

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Космических и Информационных технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ИС
_____ П. П. Дьячук
подпись инициалы, фамилия
« _____ » _____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – “Информационные системы и технологии”

Чат-бот СФУ

Руководитель

подпись, дата

доцент, к.т.н.

должность, учёная степень

И. А. Легалов

инициалы, фамилия

Выпускник

подпись, дата

Д. А. Макушев

инициалы, фамилия

Красноярск 2020

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Космических и Информационных Технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ИС
_____ П. П.
Дьячук
подпись инициалы, фамилия
« _____ » _____ 2020 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Макушеву Данилу Андреевичу

Группа: КИ16-14Б Направление: 09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: «Чат-бот СФУ»

Утверждена приказом по университету № 6499/с от 22.05.2020 г.

Руководитель ВКР: И.А. Легалов, к.т.н., доцент кафедры «Информационные системы» ИКИТ СФУ.

Исходные данные для ВКР: Требования к разрабатываемой системе, методические указания научного руководителя, учебные пособия.

Перечень разделов для ВКР: Введение, анализ предметной области, выбор средств и среды разработки, разработка чат-бота, заключение, список использованных источников, приложение.

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2013.

Руководитель ВКР

подпись, дата

И. А. Легалов

инициалы и фамилия

Задание принял к исполнению

подпись, дата

Д.А. Макушев

инициалы и фамилия студента

« ____ » _____ 2020 г

РЕФЕРАТ

Выпускная квалификационная работа по теме «Чат-бот СФУ» содержит 40 страниц текстового документа, 25 иллюстраций, 19 использованных источников.

ЧАТ-БОТ, TELEGRAM, МЕССЕНДЖЕР, СФУ, PYTHON

Актуальность.

В наше время стали очень популярны мессенджеры, и чат-боты в том числе. Чат-бот – специальный алгоритм, формирующий переписку с настоящим собеседником. Чат-боты помогают и упрощают жизнь, выполняя однотипные задачи.

Объектом исследования является чат-боты.

Предметом исследования являются функции и задачи чат-ботов.

Целью работы является разработка чат-бота СФУ для студентов и абитуриентов.

Основные задачи:

- анализ предметной области;
- выбор наполнения системы;
- реализация чат-бота;
- отладка работы чат-бота;
- внедрение чат-бота.

Практическая значимость проекта заключается в том, что с помощью данного сервиса можно найти нужную информацию быстрее и удобней, чем на других ресурсах.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Анализ предметной области	5
1.1 Мессенджеры.....	5
1.2 Чат-боты.....	9
1.3 Примеры чат-ботов	10
1.3.1 Чат-бот BotFather	11
1.3.2 Чат-бот TranslateBot.....	12
Выводы по первой главе.....	14
2 Выбор средств и среды разработки	15
2.1 Технологии разработки	15
2.1.1 Python	15
2.1.2 Telegram Bot API.....	16
2.1.3 Heroku.....	17
2.2 Среда разработки PyCharm	18
Выводы по второй главе.....	19
3 Разработка чат-бота.....	20
3.1 Регистрация чат-бота	20
3.2 Реализация главного меню и навигации.....	22
3.3 Реализация парсинга информации с сайта СФУ	26
3.4 Размещение бота на сервере Heroku	32
Выводы по третьей главе.....	34
ЗАКЛЮЧЕНИЕ	35
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	36
ПРИЛОЖЕНИЕ А	39
ПРИЛОЖЕНИЕ Б.....	40

ВВЕДЕНИЕ

В настоящее время основой всего стали информационные технологии, а в частности интернет. Сейчас уже трудно представить какую-либо сферу человеческой деятельности без использования компьютера или баз данных. Все завязано на использовании вычислительной техники и трудно уже представить нашу жизнь без всего этого. Общение тоже в свою очередь перешло из реального мира в виртуальный, мы каждый день пользуемся социальными сетями, мессенджерами и другими приложениями для общения с людьми. Это стало важным компонентом повседневной жизни.

Актуальность выпускной квалификационной работы обусловлена большой популярностью мессенджеров, и в частности чат-ботов. Чат-бот – специальный алгоритм, формирующий переписку с настоящим собеседником. Чат-боты помогают и упрощают жизнь, выполняя однотипные задачи.

Целью выпускной квалификационной работы является разработка чат-бота СФУ для студентов и абитуриентов.

Исходя из поставленной цели можно выделить следующие задачи:

- анализ предметной области;
- выбор наполнения системы;
- реализация чат-бота;
- отладка работы чат-бота;
- внедрение чат-бота.

1 Анализ предметной области

1.1 Мессенджеры

Жизнь современного человека трудно представить без интернета. В нем люди получают ценные знания, которые используют в повседневной жизни, работают или учатся. Но основной функцией интернета является общение с другими людьми. Изначально, люди общались по средствам электронной почты. Далее этот процесс эволюционировал, и теперь для общения существуют социальные сети и мессенджеры. Именно мессенджеры сейчас занимают первое место [1]. Многие исследования утверждают, что люди чаще общаются в мессенджерах, чем в других сервисах.

Самые популярные мессенджеры в мире – это Telegram, WhatsApp Messenger и Viber [2]. У всех примерно одинаковый функционал: отправка текстовых и голосовых сообщений, медиафайлов, документов, а также аудио- и видеоконференции.

Из всех мессенджеров можно выделить Telegram. Это бесплатный кроссплатформенный мессенджер для обмена текстовыми, голосовыми и видео сообщениями [3].

Из плюсов данного мессенджера можно выделить:

- шифрование всех сообщений;
- секретные чаты и самоуничтожающиеся сообщения;
- поддержка всех платформ;
- скорость передачи сообщения;
- возможность использовать чат-боты;
- встроенный редактор изображений;
- групповые конференции до 200 человек.

Именно благодаря шифрованию сообщений Telegram пользуется такой популярностью. Мессенджер использует технологию шифрования MTProto, которая включает в себя несколько протоколов шифрования. Принцип данного шифрования представлен на рисунке 1.



Рисунок 1 – Принцип работы протокола MTProto

Telegram был создан Павлом Дуровым, совместно с его братом Николаем Дуровым. Данный протокол также разработан Павлом и Николаем. 14 августа 2013 года появился первый клиент Telegram на платформе IOS. Сейчас Telegram представлен на всех популярных платформах: Android OS, IOS, Windows Phone, Windows, Linux и MacOS.

Сейчас аудитория Telegram составляет более 400 млн человек в мире [4] и более 10 млн человек в России. В нашей стране ежедневно мессенджером пользуются более 4,5 млн человек [5]. Что касается возраста, наиболее многочисленная группа пользователей - это люди в возрасте от 25 до 34 – 38% и от 18 до 24 – 27%. Так же с каждым годом увеличивается количество

пользователей более возрастной категории [6]. В основном пользователи мессенджера работают в сфере IT, маркетинга и пиара. Основные цели использования Telegram – личная и бизнес-переписки, а также использование ботов. На рисунке 2 приведены результаты исследования компании TGStat по целям использования Telegram в России.



Рисунок 2 – Основные цели использования Telegram

Интерфейс клиента для ПК состоит из списка чатов, который находится слева, выбранного чата в центре и информации о чате в правой части окна. Интерфейс работы клиента Telegram показан на рисунке 3.

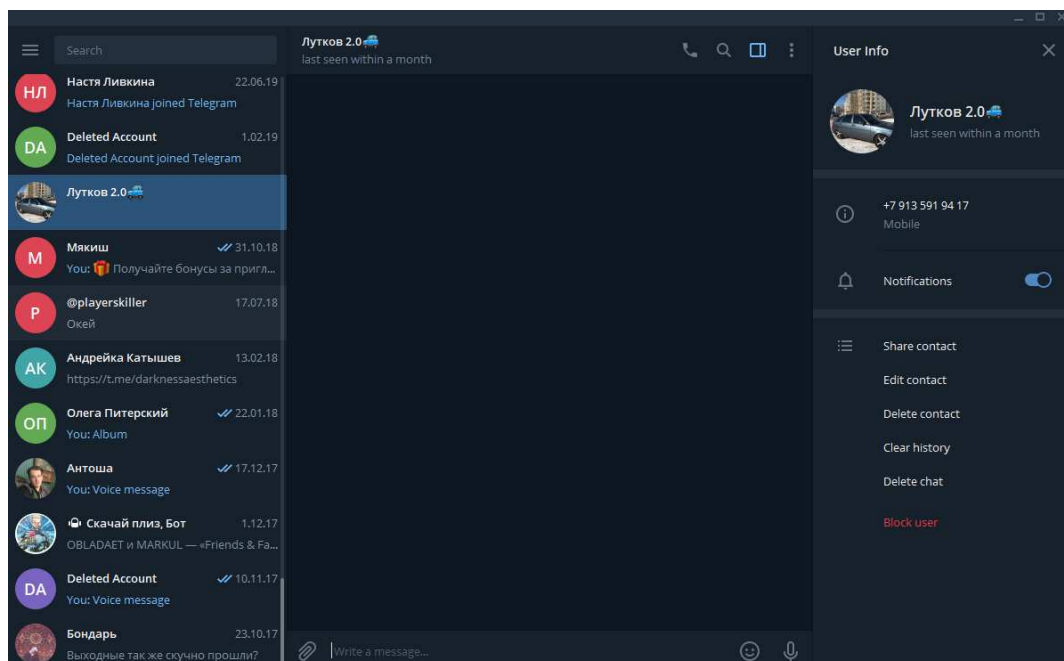


Рисунок 3 – Интерфейс клиента Telegram для ПК

В мобильной версии Telegram список чатов, выбранный чат и информация о чате находятся на разных экранах. Это обусловлено размерами и ориентацией экрана. Вид мобильной версии показан на рисунке 4.

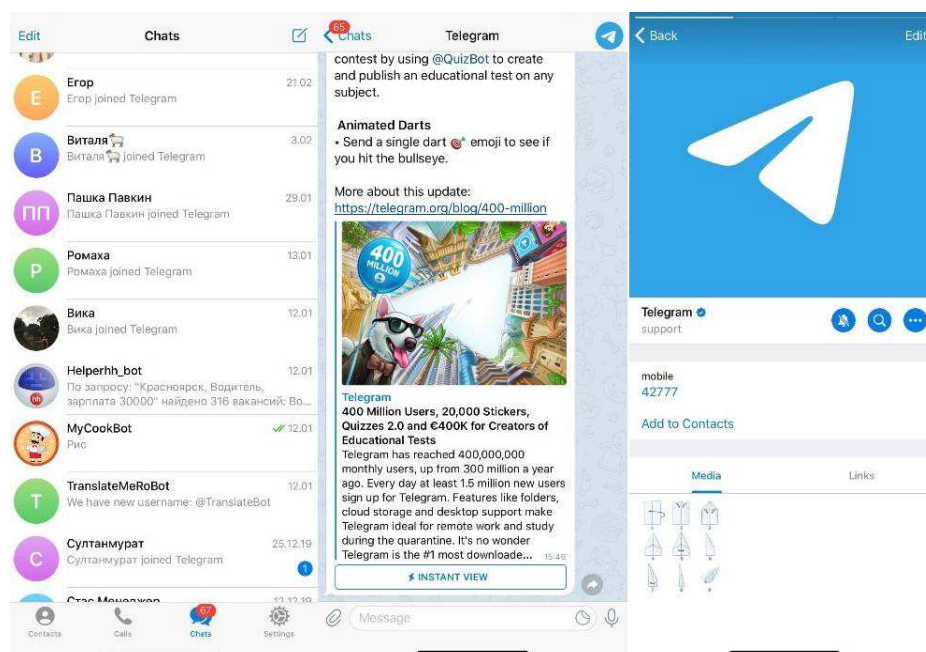
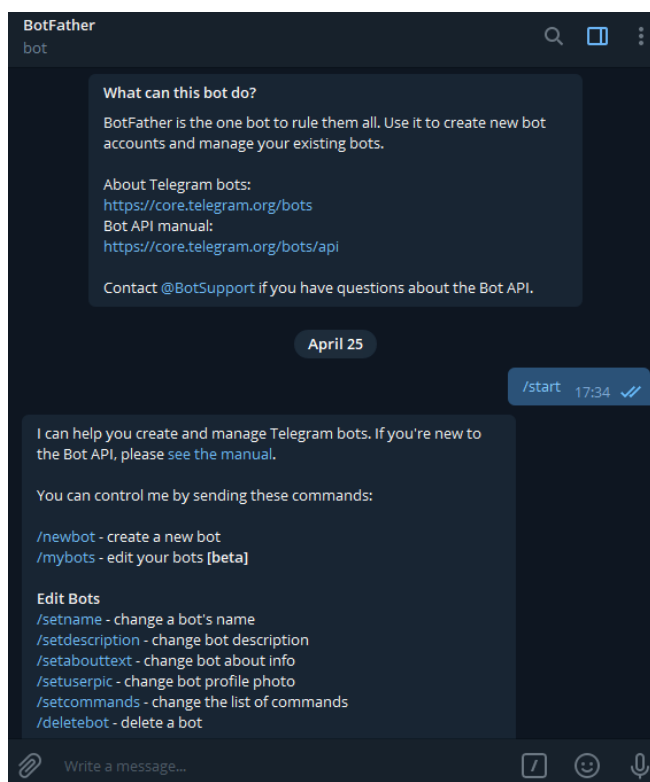


Рисунок 4 – Интерфейс клиента Telegram на IOS 13

1.2 Чат-боты

Чат-бот – программа, имитирующая разговор с пользователем [7]. Она отвечает на вопросы собеседника с помощью текстовых или голосовых сообщений. Чат-бот работает с помощью искусственного интеллекта [8] или заранее запрограммированных сценариев поведения собеседника при общении с ботом.

В мессенджере Telegram чат-бот может быть представлен с помощью кнопок навигации. Пользователь может сам выбрать, что ему нужно узнать от чат-бота [9]. Программа распознает написанные сообщения пользователя, а также определенные команды. Синтаксис команды состоит из знака «/» и текста самой командой. Бот считывает команду и выполняет действие, относящееся к этой команде. Также после написания знака «/» бот сам выводит доступные команды. Пример работы команд в чат-боте продемонстрирован рисунком 5.



По результатам исследования “State of chatbots 2018” [10] выявлено, что чат-боты очень эффективны для бизнеса. 64% опрошенных людей сказали, что круглосуточный сервис является важным преимуществом чат-ботов. Пользователь в любой момент времени может воспользоваться чат-ботом и узнать нужную ему информацию. Также 69% пользователей ответили, что для взаимодействия с компанией чаще выбирают чат-ботов, чем использование приложения.

Рассмотрим преимущества использования чат-ботов:

- эффективное взаимодействие с клиентами. Чат-боты привлекают новых пользователей и упрощают общение с действующими. Бот предоставляет только нужные данные, которые запросил пользователь, поэтому клиенты получают только релевантную информацию;
- экономичность. Чат-бот сокращает траты компании на персонал. Программу можно интегрировать для ответов на часто задаваемые вопросы, что способствует уменьшению трафика на службы поддержки клиентов;
- отслеживание данных о пользователях. Чат-бот может собирать статистику пользователей и их предпочтения для последующей аналитики. Это позволяет сегментировать аудиторию для рассылки полезной информации;
- удобство пользования. Чат-бот можно интегрировать практически в любой мессенджер, который чаще используют пользователи. Им не нужно разбираться в том, как работает данная программа.

1.3 Примеры чат-ботов

Для создания чат-бота необходимо рассмотреть существующие чат-боты. Для анализа были выбраны популярные чат-боты Telegram и разобраны их преимущества и недостатки.

1.3.1 Чат-бот BotFather

Любой чат-бот в Telegram начинается именно с BotFather. Данный бот был создан разработчиками мессенджера Telegram.

Цель бота состоит в том, чтобы сделать создание и управление чат-ботами максимально простыми. Любой пользователь может создать чат-бота, написав BotFather команду «/newbot». Далее с помощью других команд пользователь может настроить своего бота, добавить описание, фотографию или др.

Также через этот чат-бот можно получить токен для доступа к своему чат-боту. Далее этот токен нужно вписать в программный код или вставить в конструктор бота. Работа с чат-ботом BotFather показана на рисунке 6.

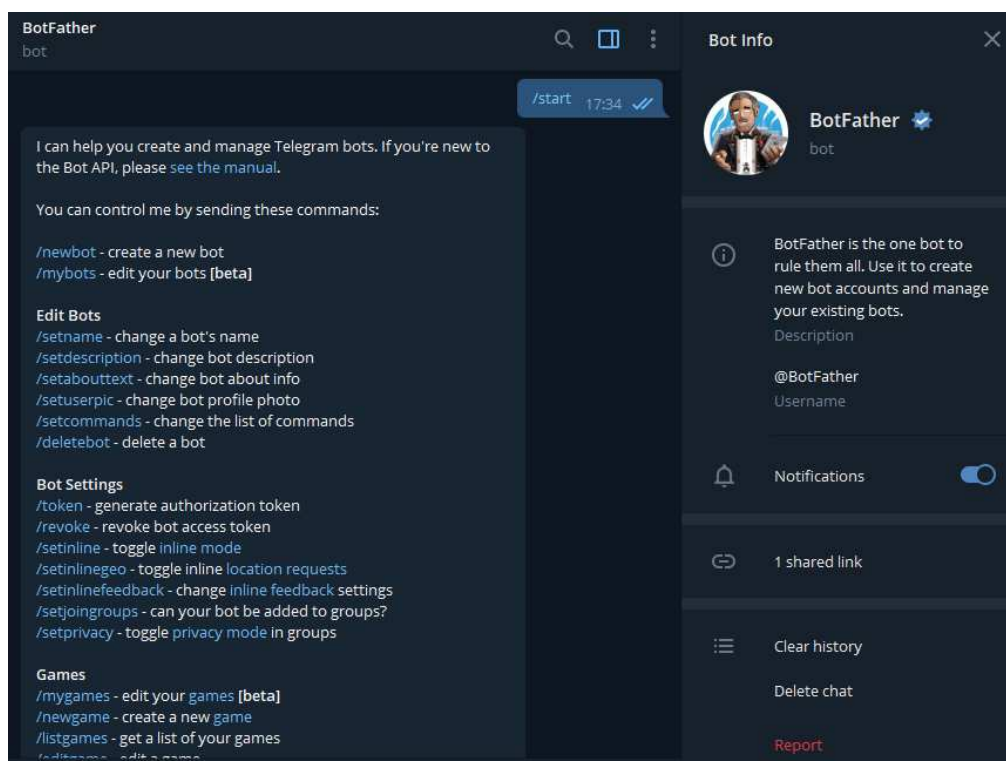


Рисунок 6 – Чат-бот BotFather

Из плюсов можно отметить большой набор команд и широкий функционал действия.

Минусом является то, что бот поддерживает только английский язык.

1.3.2 Чат-бот TranslateBot

Чат-бот TranslateBot позволяет мгновенно переводить слова. Данный бот поддерживает 22 языка перевода и использует на выбор движки от Google и Yandex.

В начале пользователь выбирает используемые языки и движок, на котором нужно выполнить перевод, а затем отправляет нужное слово или словосочетание. Чат-бот отправляет голосовое сообщение с произношением данного слова, если пользователь хочет услышать, как читается перевод. Работа TranslateBot проиллюстрирована рисунком 7.

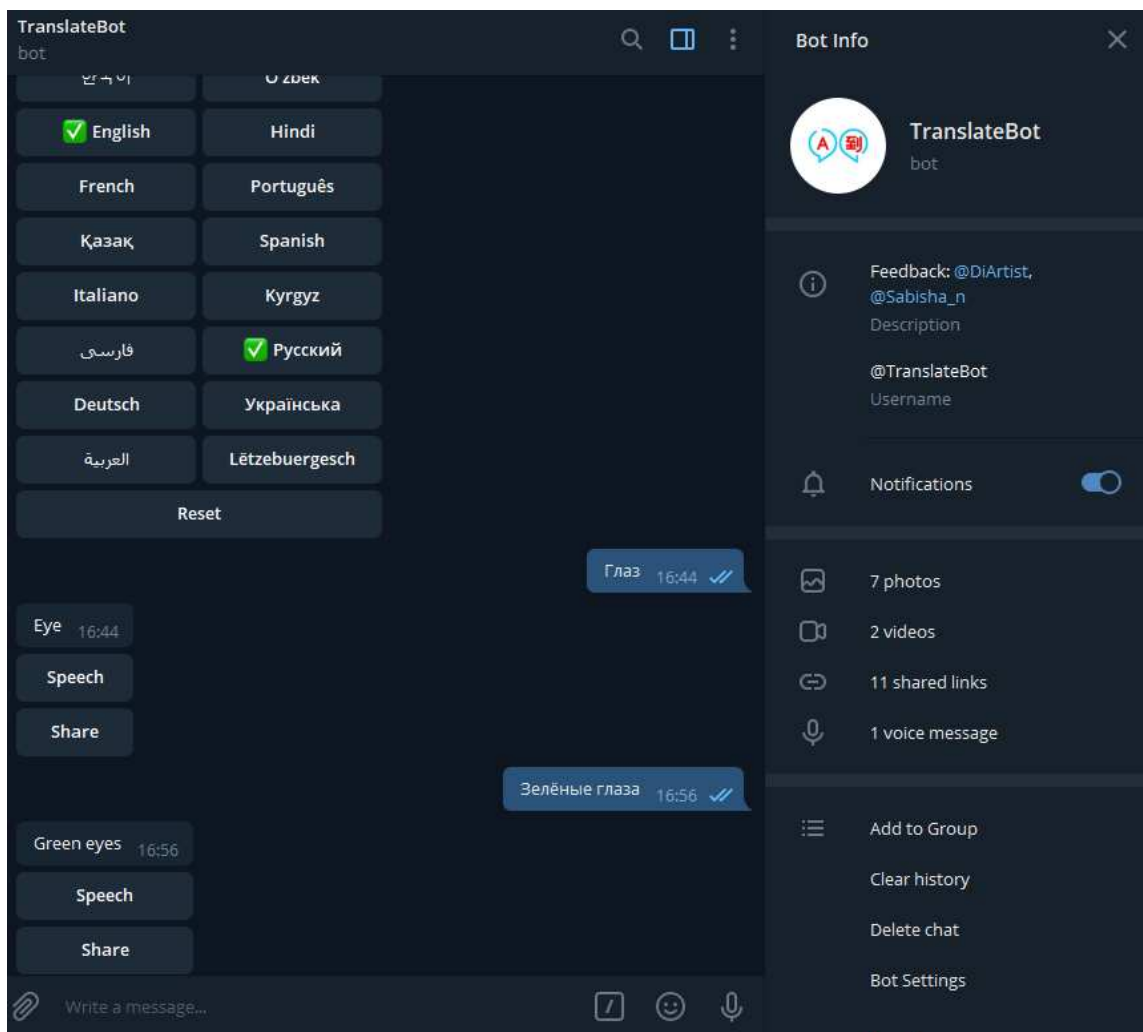


Рисунок 7 – Чат-бот TranslateBot

Плюсами данного чат-бота можно назвать:

- большое количество поддерживаемых языков;
- выбор движка перевода;
- дополнительная настройка;
- произношение перевода.

Минусами чат-бота являются:

- ошибки в связи между слов в больших предложениях;
- отсутствие узкоспециализированной терминологии.

Выводы по первой главе

В рамках анализа предметной области были разобраны существующие мессенджеры, подробно изучен мессенджер Telegram. Изучены составляющие чат-ботов и порядок их создания. Проанализированы примеры чат-ботов и выделены их плюсы и минусы. Выявлены требования будущего чат-бота. Система должна иметь:

- несколько поддерживаемых языков;
- большое количество команд;
- широкий функционал.

2 Выбор средств и среды разработки

2.1 Технологии разработки

2.1.1 Python

Python является высокоуровневым языком программирования, который был разработан в 1991 году Гвидо ван Россумом. Этот язык предлагает динамическое управление типами данных и поддерживает различные парадигмы программирования, включая объектно-ориентированную, процедурную или функциональную. В 2018 году популярность Python возросла, он стал одним из самых популярных языков [11]. Практически во всех рейтингах он входит в тройку лучших языков программирования.

Python был разработан как проект с открытым кодом, который предлагает бесплатные установочные пакеты для всех популярных платформ, таких как Unix, Windows, macOS, Android.

На данный момент существует 3 версии Python, из которых поддерживается только последняя третья версия. Версия Python 3.0 была выпущена в 2008 году.

Python имеет ряд достоинств:

- кроссплатформенный;
- простой и понятный синтаксис. Сложные программы можно записывать кратко, но в то же время понятно;
- легкий в освоении;
- большое количество поддерживаемых библиотек.

Для работы с Python необходимо установить менеджер пакетов PIP. PIP – это стандартная система управления пакетами, используемая для установки и управления программными пакетами. Преимуществом PIP является простота

использования. Для установки любого пакета необходимо в командной строке написать команду «pip install» и далее указать название пакета. Для обновления пакета необходимо к команде установки добавить «--upgrade». После чего нужный пакет обновится.

В качестве примера на рисунке 8 изображен фрагмент кода на языке Python.

```
print("Hello world!") # Hello World!
```

Рисунок 8 – Пример кода на языке Python

2.1.2 Telegram Bot API

Все Боты являются отдельными аккаунтами в мессенджере [12], которые автоматически обрабатывают отправленные им сообщения. Управлять ботами можно с помощью Telegram Bot API, используя HTTPS-запросы [13]. Принцип работы представлен на рисунке 9.

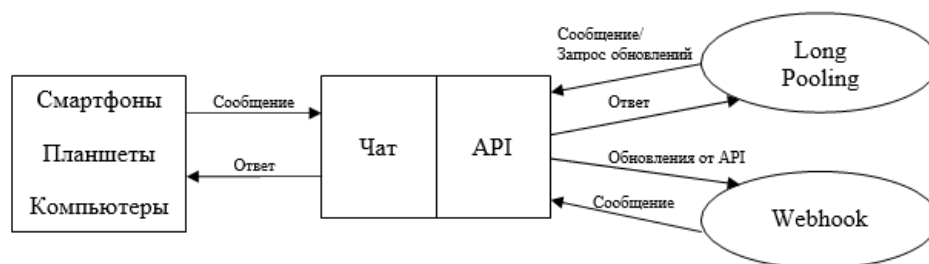


Рисунок 9 – Принцип работы чат-бота

Long Pooling и Webhook – методы получения обновлений от бота [14]. Первый метод заключается в том, что приложение опрашивает сервера Telegram на наличие каких-либо обновлений. Во втором методе сервера сами оповещают приложение о новых обновлениях, как только они появляются.

Для управления ботом посредством Bot API необходимо получить токен. Это уникальный ключ доступа к чат-боту [15]. Токен можно получить в чат-боте BotFather, который создан специально для создания и управления чат-ботами Telegram.

2.1.3 Heroku

Heroku – облачная платформа PaaS, поддерживающая множество языков программирования [16]. Она считается одной из первых облачных платформ, ее разработка началась в 2007 году. Heroku поддерживает такие языки программирования как Java, Node.js, Scala, Closure, Python, PHP, Ruby и GO. Данная облачная платформа позволяет создавать и масштабировать приложения одинаково на всех языках.

У каждого приложения, которые размещены на Heroku, имеется свое доменное имя «имя_приложения.herokuapp.com», которое используется для маршрутизации HTTP-запросов.

Преимущества использования Heroku:

- легкое начало. Он идеально подходит для небольших проектов и стартапов;
- масштабируемость;
- использование стандартных инструментов.

Основным блоком, который обеспечивает среду выполнения кода являются контейнеры `duo` – изолированные виртуализированные контейнеры UNIX. Они распределены по специальной виртуальной сетке, которая состоит из нескольких серверов [17]. Создание этих контейнеров происходит из числа выполненных контейнеров, разделенных на отдельные процессы, которые масштабируются разработчиком.

2.2 Среда разработки PyCharm

PyCharm – редактор кода, поддерживающий множество языков программирования, таких как Python, JavaScript, CoffeeScript, TypeScript, CSS и других популярных языков. PyCharm также способен проверять код на наличие ошибок и быстро исправлять их. Кроме всего этого, он имеет отладчик с графическим интерфейсом специально для Python. Благодаря отладчику есть возможность проводить тестирование с поддержкой кода.

С помощью PyCharm можно работать с различными базами данных прямо из приложения. Пользователь может редактировать код SQL, выполнять запросы, просматривать данные и вносить свои изменения в уже существующие схемы.

Кроме всего, PyCharm предоставляет поддержку различных фреймворков для веб-разработки от Python, такие как Django, Google App Engine, Pyramid и web2py.

С помощью PyCharm разработка кода становится очень продуктивной, благодаря встроенным функциям мгновенного поиска ошибок и анализа кода. Также отличной функцией является функция автодополнения, предлагающая варианты продолжения нужных команд и фраз при написании кода.

При запуске консоли в Python REPL в PyCharm появляется возможность мгновенной проверки синтаксиса с дополнительными проверками скобок и кавычек.

Среди всех преимуществ можно выделить следующие:

- удобный интерфейс;
- автодополнение;
- проверка синтаксиса;
- простая организация проектов.

Именно благодаря данным функциям выбор пал именно на IDE PyCharm.

На рисунке 10 представлен интерфейс IDE PyCharm.

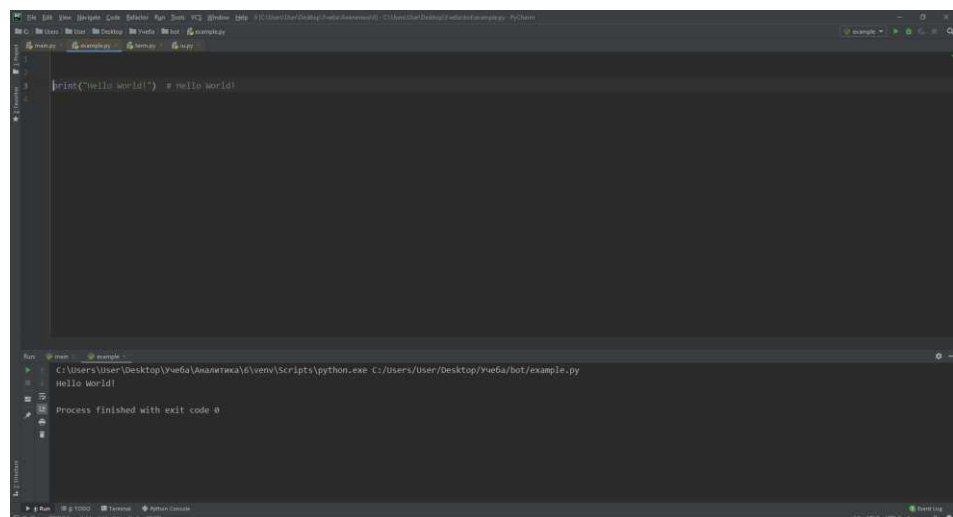


Рисунок 10 – Интерфейс IDE PyCharm

Выводы по второй главе

Разобраны преимущества языка программирования Python, IDE PyCharm, а также изучен Telegram Bot API и облачный сервис Heroku.

3 Разработка чат-бота

3.1 Регистрация чат-бота

Самым первым шагом разработки чат-бота мессенджера Telegram является регистрация бота в системе и получение токена для привязки чат-бота. Регистрация проходит в месенджере. Для создания нового бота нужно отправить команду «/newbot» чат-боту «BotFather». Этот бот создан специально для регистрации и управления всеми чат-ботами Telegram. На рисунке 11 показан процесс регистрации чат-бота.

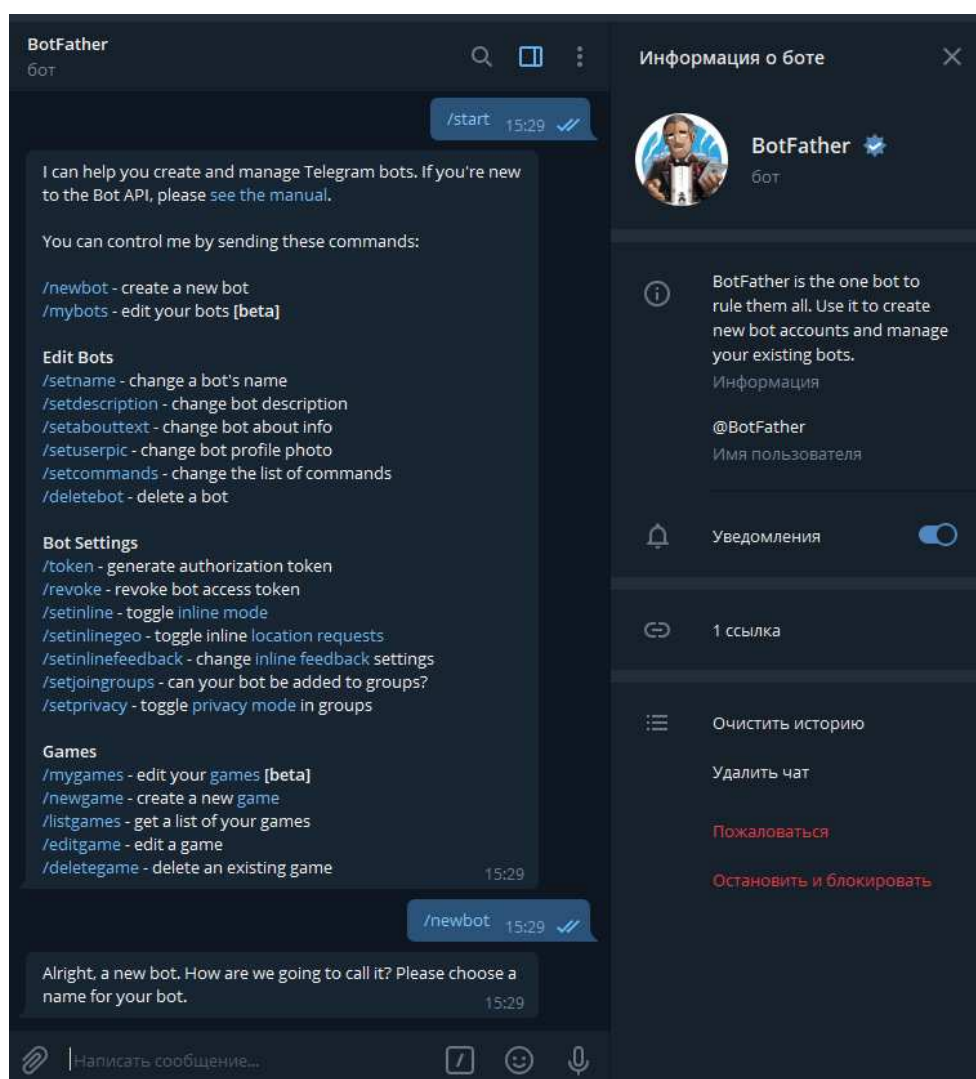


Рисунок 11 – Регистрация чат-бота

Далее нужно ввести название будущего бота. Название должно оканчиваться на слова «Bot» «_bot». После успешной регистрации «BotFather» выдает токен вашего чат-бота. Токен – это уникальный ключ доступа к вашему чат-боту. Данный токен указывается в коде нашего чат-бота.

Для настройки чат-ботов в «BotFather» используются следующие команды:

- /setname – изменение имени бота;
- /setdescription – изменение описания бота;
- /setabouttext – изменение информации о боте;
- /setuserpic – изменение фотографии бота;
- /setcommands – изменение списка команд бота;
- /setjoingroups – изменение настроек добавление бота в группы;
- /setprivacy – изменение настроек видимости сообщений в группах;
- /deletebot – удаление бота.

Кроме команд изменения настроек чат-бота, существуют команды, позволяющие изменять параметры и присваивать нужные значения:

- /token – отправка полученного ранее токена;
- /revoke – аннулирование токена доступа к боту;
- /setinline – настройка возможности вызова бота из других чатов;
- /setinlinegeo – настройка возможности передачи местоположения;

Данные настройки можно менять в любое время. Все параметры изменяются моментально у всех пользователей чат-бота. Результат настроек чат-бота продемонстрирован на рисунке 12.

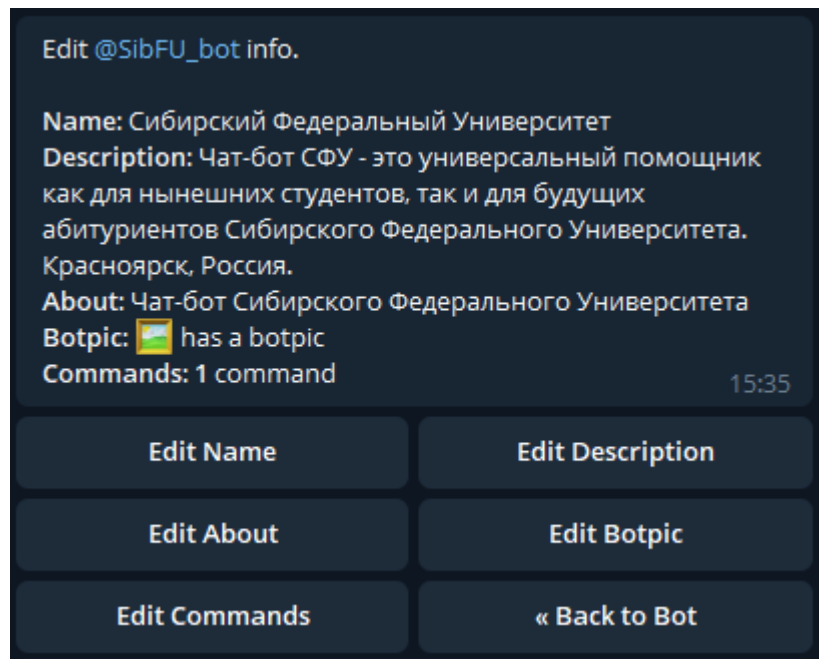


Рисунок 12 – Настройки чат-бота в «BotFather»

После регистрации и настройки чат-бота можно приступить к разработке программного кода чат-бота.

3.2 Реализация главного меню и навигации

Для удобства пользования чат-ботом используется главное меню в виде кнопок. Пользователи также могут перемещаться между функциями чат-бота через команды из списка команд. Все команды дублируют кнопки и вызывают те же самые функции [18].

Главное меню содержит следующие разделы:

- раздел для студентов;
- поступление;
- новости;
- расписание;
- другие соцсети;

- смена языка;
- сайт СФУ.

Главное меню изображено на рисунке 13.

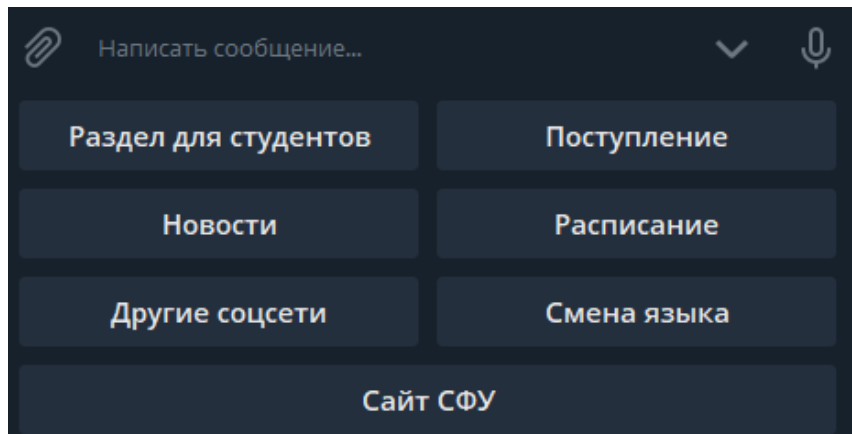


Рисунок 13 – Главное меню чат-бота

Главное меню чат-бота представляет собой кастомную клавиатуру. Данная клавиатура реализуется с помощью объекта ReplyKeyboardMarkup. Каждая клавиатура, созданная таким способом, привязана к сообщению чат-бота, с которым была отправлена. Реализация данной клавиатуры представлена на рисунке 14.

```
@bot.message_handler(commands=['menu'])
def menu(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    bth1 = types.KeyboardButton('Раздел для студентов')
    bth2 = types.KeyboardButton('Поступление')
    bth3 = types.KeyboardButton('Новости')
    bth4 = types.KeyboardButton('Расписание')
    bth5 = types.KeyboardButton('Другие соцсети')
    bth6 = types.KeyboardButton('Смена языка \n\u0001F1F7 \u0001F1FA \u0001F1FA \u0001F1F8')
    bth7 = types.KeyboardButton('Сайт СФУ')
    markup.add(bth1, bth2, bth3, bth4, bth5, bth6, bth7)
    final_message = 'Что теперь тебя интересует?'
    bot.send_message(message.chat.id, final_message, reply_markup=markup)
```

Рисунок 14 – Реализация главного меню чат-бота

При первом запуске чат-бота пользователю предлагается выбрать язык. В дальнейшем, пользователь может изменить язык в главном меню. Меню выбора языков показано на рисунке 15. После выбора нужного языка открывается главное меню чат-бота.

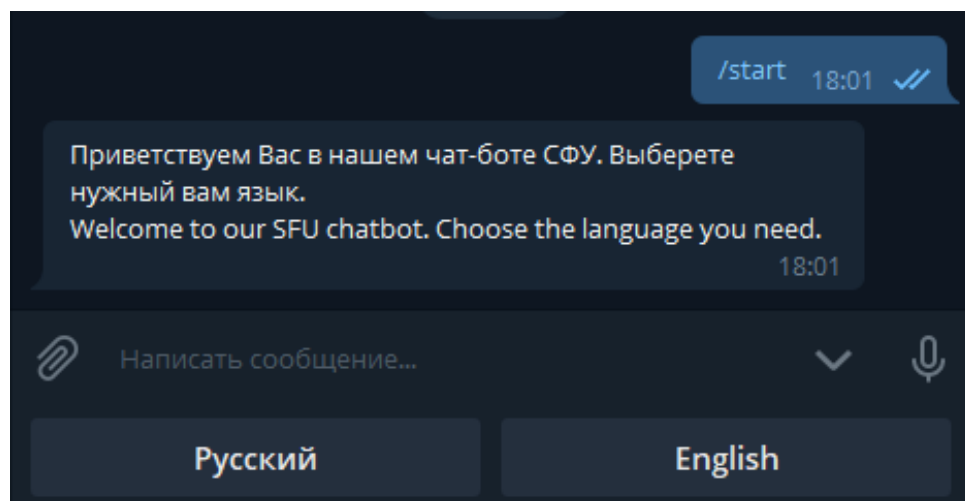


Рисунок 15 – Меню выбора языков

Следующим шагом создания чат-бота была разработка второго уровня меню чат-бота и навигация между подразделами меню. Каждый раздел главного меню открывает подразделы с информацией. Для возврата в предыдущий раздел меню необходимо нажать на кнопку «Назад в главное меню».

Раздел для студентов содержит следующие функции:

- вакансии СФУ;
- WI-FI и почта;
- Библиотека;
- учебные ресурсы.

Данный раздел представлен на рисунке 16.

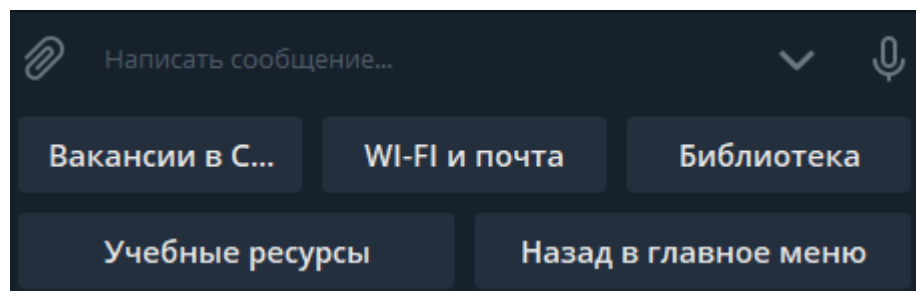


Рисунок 16 – Раздел для студентов

Каждая функция содержит информацию по данному подразделу. Информация состоит из текстового сообщения и ссылок на документы или дополнительные страницы с информацией на сайте СФУ.

Кроме разделов в нижней части главного меню располагается кнопка «Сайт СФУ». При нажатии на неё пользователю приходит сообщение с ссылкой на сайт СФУ. Реализация данной кнопки показана на рисунке 17.

Ссылки в чат-боте СФУ реализуются при помощи объектов `InlineKeyboardButton`. Такая ссылка представляет собой отдельную кнопку под сообщением, которая содержит название ссылки и URL-адрес, на который ведет данная ссылка.

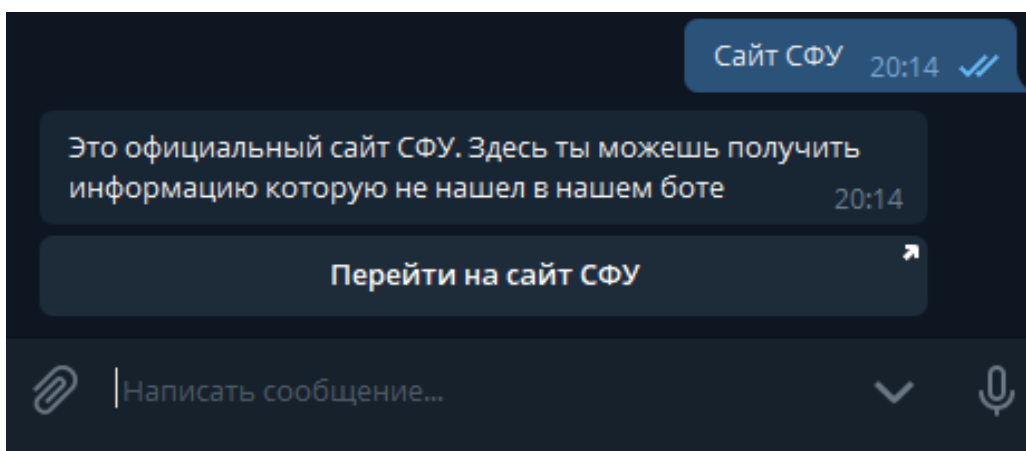


Рисунок 17- Кнопка «Сайт СФУ»

3.3 Реализация парсинга информации с сайта СФУ

Чтобы информация в чат-боте была всегда достоверная и актуальная, было принято решение брать основную информацию для популярных разделов с официального сайта СФУ. Для этого необходимо реализовать парсинг информации с помощью библиотек Requests и BeautifulSoup. Разделы, в которых используются данные библиотеки:

- новости;
- вакансии;
- расписание;
- поступление.

Все эти разделы постоянно обновляются, именно поэтому была выбрана система парсинга сайта.

Парсинг – это сбор и систематизирование информации с сайтов. Алгоритм действия парсинга следующий:

1. получение доступа к коду веб-ресурса;
2. чтение и извлечение данных;
3. предоставление извлеченных данных.

В случае парсинга с помощью Python все полученные данные с веб-ресурса хранятся в отдельном списке. С помощью различных библиотек разработчик систематизирует, изменяет и использует полученные данные [19].

Для парсинга информации про актуальные вакансии СФУ необходимо указать ссылку на страницу вакансий. Затем указать нужные тэги HTML для поиска нужной информации. После этого необходимо отсортировать нужную информацию и объединить с сообщением для пользователя. Код функции «Актуальная вакансия» представлен на рисунке 18. Перед каждым запросом система проверяет доступ к нужной веб-странице. Если страница сайта

недоступна, пользователю выводится сообщение о том, что в данный момент невозможно выполнить данный запрос. Программный код данного алгоритма представлен в приложении А.

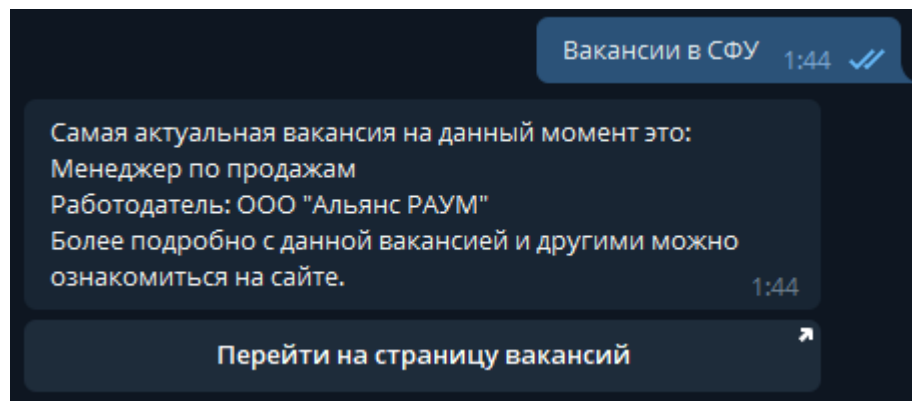


Рисунок 18 – Функция «Актуальная вакансия»

После того, как функция выполнила запрос, список данных очищается для записи следующей информации.

Для реализации функции «Новости» необходимо создать подраздел, который будет включать в себя новости по категориям:

- последние новости;
- о поступлении;
- об обучении;
- о науке;
- о внеучебной жизни;
- о спорте.

Каждая функция будет искать нужную информацию на соответствующей странице новостей по категориям. Функция «Последние новости» показывает самую актуальную новость с сайта СФУ из всех разделов. Результат работы

функции «Последние новости» продемонстрирован на рисунке 19. Программный код данного алгоритма представлен в приложении Б.

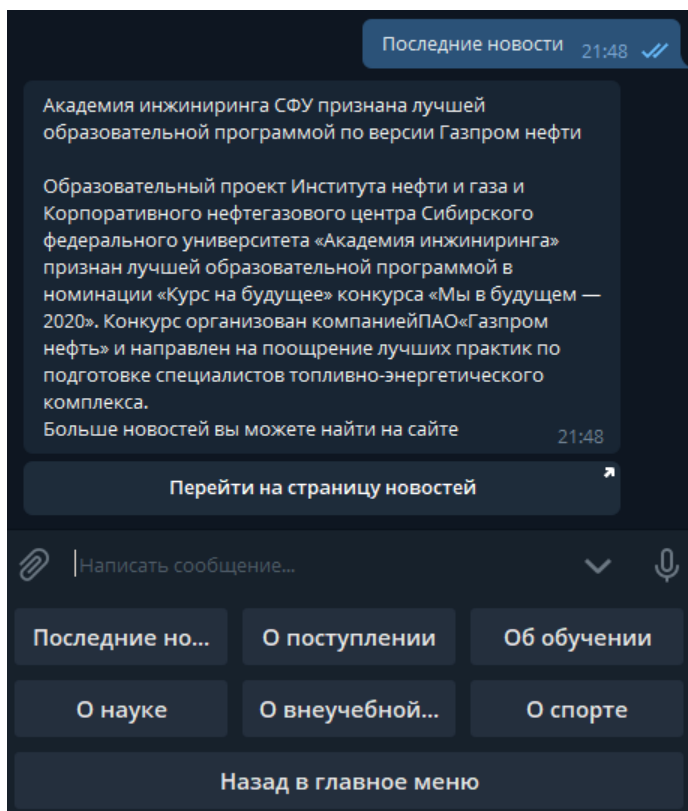


Рисунок 19 – Функция «Последние новости»

При выполнении функции «Расписание» чат-бот выводит список групп. Для получения расписания необходимо выбрать нужную группу. Результат данной функции представлен на рисунке 20.

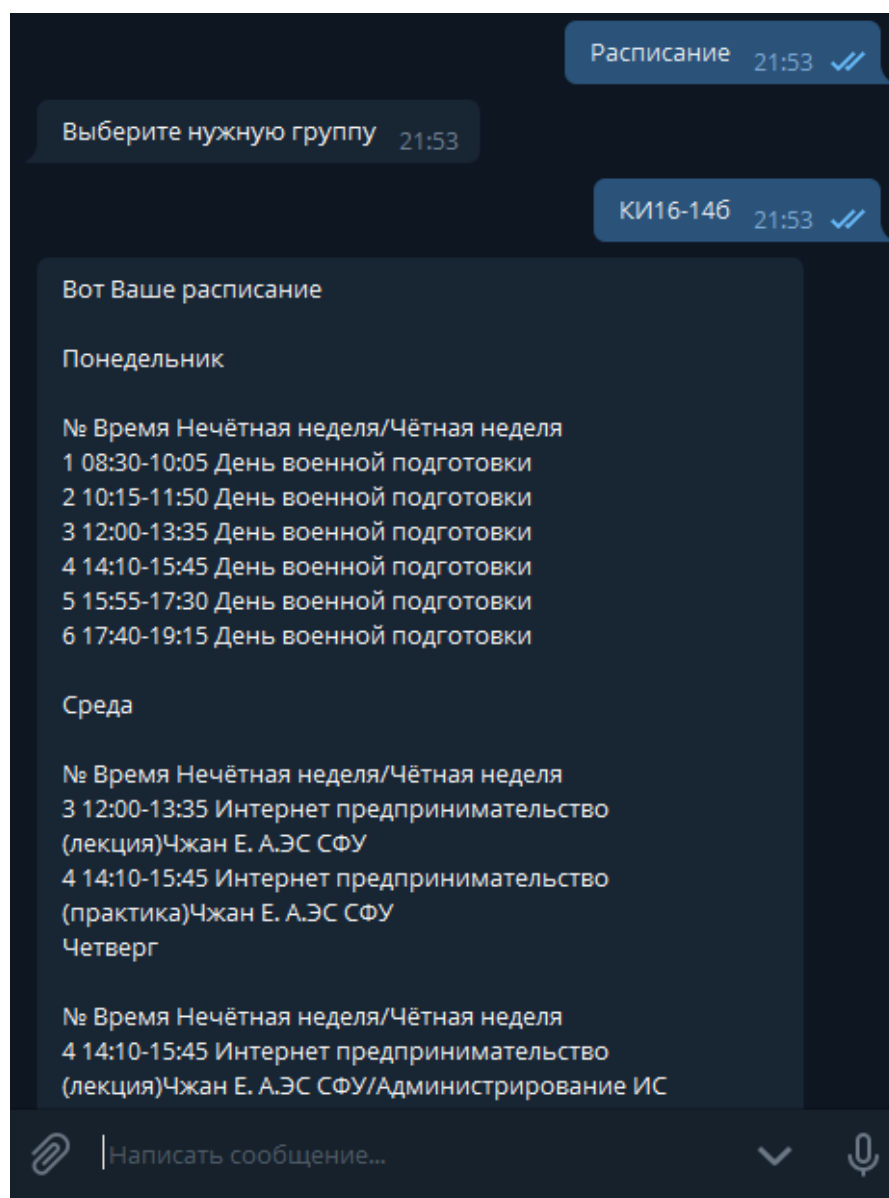


Рисунок 20 – Функция «Расписание»

При нажатии на раздел «Поступление» перед пользователем появляется меню выбора обучения «Бакалавр/специалист» или «Магистр». Результат работы данного меню показан на рисунке 21.

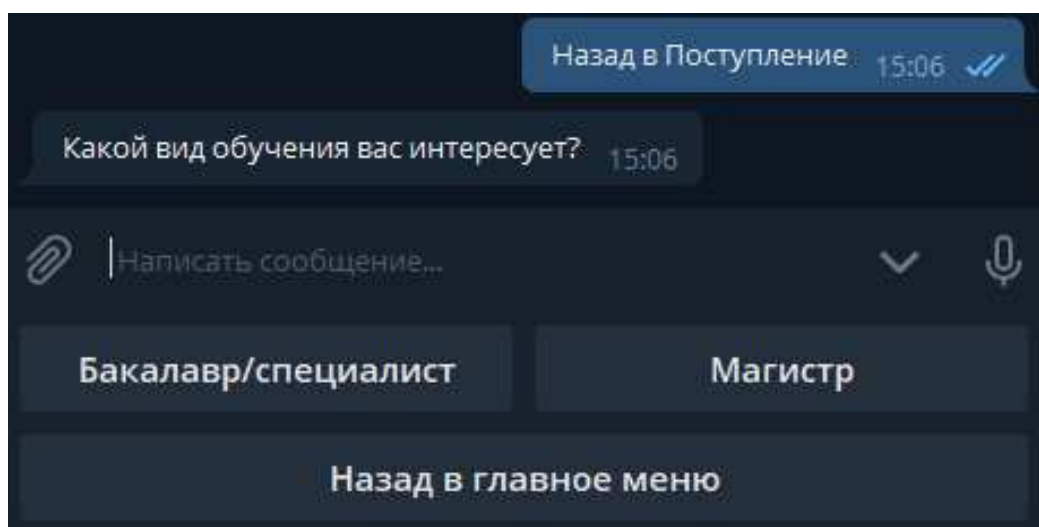


Рисунок 21 – Меню выбора вида обучения

При нажатии на кнопку «Магистр» чат-бот предлагает выбрать пользователю нужный ему раздел. Все разделы функции «Магистр» представлены на рисунке 22.

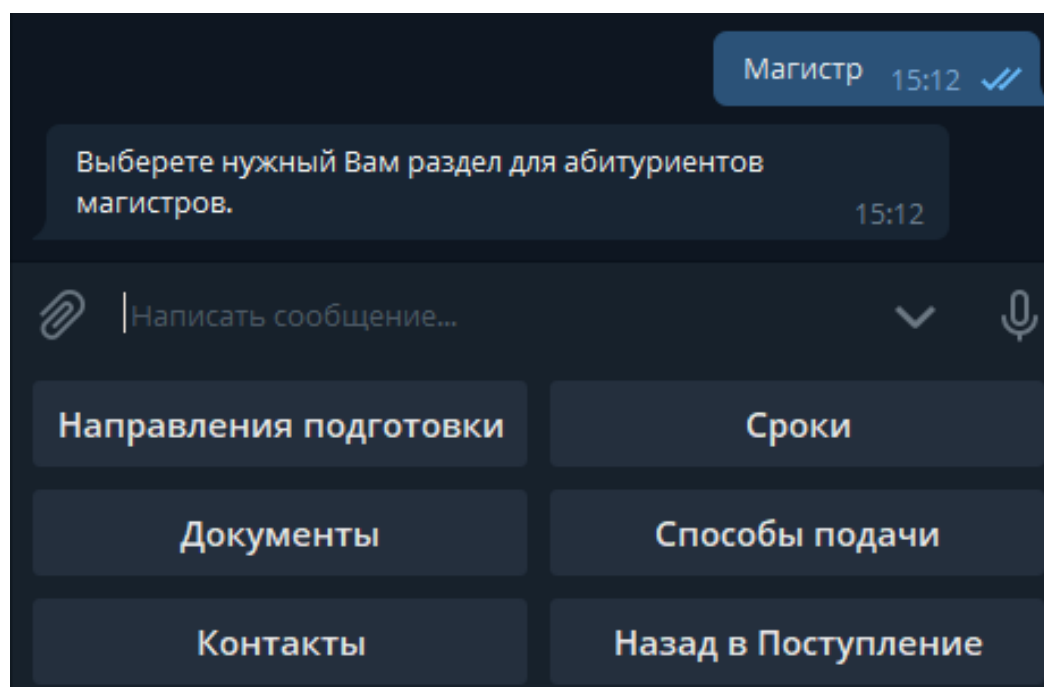


Рисунок 22 – Меню «Магистр»

При нажатии на кнопку «Бакалавр/специалист» пользователю отправляется аналогичное меню с информацией для абитуриентов бакалавриата и специалитета.

Раздел «Направления подготовки» содержит информацию о направлениях, на которые абитуриенты могут подать документы в текущем году поступления. Разделы «Сроки», «Документы», «Способы подачи» и «Контакты» содержат актуальную информацию о местах приема документов, контактах секретарей приемной комиссии, документы и правила приема абитуриентов. Пример работы функции «Сроки поступления» для магистратуры представлен на рисунке 23.

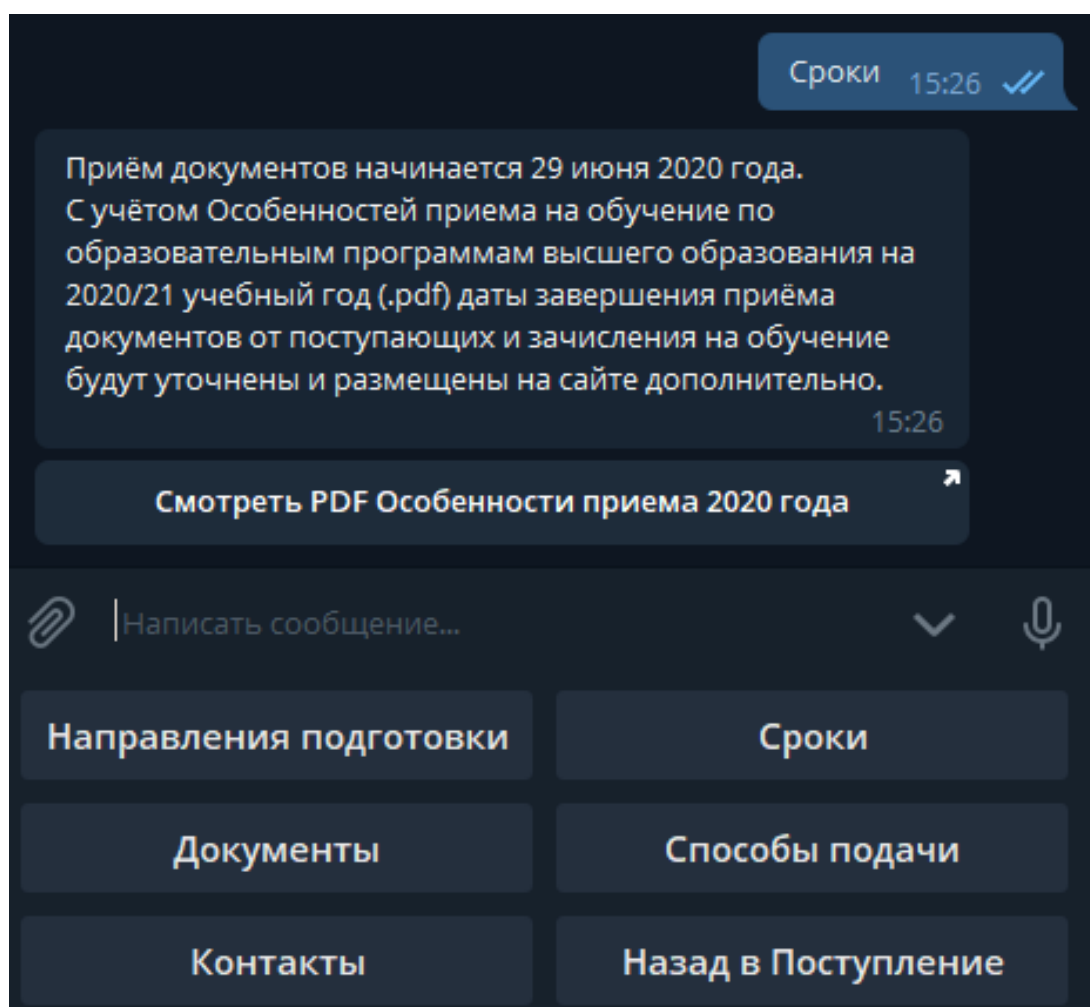


Рисунок 23 – «Сроки поступления» для магистров

После разработки всех функций необходимо добавить все команды в список команд чат-бота. Для выполнения данных действий необходимо перейти в чат-бот «BotFather» и нажать на кнопку «Edit commands». Далее необходимо отправить все команды и описание команд чат-боту сообщением. После успешного добавления команд в чат-боте СФУ будет появляться список команд. Список команд чат-бота СФУ показан на рисунке 24.

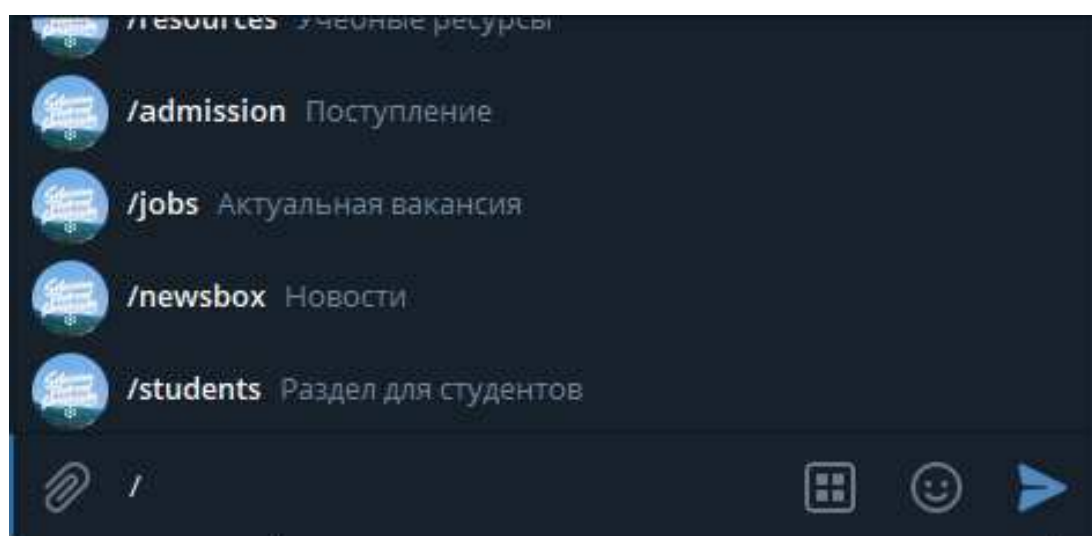


Рисунок 24 - Список команд чат-бота СФУ

3.4 Размещение бота на сервере Heroku

Для доступа к чат-боту его необходимо загрузить на удаленный сервер. В качестве сервера был выбран облачный сервис Heroku.

Для размещения чат-бота необходимо сделать следующие действия:

- Зарегистрироваться на сайте Heroku.
- Скачать и установить Heroku CLI и GIT.
- В терминале выполнить вход в аккаунт Heroku, используя команду «heroku login».

- Создать репозиторий на сайте.
- Клонировать его в папку с кодом программы с помощью команды «heroku git:clone -a имя_репозитория».
- Зафиксировать изменения при помощи команд «git add .» и «git commit -am название_сборки».
- Отправить все изменения на сервер с помощью команды «git push heroku master».

После успешного выполнения всех действий данные отправляются на удаленный сервер. При успешном развёртывании в терминале отобразится информация о режиме доступа к приложению и его статус (рисунок 25).

```
3.0
remote: -----> Discovering process types
remote:          Procfile declares types -> worker
remote:
remote: -----> Compressing...
remote:          Done: 76.8M
remote: -----> Launching...
remote:          Released v3
remote:          https://sibfubot.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sibfubot.git
 * [new branch]      master -> master

C:\Users\User\Desktop\Учеба\bot>_
```

Рисунок 25 – Результат развёртывания чат-бота на сервер Heroku

Таким образом, чат-бот СФУ был размещен на удаленном облачном сервере Heroku для быстрого и бесперебойного доступа. Теперь пользователь в любой момент может получить доступ к чат-боту.

Выводы по третьей главе

В рамках третьей главы выпускной квалификационной работы реализован чат-бот СФУ на основе мессенджера Telegram. Чат-бот соответствует выявленным требованиям:

- поддерживает русский и английский языки;
- имеет 10 основных команд;
- имеет широкий функционал действий.

После реализации чат-бот размещен на облачном сервисе Heroku для быстрого и бесперебойного доступа.

ЗАКЛЮЧЕНИЕ

С каждым днем популярность мессенджеров растет. Люди обмениваются сообщениями и медиафайлами даже на расстоянии. С популярностью мессенджеров растет и популярность чат-ботов. Они являются помощниками различных компаний для потребителей или имитируют переписку с пользователями.

Заказ еды, вызов такси, переводчик – это все уже умеют чат-боты. Пользователям не нужно скачивать много приложений для всего этого. Они могут сделать все в мессенджере. Чат-боты доступны для всех пользователей Telegram.

В рамках выпускной квалификационной работы были выполнены поставленные задачи. Были изучены популярные мессенджеры. После сравнения и анализа преимуществ и недостатков каждого был выбран мессенджер Telegram, как самый удобный в плане разработки чат-бота.

Далее были изучены чат-боты, рассмотрены их достоинства и недостатки. Анализируя другие чат-боты, были выявлены основные требования для разработки чат-бота СФУ.

Затем были выбраны технологии разработки чат-бота. Он создавался на языке программирования Python, с помощью IDE PyCharm и библиотек Requests и BeautifulSoup, а размещался на удаленном сервере Heroku.

В итоге, результатом работы является реализованный чат-бот для помощи студентам и абитуриентам Сибирского Федерального Университета.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Messina, C. 2016 will be the year of conversational commerce [Электронный ресурс] / С. Messina // Medium. – 2016. – Режим доступа: <https://medium.com/chris-messina/2016-will-be-the-year-of-conversational-commerce-1586e85e3991>
2. Рейтинг мессенджеров 2018 [Электронный ресурс]: Сервис сравнения и выбора приложений и сервисов в сфере IT. – 2018. – Режим доступа: <https://coba.tools/compilation/reiting-messendzherov-2018>
3. Аванесян Н. Л., Telegram, как пример мессенджера: возможности и перспективы развития. [Электронный ресурс] / Н. Л. Аванесян // Научный потенциал XXI века. 2017 – Режим доступа: https://elibrary.ru/download/elibrary_29653726_34734017.pdf
4. Официальный сайт Telegtam [Электронный ресурс]: Статистика – Режим доступа: <https://telegram.org/blog/400-million>
5. Telegram Analytics [Электронный ресурс]: Исследование аудитории Telegram 2019 – Режим доступа: <https://tgstat.ru/research>
6. Шмыров, В. Названы любимые мессенджеры россиян [Электронный ресурс]: В. Шмыров – Издание о высоких технологиях. – 2018 – Режим доступа: <http://www.cnews.ru/news/top/2018-02-28>
7. Что такое чат-бот: Определение и Руководство [Электронный ресурс]: SendPulse – Режим доступа: <https://sendpulse.com/ru/support/glossary/chatbot>
8. Иванов А. Д., Чат-бот в Telegram и ВКонтакте, как новый канал распространения новостей. [Электронный ресурс] / А. Д. Иванов // Волжский университет имени В. Н. Татищева. – 2016. – №3 – с. 126-132. – Режим доступа: https://elibrary.ru/download/elibrary_26673675_340583358.pdf

9. Козлов А. А., Телеграм-бот как простой и удобный способ получения информации [Электронный ресурс] / А. А. Козлов, А. В. Батищев // Территория науки. 2017. №5. с. 55-64. – Режим доступа: <https://cyberleninka.ru/article/v/telegram-bot-kak-prostoy-i-udobnyy-sposob-polucheniya-informatsii>
10. Business Insider Intelligence [Электронный ресурс]: 80% of businesses want chatbots by 2020 – Режим доступа: <https://www.businessinsider.com/80-of-businesses-want-chatbots-by-2020-2016-12>
11. Interactive: The Top Programming Languages 2018 [Электронный ресурс]: IEEE Spectrum – Режим доступа: <https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018>
12. Матвеева Н. Ю., Технологии создания и применения чат-ботов [Электронный ресурс]: Н. Ю. Матвеева, А. В. Золотарюк. – Научные записки 38 молодых исследователей. – 2018. – №1. – с. 28-30. – Режим доступа: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chat-botov>
13. Официальный сайт Telegtam [Электронный ресурс]: API – Режим доступа: <https://core.telegram.org/api>
14. Советов, Б. Я. Информационные технологии.: учебное пособие / Б. Я. Советов, В. В. Цехановский. – Москва: Высш. шк., 2003. – 263 с.
15. Telegram Bot API [Электронный ресурс]: Telegram Documents. – Режим доступа: <https://core.telegram.org/bots/api>
16. Официальный сайт Heroku [Электронный ресурс]: Облачная PaaS-платформа. – Режим доступа: <https://www.heroku.com/>.
17. Маленькие хитрости с Heroku [Электронный ресурс]: JavaRush – Режим доступа: <https://javarush.ru/groups/posts/1987-malenjkie-khitrosti-s-heroku>

18. Работа с библиотекой pyTelegramBotApi [Электронный ресурс]:
Gitbook – Режим доступа: https://groosha.gitbook.io/telegram-bot-lessons/extra/pytelegrambotapi_basics

19. Requests в Python [Электронный ресурс]: Python scripts – Режим
доступа: <https://python-scripts.com/requests#method-get-requests>

ПРИЛОЖЕНИЕ А

Код алгоритма функции «Актуальная вакансия»

```
@bot.message_handler(commands=['jobs'])
def jobs(message):
    url_job = 'http://career.sfu-kras.ru/students/job'
    params = None
    tables_jobs = requests.get(url_job, params=params)
    if tables_jobs.status_code == 200:
        soup = BeautifulSoup(tables_jobs.text, 'html.parser')
        items = soup.find_all('table', class_='table')
        job = []
        for item in items:
            job.append(item.find('tr', class_='heading heading-
section').get_text(strip=True))
            job.append(item.find('td').get_text(strip=True))

        final_message = f'Самая актуальная вакансия на данный момент
это:\n{job[0]}\nРаботодатель: {job[1]} \n' \
            f'Более подробно с данной вакансией и другими можно
ознакомиться на сайте.'
        markup = types.InlineKeyboardMarkup()
        markup.add(types.InlineKeyboardButton('Перейти на страницу вакансий',
url="http://www.sfu-kras.ru/job"))
        job.clear()
        bot.send_message(message.chat.id, final_message, reply_markup=markup)

    else:
        final_message = 'К сожалению, в данный момент мы не можем показать вам
вакансии'
        bot.send_message(message.chat.id, final_message)
```

ПРИЛОЖЕНИЕ Б

Код алгоритма функции «Последние новости»

```
@bot.message_handler(commands=['news'])
def news(message):
    url_news = 'http://edu.sfu-kras.ru/news'
    params = None
    tables_news = requests.get(url_news, params=params)
    if tables_news.status_code == 200:
        soup = BeautifulSoup(tables_news.text, 'html.parser')
        items = soup.find_all('article', class_='news-item teaser')
        new = []
        for item in items:
            new.append(item.find('h3', class_='news-item-title').get_text(strip=True)),
            new.append(item.find('p', class_='last').get_text(strip=True))

        final_message = f'{new[0]}\n\n{new[1]}\nБольше новостей вы можете найти на
сайте'
        markup = types.InlineKeyboardMarkup()
        markup.add(types.InlineKeyboardButton('Перейти на страницу новостей',
url="http://news.sfu-kras.ru/"))
        bot.send_message(message.chat.id, final_message, reply_markup=markup)
        new.clear()
    else:
        final_message = 'К сожалению, в данный момент мы не можем показать вам новости.
Проблемы на сайте.'
        bot.send_message(message.chat.id, final_message)
    menu(message)
```

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Космических и Информационных технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

П. П. Дьячук
подпись инициалы, фамилия

« 26 » 06 2020 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – “Информационные системы и технологии”

Чат-бот СФУ

Руководитель



подпись, дата

доцент, к.т.н.

должность, учёная степень

И. А. Легалов

инициалы, фамилия

Выпускник



подпись, дата

Д. А. Макушев

инициалы, фамилия

Красноярск 2020