*Аннотация. В статье представлен разработанный проект системы автоматизированного тестирования на базе Python + Django с пользовательским интерфейсом Telegram-бот. Система может использоваться как преподавателями для создания, редактирования и импорта тестов, а также для контроля и оценивания результатов прохождения тестирования, так и студентами для прохождения тестов и отслеживания собственного прогресса.*

*Ключевые слова: тест, автоматизированная система тестирования, база данных, Python, Django, Telegram-бот.*

Современная система образования предъявляет высокие требования к контролю и оцениванию результатов освоения образовательных программ учащимися. Среднее профессиональное образование, в особенности в сфере информационных технологий, не должно отставать от современного научно-технического прогресса, основными тенденциями которого являются использование облачных хранилищ, автоматизированная обработка данных, интегрированные программные продукты, использующие целый набор программных сред и оболочек.

В связи с этим особое внимание следует уделить автоматизированным системам тестирования с возможностью многопользовательского доступа и облачным хранением.

*Целью разработки* данной системы является автоматизация процесса разработки и редактирования тестов преподавателями, а также их прохождения студентами и оценивания.

В соответствии с целью были поставлены следующие *задачи*:

- разработать удобный пользовательский интерфейс для преподавателей для создания вопросов и ответов на них, а также добавления/импорта уже существующих тестов из файлов электронных таблиц;

- создать возможность мониторинга результатов прохождения теста студентами;

- реализовать практичный и удобный интерфейс для прохождения тестирования.

В качестве платформы для прохождения тестирования был выбран мессенджер Telegram, основными плюсами которого являются:

- наличие ботов;

- удобное и понятное API для взаимодействия пользователя с ботом [4, с. 68];

- отсутствие рамок в процессе реализации.

Однако для составления тестов и доступа к базе данных вопросов для тестирования Telegram является не очень удобным инструментом, кроме того, могут возникнуть сложности отображения дополнительной информации.

Поэтому в качестве основного инструмента для преподавателей была выбрана админ-панель на Django. Основные ее плюсы:

- полный контроль над базой данных, включая удаление/редактирование записей;

- возможность простой интеграции, связь как и с кодом бота, так и с базой данных.

**Описание продукта.**

Готовый проект представляет собой aдмин-панель, разработанную на Python+ Django [1, c. 31-32; 2, с. 24], с доступом к базе данных PostgreSQL [3, с. 94] и бот в мессенджере Telegram.

Внутри бота пользователь-студент имеет возможность пройти регистрацию, указав свое имя и фамилию, а также указав группу, ссылка на которую изначально находится в QR-коде, отсканированным учеником. После регистрации ему остается ждать начала тестирования, управление которым осуществляет преподаватель.

Преподаватель имеет возможность начать тестирование по заранее выгруженным через админ-панель вопросам, а также напрямую через бота, выбрав группу и категории вопросов, необходимых для тестирования.

После того, как преподаватель нажал на кнопку «Начать тестирование», всем студентам выбранной группы отправляется уведомление с предложением приступить к тестированию. После прохождения теста система подсчитывает количество правильных ответов студента и отправляет отчет преподавателю для дальнейшего оценивания.

**Функционал продукта.**

Для реализации данного проекта была разработана база данных на PostgreSQL. На рисунке 1 представлена модель разработанной базы данных.

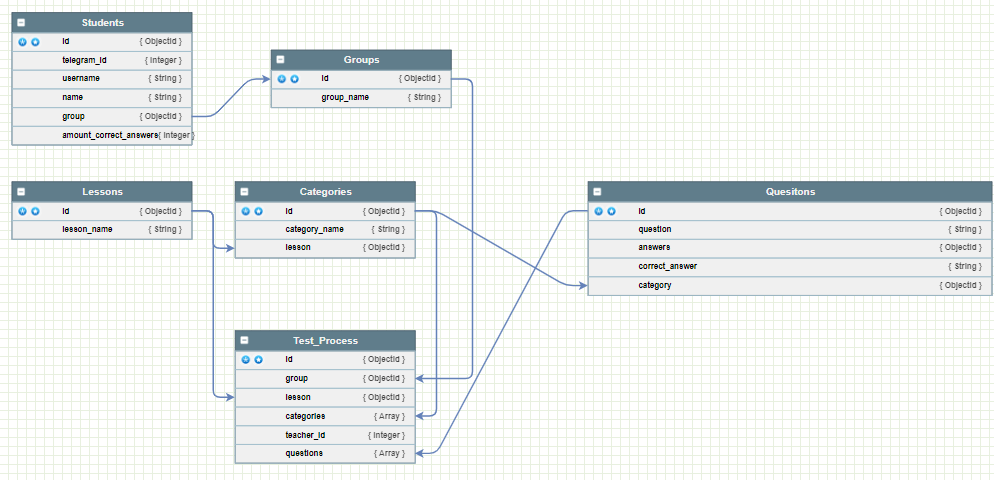


Рис. 1 Модель БД

Таблица студентов (Students) содержит в себе следующие поля:

* telegram\_id –уникальный идентификатор пользователя внутри Telegram, необходимый для взаимодействия со студентом, отправки сообщений, рассылки;
* username –имя пользователя студента, необходимое для связи с учеником путем отправки сообщений;
* name – имя студента, используемое при отправке сообщений, содержащих обращение, и для отправки результатов преподавателю;
* group – группа студента, необходимая для процесса тестирования, чтобы студент группы A не начал проходить тестирование для студентов группы B;
* amount\_correct\_answers – счетчик правильных ответов, актуальный только на время тестирования.

Таблица Групп (Groups) содержит поле group\_name - название группы

Таблица Дисциплин (Lessons) содержит поле lesson\_name - название дисциплины.

Таблица Категорий Вопросов (Categories) содержит следующие поля:

* category\_name –название категории, отображаемое преподавателю при выборе параметров тестирования;
* lesson – внешний ключ, служащий для связи каждой категории с определенной дисциплиной.

Таблица Вопросов (Questions) содержит следующие поля:

* question –формулировка вопроса (отображается студенту при процессе тестирования);
* answers – предложенные ответы в виде списка;
* correct\_answer – правильный ответ, который сравнивается с выбранным ответом студента на вопрос;
* Category – внешний ключ, связывающий каждый вопрос с определенной категорией.

Таблица процесса тестирования (Test\_Process) необходима для контролирования и более удобного тестирования студентов и содержит следующие поля:

* Group – группа, для которой проводится тестирование;
* Lesson – дисциплина, по которой проводится тестирование;
* Categories – список категорий, выбранных для тестирования по дисциплине;
* Teacher\_id – идентификатор пользователя Telegram, необходимый для отправления преподавателю итогов тестирования внутри мессенджера;
* Questions – список внешних ключей вопросов, связанных с категориями и необходимых для отображения вопросов и ответов на них.

Инструментом для взаимодействия и контроля базы данных была выбрана админ-панель на Django, внешний вид которой представлен на рисунке 2.

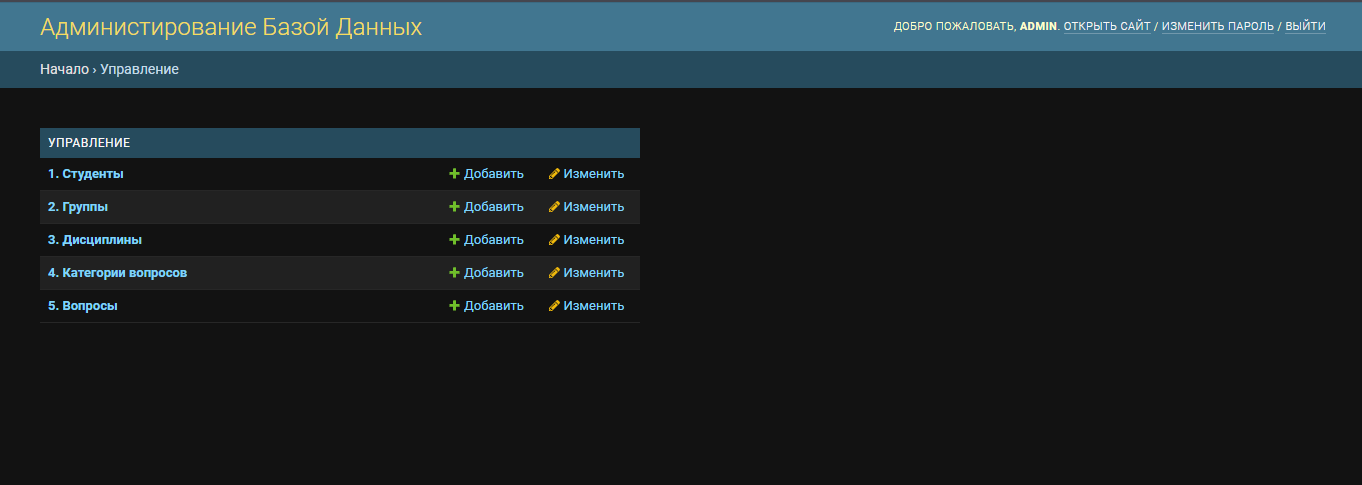


Рис. 2. Внешний вид админ-панели

Панель позволяет взаимодействовать с вышеописанными таблицами.

Выбрав необходимую таблицу, можно просмотреть уже существующие записи (пример см. на рис. 3).

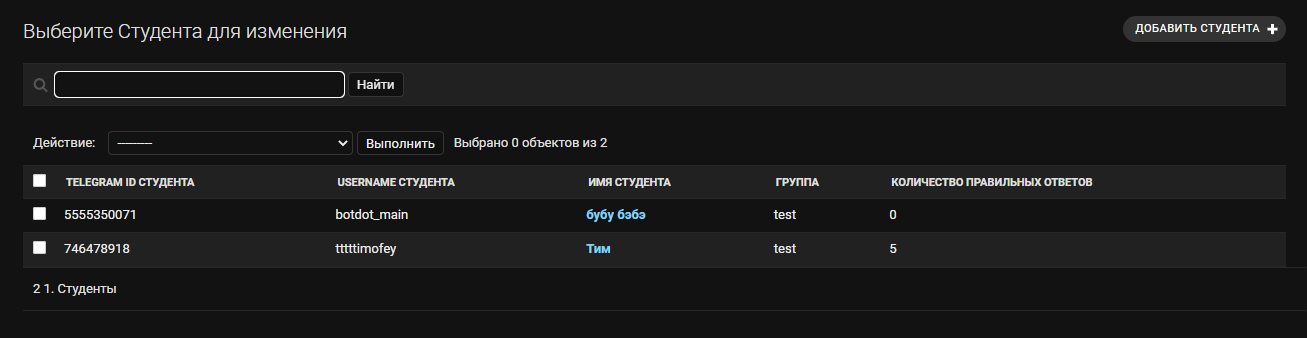


Рис. 3. Просмотр существующий записей

Кроме того, есть возможность создать вручную новую запись, например, внести в систему данные нового студента (рис. 4)

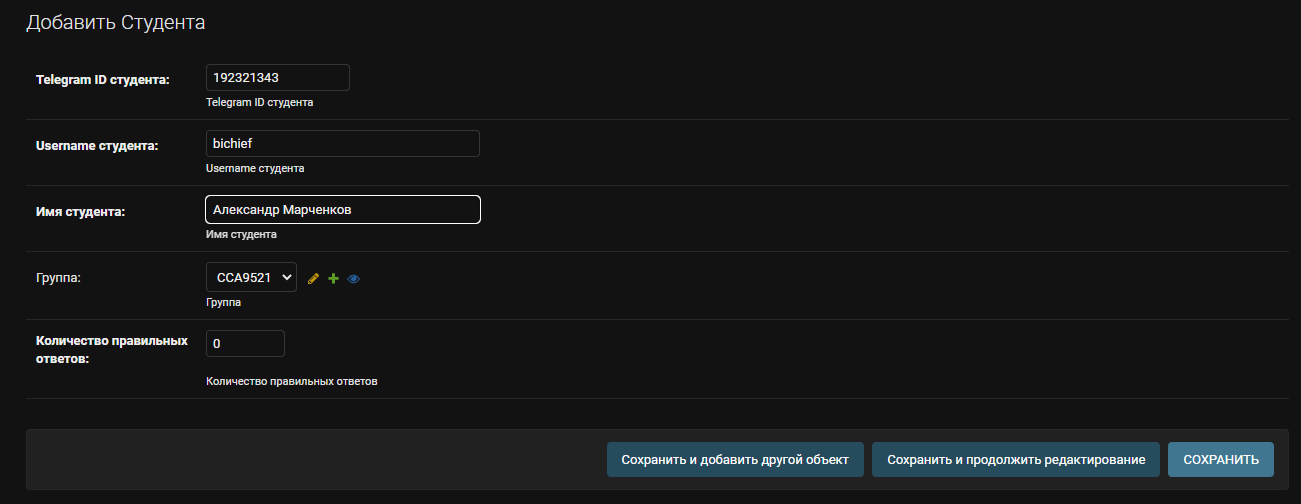


Рис. 4. Внесение данных нового студента

Особенностью данной панели является автоматизированное добавление вопросов и ответов к ним. Была разработана система, позволяющая преподавателю, заполнив таблицу Excel по определенному шаблону, выгрузить ее на сервер, где программа добавит новые записи, распределив столбцы электронной таблицы по соответствующим полям (рис. 5).

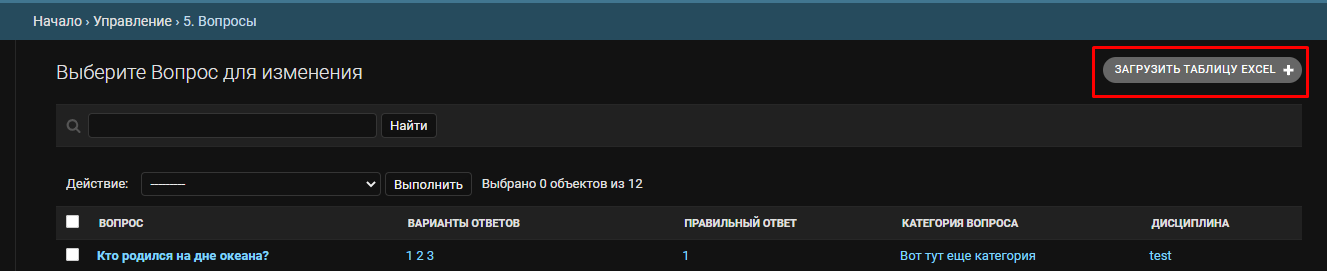


Рис. 5. Возможность импорта таблицы Excel

Дополнительной особенностью является процесс генерации QR-кода для определенной группы двумя способами.

Первый способ – при помощи админ-панели. В случае, если преподаватель создает новую запись в таблице Группы, у него есть возможность сохранить и сгенерировать QR-код, который в дальнейшем будет доступен группе студентов, проходящих тестирование. (рис. 6)

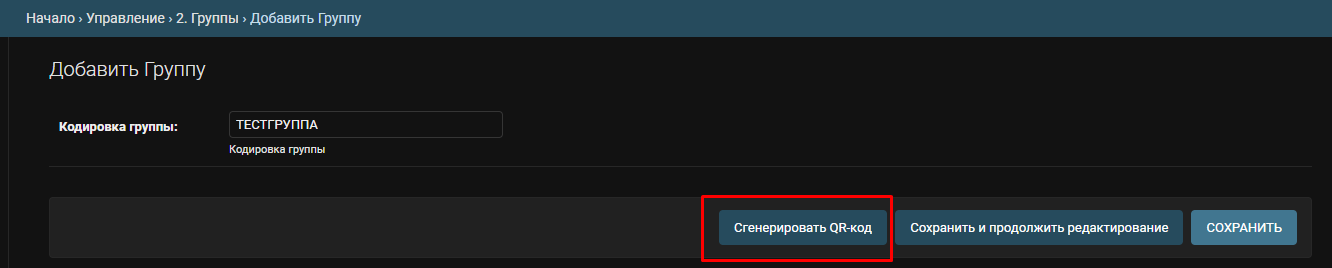


Рис. 6. Генерация QR-кода в админ-панели

Второй способ – при помощи бота в Telegram. После отправки команды ‘/qr пройдет проверка, является ли пользователем преподавателем по telegram\_id, записанному в виде констант внутри кода. После этого у преподавателя бот запрашивает информацию о названии группы и в ответ отправляет сгенерированный QR-код группы. (рис. 7)

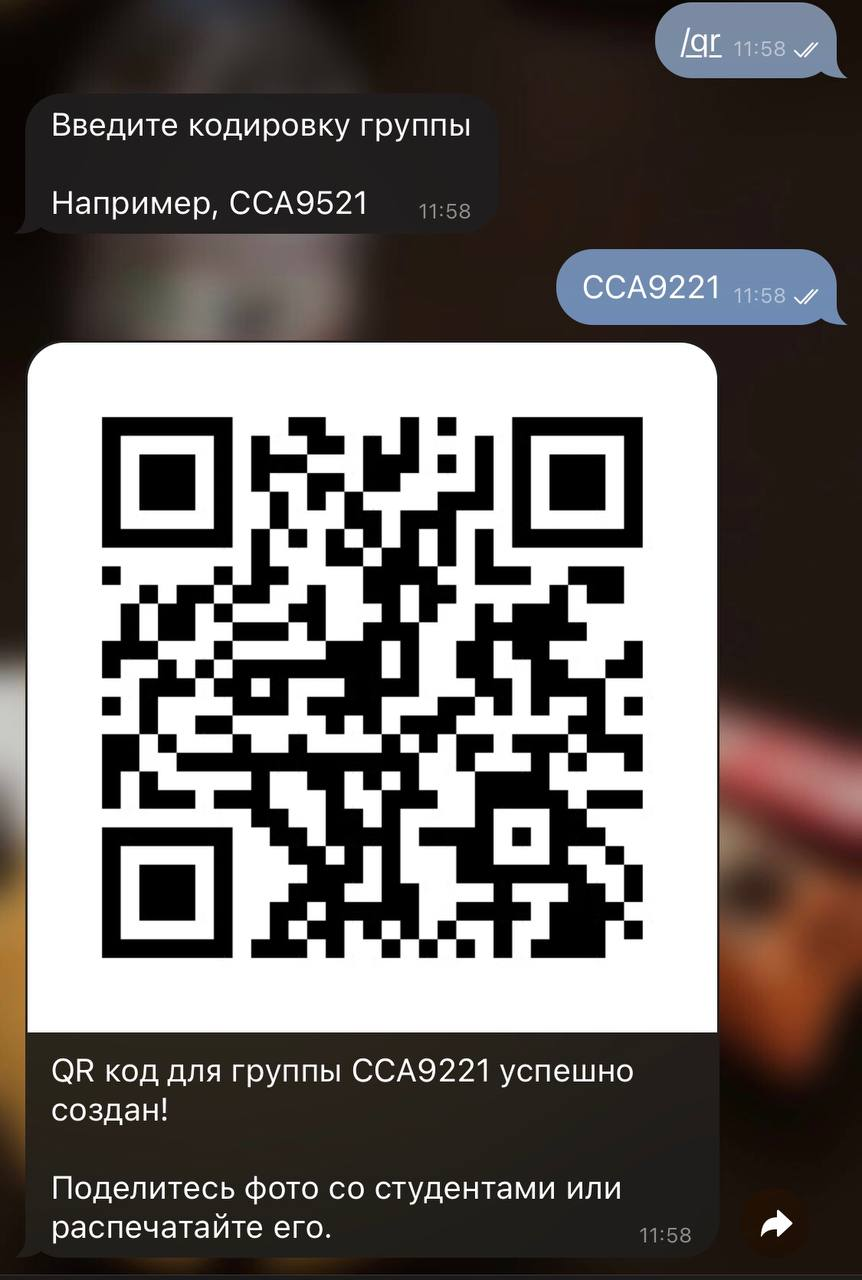


Рис. 7. Генерация QR-кода в боте

Главная функция бота Telegram заключается в регистрации внутри системы. Изначально пользователь сканирует QR-код, в котором находится код группы, в которой он обучается (рис. 8).

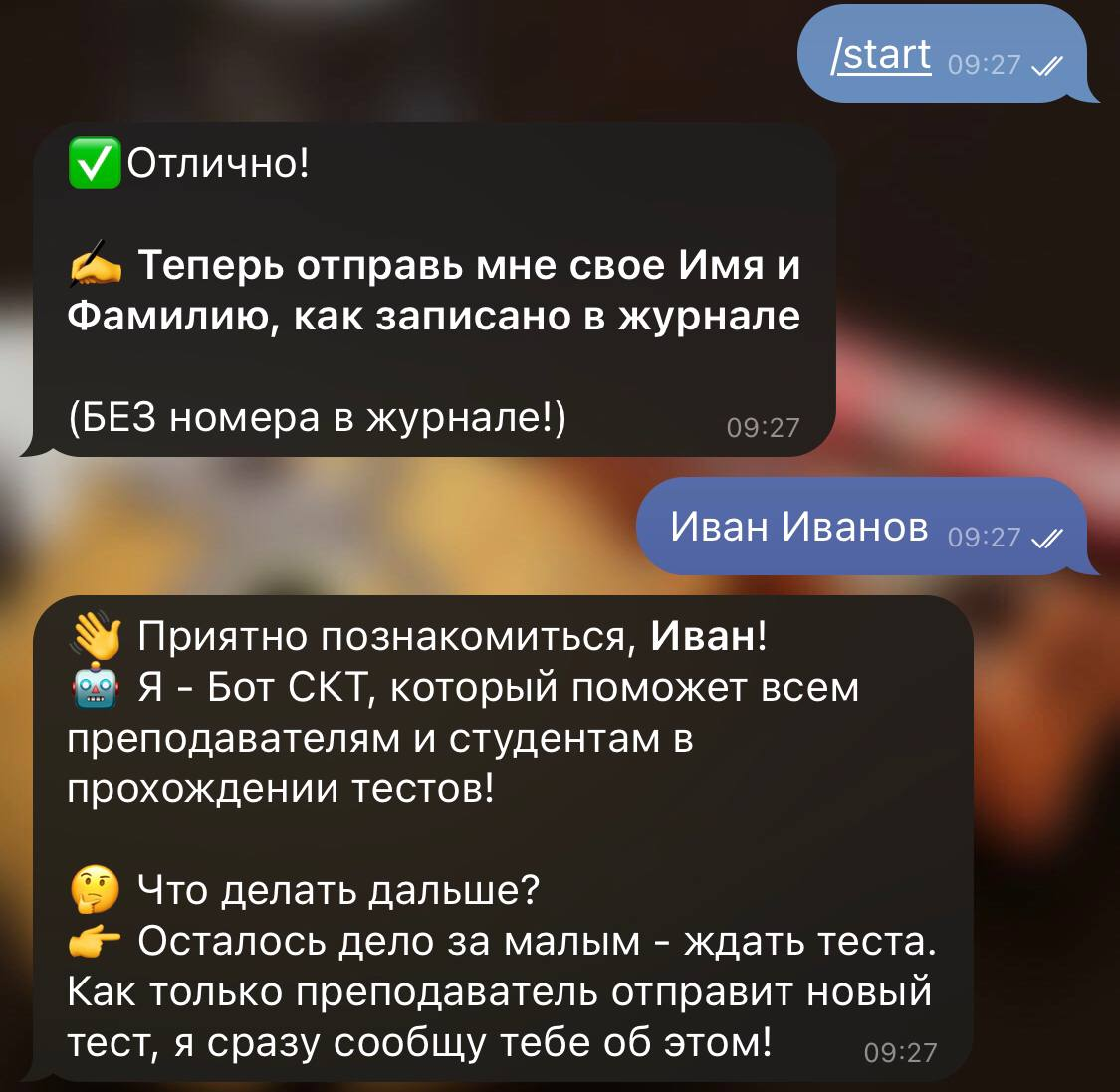


Рис. 8. Пример регистрации студента

В случае, если другой пользователь захочет получить доступ к боту, не сканируя QR-код, то бот вернет ему ответ о том, что бот доступен только студентам СКТ. (рис. 9)

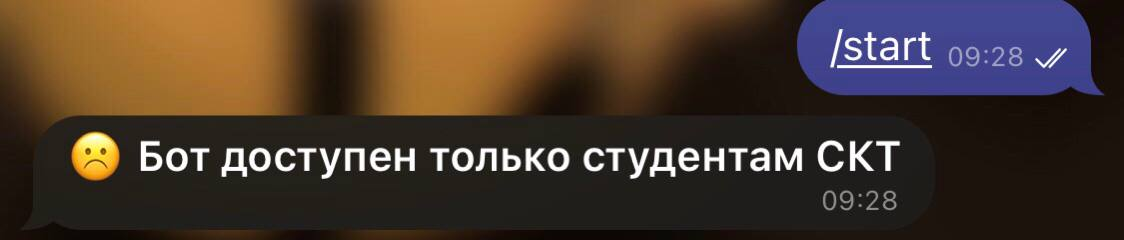


Рис. 9. Проверка на сканирование QR-кода

Если студент отсканирует QR-код с кодировкой несуществующей группы, которой нет в таблице Групп в базе данных, то ему бот вернет ответ о том, что кодировка некорректна. (рис. 10)

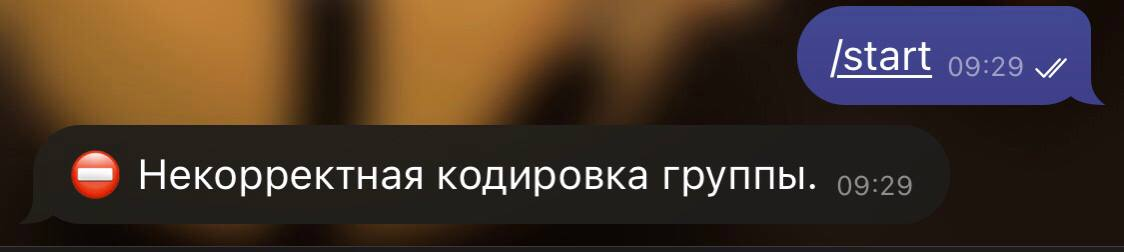


Рис. 10. Проверка на правильность QR-кода

После регистрации студент получает возможность пройти тестирование.

Преподаватель, которому был выдан доступ к команде */test* отправляет ее и получает следующий результат (рис. 11)

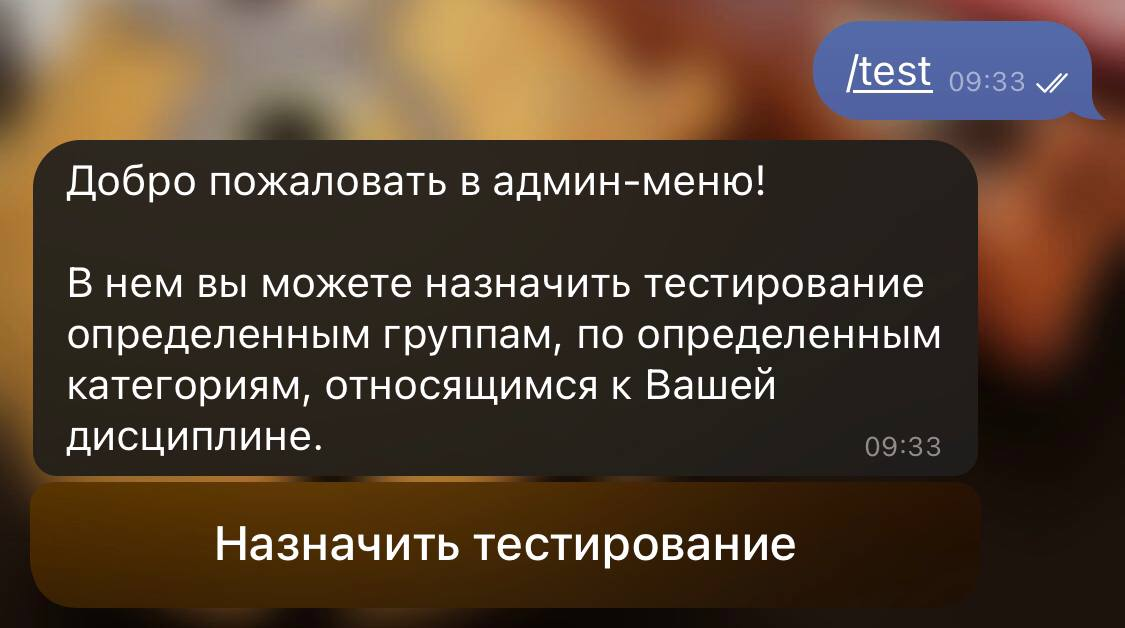


Рис. 11. Админ-меню в боте для преподавателей

После нажатия на кнопку «Назначить тестирование» преподавателю отображается список групп, по которым возможно начать тестирование, дисциплин, а также категории вопросов. (рис. 12)

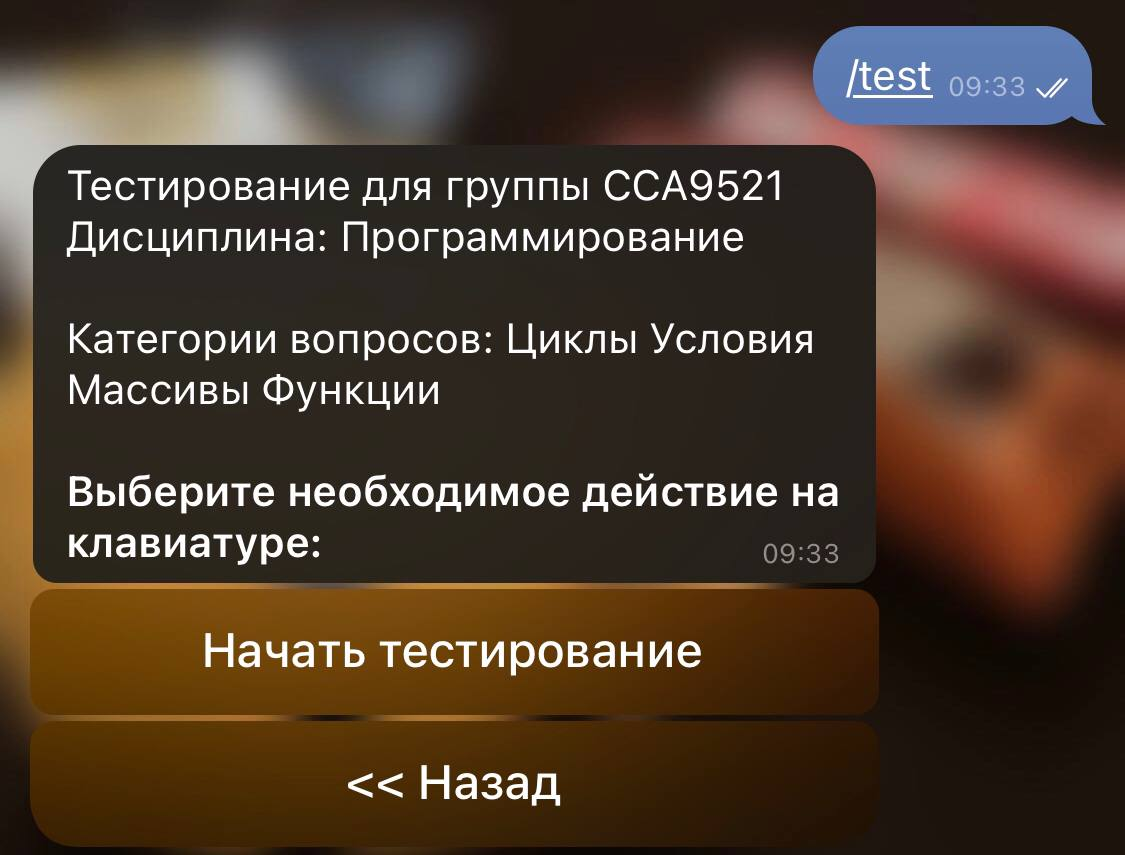


Рис. 12. Финальное сообщение перед тестированием

После нажатия кнопки «Начать тестирование» начинается рассылка теста и студенты получают следующее сообщение. (рис. 13)

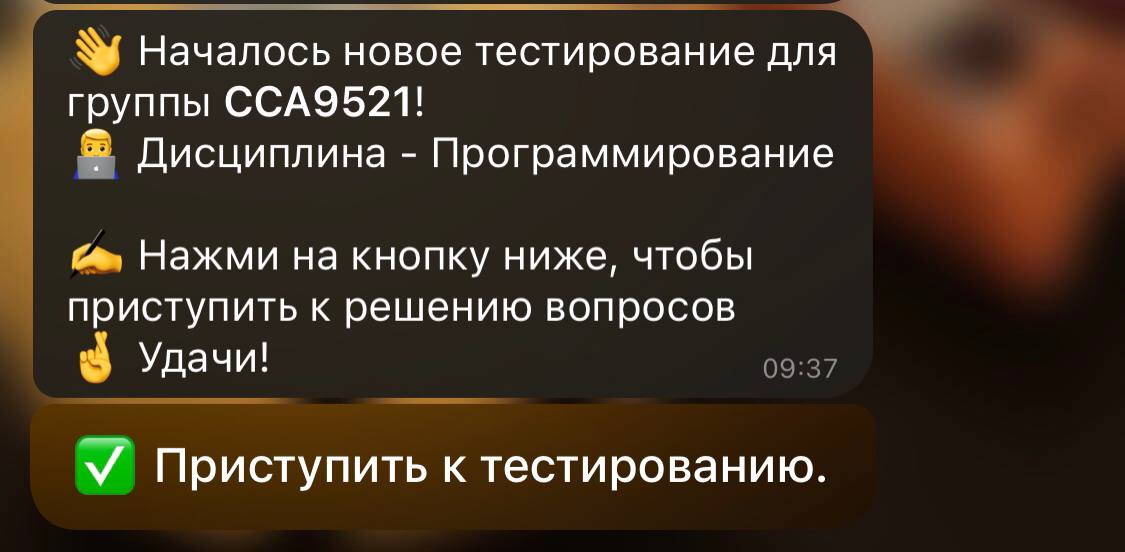


Рис. 13. Сообщение о начале тестирования

Студент может выполнять тест с помощью смартфона и после запуска теста получает доступ к его вопросам. (рис. 14)

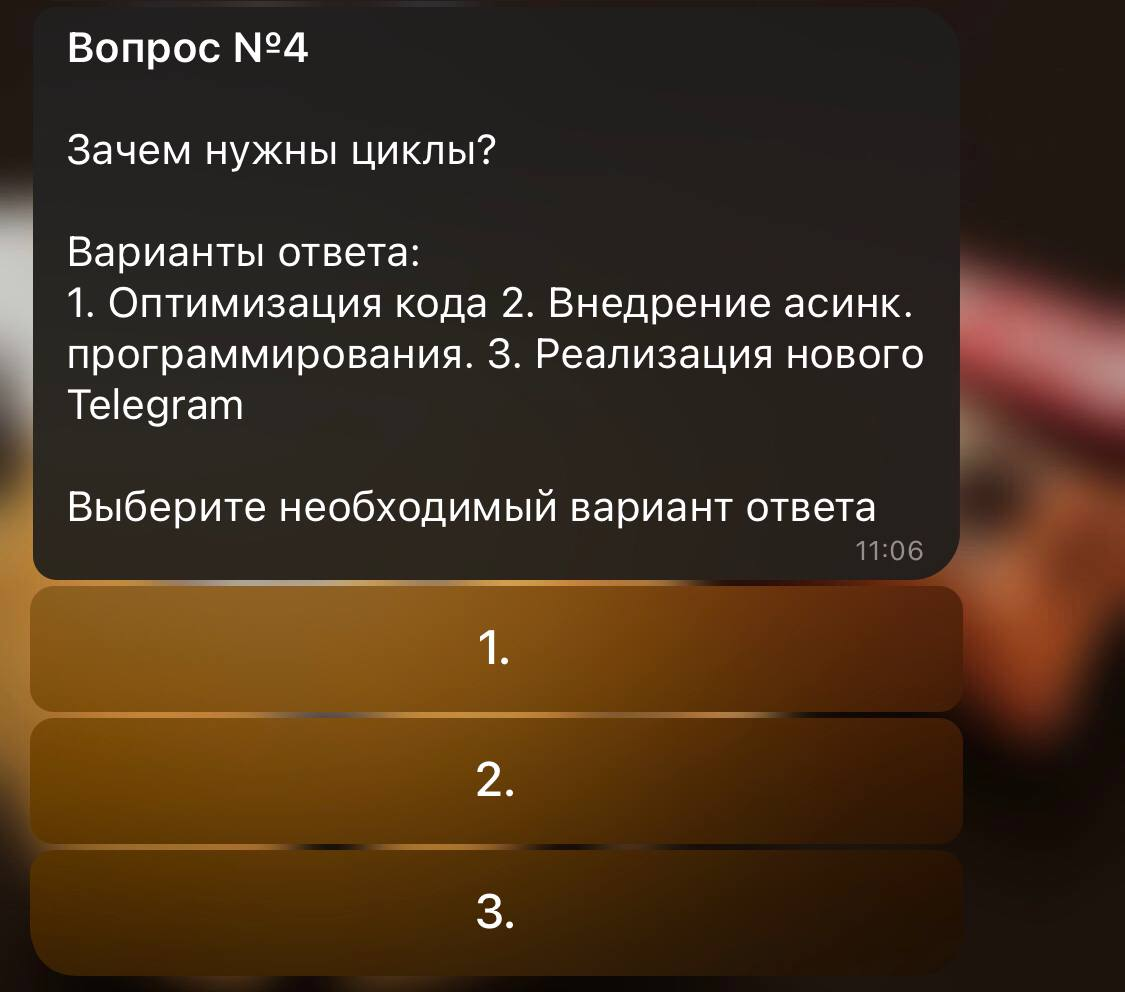


Рис. 14. Пример вопроса тестирования

После завершения тестирования студент получает сообщение о результатах, которые также отправляются и преподавателю (рис. 15)

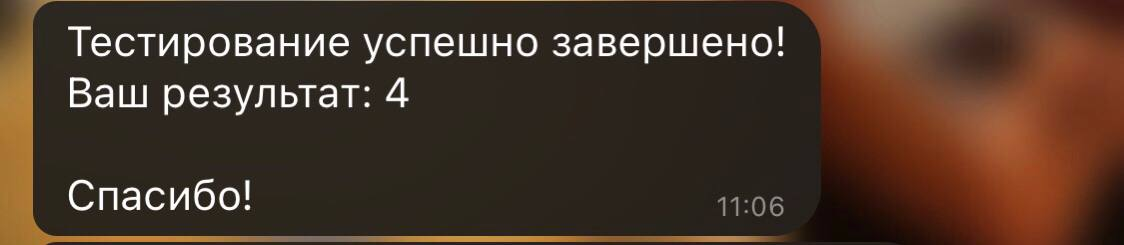


Рис. 15. Итог тестирования

Кроме того, в дальнейшем планируется развивать данный проект в следующих направления:

- интеграция в админ-панель статистики тестирования в режиме реального времени (преподаватель будет иметь возможность отслеживать прогресс студента по мере прохождения теста, а также видеть полную статистику по результатам его выполнения);

- возможность рассылки сообщений через админ-панель (индивидуальной и групповой), что может быть использовано в том числе и для дистанционного обучения.

- изменение внешнего вида админ-панели (добавление средств визуализации текущих тестирований, активных пользователей и т.д.)

Представленный проект можно использовать для промежуточного и итогового тестирования студентов. Особенностью данной системы является возможность удаленного доступа, использования личных смартфонов, что ускоряет процесс прохождения тестирования. Кроме того, использование широко известного мессенджера Телеграмм позволит упростить использования программы в связи с отсутствием необходимости устанавливать дополнительное программное обеспечение

**Список использованной литературы**

1. Доусон, М. Программируем на Python / М. Доусон. – СПб.: Питер, 2014. - 416 с. - ISBN: 978-5-4461-1386-6.
2. Дронов, В.А. . Django 2.1. Практика создания веб-сайтов на Python / В.А. Дронов. – СПб.: БХВ-Петербург, 2019. - 672 с. - ISBN: 978-5-9775-4058-2
3. Моргунов, Е.П. PostgreSQL. Основы языка SQL: учебное пособие / Е.П. Моргунов. – СПб.: БХВ-Петербург, 2018. - 336 с. - ISBN: 978-5-9775-4022-3
4. Фаулер, М. Asyncio и конкурентное программирование на Python/пер. с англ. А.А. Слинкина / М. Фаулер. – М.: ДМК-пресс, 2022. - 398 с. - ISBN: 978-5-93700-166-5