**Государственное бюджетное общеобразовательное учреждение города Москвы Школа № 618**

**<<ЧАТ-БОТ В TELEGRAM ДЛЯ ШКОЛЬНОГО РАСПИСАНИЯ И ДОМАШНИХ ЗАДАНИЙ>>**

**Участники проекта:**

8Д класс ГБОУ Школа № 618

Нураков Данакер

Мирзоян Рубен

Бекасов Евгений

**Руководитель проекта:**учитель информатики ГБОУ Школа № 618  
Баранов Михаил Сергеевич

**Москва, 2026г**

**Оглавление**

**1.Введение................................................................................3**

**2.Цель и задачи проекта...........................................................5**

**3.Описание проекта..................................................................6**

**4.Описание алгоритма программы..........................................7**

**5.Алгоритм работы программы (блок-схема)..........................8**

**6.Обработка основных команд(блок-схемы)...........................9**

**7.Обработка нажатия кнопок(интерактивного меню)..............12**

**8.Скриншоты работы бота..........................................................13**

**9.Выводы......................................................................................17**

**10.Список используемой литературы..........................................18**

**2**

**Введение**

**Актуальность**

В современном учебном процессе ученики часто сталкиваются с потерей информации о домашних заданиях в чатах и на листочках, сложностью отслеживания изменений в расписании, а также отсутствием единого места для учебных материалов. Родители также испытывают трудности из-за неполной информированности об учебном процессе и зависимости от сообщений ребенка. Решением этих проблем является создание единой автоматизированной системы в привычном мессенджере.

**Почему была выбрана эта тема**

Telegram является популярным и доступным мессенджером среди школьников, учителей и родителей, что делает его удобной платформой для внедрения образовательных инструментов. Автоматизация рутинных процессов в школе позволит повысить организованность всех участников учебного процесса.

**Анализ представленных на рынке аналогов**

На рынке существуют различные решения для организации учебного процесса:

1. Электронные дневники (например, «Дневник.ру», «МЭШ»)
   * Требуют отдельной регистрации и входа через браузер или приложение.
   * Часто имеют сложный интерфейс.
   * Не интегрированы с мессенджерами.
2. Школьные чаты в WhatsApp/Telegram
   * Информация теряется в потоке сообщений.
   * Нет структурированного хранения данных.
   * Нет автоматических уведомлений.
3. Специализированные приложения для расписания (например, «School Timetable»)
   * Требуют установки отдельного приложения.
   * Нет функций для домашних заданий и уведомлений.
   * Часто платные или с ограниченным функционалом.

**3**

**Сравнение с нашим продуктом:**

|  |  |  |
| --- | --- | --- |
| **Критерий** | **Аналоги (электронные дневники, чаты)** | **Наш Telegram-бот** |
| Доступность | Требует отдельного приложения/входа | Работает прямо в Telegram |
| Удобство использования | Сложный интерфейс | Простой интуитивный интерфейс |
| Уведомления | Ограничены или отсутствуют | Автоматические напоминания |
| Хранение ДЗ и расписания | Разрозненно | Централизованная база данных |
| Бесплатность для пользователей | Часто платные или с ограничениями | Полностью бесплатен |
| Мобильность | Зависит от приложения | Доступен на любом устройстве с Telegram |

**4**

**Цель проекта**

Создать удобный и функциональный Telegram-бот для автоматизации доступа к школьному расписанию и домашним заданиям.

**Задачи проекта**

1. Разработать архитектуру и схему работы бота.
2. Реализовать функции:
   * Хранение и отображение расписания.
   * Ведение списка домашних заданий.
   * Систему уведомлений.
3. Создать интуитивно понятный интерфейс.
4. Обеспечить разный уровень доступа (ученик/учитель/родитель).
5. Протестировать работу всех функций.
6. Подготовить документацию и инструкцию по использованию.

**Используемые инструменты и технологии**

1. Язык программирования: Python 3.9+
2. Фреймворк: python-telegram-bot
3. База данных: SQLite / PostgreSQL
4. Хостинг: Heroku / VPS / PythonAnywhere
5. Внешние API: Telegram Bot API, Google Sheets API (опционально)
6. Инструменты разработки: Git, VS CodeQL, SiteBrowser
7. Диаграммы: [draw.io](https://draw.io/)

**5**

**ОПИСАНИЕ ПРОЕКТА**

Разработанный Telegram-бот для школьного расписания и домашних заданий представляет собой программное решение, предназначенное для автоматизации доступа учеников, учителей и родителей к учебной информации через популярный мессенджер Telegram. Бот функционирует как интерактивный помощник, который централизованно хранит, обрабатывает и предоставляет данные о расписании уроков и домашних заданиях.

**Ключевые характеристики проекта:**

1. Архитектура и взаимодействие:

· Пользователь отправляет команды боту через Telegram-интерфейс.

· Бот обрабатывает запросы с использованием библиотеки python-telegram-bot.

· Данные хранятся в локальной базе SQLite (с возможностью миграции на PostgreSQL).

· Реализована система ролей: ученик, учитель, родитель — с разным уровнем доступа к функциям.

2. Основные функциональные модули:

· Модуль расписания: позволяет получать расписание на сегодня, завтра и на всю неделю.

· Модуль домашних заданий: учителя могут добавлять задания, ученики — просматривать их, родители — отслеживать.

· Модуль уведомлений: автоматические напоминания о начале уроков и новых заданиях.

· Модуль настроек: пользователи могут настраивать время и частоту уведомлений.

3. Техническая реализация:

· Бот написан на языке Python версии 3.9+.

· Для развертывания использован хостинг Heroku, что обеспечивает круглосуточную доступность.

· Интерфейс реализован с помощью кнопок и текстовых команд для максимальной простоты использования.

· Все изменения фиксируются в репозитории GitHub для контроля версий.

4. Преимущества решения:

· Не требует установки дополнительных приложений — работает прямо в Telegram.

· Кроссплатформенность: доступен на смартфонах, планшетах и компьютерах.

· Нулевая стоимость использования для всех участников.

· Простота масштабирования и интеграции с внешними сервисами (например, Google Sheets для импорта расписания).

**Процесс работы:**

1. Пользователь запускает бота командой /start.

2. Бот приветствует пользователя и показывает доступные команды.

3. При выборе команды (например, /today) бот запрашивает данные из базы и возвращает форматированный ответ.

4. Учителя могут добавлять домашние задания через команду /homework, после чего ученики автоматически получают уведомления.

5. Система уведомлений работает по расписанию, напоминая о начале уроков.

Проект успешно прошёл тестирование и готов к внедрению в учебный процесс, демонстрируя эффективность цифровизации рутинных школьных задач.

**6**

**Описание алгоритма программы**

**Словесное описание работы программы**

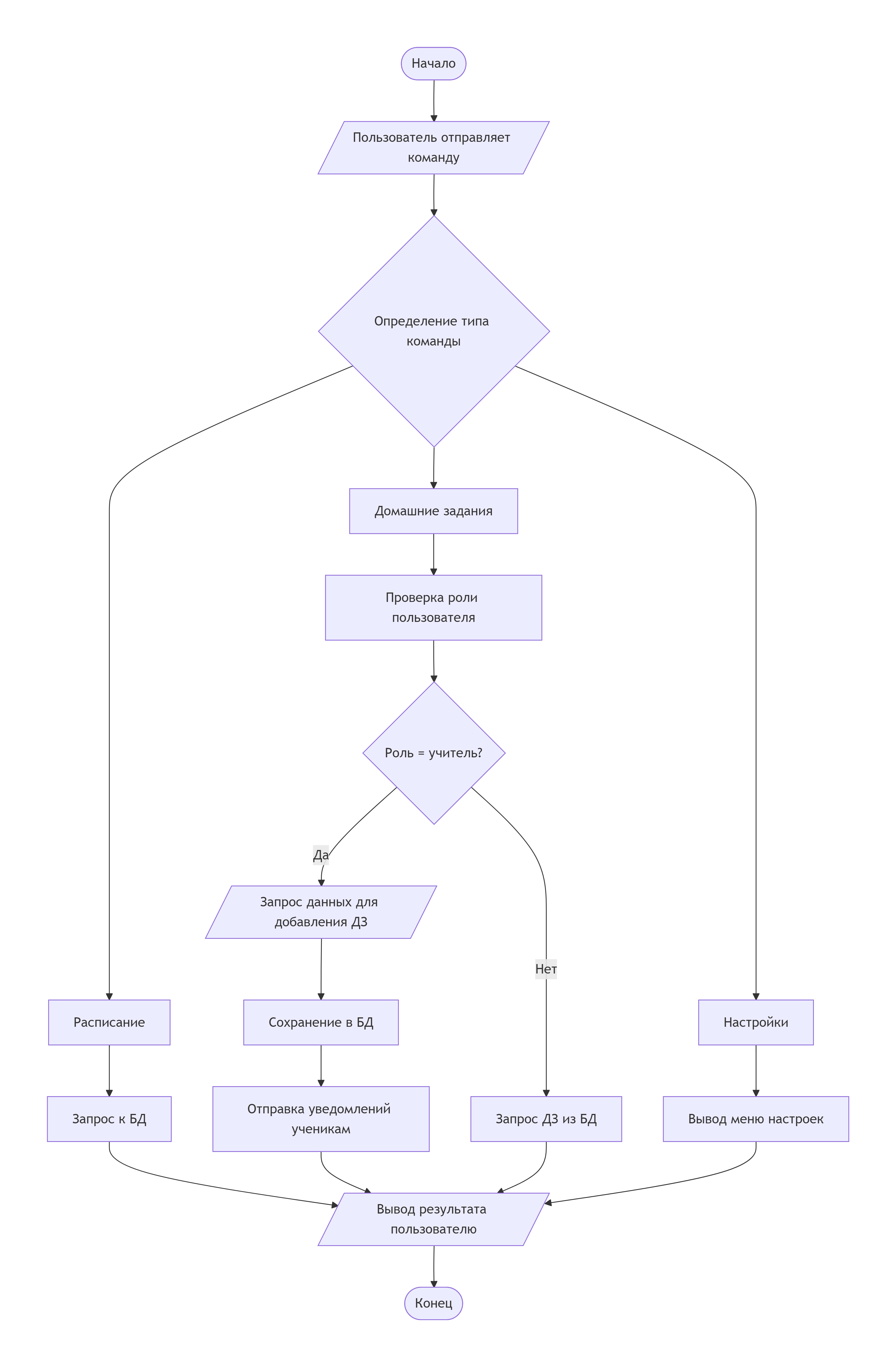
Пользователь взаимодействует с ботом через команды в Telegram. Бот обрабатывает запросы, обращается к базе данных для получения или сохранения информации и отправляет ответ пользователю. Основные сценарии:

1. Запрос расписания → поиск в БД → вывод.
2. Добавление домашнего задания учителем → сохранение → уведомление ученикам.
3. Напоминание о начале урока → автоматическая отправка.

**7**

**Алгоритм работы программы (блок-схема)**

Рис. 1. Алгоритм работы бота

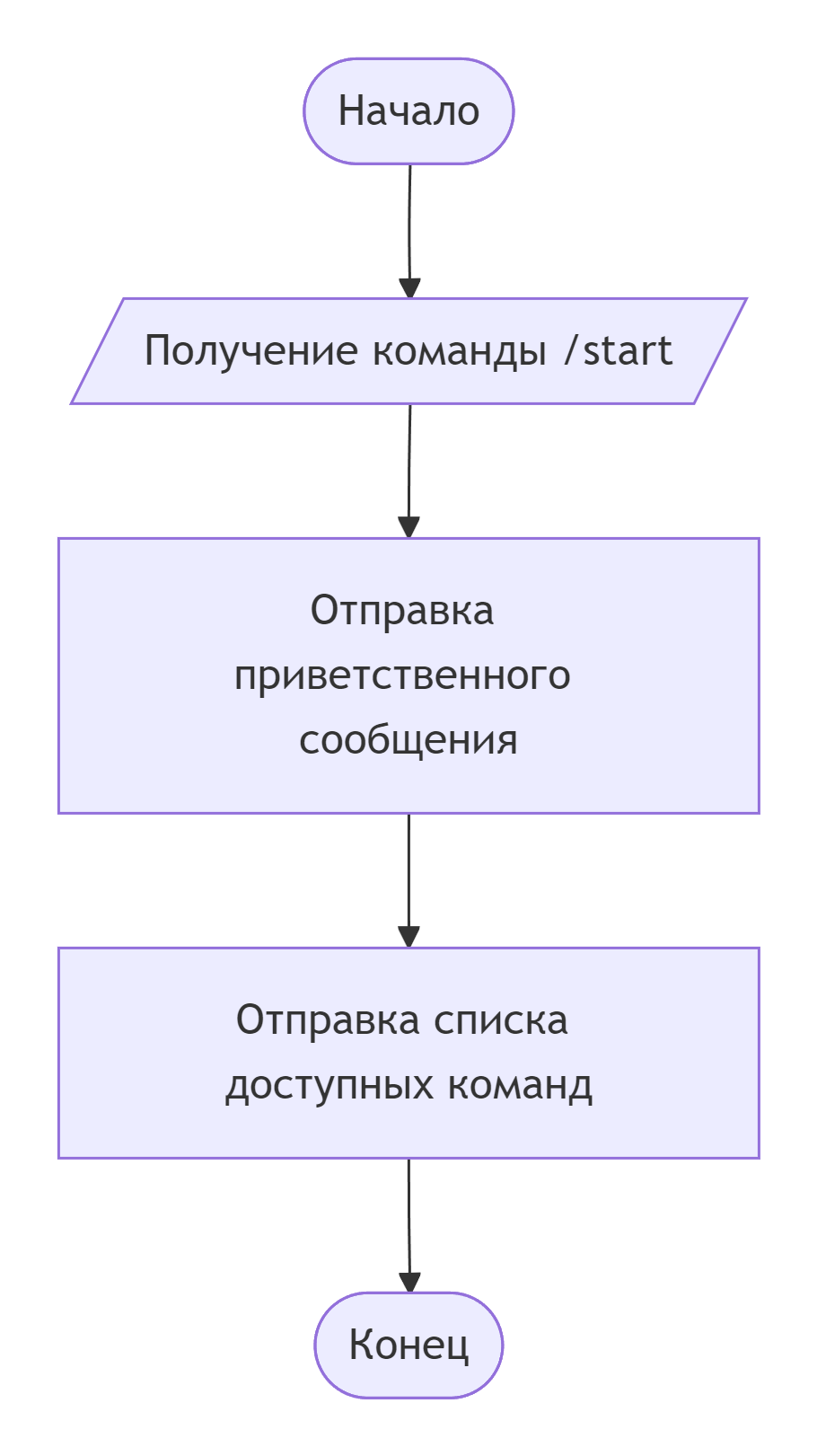


**8**

**Обработка основных команд (блок-схемы)**

**Команда /start**

Рис. 2. Обработка команды /start

****

**9**

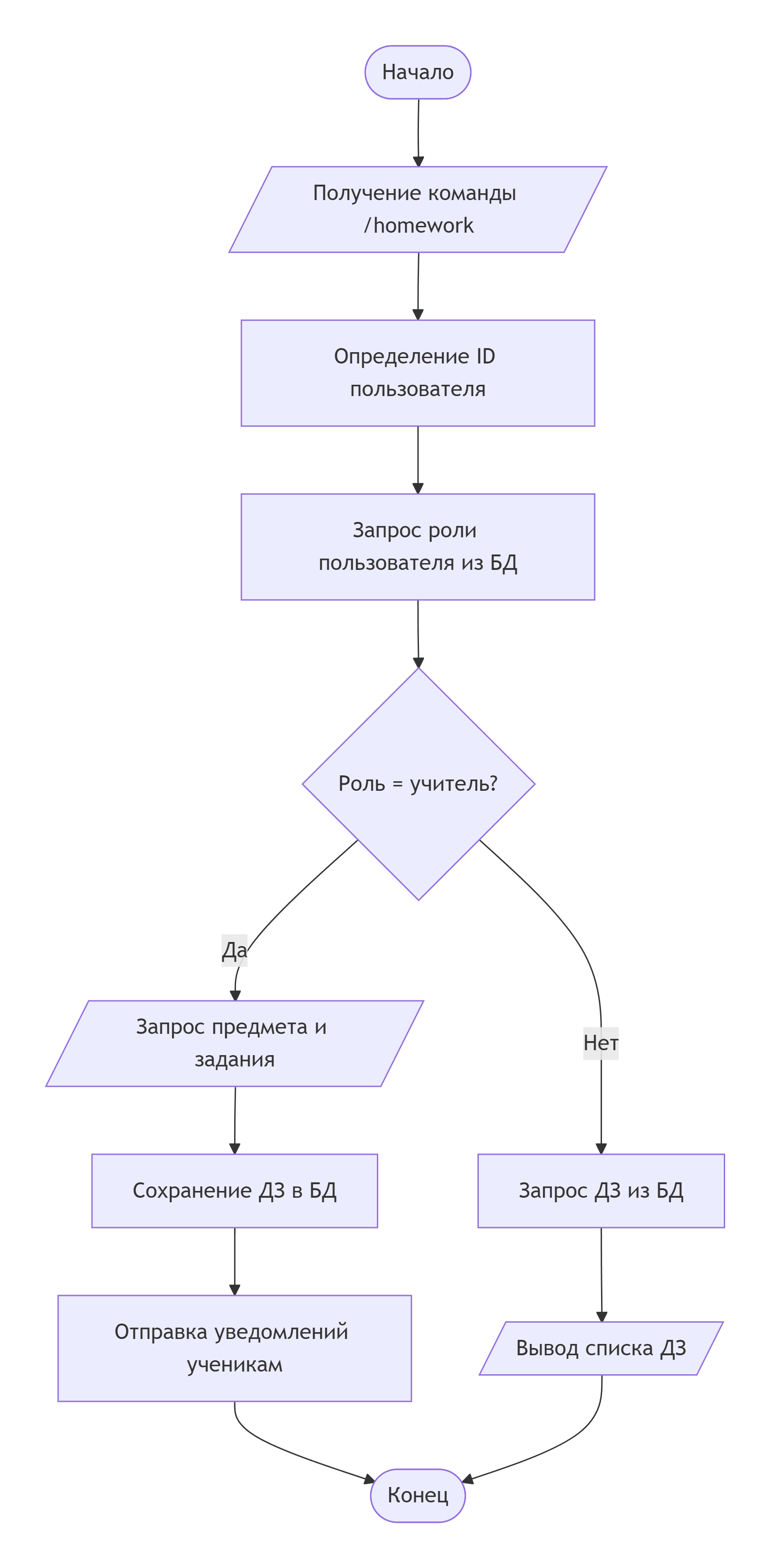
**Команда /today**

Рис. 3. Обработка команды /today

**10**

**Команда /homework**

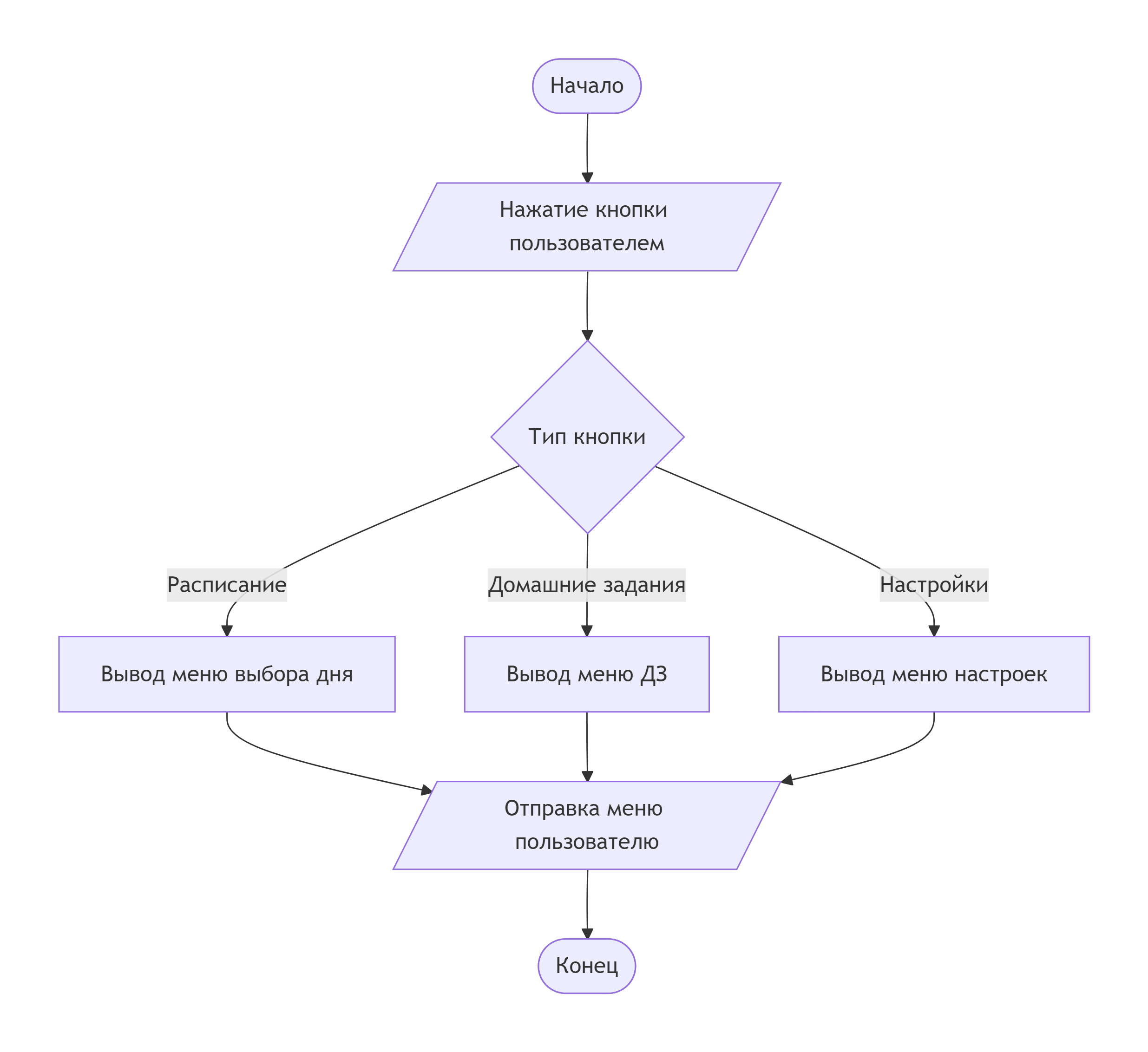
Рис. 4. Обработка команды /homework

****

**11**

**Обработка нажатия кнопок (интерактивного меню)**

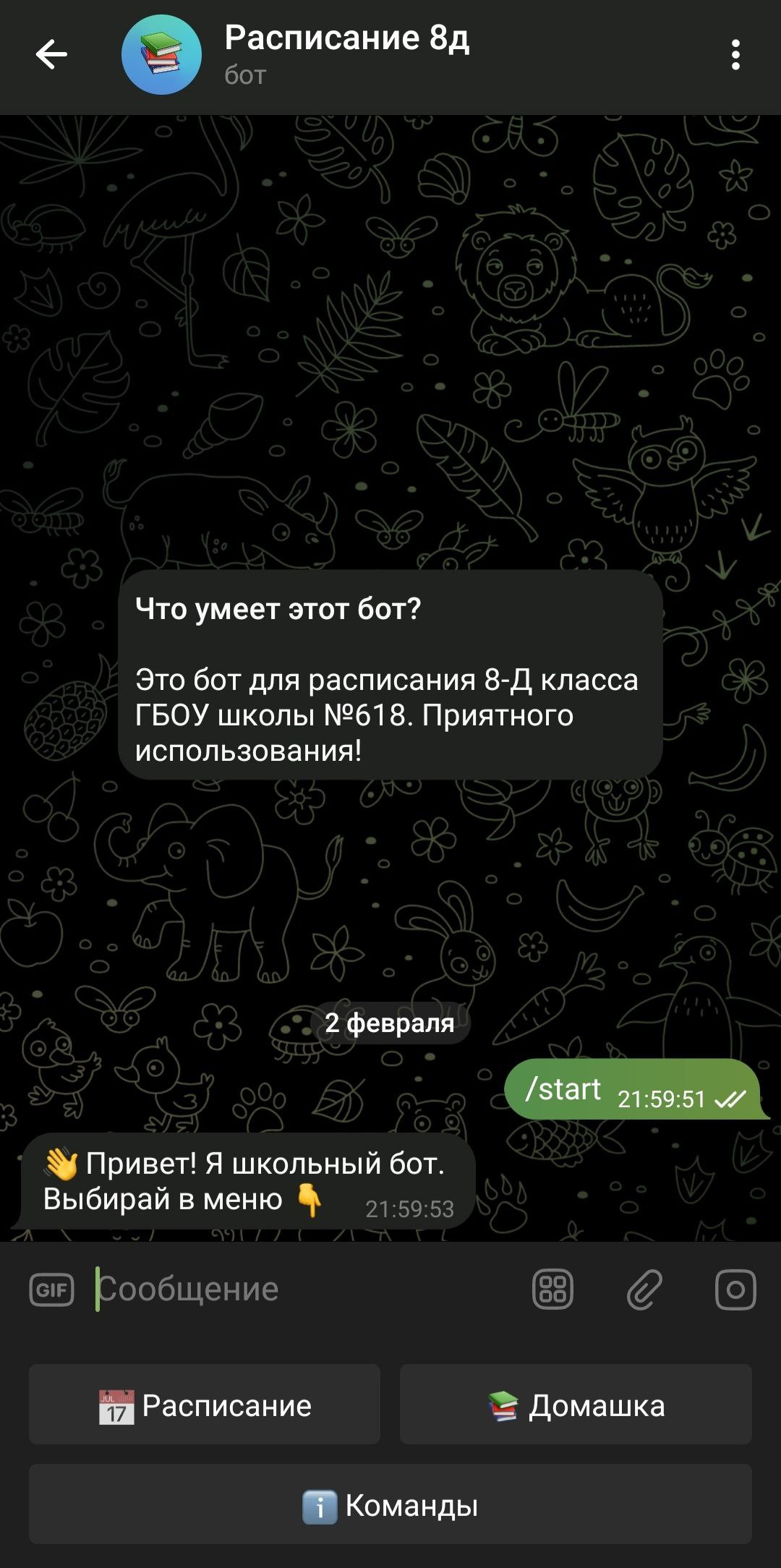
Рис. 5. Обработка нажатия кнопок

****

**12**

**Скриншоты работы бота**

Рис. 6. Начало работы с ботом (команда /start)

****

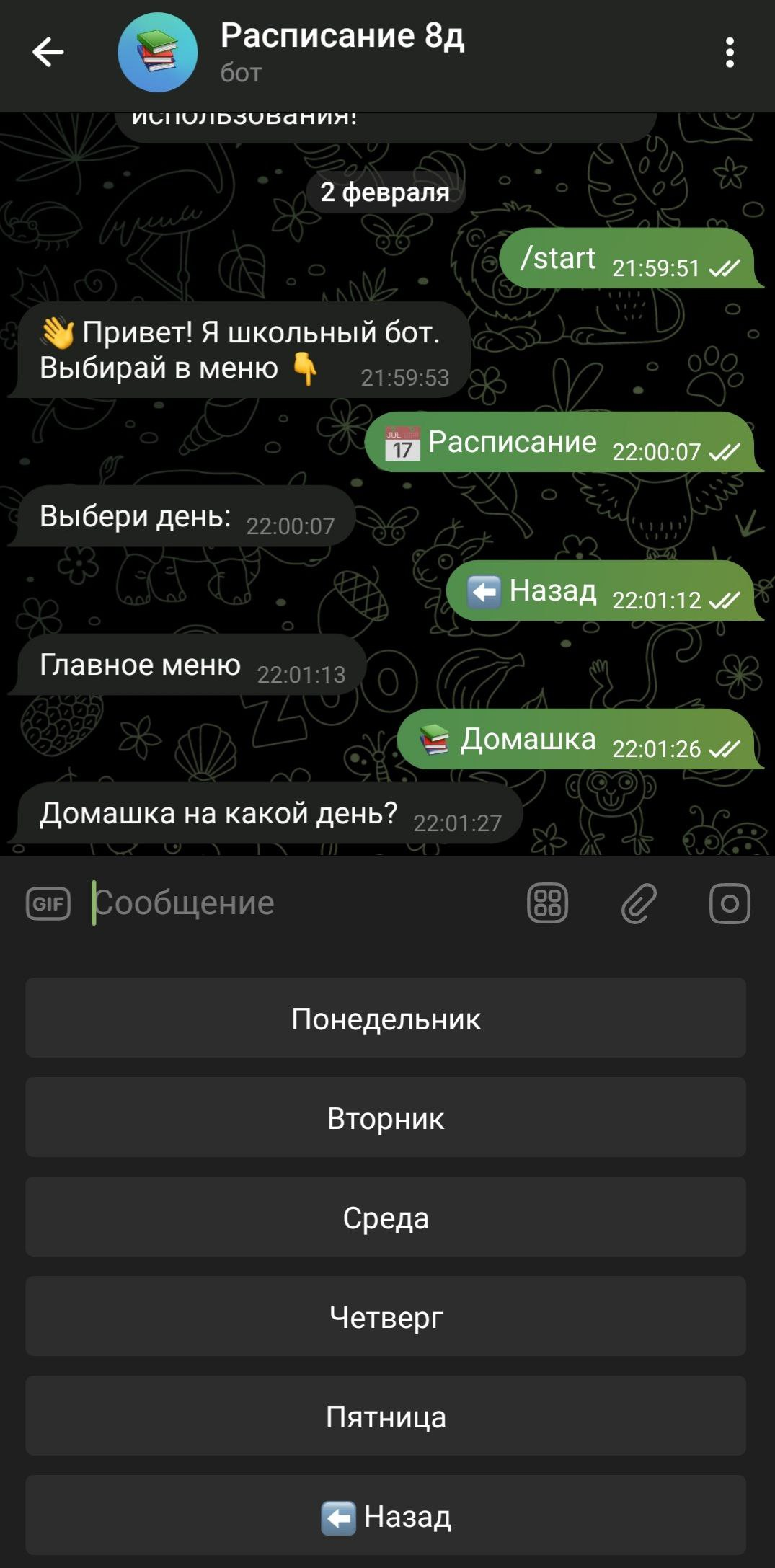
**13**

Рис. 7. Получение расписания на неделю

****

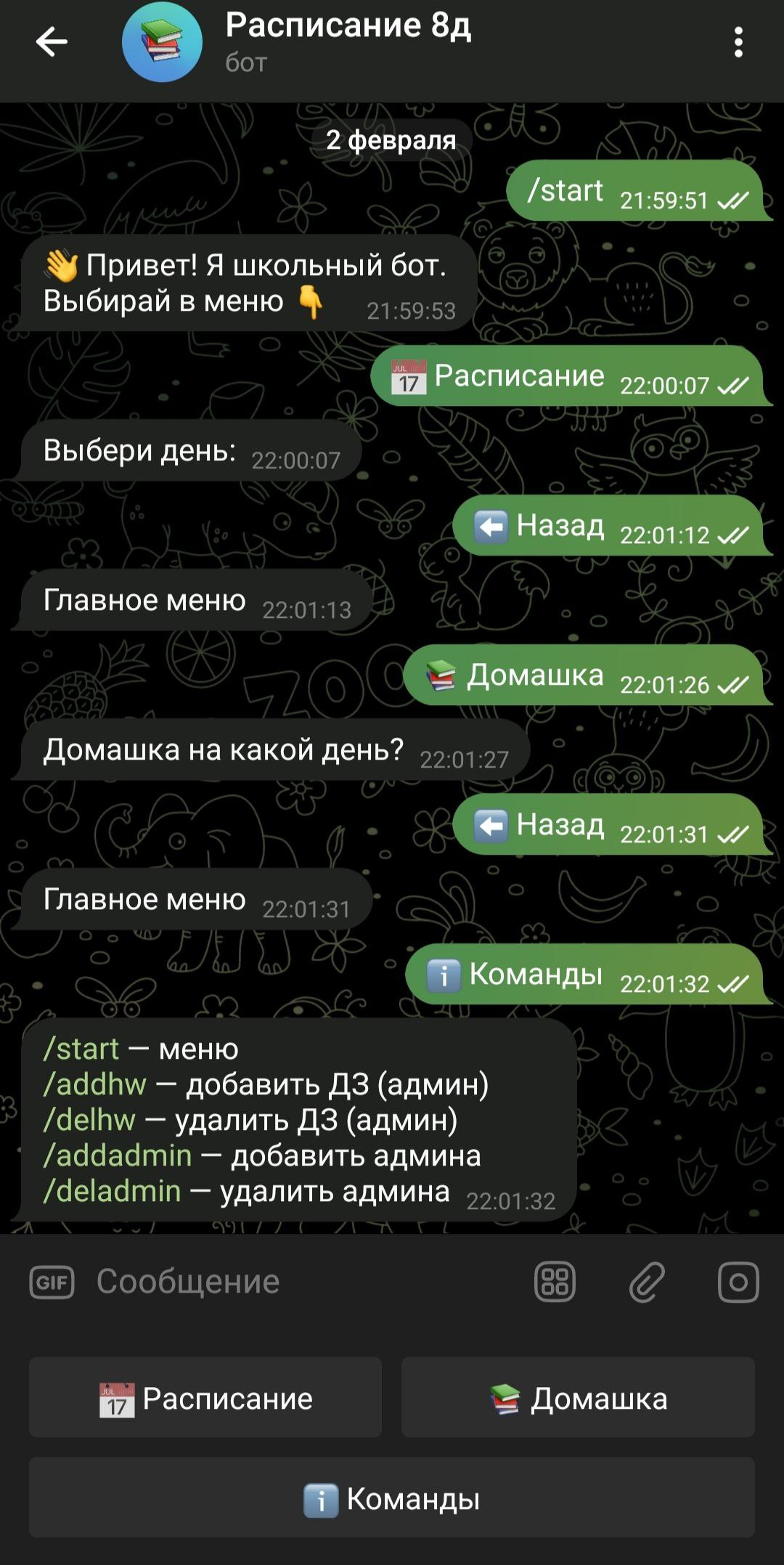
**14**

Рис. 8. Меню домашних заданий



**15**

Рис. 9. Список Команд

****

**16**

Выводы

Что получилось в итоге работы над проектом

Разработанный чат-бот успешно решает поставленные задачи: обеспечивает централизованное хранение расписания и домашних заданий, упрощает коммуникацию между учителями, учениками и родителями, а также автоматизирует рутинные процессы. Бот протестирован и готов к использованию в учебном процессе.

Чем ваш продукт отличается от аналогов

1. Интеграция с Telegram – не требует установки дополнительных приложений.
2. Простота использования – интуитивный интерфейс на основе команд и кнопок.
3. Автоматические уведомления – напоминания о начале уроков и новых заданиях.
4. Разграничение прав доступа – разные возможности для учеников, учителей и родителей.
5. Бесплатность – полное отсутствие платных функций для пользователей.
6. Кроссплатформенность – работает на любом устройстве с Telegram.

Что удалось реализовать при работе

1. Полный набор команд для работы с расписанием и домашними заданиями.
2. Базу данных SQLite для хранения информации.
3. Систему ролей (ученик, учитель, родитель).
4. Автоматические уведомления.
5. Интерактивное меню с кнопками.
6. Развертывание бота на хостинге.

**17**

Список используемой литературы

1. Официальная документация Telegram Bot API.  
   URL: <https://core.telegram.org/bots/api>
2. Документация библиотеки python-telegram-bot.  
   URL: <https://python-telegram-bot.org/>
3. Официальная документация по языку Python.  
   URL: <https://docs.python.org/3/>
4. Руководство по работе с SQLite.  
   URL: <https://www.sqlite.org/docs.html>
5. Статьи по разработке Telegram-ботов на Habr.  
   URL: <https://habr.com/ru/hub/telegram_bots/>
6. Учебные материалы по Git и GitHub.  
   URL: <https://docs.github.com/>
7. Документация по хостингу Heroku.  
   URL: <https://devcenter.heroku.com/>
8. Материалы по интеграции с Google Sheets API.  
   URL: <https://developers.google.com/sheets/api>
9. Статьи по проектированию баз данных.  
   URL: <https://www.sqlstyle.guide/>
10. Официальный сайт Visual Studio Code.  
    URL: <https://code.visualstudio.com/docs>
11. Платформа [draw.io](https://draw.io/) для создания диаграмм.  
    URL: <https://app.diagrams.net/>

Ссылка на репазиторий

https://github.com/o2547817-sketch/project

**18**