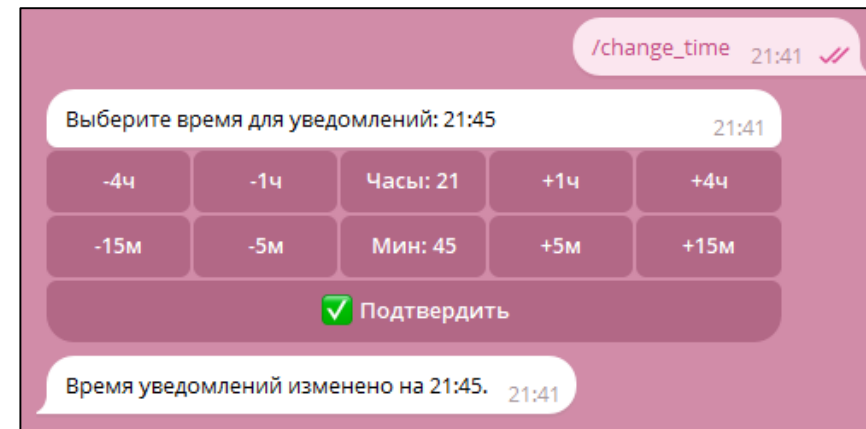
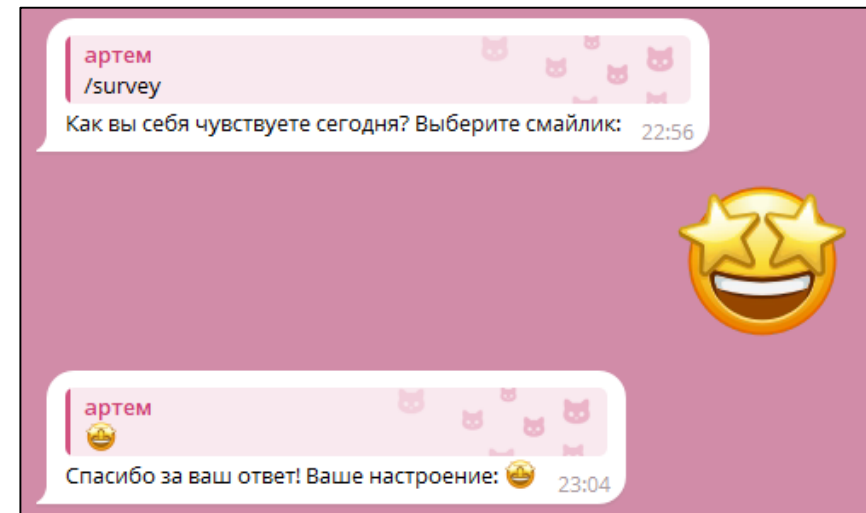
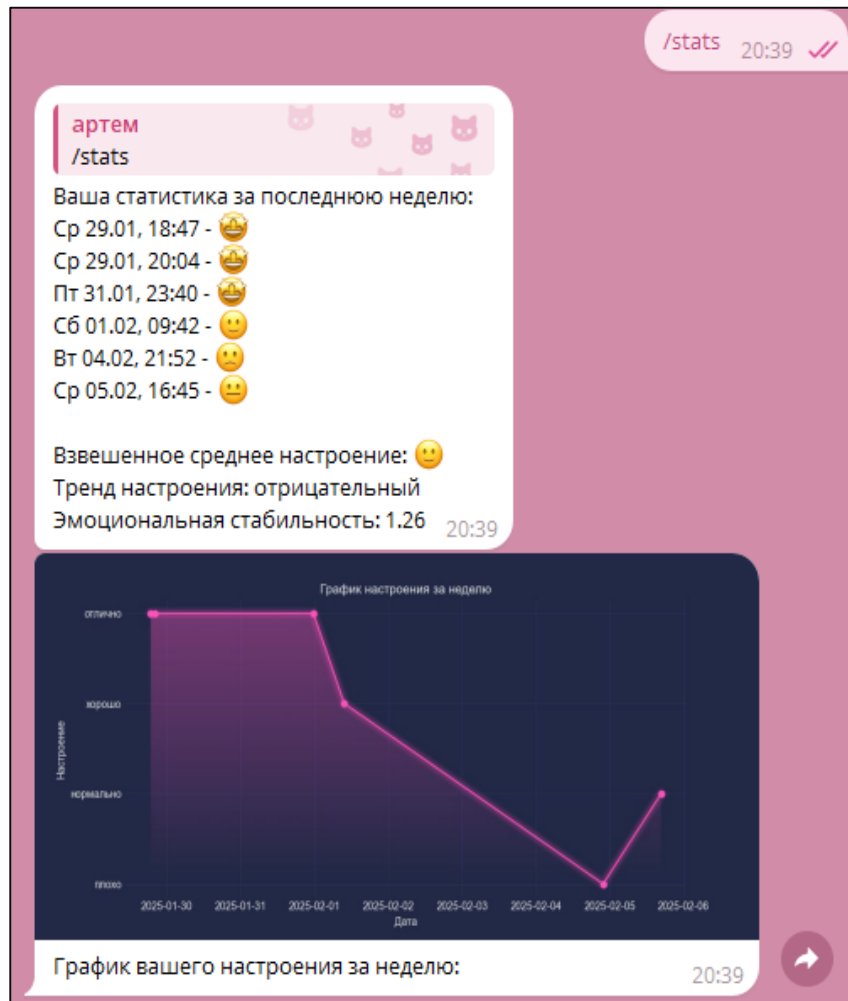


Рис. 19, 20, 21. Демонстрация работы бота

# Проблема с часовыми поясами

(>> вставить скрины с демонстрацией)

(>> вставить скрины с решением)

# Перспективы развития проекта

1. Комментарии и заметки как часть прохождения опроса. В недельных результатах нейросеть соберет все комментарии в один текст, «сожмет» его и направит пользователю.
2. Нейросеть, которая будет общаться с пользователем, поддерживать и давать мотивацию. Появится возможность рассказать о наблевшем или о своих успехах.
3. Поддержка нескольких языков.
4. Дополнительный функционал: отслеживание сна, питания, физической активности, установка целей и так далее.

# Заключение

1. Изучена техническая литература по теме и основные концепции создания Telegram-ботов на Python.
2. Спроектирован чат-бот для отслеживания эмоционального состояния.
3. Разработан и протестирован сам Telegram-бот.

Цель проекта достигнута, все задачи выполнены

# Список литературы

1. Байкова, К. Д. АНАЛИЗ И РАЗРАБОТКА ФУНКЦИОНАЛЬНОГО TELEGRAM-БОТА / К. Д. Байкова, П. А. Медведева // Молодой исследователь Дона. – 2021. – № 6. – С. 15-20. – ISSN 2500-1779
2. Какие бывают Телеграм-боты и для чего они нужны : сайт. – URL: <https://practicum.yandex.ru/blog/telegram-boty-kak-rabotayut-i-kak-nastroit/> (дата обращения: 12.01.2025)
3. Курс "Телеграм-боты на Python и AIogram" : сайт. – URL: <https://stepik.org/120924> (дата обращения: 22.01.2025)
4. Любанович, Б. Простой Python. Современный стиль программирования. / Б. Любанович. – Санкт-Петербург : Питер, 2016. – 477 с. – ISBN 978-5-496-02088-6.
5. О персональных данных : Федеральный закон от 27.07.2006 N 152-ФЗ // Собрание законодательства РФ. – 2006. – № 31, ч. 1. – Ст. 3451.

# Список литературы

6. Часовой пояс в ботах для Telegram : сайт. – URL: <https://4xpro.ru/profblog/telegram-timezones/> (дата обращения: 02.02.2025)
7. Шумилина, М. А. РАЗРАБОТКА ЧАТ-БОТА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON В МЕССЕНДЖЕРЕ «TELEGRAM» / М. А. Шумилина, А. В. Коробко // Научные известия. – 2022. – № 28. – С. 47-54. – ISSN 2413–6492
8. aiogram : сайт. – URL: <https://aiogram.dev/> (дата обращения: 19.12.2024)
9. BotFather : сайт. – URL: <https://telegram.me/BotFather> (дата обращения: 22.01.2025)
10. docker : сайт. – URL: <https://docs.docker.com> (дата обращения: 20.12.2024)
11. matplotlib : сайт. – URL: <https://matplotlib.org/stable/index.html> (дата обращения: 29.01.2025)
12. Python : сайт. – URL: <https://www.python.org/> (дата обращения: 19.12.2024)