**Содержание:**

[**Цель**](#_joy88qaf7ddc) **4**

[**Основные определения**](#_u54e41eqe1iz) **4**

[**Сокращения:**](#_m94xfji3lfy6) **5**

[**Введение**](#_lf5aow8366k6) **5**

[**Требования к проекту**](#_yaxn19ybwswh) **6**

[**Проектирование системы**](#_758d48x0ts5t) **7**

[**Выбор технологий**](#_26in1rg) **14**

[Выбор языка программирования](#_lnxbz9) 14

[Выбор используемых библиотек](#_3uaa8pb8kqb2) 15

[**Описание технический решений**](#_arrj4eqm598) **16**

[**Заключение**](#_44sinio) **18**

[**Список используемой литературы**](#_2jxsxqh) **18**

# 

# **Цель**

Создание чат-бота в приложении Telegram для работы с субтитрами и изучения незнакомых пользователю слов на иностранном языке.

# Основные определения

**Telegram** — бесплатный кроссплатформенный мессенджер для смартфонов и других устройств, позволяющий обмениваться текстовыми сообщениями и медиафайлами различных форматов.

**API** (***a****pplication* ***p****rogramming* ***i****nterface*) — набор готовых классов, процедур,функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах.

**SQL** ( *structured query language*) — язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

**Система управления базами данных (СУБД)** — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

**PaaS-платформа** (place as a service) - модель предоставления облачных вычислений, при которой потребитель получает доступ к использованию информационно-технологических платформ: операционных систем, систем управления базами данных, связующему программному обеспечению, средствам разработки и тестирования, размещенным у облачного провайдера.

**Heroku** - облачная PaaS-платформа, поддерживающая ряд языков программирования.

**Бот** — специальная программа, выполняющая автоматически и/или по заданному расписанию какие-либо действия через интерфейсы, предназначенные для людей.

# Сокращения:

*БД* – база данных;

# Введение

Многие в наше время смотрят сериалы и фильмы в оригинале без русского дубляжа. Но хорошо знают язык далеко не все. Переводить слова по мере просмотра фильма порой неудобно, так почему бы не посмотреть незнакомые Вам слова перед самим просмотром. Запомнить их значение и без постоянных заглядываний в словарь или переводчик посмотреть очередную серию сериала или фильма.

В этом и есть суть моего проекта. Пользователю будет дана возможность поиска субтитров по названию и предоставлен список незнакомых ему слов вместе с определением и переводом к ним.

Для удобства предложенный функционал будет предоставляться ботом в Telegram. Так пользователю не придется скачивать отдельное приложение, которое занимает место на устройстве, создавать отдельный аккаунт для сервиса и т.д. Весь функционал будет представлен в чате с ботом.

Также будет обеспечена кроссплатформенность и возможность быстрого развертывания бота на новом сервере.

# Требования к проекту

1. Легкость и простота развертывания бота на новом сервере
2. Загрузка субтитров с ресурса www.opensubtitles.org
3. Получать карточки для изучения неизвестных слов
4. Иметь возможность изучать слова для каждого фильма по отдельности
5. Предоставить круглосуточный доступ к боту

# 

# 

# 

# Проектирование системы

Система состоит из четырех модулей:

1. Telegram бот
2. Сервер
3. База данных
4. Скрипт

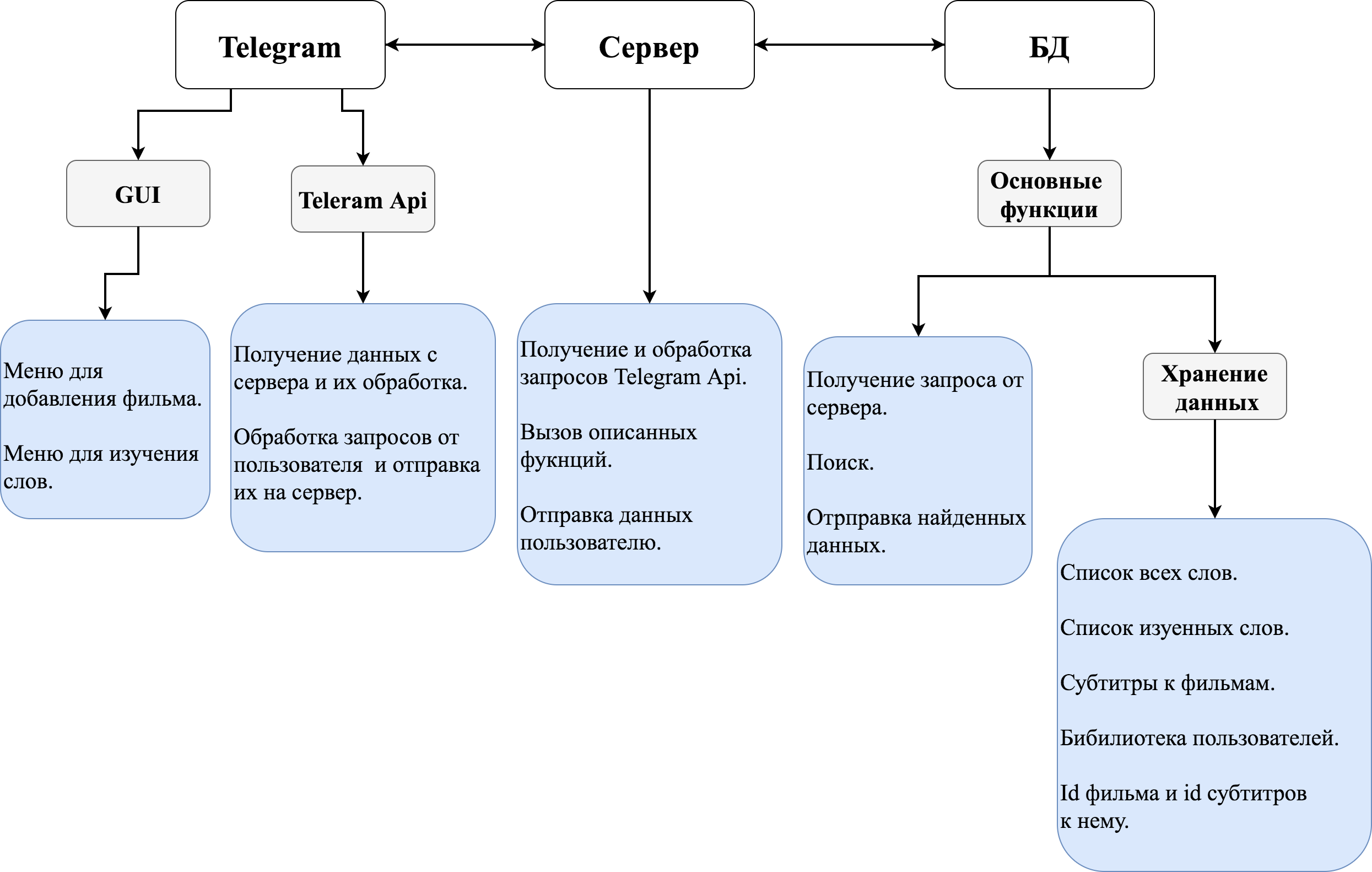


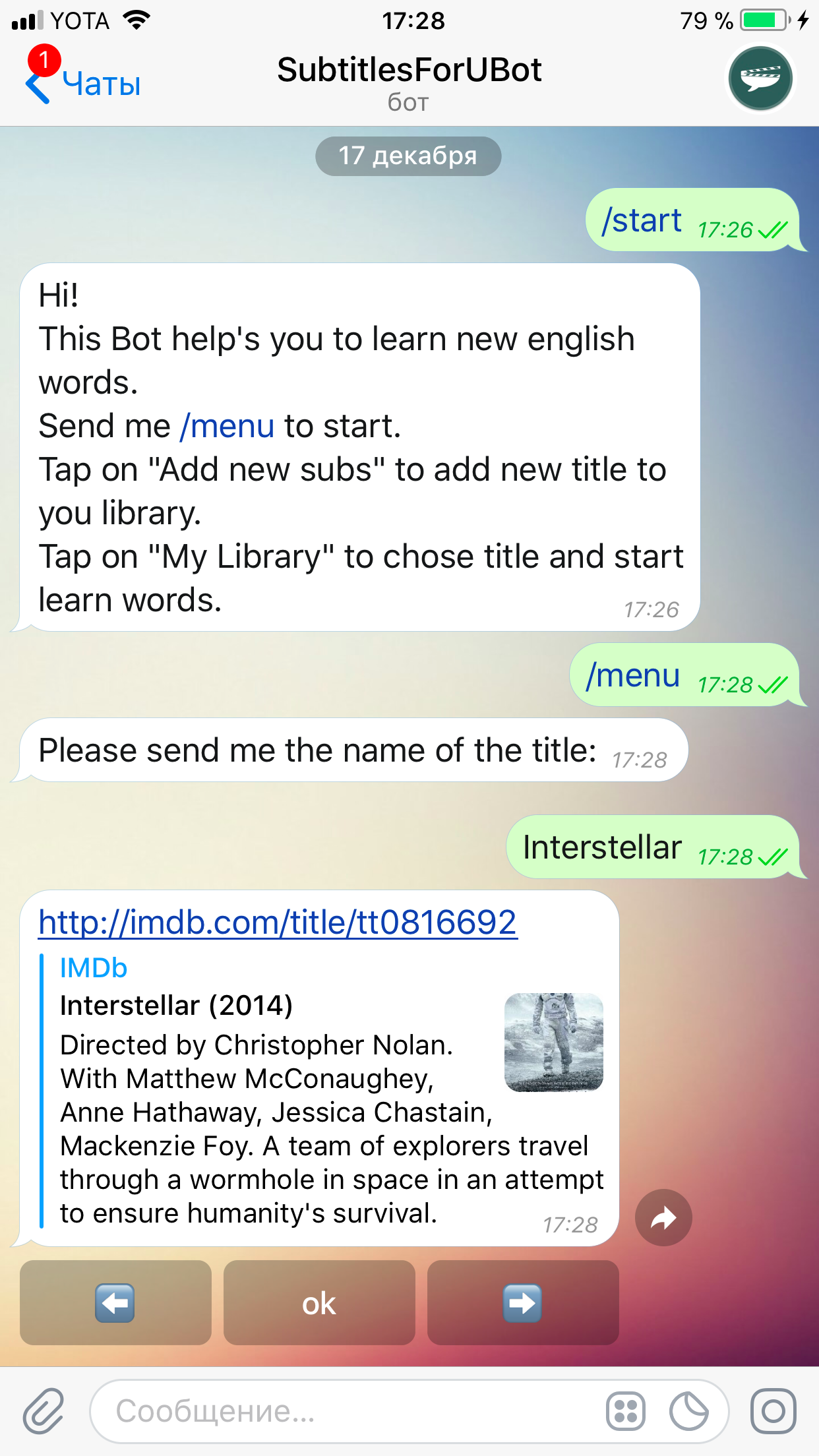
Рисунок 1 - Схема связи модулей системы

**Telegram бот**

Весь функционал бота доступен в чате с ним.

Для доступа к функционалу бота достаточно отправить /menu в чат.

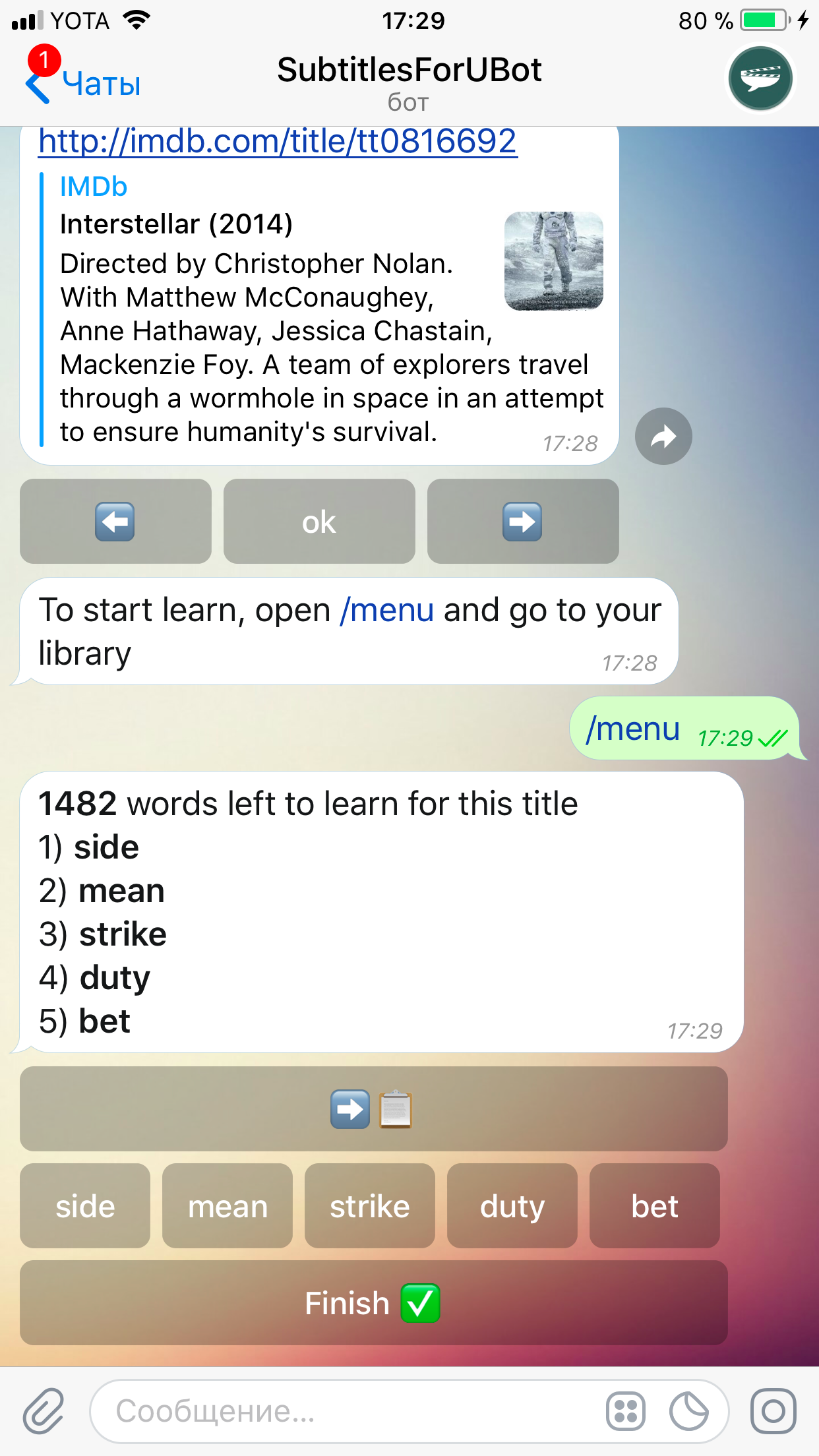
Изучение слов происходит для конкретного фильма или серии сериала. Поэтому для начала пользователь должен добавить фильма к себе в библиотеку.

Рисунок 2 - Экран добавления фильма в библиотеку

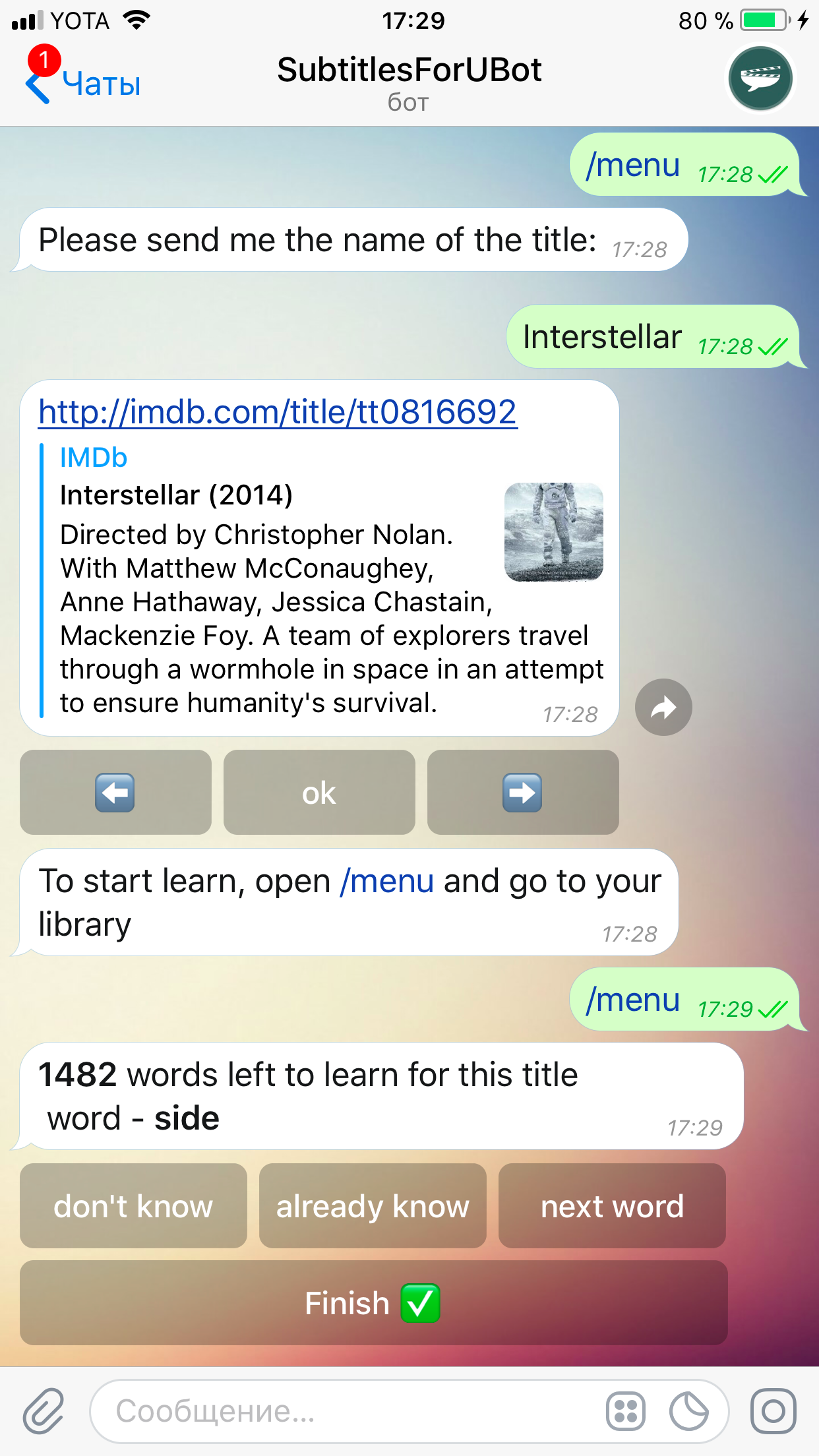
После успешного добавления, пользователю предлагается перейти в библиотеку, где хранятся все добавленные им серии и фильмы.

Бот предоставляет два способа изучения слов:

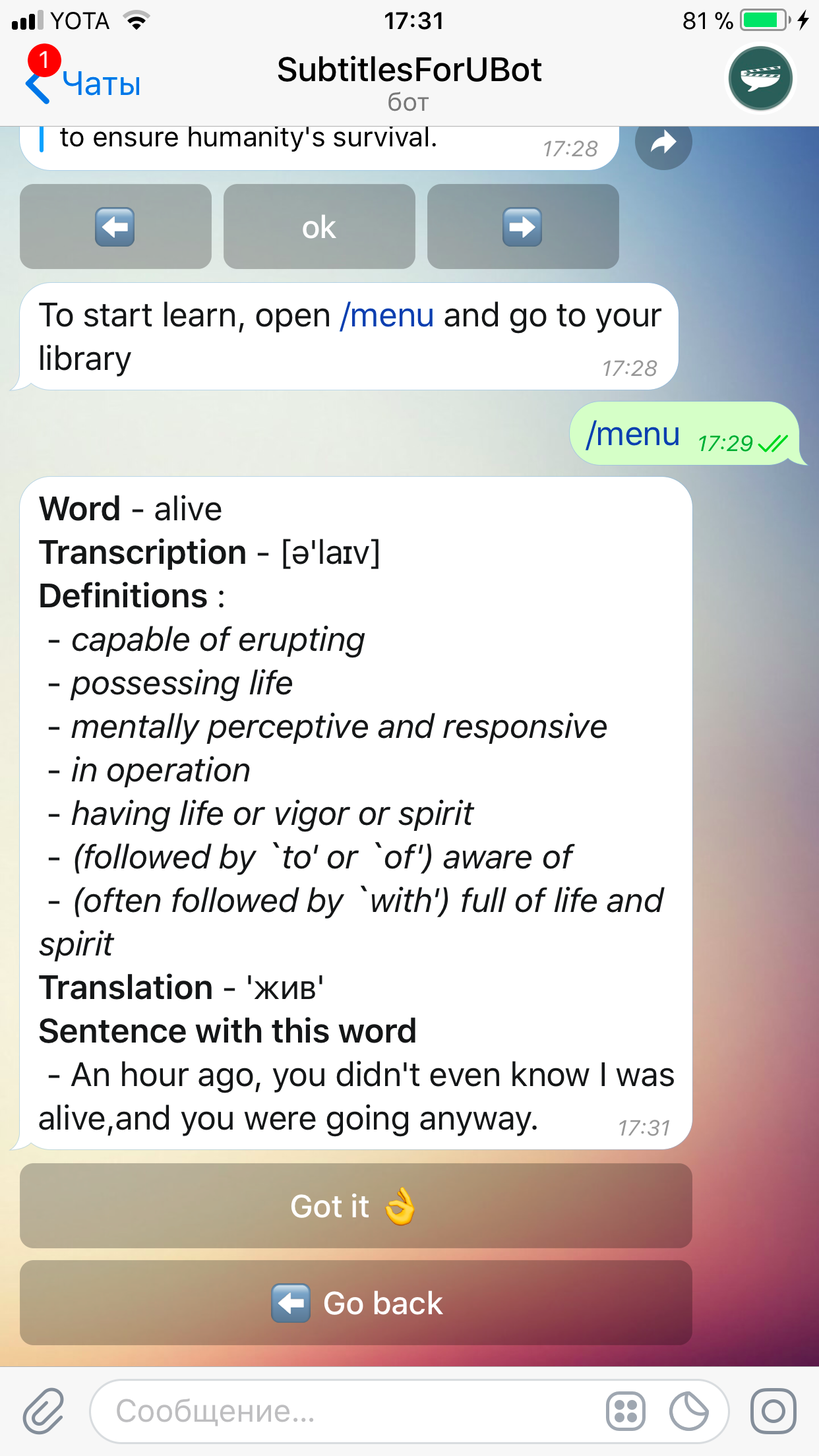
1. Списками по 5 слов

Рисунок 3 - Экран изучения слов списком

1. Слова по отдельности

Рисунок 4 - Экран изучение слов по отдельности

Выбрав определенное слово, пользователь получает карточку с транскрипцией, определениями, переводом и примером предложения для этого слова.

Рисунок 5 - Экран карточки со словом 

В данной карточке можно вернуться обратно к изучению слов, а также пометить слово как изученное. Сама карточка остается в чате, для того, чтобы пользователь мог быстро просмотреть историю сообщений в боте и вспомнить слова, которые он изучал.

**Сервер**

Бот располагается на облачной платформе Heroku, что обеспечивает его непрерывную работу.

На сервере запущен код на языке программирования “Python”.

Основным классом, описывающим работу телеграмм-бота, является класс SubsBot, в нем описаны методы взаимодействия с пользователем, методы поиска фильмов/сериалов и субтитров к ним.

Отдельными классами (OpenSubtitles, OMDB) предоставлены интерфейсы для работы с сайтом opensubtitles.org и imdb.com .

Также в отдельных файлах описаны функции для обработки скачанных субтитров, получения определений и перевода слов.

Для работы с базой данных написан отдельный класс DataBase.

**База данных**

В качестве базы данных используется СУБД PostgreSQL. БД располагается на Heroku.

В базе данных 5 таблиц:

1. Таблица words:

* id слова
* слово
* определение слова

1. Таблица user\_words

* id пользователя
* id слова

1. Таблица user\_films

* id пользователя
* id фильма

1. Таблица subtitle\_words

* id фильма
* id слова
* предложение с этим словом

1. Таблица subtitle\_imdb

* id субтитров
* id фильма
* сезон
* эпизод

Каждое слово имеет уникальный id, что позволяет экономить место в БД , достаточно единожды занести его в таблицу words, а в последствии использовать его id.

Рассмотрим как происходит взаимодействие с БД ,на примере добавления пользователем фильма .

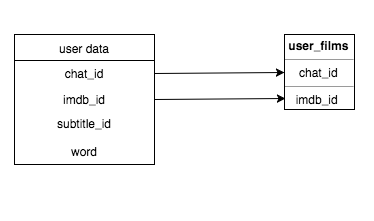
1. Если к данному фильму удается скачать субтитры , то происходит добавление фильма в библиотеку пользователя. 

Рисунок 6 - Схема добавления фильма в БД

1. Идет поиск фильма в таблице subtitle\_words , если его там нету, то слова записываются в эту таблицу. Для этого происходит вызов функции GetWordID(), которая, возвращает id слова из таблицы words.

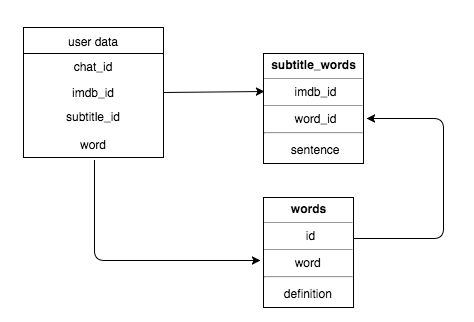


Рисунок 7 - Схема добавления слова в БД

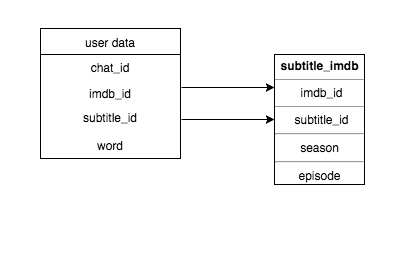
1. Id субтитров и id фильма в проверяются на вхождение в таблицу subtitle\_imdb и добавляются туда, в случае их отсутствия.

Рисунок 8 - Схема добавления id фильма и id субтитров в БД

**Скрипт**

Для простоты установки данного бота на сервер был написан shell script, скрипт, который устанавливает все необходимые библиотеки и создает таблицы для базы данных. Упрощая таким образом развертку бота на новом сервере.

# Выбор технологий

## **Выбор языка программирования**

Для курсового проекта при выборе языка программирования было произведено сравнение между двумя языками программирования: “Python” и “C++”.

“Python” - высокоуровневый интерпретируемый язык программирования, преимущества которого заключаются в следующих характеристиках:

* динамическая типизация
* автоматическое управление памятью
* простой синтаксис кода
* большое количество сторонних библиотек

Отрицательной чертой данного языка программирования является его низкая производительность.

“C++” - компилируемый, сильно типизированный язык программирования, плюсы которого проявляются в нижеперечисленных характеристиках:

* высокая производительность
* возможность работы на низком уровне с памятью, адресами, портами.
* кросс-платформенность

Из минусов стоит отметить:

* сложность привязывания библиотек.

В данном курсовом проекте необходимо, чтобы для языка программирования было предоставлено Telegram API для написания телеграмм-бота и его функционала. Помимо Telegram API, должны присутствовать библиотеки для работы с базой данных и для символьной и статистической обработки естественного языка. Следует отметить, что к проекту нет требований к производительности. Именно поэтому для данного курсового проекта был выбран язык программирования – «Python».

## **Выбор используемых библиотек**

Далее приведен список используемых библиотек, необходимых для реализации телеграмм-бота:

* PostgreSQL - объектно-реляционная база данных;
* Psycopg2 - библиотека для «Python», предоставляющая интерфейс для работы с базой данных PostgreSQL, выполнения SQL-запросов к базе данных;
* Python-telegram-bot - библиотека для «Python» предоставляемая API для работы с Telegramm. С помощью этой библиотеки реализован телеграмм-бот. У библиотеки имеется подробная документация;
* NLTK - пакет библиотек и программ для символьной и статистической обработки естественного языка, написанных на языке программирования «Python»;
* Requests - библиотека для «Python», предоставляющая удобный интерфейс для работы с http запросами;
* Omdb - библиотека для работы с OMDb API, API для сайта imdb.com;
* IMDbPY - представляет собой библиотеку для «Python», предоставляющей интерфейс для работы с базой данных IMDb .

# Описание технический решений

Telegram бот размещен на облачной платформе Heroku. После его создания, следует настроить автоматическое развертывание.

Созданный проект подключается к существующему git-репозиторию с сервером. Для этого выбирается метод развертывания GitHub.

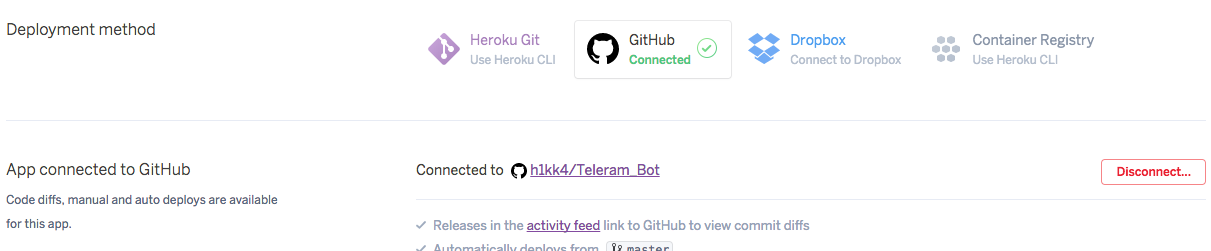
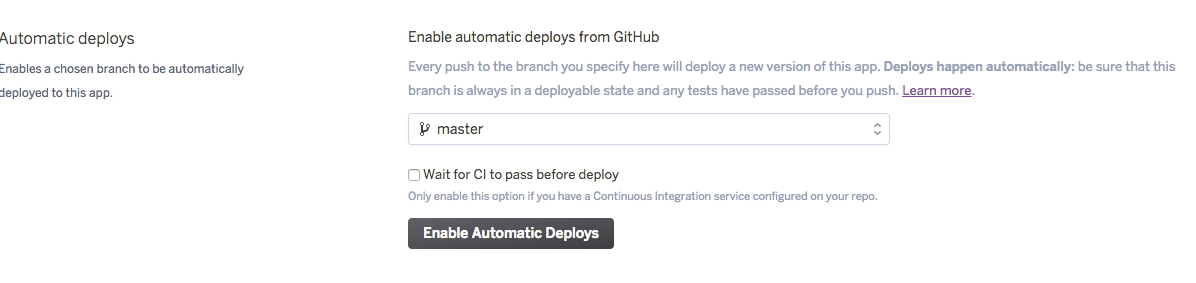


Рисунок 9 - Подключение к существующему репозиторию

Для настройки автоматического развертывания с git-репозитория следует нажать ‘Enable Automatic Deploys’.

Рисунок 10 - Подключение автоматического развертывания

Таким образом, после любого изменения в репозитории на GitHub проект будет автоматически обновляться на Heroku.

В процессе разработки и тестирования проекта выявлялись различные проблемы.

Первая проблема проявлялась при одновременной использовании telegram бота несколькими людьми. Команды, вызванные одним пользователем могли принимать сообщения от другого пользователя. В сложившейся ситуации была создана структура, которая хранит флаги для корректной работы команд. Структура данных флагов связана с идентификатором чата, которые, в свою очередь, являются уникальными.

Следующей проблемой, с которой пришлось столкнуться, является нестабильная работа API ​opensubtitles.org. Метод “SearchMoviesOnIMDB” , который должен возвращать id фильма с сайта imdb.com, стал некорректно работать. Причиной этому явились проблемы на самом сайте. Было решено отказаться от поиска через API от ​opensubtitles.org и воспользоваться методом поиска через библиотеку IMDbPY. При переходе на IMDbPY также стал доступен поиск фильмов на русском языке.

Во время поиска субтитров к серии сериала так же возникли проблемы. API ​opensubtitles.org при запросе не выдавало полного списка всех доступных субтитров к сериями сериала. Тогда было решено для начала находить id серии с сайта imdb.com, а уже после искать субтитры. Упомянутая выше библиотека IMDbPY не годится для поиска id серий к сериалу, так как способна возвращать лишь полный список серий к сериалам, вследствии чего работает слишком медленно. Для решения данной проблемы было решено воспользоваться библиотекой omdb.

# Заключение

В процессе выполнения курсового проекта был реализован Телеграмм-бот, помогающий пользователям изучать незнакомые им иностранные слова к интересующих их субтитрам. Телеграмм-бот был успешно запущен на сервисе Heroku. В дальнейшем планируется развивать и улучшать проект. Будет добавлена поддержка других языков и добавлены тесты, для повторения изученных пользователем слов.

# 

# Список используемой литературы

1. Грабер, М. Введение в SQL/М. Грабер - М.:ЛОРИ, 1996. - 375 с.
2. API for opensubtitles.org [Электронный ресурс] Режим доступа: http://trac.opensubtitles.org/projects/opensubtitles/wiki/XMLRPC, свободный
3. Python Telegram Bot’s documentation [Электронный ресурс] Режим доступа: https://python-telegram-bot.readthedocs.io/en/stable/, свободный

# Natural Language Toolkit [Электронный ресурс] Режим доступа: http://www.nltk.org/api/nltk.html, свободный

1. IMDbPY documentattion [Электроннный ресурс] Режим доспупа: https://github.com/alberanid/imdbpy/tree/master/docs, свободный
2. OMDB documentation [Электронный ресурс] Режим доступа:

https://media.readthedocs.org/pdf/omdbpy/latest/omdbpy.pdf, свободный