

Автоматизация проверки работ студентов с помощью Telegram бота

Содержание

1	Кратко о целях и функционале	1
2	Детали функционирования	2
2.1	Взгляд студента	2
2.2	Взгляд преподавателя	7
2.3	Взгляд разработчика	9

1 Кратко о целях и функционале

Целью проекта является упрощение процесса оценки преподавателем качества работы студентов. В качестве **метода** достижения цели был выбран чат-бот в Telegram. На данный момент реализованный бот обладает следующими функциями:

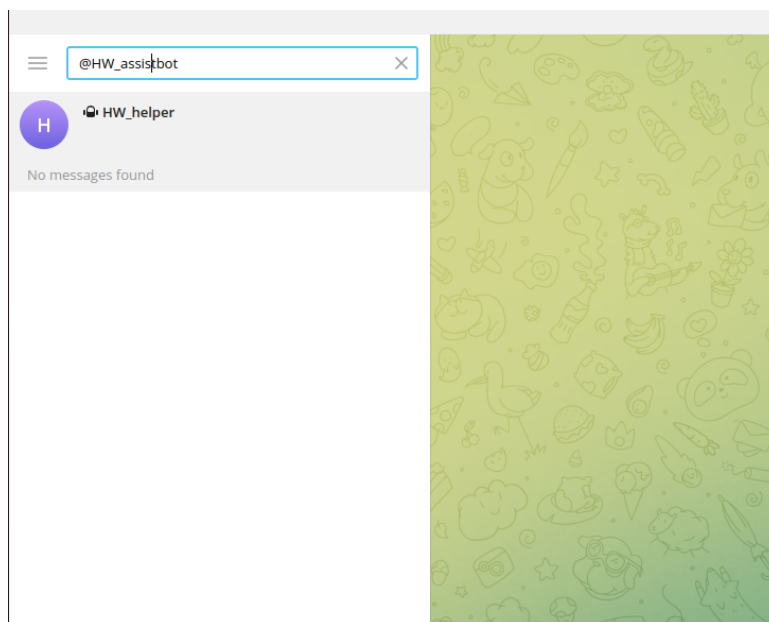
- регистрация пользователя,
- получение списка доступных заданий,
- получение информации о доступных заданиях,
- выдача заданий,
- отправка выполненных заданий,
- проверка ответов студента и выставление баллов,
- просмотр статистики: сколько баллов проставлено за каждое задание,
- получение архива решений преподавателем,
- формирование отчёта для преподаателя.

Ниже приведено подробное описание перечисленных функций.

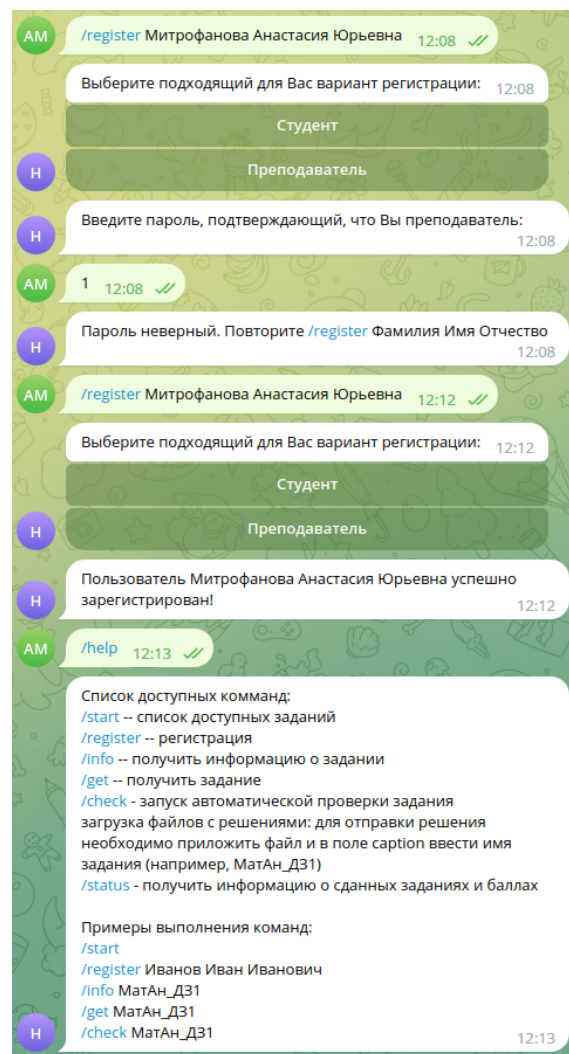
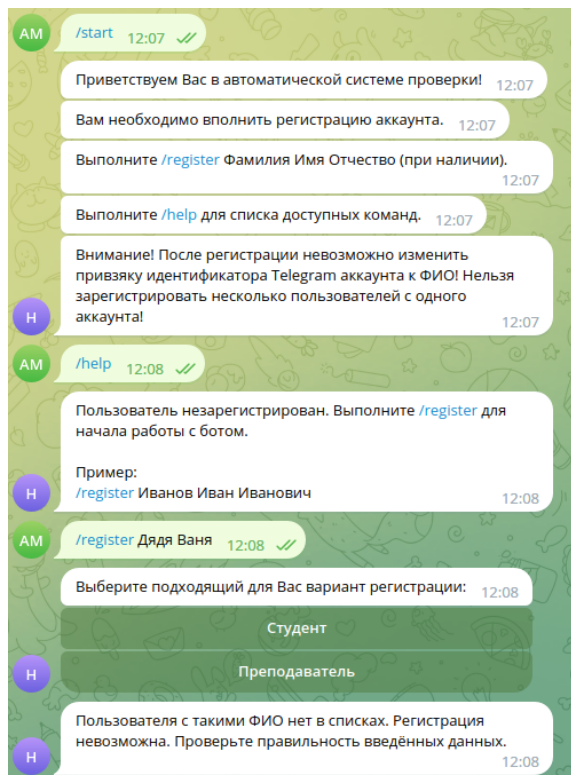
2 Детали функционирования

2.1 Взгляд студента

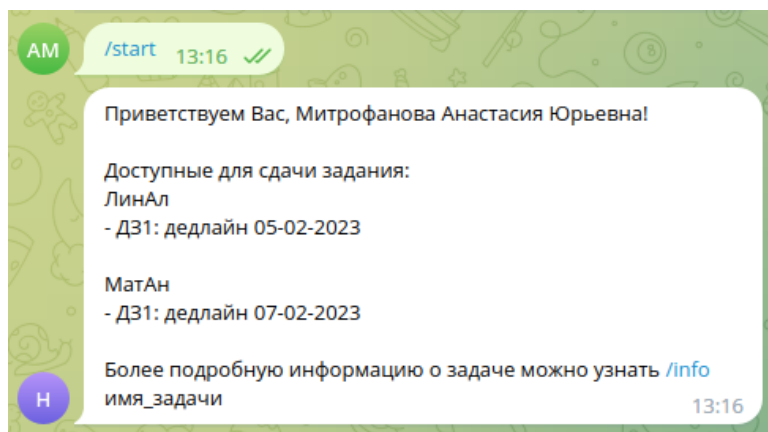
Уникальное имя пользователя бота — @HW_assistbot, по этому имени студент может найти бота в Telegram. При этом никнейм — HW_helper. Никнейм и имя пользователя могут быть изменены позднее.



При запуске чата бот приветствует пользователя и просит зарегистрироваться. Обратите внимание, что на данный момент бот доступен не всегда, а только в то время, когда на компьютере разработчика запущен соответствующий код. Для регистрации студенту нужно ввести свои ФИО. Предполагается, что ботом будут пользоваться студенты, занесённые в специальную таблицу на компьютере преподавателя/разработчика, поэтому пользователи, чьих ФИО нет в данной таблице, не смогут пройти дальше данного этапа. Возможен вариант с последующим размещением бота на сервере.

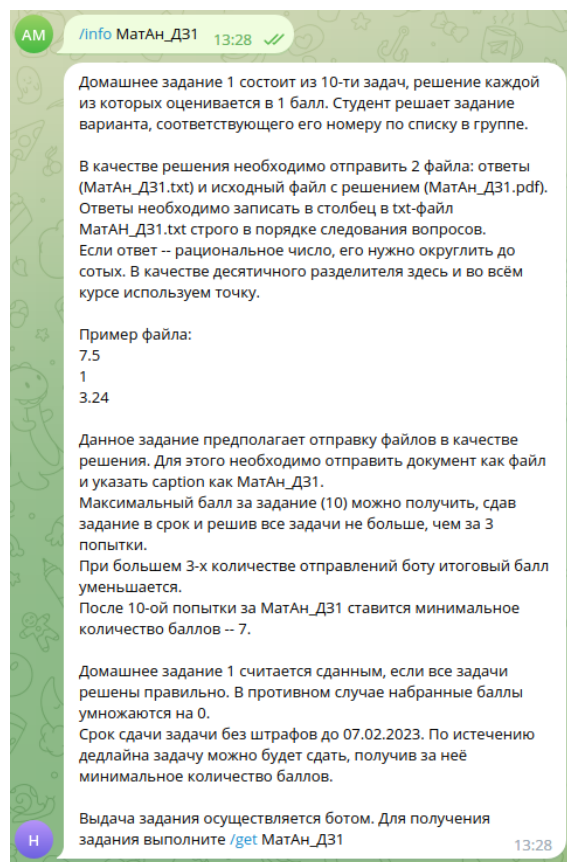
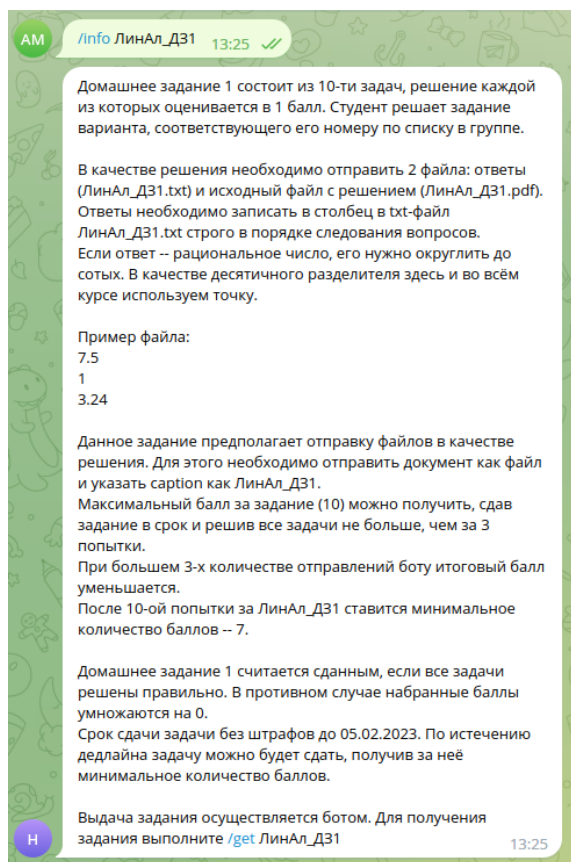


Для просмотра списка доступных заданий студент должен написать боту команду /start. Бот находит соответствие уникального идентификатора пользователя Telegram фамилии, имени и отчеству конкретного студента. Поэтому, обращаясь к пользователю, бот будет использовать именно его ФИО. Этот же факт делает невозможной регистрацию и сдачу работ нескольких студентов с одного аккаунта.

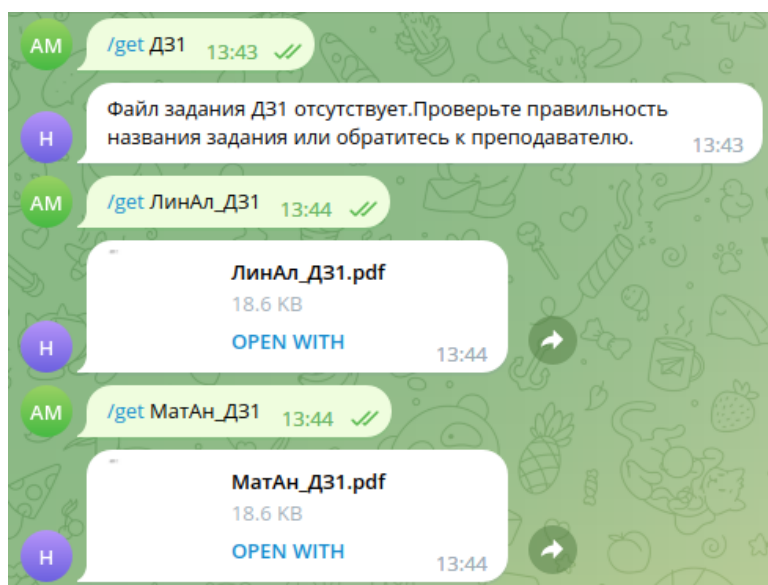


Получение информации о задании поддерживается командой /info. В ответ на свой запрос студент получит сообщение, содержащее текст из подготовленного преподавателем

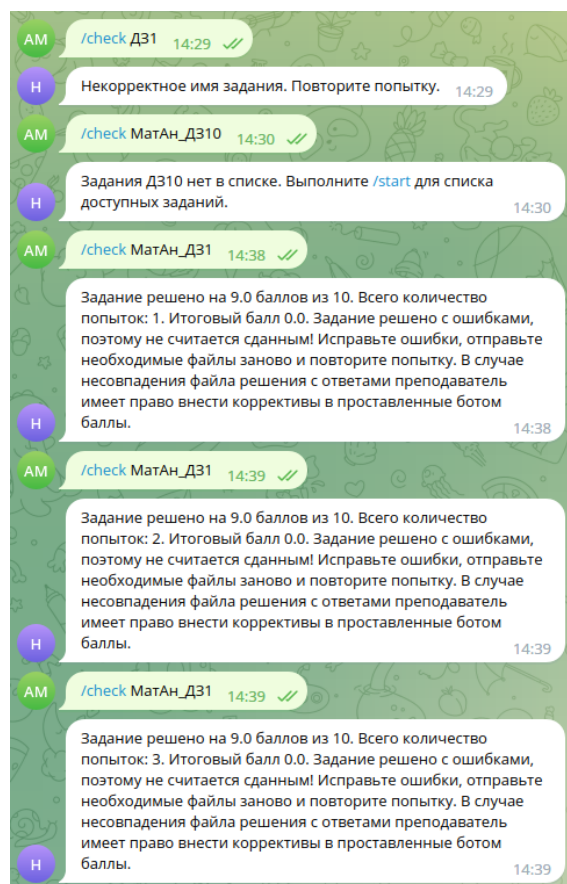
текстового файла формата .txt. Текстовый файл может содержать комментарии к получению, организации сдачи задания, или само задание.

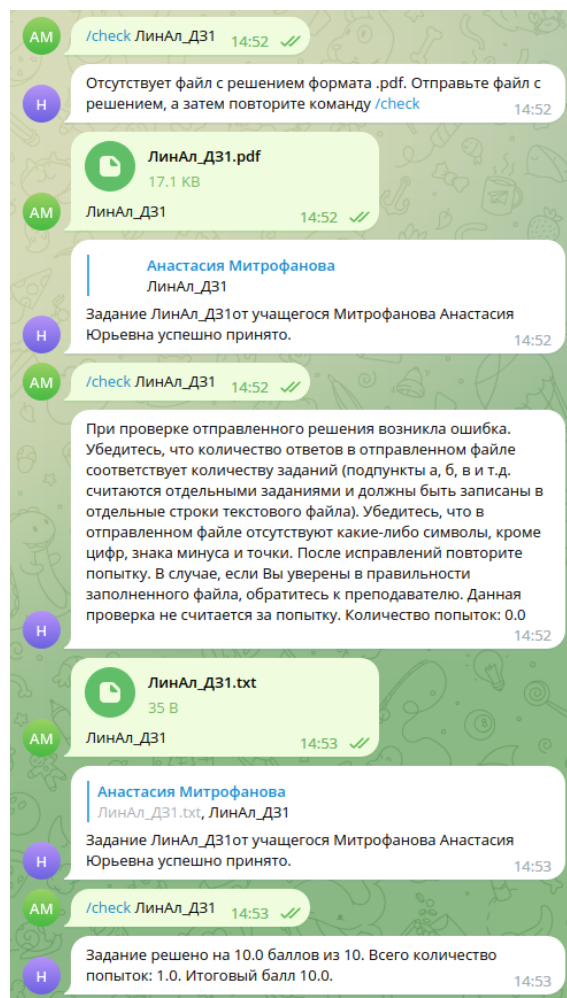
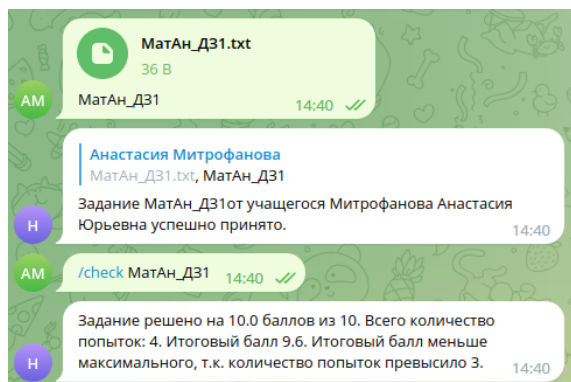


В случае, если это предусмотрено преподавателем, студент может получить задание у бота, отдав тому команду `/get`. На данный момент предполагается, что файл задания один на всех студентов группы. В дальнейшем возможно реализовать присвоение варианта студенту при регистрации или генерацию индивидуальных заданий.

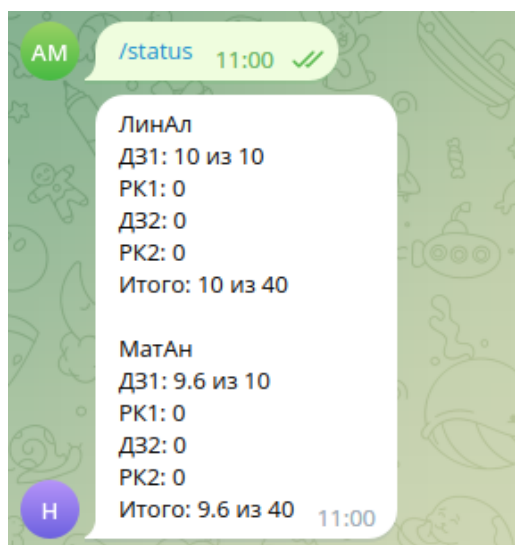


Студент может отправить боту файлы со своими решениями, в числе которых обязательно текстовый файл формата .txt (в случае его отсутствия бот отправит студенту соответствующее сообщение при проверке). Отправив боту команду /check, студент может проверить свои ответы с теми, что хранятся в текстовом файле на компьютере преподавателя/разработчика или на сервере. Ниже на рисунке выставление баллов осуществлялось по описанной выше на рисунках тактике.





Наконец, студент может узнать информацию о своих баллах, отправив боту сообщение с командой /status.



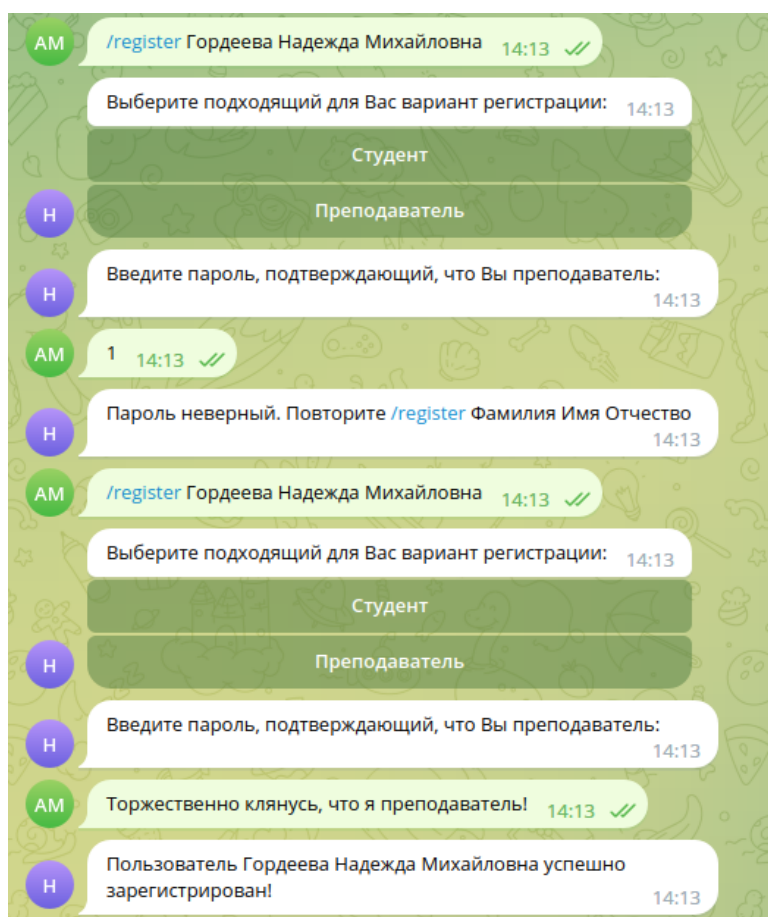
2.2 Взгляд преподавателя

Для корректной работы бота от преподавателя требуется

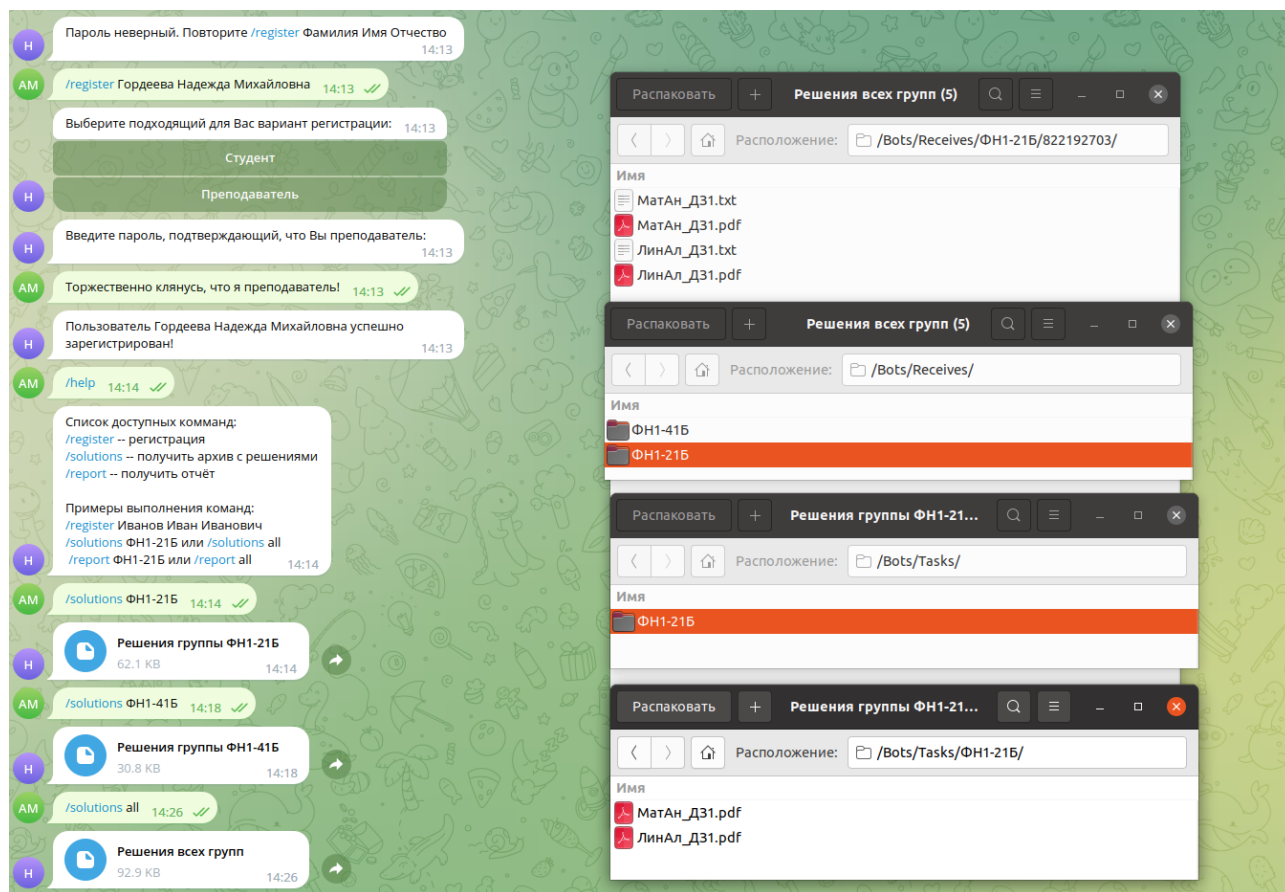
- предоставить полный список студентов с указанием их ФИО и номера группы;
- предоставить названия заданий, крайних сроков сдачи, минимальных и максимальных баллов за эти задания, указания на возможность автоматической проверки;
- иметь файлы с заданиями и ответы на них;
- создать файлы с информацией о заданиях.

После проверки заданий ботом составляется таблица с указанием баллов. Файлы, отправленные студентами, на данный момент сохраняются на компьютере разработчика.

Для регистрации преподавателю необходимо знать пароль (один для всех преподавателей).



Т.к. бот может проверить лишь численные ответы, то проверка самих решений студентов остаётся обязанностью преподавателя. С помощью бота же можно получить отчёт, применив для этого команду `/report`, опираясь на который можно потратить меньше времени на проверку работ, также получаемых в архиве командой `/solutions`.



Report_ФН1-215_МатАн (6).xlsx - LibreOffice Calc

ФайлПравкаВидВставкаФорматСтилиЛистДанныеСервисОкноСправка

Calibri11ЖКЧА

fxΣ=

74%

A1

Математический анализ. ФН1-215																		
Студент			Д31			РК1			Д32			РК2			Итого			
Фамилия	Имя	Отчество	балл	min	max	балл	min	max	балл	min	max	балл	min	max	балл	min	max	
0	Митрофанова	Анастасия Юрьевна	9,6	7	10	8	5	10	10	7	10	9,5	5	10	37,1	24	40	
1	Байбиков	Анастасия Борисовна	7	7	10	10	5	10	10	7	10	6	5	10	33	24	40	

Лист 1 из 1

PageStyle_Sheet1

Русский

Среднее значение: Сумма: 0

AM /report ФН1-215 14:30 ✓

Report_ФН1-215_МатАн.xlsx5.8 KB14:30

Report_ФН1-215_ЛинАл.xlsx5.8 KB14:30

2.3 Взгляд разработчика

Код бота представлен в виде процедур, входными данными для которых являются сообщения (и сопутствующая информация, такая как id и имя отправителя в Telegram, дата отправки и т.д.), и некоторых вспомогательных функций, код которых представлен ниже.

```
def get_user_data(user_id, register_table, real_table):  
    """Возвращает личные данные пользователя"""  
    registered_users = pd.read_excel(register_table)  
    real_user_data = pd.read_excel(real_table)  
    user_data = registered_users[registered_users['id'] == user_id]  
    surname = list(user_data['Фамилия'])[0]  
    name = list(user_data['Имя'])[0]  
    patronymic = list(user_data['Отчество'])[0]  
    if real_table == 'MyStudents.xlsx':  
        condition = (real_user_data['Фамилия'] == surname) & \  
                    (real_user_data['Имя'] == name) & \  
                    (real_user_data['Отчество'] == patronymic)  
        group = list(real_user_data[condition]['Группа'])[0]  
        return surname, name, patronymic, group  
    else:  
        condition = (real_user_data['Фамилия'] == surname) & \  
                    (real_user_data['Имя'] == name) & \  
                    (real_user_data['Отчество'] == patronymic)  
        group = list(real_user_data[condition]['Группа'])  
        code = list(real_user_data[condition]['Код предмета'])  
        return surname, name, patronymic, group, code  
  
def check_if_registered(user_id, mode='universal'):  
    """Возвращает True или False в зависимости от наличия id пользователя  
    в таблице с зарегистрированными студентами/преподавателями"""  
    registered_students = pd.read_excel('RegisteredStudents.xlsx')  
    registered_teachers = pd.read_excel('RegisteredTeachers.xlsx')  
    if mode == 'universal':  
        if user_id in list(registered_students['id']) + \  
            list(registered_teachers['id']):  
            return True  
        else:  
            return False  
    elif mode == 'student':  
        if user_id in list(registered_students['id']):  
            return True  
        else:  
            return False  
    else:  
        if user_id in list(registered_teachers['id']):  
            return True  
        else:  
            return False
```

```
def save_zip(arch, folder_list, mode, subjects):  
    """Сохраняет архив с решениями"""  
    z = zipfile.ZipFile(arch, mode, zipfile.ZIP_DEFLATED, True)  
    for add_folder in folder_list:  
        for root, dirs, files in os.walk(add_folder):  
            for file in files:  
                if file.split('_')[0] in subjects:  
                    path = os.path.join(root, file)  
                    z.write(path, os.path.relpath(os.path.join(root, file),  
                                                    os.path.join(root_dir, '..')))  
    z.close()
```

На личном ПК разработчика при этом находятся таблицы (формат .xlsx), с которыми осуществляет работу бот, со следующим содержанием:

- список студентов (ФИО и группа),
- список преподавателей (ФИО, группа и код преподаваемого предмета (если предметов и групп несколько, то ФИО преподавателя дублируются)),
- зарегистрированные студенты (ФИО, id, username),
- зарегистрированные преподаватели (ФИО, id, username),
- информация о заданиях (группа, код предмета (например, МатАн), короткое название задания (например, ДЗ1), даты выдачи и крайних сроков, минимальный и максимальный баллы, возможность автопроверки),
- информация о полученных баллах (по листу на каждый предмет) (id, ФИО, первичный балл, наличие файлов .txt и .pdf, дата сдачи и количество попыток, вес, итоговый балл)