

Постановка задачи

Идея продукта - telegram-бот для отправки аналога обычных стикеров между пользователями в виде аудио-стикеров - коротких голосовых сообщений с забавным содержанием.

Проблема:

В мессенджере Telegram пользователи могут обмениваться стикерами - небольшими изображениями (статичными или анимированными), которые выражают эмоции и могут быть использованы в разговоре. Однако, в текущей реализации стикеров ограничены только графической составляющей, что может быть ограничением для пользователей, которые хотят делиться аудио-контентом или звуковыми эффектами.

Подход к решению:

Для решения данной проблемы предлагается разработать телеграм бота, который позволит пользователям создавать и отправлять аудио-стикеры. Аудио-стикеры будут аналогичны обычным стикерам, но вместо изображений они будут содержать звуковые файлы или аудиозаписи.

Основные функции и возможности бота:

1. Создание аудио-стикеров: Пользователи смогут загружать собственные аудиофайлы или аудиозаписи и привязывать их к определенному стикеру, задав имя загружаемого стикера.
2. Хранилище аудио-стикеров: Бот будет иметь хранилище для сохранения загруженных аудио-стикеров, чтобы пользователи могли легко выбрать и отправлять доступные стикеры.
3. Отправка аудио-стикеров: Пользователи смогут выбрать нужный аудио-стикер из хранилища и отправить его в чаты или приватно другим пользователям. Аудио-стикер будет работать аналогично обычным стикерам - при нажатии на него, будет проигрываться связанный с ним звуковой файл.
4. Удобство использования: Бот будет предоставлять простой и понятный интерфейс, который позволит пользователям легко создавать, сохранять и отправлять аудио-стикеры. Он будет иметь интуитивно понятные команды и меню для доступа к разным функциям бота.
5. “Социальные” функции: Бот будет иметь функции социальной сети, такие как возможность добавить в чат или себе в коллекцию набор

стикеров другого автора, поделиться набором стикеров с другими пользователями. Это позволит пользователям обмениваться своими творческими работами, находить новых интересных создателей аудио-стикеров и взаимодействовать с ними.

Результат работы

В конечном счете был реализован следующий набор функций, которые может использовать пользователь:

1. Создание наборов стикеров
2. Удаление наборов стикеров
3. Добавление стикера в набор
4. Удаление стикера из набора
5. Возможность просмотра всех доступных пользователю наборов
6. Возможность поделиться стикерпаком с другими пользователями или добавить стикерпак другого пользователя себе в коллекцию

Telegram API используется для взаимодействия приложения, базы данных и хранилища с ботом и данными чатов, в которых он используется. С помощью этой технологии производятся асинхронные запросы к БД, хранилищу для сохранения или извлечения данных, которые затем в ответ передаются в чат, из которого делается запрос.

```
private readonly Client client;

private readonly TgApiCommandService tgApiCommandService;
private readonly TgApiCommandResultHandlerService tgApiCommandResultHandlerService;

private readonly Dictionary<long, UserInfo> userInfoByChatId = new();

private readonly ILog log;

private readonly IUsersRepository userRepository;

public Queue<Update> Requests { get; } = new();

public TgApiGatewayService(
    Client client,
    TgApiCommandService tgApiCommandService,
    TgApiCommandResultHandlerService tgApiCommandResultHandlerService,
    ILog log,
    IUsersRepository userRepository)
{
    this.client = client;
    this.tgApiCommandService = tgApiCommandService;
    this.tgApiCommandResultHandlerService = tgApiCommandResultHandlerService;
    this.log = log;
    this.userRepository = userRepository;
}
```

Для перемещения между элементами меню бота используется принцип сохранения состояния пользователя:

```
1 namespace VoiceStickersBot.Core.CommandArguments;
2
3 public enum UserState
4 {
5     WaitFile,
6     WaitPackName,
7     WaitStickerName,
8     WaitStickerChoice,
9     WaitPackId,
10    NoWait
11 }
```

Для каждого действия пользователя описано поведение приложения и набор действий, которые можно выполнять в текущем состоянии. Например, пользователь хочет добавить новый стикер в уже созданный набор; для этого он выбирает соответствующие опции в меню, и в момент, когда нужно прикрепить аудио файл для формирования стикера, приложение запоминает, что пользователь находится на этапе отправления файла, и ожидает от пользователя только действий, связанных с отправкой файла или отменой проводимой в боте операции.

```

else if (userInfoByChatId.TryGetValue(chatId, out var userInfo)
        && userInfo.State == UserState.WaitStickerName)
{
    var stickerPackId = userInfo.StickerPackId;
    var args = new List<string> { stickerPackId, message.Text };
    context = new QueryContext("AS", "SendFileInstr", args, chatId);
}
else if (userInfoByChatId.TryGetValue(chatId, out userInfo)
        && userInfo.State == UserState.WaitPackName)
{
    var args = new List<string> { message.Text };
    context = new QueryContext("CP", "AddPack", args, chatId);
}
else if (userInfoByChatId.TryGetValue(chatId, out userInfo)
        && userInfo.State == UserState.WaitPackId)
{
    var args = new List<string> { message.Text };
    context = new QueryContext(
        "SP",
        "ImportPack",
        args,
        chatId,
        chatType.ToString());
}

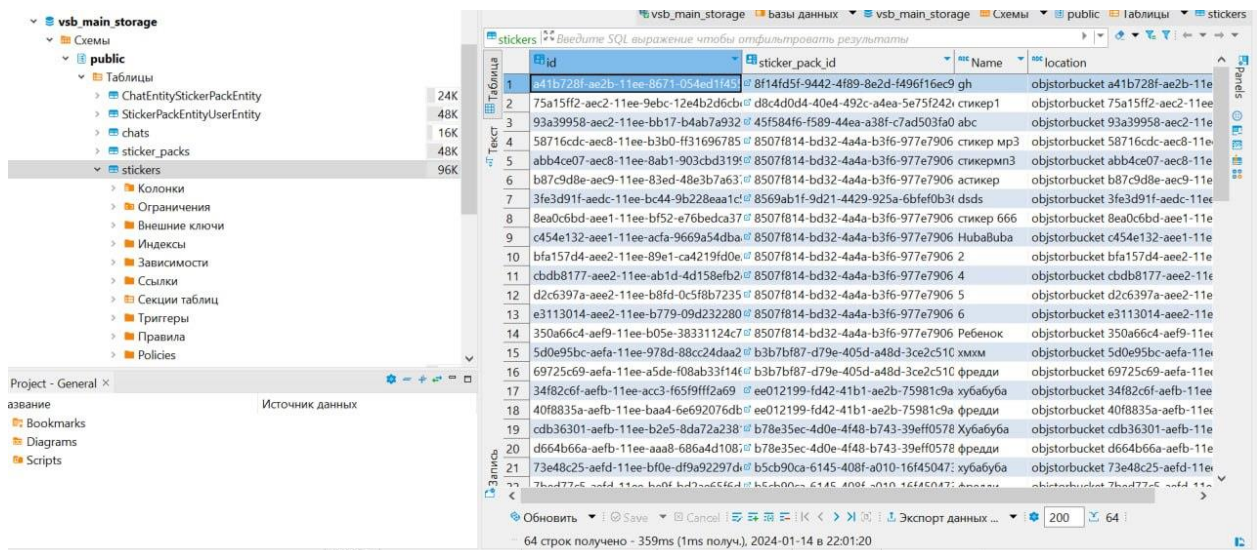
```

Это делает поведение приложения предсказуемым и позволяет избежать ошибок, связанных с различными сценариями использования приложения.

Хранение данных о создателях наборов, самих наборах и стикерах в них производится с помощью базы данных PostgreSQL.



Для того, чтобы идентифицировать каждый набор и каждый стикер, информация о них записывается в базу данных под уникальным Guid, созданным из временной метки, отражающей время создания набора или стикера.



Индексы:

Название	Column	Таблица	По возрастанию	Значение NULL	Уникальный	Клс
> IX_stickers_Name_id	Name, id	stickers	—	—	[]	
> IX_stickers_sticker_pack_id	sticker_...	stickers	—	—	[]	
> PK_stickers	id, stick...	stickers	—	—	[v]	

Хранение стикеров производится в “сыром” виде - массивы байт сохраняются под одним Guid в хранилище Yandex Object Storage, после чего извлечь их можно, сделав запрос соответствующих данному стикеру данных к базе данных из приложения и обратиться к хранилищу по метке Guid, привязанной к соответствующему стикеру.

Примерно так выглядит итоговое хранилище стикеров:

Имя	Размер	Класс хранилища	Последнее изменение
02b9c33d-a7c1-11ee-b08b-e73bdc45c22	101.69 КБ	Стандартное	10.01.2024, в 10:49
071e8b01-a984-11ee-89e5-b0c2aaddc65	56.19 КБ	Стандартное	10.01.2024, в 11:46
09bd52b3-ac26-11ee-ba40-3ac7e8e302d9	101.69 КБ	Стандартное	07.01.2024, в 11:29
0cc3d6a0-ad4b-11ee-8789-26889345c88f	5.78 КБ	Стандартное	07.01.2024, в 15:53
123e63b8-a901-11ee-8abd-d3491bd83cd	298.31 КБ	Стандартное	09.01.2024, в 20:09
13ae40ee-acbb-11ee-81e6-fccf9132620f	81.48 КБ	Стандартное	06.01.2024, в 22:43
1479cd9c-a900-11ee-a77e-82b144fab30	813.48 КБ	Стандартное	09.01.2024, в 20:02
1b54xc28-ad26-11ee-9035-902e74e0f616	23.38 КБ	Стандартное	07.01.2024, в 11:29
2389d97d-acc8-11ee-9270-1a5e9f5abe11	23.38 КБ	Стандартное	07.01.2024, в 00:16
24eef75e-a902-11ee-8e72-577a702b0fa1	77.96 КБ	Стандартное	09.01.2024, в 20:17
262a78b5-acb5-11ee-a457-8e9b6eaa2c6	49.03 КБ	Стандартное	06.01.2024, в 22:43
2932b6a0-a7c1-11ee-b6d3-672b2dc8b6b	23.38 КБ	Стандартное	10.01.2024, в 10:50
2964bae9-ad93-11ee-8070-7a5c1a4eeba2	37.87 КБ	Стандартное	08.01.2024, в 00:30
297e13d-acc5-11ee-a25e-c519615e03c5	49.03 КБ	Стандартное	07.01.2024, в 00:59
2a5ecd57-a771-11ee-9244-556568cc8b9	48.16 КБ	Стандартное	10.01.2024, в 11:12

Ссылки

<https://docs.google.com/document/d/1X8J6XbW44eahXNJzP5vcm5EL4eFlK8lq0K2RNFIP7vI/edit?usp=sharing> - описание продукта (стартовая итерация);

<https://www.figma.com/file/21dihmpiUbOORMMsGnaXUW/Untitled?type=design&node-id=0%3A1&mode=design&t=X3O4YMo86mKbMjR9-1> - описание MVP, выполнено в Figma для наглядности выполнения сценариев использования продукта;

<https://cloud.yandex.ru/ru/services/storage> - технология, позволившая реализовать хранение данных бота и стикеров в облаке. Для реализации технологии в проекте важную роль сыграла документация;

<https://t.me/thmusicbot> - бот, которым мы вдохновлялись в плане удобства пользования и общего подхода к UX, его функциональность не была скопирована участниками проекта.