**ОТЧЁТ ПО РАБОТЕ В ПЕРИОД С 21.05.2024 ПО 27.05.2024**

*Группа 231-3210, Московский Политех, команда №4:*

1. *Самохвалов Вячеслав Дмитриевич*
2. *Балезина Мария Владимировна*
3. *Басенко Каролина Евгеньевна*
4. *Рожкова Елизавета Михайловна*

*Тема проекта: Telegram Bot для автоматизированной очереди "QUEUE-3210". Источник:* [*https://www.freecodecamp.org/news/how-to-create-a-telegram-bot-using-python/*](https://www.freecodecamp.org/news/how-to-create-a-telegram-bot-using-python/)*. Основа проекта:* [*https://habr.com/ru/articles/806471/*](https://habr.com/ru/articles/806471/)*.*

**ПЛАН РАБОТЫ НАД ПРОЕКТОМ:**

1. **Поставлена цель проекта**: изучить процесс и алгоритм создания бота в Telegram, изучить процесс написания API на языке программирования PHP. На основе полученных знаний реализовать бота в Telegram, который будет упрощать взаимодействие с автоматизированной очередью "QUEUE-3210".
2. **Выделены основные задачи проекта:**

1. Внесение корректировок в существующую БД (коэффициент важности: 0.2)

2. Написание API для получения информации о дисциплинах, студентах и очереди (коэффициент важности: 0.7)

3. Тестирование API на уязвимости (коэффициент важности: 0.3)

4. Добавление возможности прикрепления аккаунта в Telegram к учетной записи в веб-приложении (коэффициент важности: 0.5)

5. Создание бота в Telegram через Bot Father (коэффициент важности: 0.2)

6. Написание каркаса бота (коэффициент важности: 0.8)

7. Установление связи между ботом и API (коэффициент важности: 1)

8. Конфигурация бота в соответствии с API (коэффициент важности: 1)

9. Тестирование бота (коэффициент важности: 0.5)

\*10. Внедрение функционала админа в бот (коэффициент важности: 0)

\*11. Редактирование списка дисциплин из бота (коэффициент важности: 0)

*Условные обозначения: «\*» –* *необязательная задача, которая будет выполнена только при условии успешного выполнения всех предыдущих задач.*

1. **Распределены задачи проекта между участниками команды:**

Самохвалов Вячеслав Дмитриевич – задачи № 2, 7

Балезина Мария Владимировна – задачи № 3, 9

Басенко Каролина Евгеньевна – задачи № 1, 4, 8

Рожкова Елизавета Михайловна – задачи № 5, 6

1. **Определен календарный план выполнения задач:**

Задача №1 – 13.05.2024

Задача №2 – с 14.05.2024 по 15.05.2024

Задача №3 – 16.05.2024

Задача №4 – с 17.05.2024 по 18.05.2024

Задача №5 – 19.05.2024

Задача №6 – с 20.05.2024 по 22.05.2024

Задача №7 – с 23.05.2024 по 25.05.2024

Задача №8 – с 26.05.2024 по 28.05.2024

Задача №9 – 29.05.2024

1. **Работа над задачами проекта.** Ниже приведены краткие сведения о том, каким образом была выполнена та или иная задача.

**ВЫПОЛНЕНИЕ ЗАДАЧИ №6:**

Был разработан каркас бота. В процессе работы над ботом, было принято решение об удалении каждого предыдущего сообщения, при отправке ботом нового сообщения. Такая методика позволяется сохранять чат с ботом «чистым». Также было принято решение использовать только Inline-кнопки, то есть кнопки под самим сообщением. Благодаря этому нам не нужно обрабатывать никакие сообщения кроме команды «/start»/. Фрагмент кода обработчика данной команды представлен на рис. 1. Исходный код находится в ветке «bot» репозитория проекта: <https://github.com/qelio/queue-3210/tree/bot>.

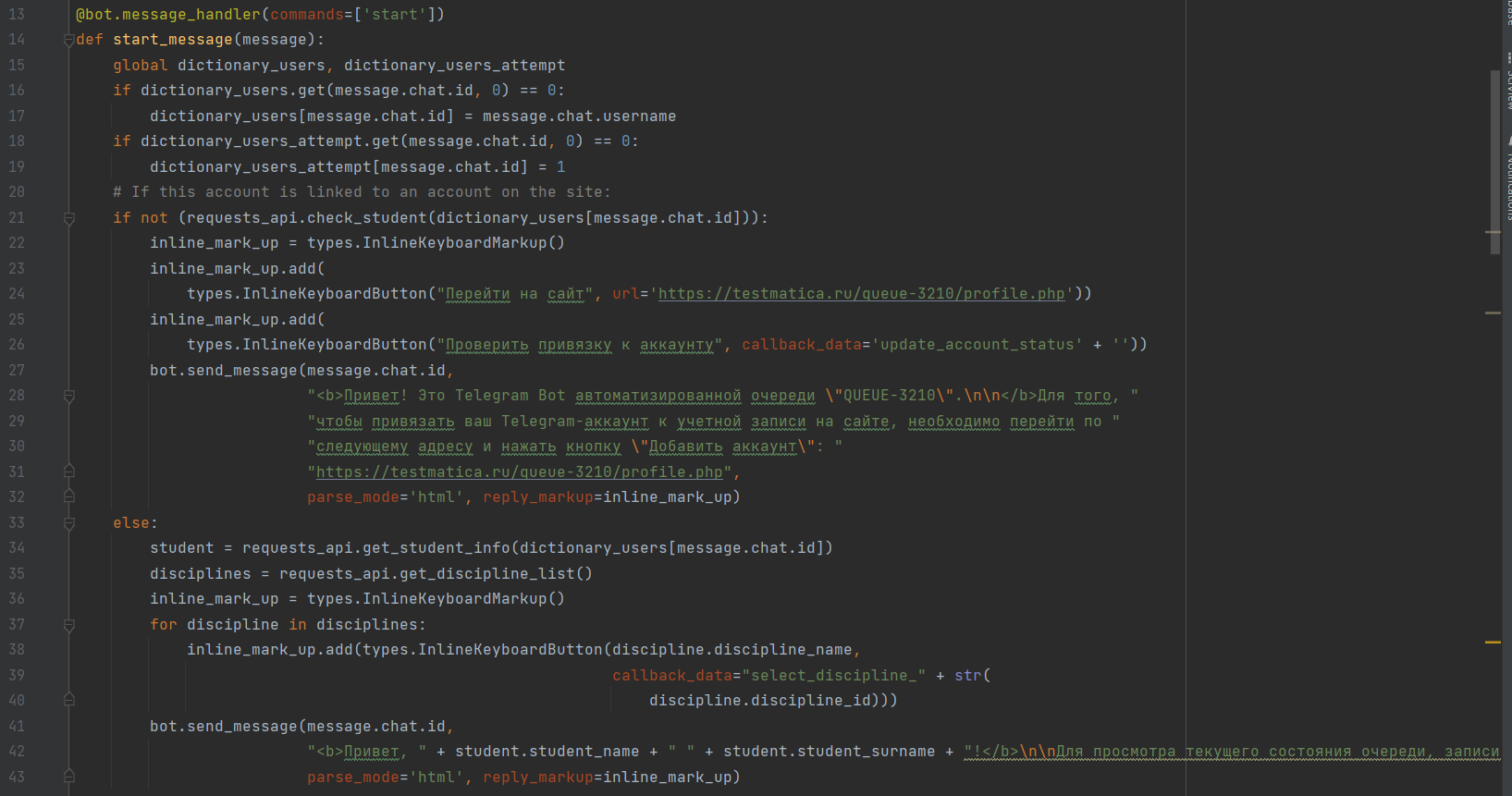


Рис. 1. Фрагмент кода обработчика команды «/start»

**ВЫПОЛНЕНИЕ ЗАДАЧИ №7:**

Далее в отдельном файле «requests\_api.py» были написаны функции для получения информации от ранее написанного API (см. ветку «api»). Также были созданы отдельные классы, такие как, «Student», «Discipline», «SharedQueue», предназначенные для хранения информации о студенте, дисциплине и состоянии очереди, соответственно. На рис. 2 и рис. 3 представлены фрагменты кода для проверки информации о наличии пользователя с заданным TG-логином в базе данных и получении информации о пользователе по его TG-логину. Исходный код находится в ветке «bot» репозитория проекта: <https://github.com/qelio/queue-3210/tree/bot>.

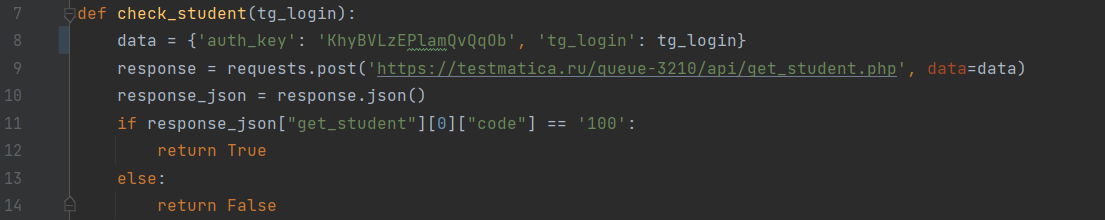


Рис. 2. Фрагмент кода для проверки информации о наличии пользователя с заданным TG-логином в базе данных

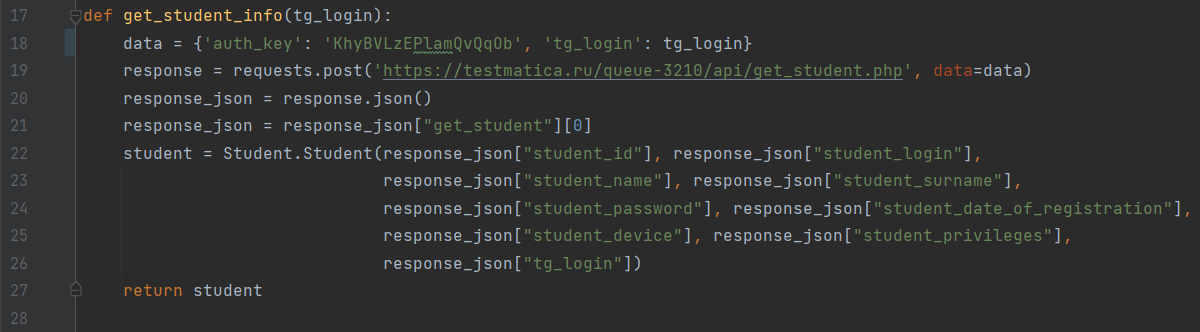


Рис. 3. Фрагмент кода для получения информации о пользователе по его TG-логину

**ВЫПОЛНЕНИЕ ЗАДАЧИ №8:**

После реализации функций для взаимодействия с API, был реализован основной функционал бота. То есть проверка TG-логина на привязку к аккаунту в автоматизированной очереди (как в начале диалога, так и при выполнении любой команды, на случай отвязки аккаунта студента от аккаунта в Telegram в процессе взаимодействия с ботом, см. рис. 4), получение и вывод списка доступных дисциплин в виде кнопок (см. рис. 5), получение состояния очереди для конкретной дисциплины, запись в очередь по конкретной дисциплине и удаление из нее (см. рис. 6).

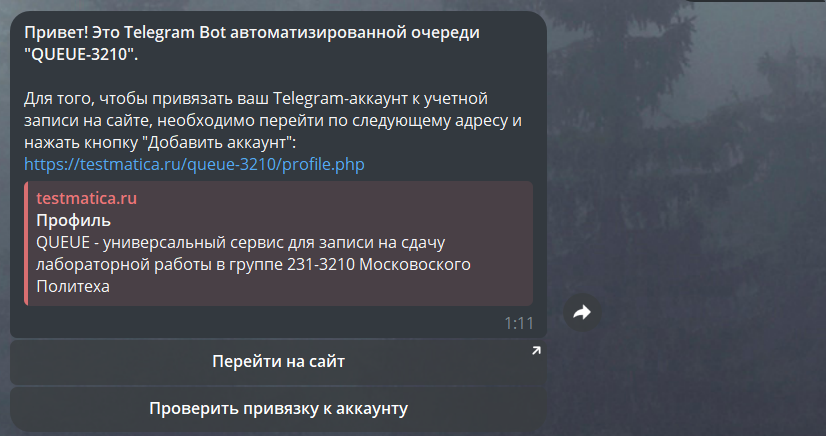


Рис. 4. Привязка аккаунта в боте

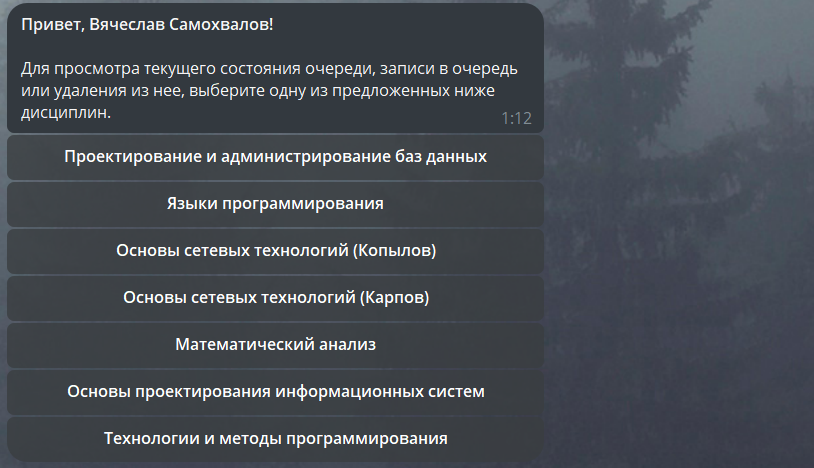


Рис. 5. Вывод доступных дисциплин в виде кнопок

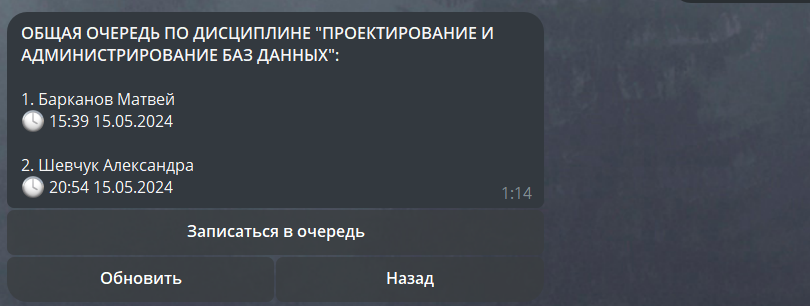


Рис. 6. Получение состояния очереди для конкретной дисциплины

Демонстрация функционала бота в видео-формате представлена по ссылке: <https://cloud.mail.ru/public/3fr4/hX8G83uKr>. Скрипт для работы бота был запущен на домашнем Linux сервере (Debian 12).

**ВЫПОЛНЕНИЕ ЗАДАЧИ №9:**

Было проведено тестирование бота и выявлены следующие проблемы:

1. Возможен некорректный ввод имени пользователя Telegram на сайте, то есть ввод имени пользователя с символом «@». Решение: внедрить на сайт функционал проверки введенной пользователем строки на наличие данного символа и удаление его в случае положительного результата проверки.
2. Возможно падение бота по истечении времени ожидания ответа от сервера. Решение: необходимо внедрить функцию перезапуска Python-скрипта при его завершении из-за ошибки. На данный момент ведутся работы по устранению данной ошибки.