**Вступление**

В начале сентября электронный дневник МЭШ(Московская Электронная Школа) крайне плохо работал: постоянно падал, многие учителя не могли загрузить в него домашнее задание и тому подобное. Хоть это и давало мне давало дополнительный повод не делать домашнее задание, но все-таки вызывало дискомфорт. Так же мне никогда не нравилось, когда какую-то очень важную информацию скидывали в классные чаты, которые мне было неудобно читать и анализировать каждое написанное там сообщение на наличие какой-либо полезной для меня информации. А поскольку мой класс очень общительный и написать в чат более пяти сотен сообщений в день для них не составляет труда, то для меня это стало серьезной проблемой. Так что искать что-то важное среди сотен сообщений и десятков фотографий меня всегда раздражало. Таким образом мне пришла идея создать свой классный журнал, который будет автономным от МЭШ и будет содержать не только домашнее задание, а еще и полезную организационную информацию по поводу классных мероприятий, дней рождений учеников и тому подобного. Телеграм, как платформу для реализации моей задумки, был выбран, поскольку я посчитал его удобным для пользователя и у меня уже имелся опыт написания ботов для него на все том же aiogram'е. Согласитесь, в конце урока зайти в Телеграм, сделать несколько нажатий и написать домашнее задание в бота не составит трудности, как и потом посмотреть его, и это будет в разы удобнее, чем листать классные чаты в поиске нужной тебе информации.

**Описание работы бота**

Для начала хочется отметить, что на момент последней редакции данного текстового файла, работа над проектом еще не была завершена, поэтому написанная ниже информация может устареть ввиду того, что многие аспекты технической части проекта могут быть переработаны и переделаны. Также хочу отметить, что некоторые аспекты, в отношении которых еще не проводилось активной работы, могут быть описаны достаточно поверхностно, особенно по сравнению с теми аспектами, которые уже готовы или находятся в разработке прямо сейчас. Заранее благодарю за понимание.

**База данных и её структура**

Телеграм бот OKlJournail связан с базой данных, куда загружаются вся информация, необходимая для долго хранения. Это данные о пользователях, домашнее задание, списки классов, дополнительная информация, данные о подписчиках на класс и так далее. Сами данные структурируются на 2 глобальные таблицы: таблица пользователей и таблица профилей классов. К таблице профилей классов привязано еще 4 таблицы для домашнего задания, дополнительной информации, списка учеников и списка подписчиков. Все эти таблицы уникальны для каждого профиля класса и генерируются автоматически. И того вся структура базы данных, привязанной к боту, выглядит следующим образом (см. схему 1):

Схема 1

Таблица пользователей «users» состоит из двух столбцов: столбца «user\_id» и столбца «full\_name». В столбце «user\_id» хранится уникальный id пользователя. В столбце «full\_name» хранится полное имя пользователя в Телеграме.

Таблица профилей классов «classes» состоит из семи столбцов: столбца «class\_id», столбца «class\_name», столбца «school\_name», столбца «task\_list\_id», столбца «hw\_list\_id», столбца «students\_list\_id» и столбца «subscribers\_list\_id». В столбце «class\_id» хранится id профиля класса, который генерируется рандомно в диапазоне от 100000 до 999999 и при этом уникален. В столбце «class\_name» хранится название класса, которое задается пользователем с клавиатуры. В столбце «school\_name» хранится название школы, которое задается пользователем с клавиатуры. В столбцах «task\_list\_id», «hw\_list\_id», «students\_list\_id» и «subscribers\_list\_id» хранятся уникальные названия для таблиц для этого профиля класса, в которых хранятся дополнительная информация, домашнее задание, список учеников и список подписчиков соответственно, и которые генерируются по шаблону «ts\_уникальное число от 100000 до 999999», «hw\_уникальное число от 100000 до 999999», «st\_уникальное число от 100000 до 999999» и «sb\_уникальное число от 100000 до 999999» соответственно.

Таблица для дополнительной информации с названием по шаблону «ts\_уникальное число от 100000 до 999999» состоит из трех столбцов: столбца «task», столбца «term\_date» и столбца «term\_day». В столбце «task» хранится информация, заданная пользователем. В столбце «term\_date» храниться дата крайнего срока, заданная пользователем с клавиатуры. В столбце «term\_day» храниться день крайнего срока, заданный пользователем с клавиатуры.

Таблица для домашнего задания с названием по шаблону «hw\_уникальное число от 100000 до 999999» состоит из четырех столбцов: столбца «subject», столбца «task», столбца «term\_date» и столбца «term\_day». В столбце «subject» хранится название предмета, заданное пользователем с клавиатуры. В столбце «task» хранится содержание домашнего задания, заданное клавиатуры. В столбце «term\_date» храниться дата крайнего срока, заданная пользователем с клавиатуры. В столбце «term\_day» храниться день крайнего срока, заданный пользователем с клавиатуры.

Таблица для списка учеников с названием по шаблону «st\_уникальное число от 100000 до 999999» состоит из двух столбцов: столбца «surname\_name» и столбца «birthday». В столбце «surname\_name» хранится настоящая фамилия и имя ученика, заданные пользователем с клавиатуры. В столбце «birthday» хранится день рождения ученика, заданный пользователем с клавиатуры.

Таблица для списка подписчиков на профиль класса с названием по шаблону «sb\_уникальное число от 100000 до 999999» состоит из двух столбцов: столбца «user\_id» и столбца «surname\_name». В столбце «user\_id» хранится уникальный id пользователя. В столбце «surname\_name» хранится настоящая фамилия и имя ученика, заданные пользователем с клавиатуры.

**Использованные библиотеки**

В основном мною были задействованы 2 библиотеки: aiogram и sqlite3. С помощью библиотеки aiogram мною был написан сам функционал взаимодействия бота с пользователем и является основной библиотекой данного проекта. Aiogram был выбран мной, поскольку является наиболее технически грамотным по сравнению с другими библиотеками для ботов в Телеграме (например Telebot’ом) и поскольку у меня уже был неплохой опыт работы с этой библиотекой, как уже и было сказано ранее. Библиотека sqlite3 позволяет мне подключать базы данных к моей программе и взаимодействовать с ними при помощи языка SQL. Также мной были использованы такие библиотеки как random.

**Код и его работа**

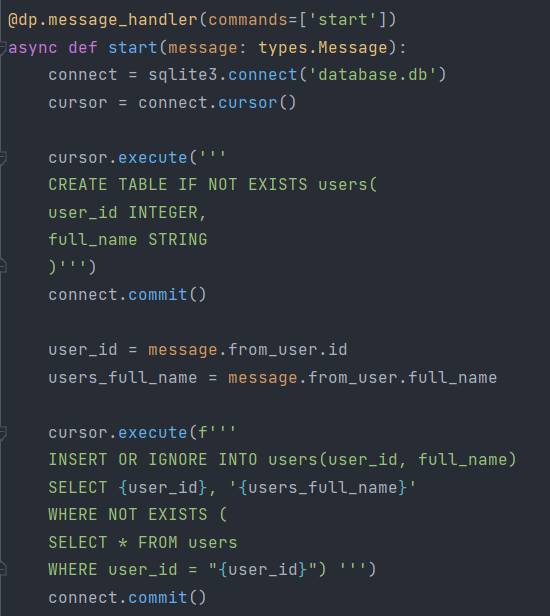
Начало работы бота начинается с ввода команды /start. При вводе этой команды программа (см. рис.1) подключает базу данных. Затем при помощи языка SQL проходит проверка на существование в ней таблицы users. Если такой нет, то она создается с заданными параметрами. Далее в таблицу users добавляются данные пользователя, если их там нет. 

рис.1

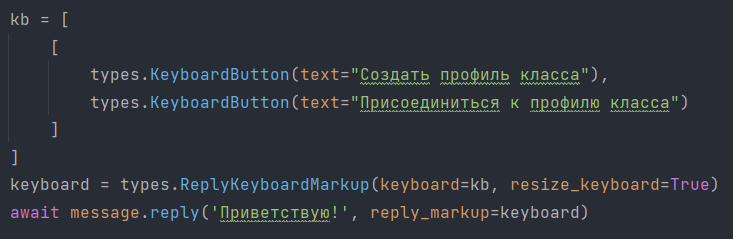
Затем бот выводит клавиатуру с вариантами последующего ответа и отвечает пользователю (см. рис.2)

рис.2

По итогу в чате с ботом это выглядит следующим образом (см. рис.3):



рис.3