

Телеграмм бот "Расписание"

Выполнил студент группы

Б9121-09.03.04 (5)

Шевелев Р.В.

Проблема

Как студенты сейчас ориентируются в расписании?

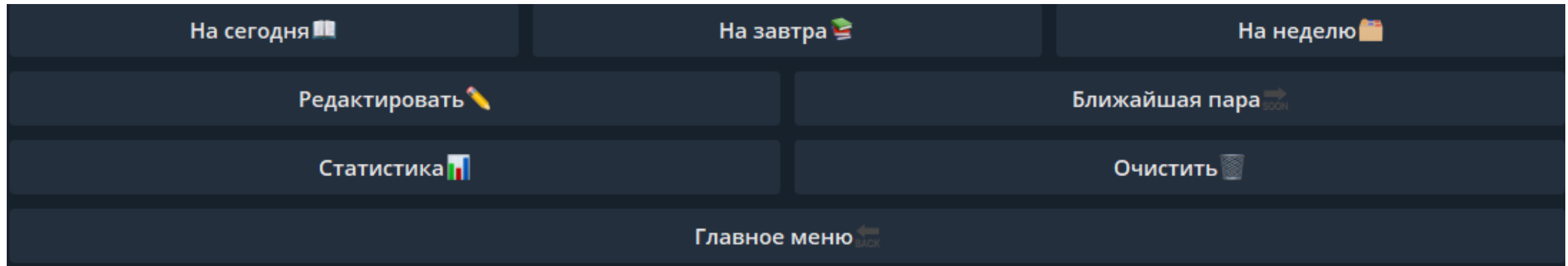
- Пользуются расписанием, которое предоставляет `univer.dvfu`
- Записывают расписание вручную в календарь или блокнот.
- Легко забывают о дисциплинах по выбору.

Решение

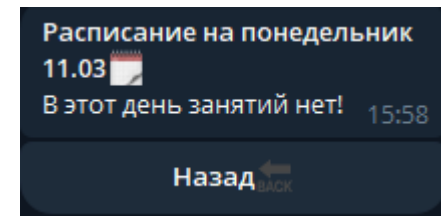
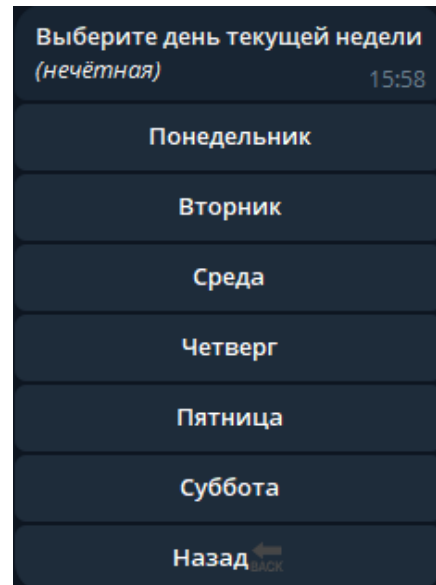
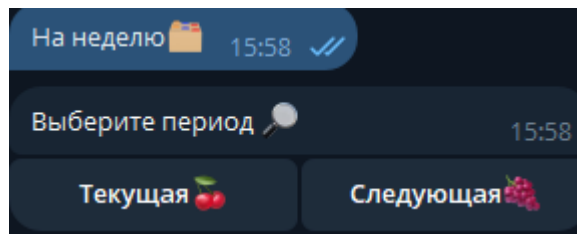
Телеграмм-бот для расписания студентов:

- **Заполнение расписания:** Студент заполняет расписание о всех занятиях, включая дисциплины по выбору, дополнительные активности и консультации.
- **Формирование удобного расписания:** Бот формирует расписание в удобном для студента формате, например, в виде календаря или списка, а также просто формирует ответ в диалоге с пользователем
- **Уведомления о предстоящих занятиях:** Бот присылает уведомления о предстоящих занятиях, чтобы студент не пропустил их. Настройка уведомлений гибкая: от 5 до 60 минут.

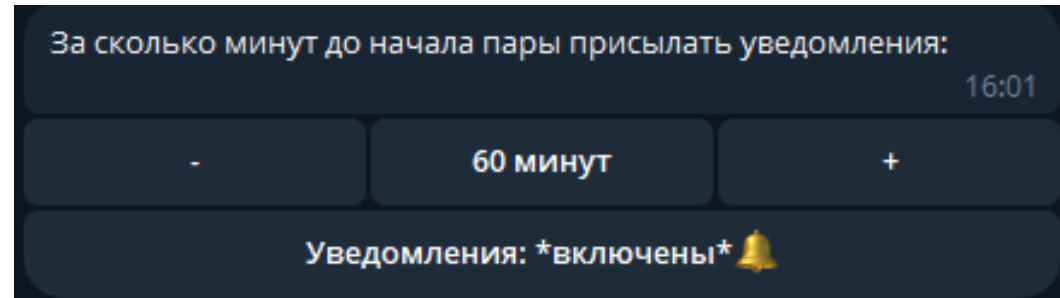
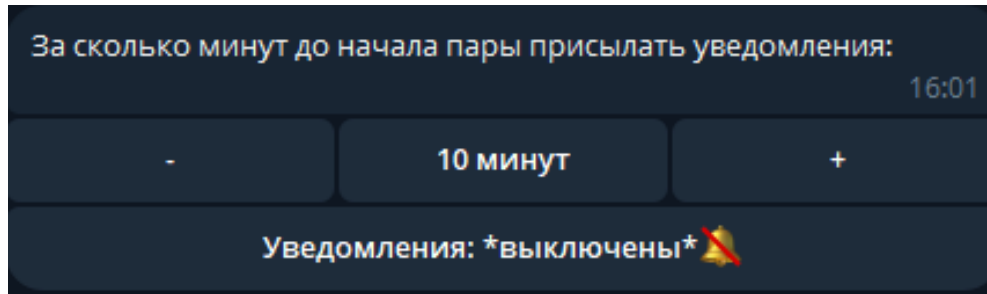
Главная панель пользователя



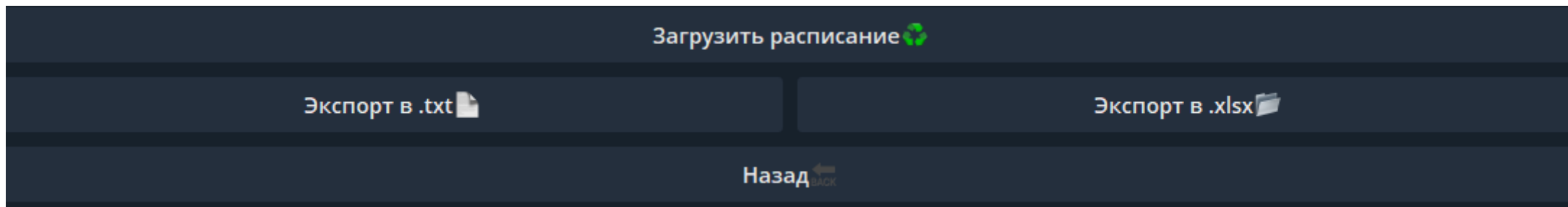
Диалог с пользователем



Настройка уведомлений



Выгрузка расписания в файлы



Реализация

- В ходе разработки бота были использованы библиотеки:
- Aiogram – библиотека для работы с Telegram API
- Asyncio – библиотека для асинхронного вызова функций
- Pytz – библиотека для работы со временем (даты и настройка часовых поясов для сервера)
- Redis – память, которую использует бот в процессе работы
- Sqlite3 – встроенная библиотека для работы с базой данных
- Прочие библиотеки для работы с excel файлами

Код был написан в функциональном стиле.

Код создания базы данных

```
import sqlite3
from config import database_path, sql_path

def execute_query(query_path):
    with open(query_path, 'r', encoding='utf-8') as file:
        queries = file.read().split(';')

    connection = sqlite3.connect(database_path)
    cursor = connection.cursor()

    try:
        for query in queries:
            if query.strip():
                cursor.execute(query)
        connection.commit()
    finally:
        connection.close()

if __name__ == "__main__":
    execute_query(sql_path)
```

Код создания нескольких таблиц, используемых в БД

```
CREATE TABLE IF NOT EXISTS users (  
    user_id INTEGER PRIMARY KEY,  
    username VARCHAR(255),  
    notifications BOOLEAN DEFAULT false,  
    minutes_before INT DEFAULT 10,  
    registration_date TEXT DEFAULT CURRENT_TIMESTAMP NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS lesson_times (  
    lesson_time_id INTEGER PRIMARY KEY AUTOINCREMENT,  
    start_time TIME NOT NULL,  
    end_time TIME NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS week_parity (  
    week_parity_id INTEGER PRIMARY KEY AUTOINCREMENT,  
    day_of_week INT NOT NULL,  
    is_odd_week BOOLEAN NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS schedule (  
    user_id INTEGER NOT NULL,  
    week_parity_id INTEGER NOT NULL,  
    tutor_name VARCHAR(255) NOT NULL,  
    subject_name VARCHAR(100) NOT NULL,  
    is_practice BOOLEAN NOT NULL,  
    subject_priority INTEGER NOT NULL,  
    subject_place VARCHAR(20) NOT NULL,  
    lesson_time_id INTEGER NOT NULL,  
    FOREIGN KEY (week_parity_id) REFERENCES week_parity(week_parity_id) ON DELETE RESTRICT,  
    FOREIGN KEY (lesson_time_id) REFERENCES lesson_times(lesson_time_id) ON DELETE RESTRICT,  
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE  
);
```


Функции из python-пакета, которые используются

```
def get_random_book_emoji(book_emojis):
    if not book_emojis:
        return "\U0001F4D9"
    random_book = random.choice(book_emojis)
    book_emojis.remove(random_book)
    return random_book

def get_week_title(data):
    return f"<b>Выберите день {'текущей' if data == 'this_week' else 'следующей'} недели</b>\n" \
           f"{'<i>(нечётная)</i>' if get_week_parity() == (data == 'this_week') else '<i>(чётная)</i>'}"
```

```
def get_week_parity():
    academic_year_start = datetime(datetime.now().year, start_month, start_day)
    current_date = datetime.now()
    days_since_start = (current_date - academic_year_start).days
    current_week_number = days_since_start // 7 + 1
    week_parity = 0 if current_week_number % 2 == 0 else 1
    return week_parity
```

Что можно улучшить в уже написанном коде?

Если еще раз взглянуть на проект, то можно улучшить следующее:

Для всех функций добавить тип возвращаемого значения, так как иногда можно получить неожиданный тип данных.

Добавить больше комментариев к некоторым функциям для программистов, так как многие из них выполняют объемный функционал.

Разбить сложные функции на более мелкие для повышения читаемости и возможности более легкой поддержки кода.

Преимущества использования бота

- Удобство: Бот всегда под рукой. Если расписание на группу одинаковое, то заполнив его один раз, можно поделиться со всем потоком/группой.
- Актуальность: показывает расписание, которое сформировал сам студент.
- Надежность: бот не позволит вам забыть о ваших занятиях.
- Доступность: бесплатно.

Спасибо за внимание!