

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и Информационных Технологий  
институт

Информационные системы  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

Л.С. Троценко  
подпись                      инициалы, фамилия

«13» июня 2018 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 Информационные системы и технологии

Бот-помощник для мессенджера Telegram

Руководитель

\_\_\_\_\_

подпись, дата

доцент, к.т.н  
должность, ученая степень

И.А. Легалов  
инициалы, фамилия

Выпускник

\_\_\_\_\_

подпись, дата

М.С. Русин  
инициалы, фамилия

Нормоконтролер

\_\_\_\_\_

подпись, дата

ст. преподаватель  
должность, ученая степень

Ю.В. Шмагрис  
инициалы, фамилия

Красноярск 2018



Студенту Русину Михаилу Сергеевичу

Группа: КИ14-13Б Направление: 09.03.02 Информационные системы и технологии

Тема выпускной квалификационной работы: Бот-помощник для мессенджера Telegram

Утверждена приказом по университету № 4896/с от 05.04.2018г.

Руководитель ВКР: И. А. Легалов, к.т.н., доцент кафедры Информационные системы Института Космических и Информацонных технологий

Исходные данные для ВКР: Материалы по используемым технологиям, рекомендации руководителя

Перечень разделов ВКР: Введение, теоретические основы, выбор технологий и среды разработки, разработка информационной системы, заключение, список использованных источников, приложение А

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2007

Руководитель ВКР

\_\_\_\_\_   
подпись

И.А. Легалов  
инициалы и фамилия

Задание принял к исполнению

\_\_\_\_\_   
подпись

М.С. Русин  
инициалы и фамилия

«15» марта 2018г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме: «Бот-помощник для мессенджера Telegram» содержит 41 страницу текстового документа, 23 рисунка, 2 таблицы, 25 использованных источников, 1 приложение.

ЧАТ-БОТ, TELEGRAM, МЕССЕНДЖЕР, ЗИМНЯЯ УНИВЕРСИАДА  
2019

Целью выпускной квалификационной работы является разработка бота-помощника для мессенджера Telegram.

### Основные задачи:

- анализ выбранной предметной области;
- сравнение имеющихся аналогов чат-ботов;
- выбор технологий и среды разработки;
- разработка чат-бота на платформе Telegram.

### Основные результаты:

- изучены мессенджеры, а также их достоинства и недостатки;
- изучены аналоги чат-ботов;
- разработан авторский чат-бот.

## СОДЕРЖАНИЕ

Введение.....	4
1 Теоретические основы.....	5
1.1 Мессенджеры.....	5
1.2 Чат-боты.....	9
1.3 Обзор аналогов чат-ботов .....	11
1.3.1 Чат-бот каршеринга YouDrive .....	11
1.3.2 Чат-бот «Спорт-Экспресс».....	12
1.3.3 Бот для заказа такси Maxim .....	13
1.3.4 Чат-бот «Какой счет?».....	14
2 Выбор технологий и среды разработки .....	16
2.1 Выбранные технологии разработки .....	16
2.1.1 Node.js.....	16
2.1.2 Telegram Bot API .....	18
2.1.3 Heroku.....	19
2.1.4 MongoDB.....	20
2.2 Среда разработки.....	23
2.2.1 Visual Studio Code .....	23
3 Разработка информационной системы .....	25
3.1 Регистрация чат-бота для Telegram Bot API.....	25
3.2 Реализация клиентской части чат-бота.....	27
3.2 Проектирование и реализация базы данных чат-бота.....	30
3.3 Реализация серверной части чат-бота .....	31
3.4 Размещение чат-бота на удаленном сервере Heroku .....	35
Заключение .....	37
Список использованных источников .....	38
Приложение А .....	41

## ВВЕДЕНИЕ

Интернет в современных условиях – это универсальная среда для общения, развлечений и обучения. В настоящее время в мире существует большое количество средств, форм и способов общения, и немалая часть из них так или иначе связана с современными техническими возможностями, которые, в частности, представлены использованием глобальной компьютерной сети. Интернет кроме источника разнообразной и полезной для пользователей информации также является основной формой виртуального общения. Связь с друзьями и родственниками, контакты с рабочими партнерами, новые знакомства – все это является важным компонентом повседневной жизни современного человека, причем выбор наиболее удобных способов онлайн-общения у пользователя достаточно большой.

Актуальность выпускной квалификационной работы обусловлена высокой популярностью мессенджеров и таких средств автоматизации как чат-боты среди пользователей сети Интернет. Чат-боты позволяют упростить ежедневные рутинные задачи, такие как получение информации о погоде, пробках, последних новостях и другие. Главным достоинством относительно классических приложений является возможность совмещения всех возможностей на платформе одного мессенджера. Кроме этого в 2019 году в городе Красноярске будет проводиться Зимняя Универсиада, что послужило основанием для разработки авторского чат-бота, посвященного этой тематике.

Целью выпускной квалификационной работы является разработка бота-помощника для мессенджера Telegram.

Исходя из поставленной цели, были поставлены следующие задачи:

- анализ выбранной предметной области;
- сравнение имеющихся аналогов чат-ботов;
- выбор технологий и среды разработки;
- разработка чат-бота на платформе Telegram.

# 1 Теоретические основы

## 1.1 Мессенджеры

Свое развитие интернет-сервисы для общения начали с чатов, потом мессенджеры, потом социальные сети, но с недавних пор мессенджеры вновь возглавили список самых перспективных сервисов. Это подтверждаю исследования компании «Билайн» в 2017 году: количество пользователей мессенджеров в Российской Федерации увеличилось на 7%. Одновременно с этим среднее количество мессенджеров, используемых одним пользователем, увеличилось за 2017 год с 1,73 до 3 штук. Мессенджеры используются преимущественно взрослыми, так на долю лиц до 18 лет приходится 7,7% от общего количества аудитории, пользователи в возрасте от 18 до 25 лет составляют 9,8%, от 25 до 35 лет – 32,3%, от 35 до 45 лет – 26,8%. При этом 13,1% пользователей мессенджеров находятся в возрасте от 45 до 55 лет, а 7,6% — от 55 до 64 лет. И всего 2,7% аудитории мессенджеров находится в возрасте старше 64 лет [11].

Причина повторной волны популярности мессенджеров – изменения в области мобильного Интернета: высокие скорости, намного ниже цены, чем ранее, широкое распространение смартфонов. Крис Мессина в статье «2016 will be the year of conversational commerce»[1] назвал 2016 год – годом диалоговой или разговорной коммерции. Понимая под этим понятием чаты, мессенджеры или другие интерфейсы на естественном языке (т.е. голосом), которые обеспечивают взаимодействие людей с торговыми брендами или услугами. Конечным результатом он видит то, что пользователи будут общаться с брендами и компаниями через мессенджеры. При этом, по его мнению, неважно, с кем будут общаться люди – с сотрудником компании, с ботом или с их комбинацией.

Все современные мессенджеры имеют множество схожих функций, таких как отправка сообщений, медиафайлов, аудио- и видео общение.

В настоящее время существует большое разнообразие мессенджеров, среди которых стоит отметить некоторые из них, а также их преимущества и недостатки (Приложение А).

Особого внимания заслуживает мессенджер Telegram.

Telegram – это кроссплатформенное приложение, разработанное на языке программирования C++, позволяющее обмениваться сообщениями и файлами большинства форматов. Мессенджер использует специально разработанную серверную часть с закрытым кодом, работающую на серверах Германии и США. Telegram имеет ряд преимуществ:

- приватность – все чаты зашифровываются, а сообщения уничтожаются через обозначенное время;
- быстрота – скорость доставки сообщений выше, чем у аналогов;
- распределение – сервера Telegram расположены по всему миру, что повышает отказоустойчивость;
- открытость – использование открытого протокола MTProto и API, бесплатных для всех;
- отсутствие подписок и рекламы;
- отсутствие ограничений на размер сообщений и вложенных файлов.

Из минусов можно отметить отсутствие видеозвонков.

Основателями мессенджера Telegram являются братья Павел и Николай Дуровы. Павел является финансовой и идеологической опорой проекта, а Николай занят техническими аспектами.

Особенностями данного мессенджера являются секретные чаты и специально разработанный протокол шифрования MTProto (рисунок 1)



Рисунок 1 – Принцип работы протокола шифрования MTProto

Секретные чаты нужны для людей, которые хотят обеспечить себя максимально безопасным общением. Помимо использования основного протокола все сообщения шифруются ключами по принципу от устройства до устройства. Никто не перехватит и не расшифрует сообщения, включая сотрудников Telegram. Сообщения из таких чатов пересылать нельзя и они не хранятся на серверах приложения. Но у секретных чатов есть два недостатка – это невозможность открыть чат на устройстве, которое не участвовало в его создании, и никто не защищен от скриншота интерфейса Telegram с открытым секретным чатом.

Мессенджер Telegram доступен для всех популярных платформ, например, Android OS, iOS, Windows Phone, Linux и другие. Так же в большинстве версия для разных операционных систем доступны как минимум английский, русский, польский и немецкий языки.

Примеры интерфейса десктопного приложения на Windows 10 и iOS изображены на рисунках 2 и 3.

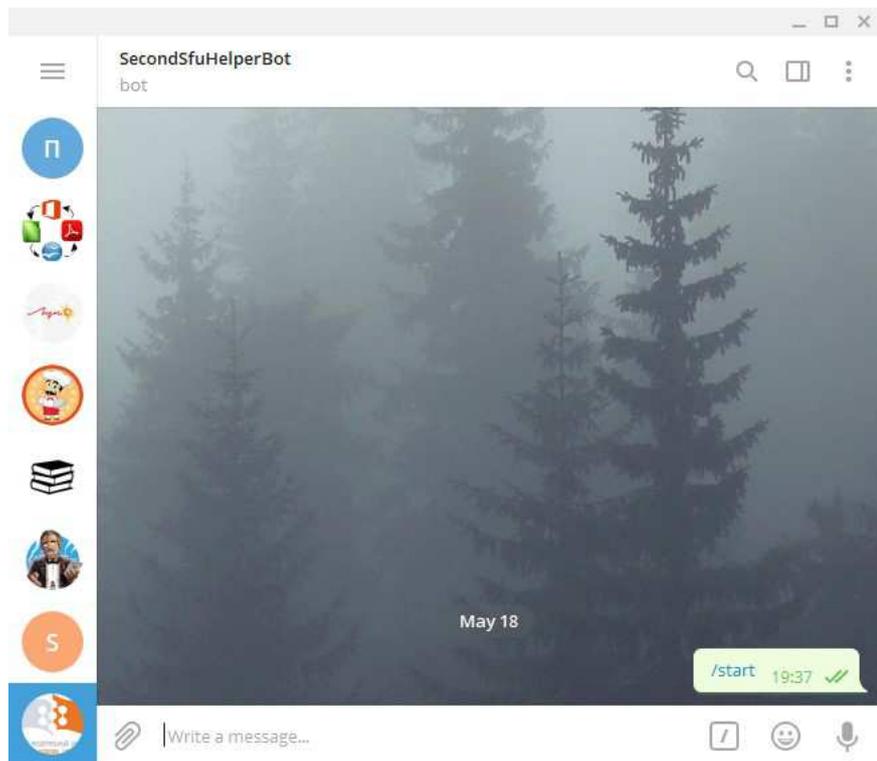


Рисунок 2 – Интерфейс Telegram на Windows 10

Также в большинстве версиях для разных операционных систем доступны как минимум английский, русский, польский и немецкий языки.



Рисунок 3 – Интерфейс Telegram на iOS 11

В последнее время идет Telegram ведет активную разработку своей блокчейн-платформы Telegram Open Network и криптовалюты Gram на ее основе.

## **1.2 Чат-боты**

В последнее время в мессенджерах набирают популярность такие сервисы как чат-боты. Сам термин «чат-бот» был придуман Майклом Молдина в 1994 г. для описания разговорных программ.

Изучив современное состояние использования чат-ботов в мессенджерах, можно прийти к выводу, что чат-боты являются универсальными средствами, способными к решению разнообразных задач – от общения до развлечений, от предоставления медицинской консультации до заказа товаров и услуг посредством специализированных прикладных решений, от распознавания эмоций до решения сложных консалтинговых задач в службах поддержки клиентоориентированных информационных систем.

В любом случае, в независимости от платформы, чат-бот – это прикладная программа, которая, получая информацию от пользователя, формирует корректные, логически обоснованные ответы.

На сегодняшний момент существует огромное разнообразие чат-ботов. Некоторые из возможных вариантов представлены ниже:

- игровые чат-боты (квесты, ролевые игры);
- рекламные чат-боты (Хрустteam, Coca-Cola);
- новостные чат-боты (Коммерсантъ, Meduza);
- чат-боты для доставок, магазинов, сервисов (такси «Максим», пиццерия «Папа Джонс», бронирования поездов и самолетов «Туту.рф»);
- чат-боты для консультации и поддержке клиентов (банк «Точка», каршеринг «UrentCar») и другие.

В последнее время большие возможности открывают информационные технологии, связанные с чат-ботами. Они настолько прочно вошли в жизнь людей, что применяются во всех сферах человеческой деятельности, и с каждым днем их роль все больше увеличивается. В первую очередь это объясняется тем, что основную часть своего времени люди проводят за смартфоном в email-клиентах и мессенджерах.

Боты – это программы, которые выполняют различные задачи для пользователя, находящегося в мессенджере. Бот выглядит как обычный чат, однако общение происходит не с человеком, а с программой, которая может принять заказ на вызов машины, если это – бот такси, или прислать свежие статьи, если это – новостной бот, или заказать доставку еды на дом, если это – ресторанный бот. Не так давно чат-боты обрели большую популярность, перевоплотившись из развлечения в более серьезную вещь, так как они, в основном, стали использоваться для решения серьезных бизнес-задач. В эпоху информационных технологий – это нормально явление, а тем более – сети Интернет, ведь общество давно перешло на «новое» и деловое, и неделовое общение. Во-первых, чат-боты – это «платформы» для решения бизнес-задач. Во-вторых, чат-бот – это приложение, которое поддерживает диалог с пользователем, выбирая ответы из базы данных: вы спрашиваете, где пообедать и тут же получаете мгновенный ответ. Кроме того, чат-боты выполняют множество полезных функций по исполнению рутинных операций, поиску информации, объединению данных, работе с клиентурой.

Чат-бот как виртуальный собеседник имеет базу знаний, которая представляет собой наборы возможных вопросов пользователя и соответствующих им ответов. Наиболее распространенными вариантами для получения нужного ответа являются ключевые слова, совпадение фразы, совпадение контекста. Всегда существуют какие-то простые и легкие по выполнению дела, на которые не хочется тратить время. Тут на помощь всегда могут прийти чат-боты. Век пишущих машинок, ходьбы по библиотекам и бесконечных очередей в книжные магазины канул в Лету. Теперь для сбора

информации можно использовать чат-ботов. Конечно, это делается при помощи общения с людьми, но отнюдь не всегда. Например, в рамках какого-либо мероприятия чат-боты могут сообщать всем участникам новости и предоставлять справочную информацию.

### **1.3 Обзор аналогов чат-ботов**

#### **1.3.1 Чат-бот каршеринга YouDrive**

Данный бот был разработан программистами каршеринг-сервиса YouDrive. Сценарии и функции были разработаны на основе анализа обращений клиентов через другие каналы связи – телефон, соцсети, электронную почту.

Цель бота состояла в том, чтобы сделать процесс уведомления о проблемах с арендованным транспортным средством максимально простыми и удобным для пользователей, а также повысить скорость обработки сообщений сотрудниками сервиса, тем самым снизив время ожидания для клиентов.

Для удобства пользователей бот имеет стандартный набор наиболее часто задаваемых вопросов и соответствующих ответов на них. Также имеется форма обратной связи, которая автоматически свяжет вас со специалистом по интересующему вопросу. Интерфейс чат-бота изображен на рисунке 4.

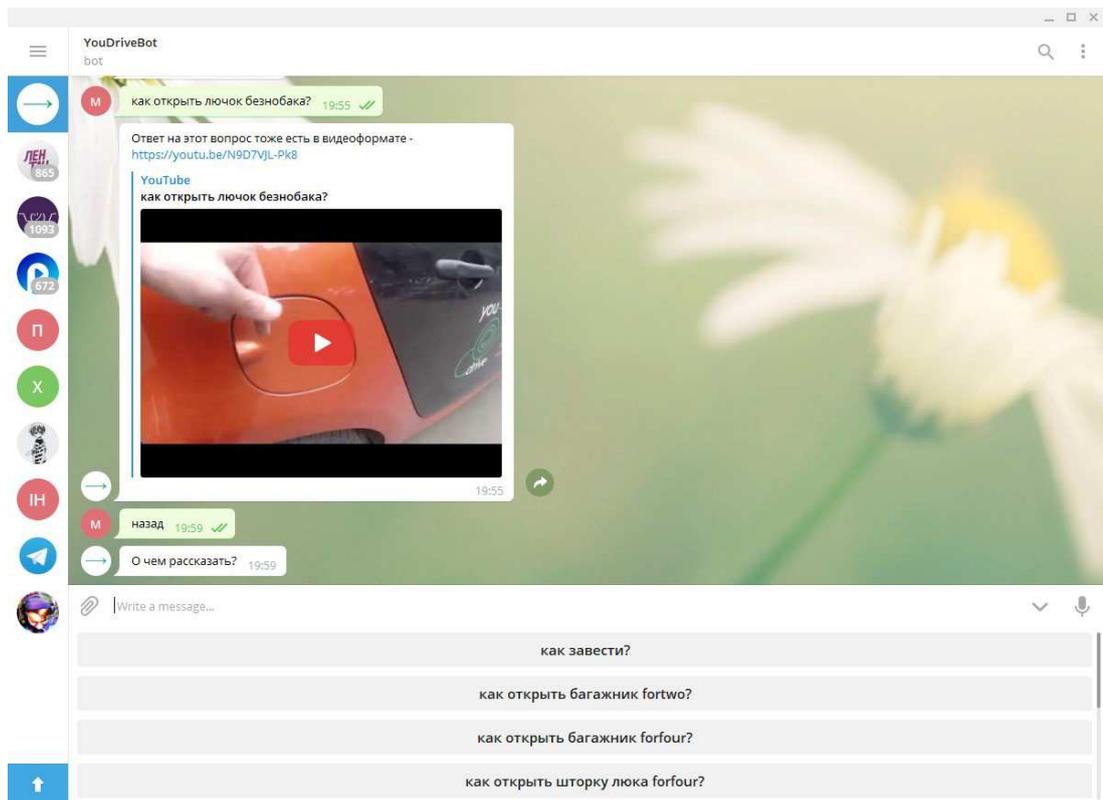


Рисунок 4 – Функционал чат-бота YouDriverBot на платформе Telegram для Window 10

### 1.3.2 Чат-бот «Спорт-Экспресс»

Чат-бот Спорт-Экспресс представляет собой новостного бота с уклоном в спортивную тематику. Главными его задачами являются:

- рассылка главных новостей в области спорта;
- вывод списка трансляций на текущий день;
- вывод списка результатов на текущий день (рисунок 5).



Рисунок 5 – Пример интерфейса чат-бота Спорт-Экспресс

Из плюсов можно отметить:

- максимально упрощенный интерфейс;
- полная сводка о результатах всех событий на текущий день.

Минусами данного чат-бота являются:

- возможность вывода результатов только на текущий день;
- отсутствие ссылок на трансляции;
- ограниченное количество спортивных дисциплин.

### 1.3.3 Бот для заказа такси Maxim

Данный чат-бот позволяет заказать такси без использования специального приложения. У бота есть всего 4 команды:

- /start – стандартная обязательная команда для активации чат-бота;
- /order – команда для заказа такси;
- /cancel – отменить текущий заказ;
- /forget – удалить имеющуюся информацию о пользователе.

После активации бота и отправки команды для заказа такси будет предложено ввести номер мобильного телефона. Это необходимо для защиты бота от атак и идентификации пользователей. После чего на мобильное устройство приходит SMS с проверочным кодом. Далее бот предлагает выбрать город и адрес из списка его частых поездок или указать другой. Если в настройках мобильного устройства включено передача текущих геоданных, то пользователь может переслать свои координаты, а бот сам попытается найти соответствующий адрес пользователя (рисунок 6).

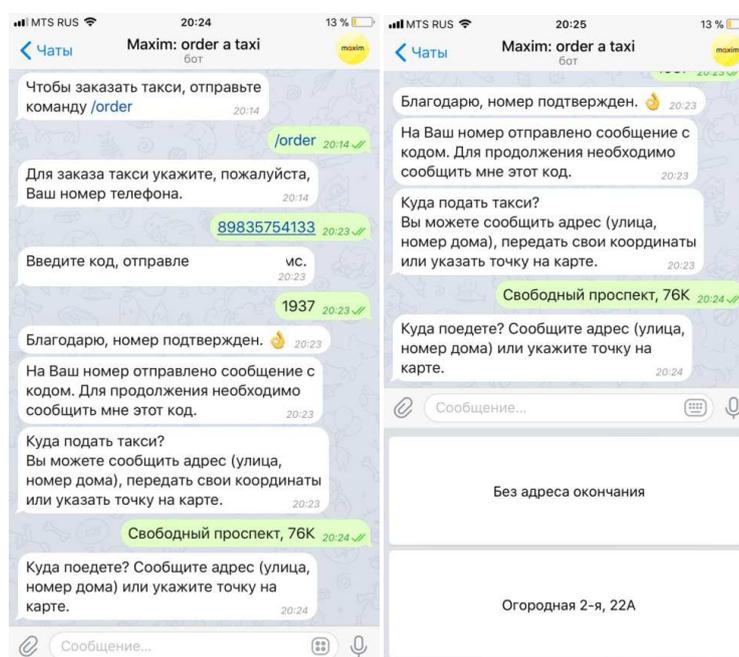


Рисунок 6 – Функционал чат-бота TaxiMaximBot на платформе Telegram для iOS 11.

### 1.3.4 Чат-бот «Какой счет?»

Данный чат-бот присылает результаты футбольных матчей коллективов, на которые подписан клиент. После выбора команд чат-бот будет в автоматическом режиме присылать оповещения после очередного события (рисунок 7).

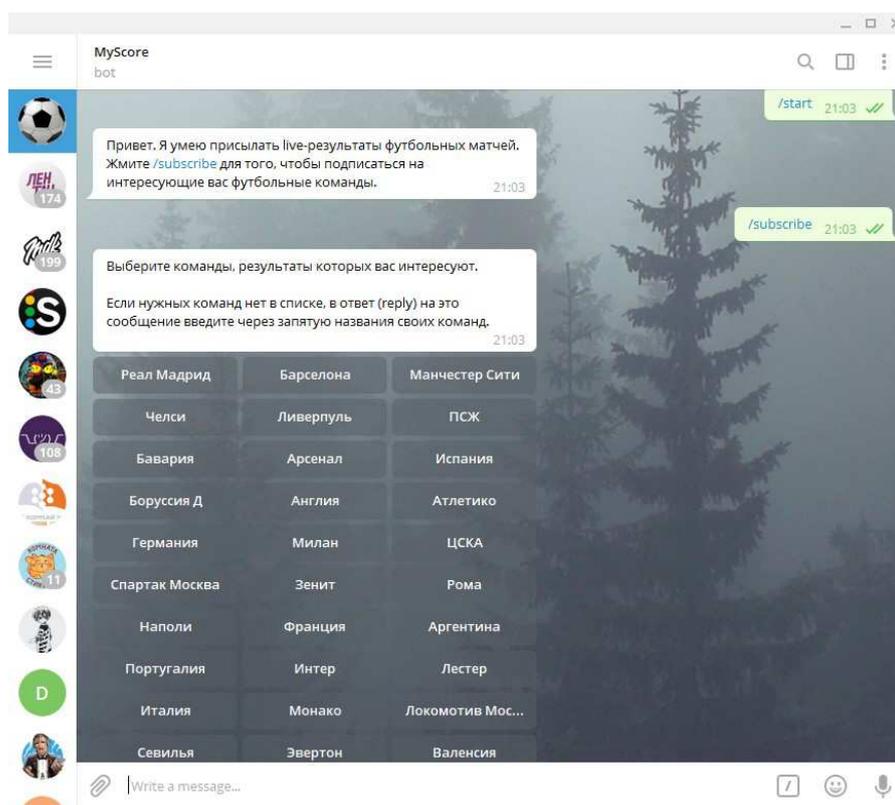


Рисунок 7 – Интерфейс чат-бота «Какой счет?»

Плюсами данного чат-бота являются:

- полный список футбольных коллективов мира (футбольные клубы и сборные стран);
- возможность получать обновления в режиме реального времени.

Однако существуют и минусы:

- невозможно отредактировать список выбранных футбольных коллективов;
- информация ограничена только результатами футбольных матчей.

## 2 Выбор технологий и среды разработки

### 2.1 Выбранные технологии разработки

#### 2.1.1 Node.js

Node.js – программная платформа на движке V8, который транслирует JavaScript-код в машинный. Этот движок превращает узкоспециализированный язык JavaScript в язык общего назначения. В Node.js появилась возможность взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, использовать сторонние библиотеки, созданные при помощи разных языков программирования, вызывая их из JavaScript-кода.

Node.js используется преимущественно как back-end язык программирования, выполняя роль сервера, но так же есть возможность разработки оконных десктопных приложений при помощи специальных фреймворков, например, NW.js, AppJS или Electron. Из особенностей можно отметить возможность использовать язык Node.js для разработки приложений в сфере «интернета вещей» (Internet of Things, IoT). В последнее время мы все чаще встречаемся с смарт-браслетами, колонками и любой другой «умной» техникой, которая в теории может быть запрограммирована на Node.js.

История Node.js началась с разработки нового JavaScript-движка V8 в датском отделении компании Google. Ведущий разработчик Lars Bak выделил основные проблемы, которые должны были решены в движке, это производительность и масштабируемость. Первая лабораторная версия была готова 3 июля 2008 года, а позже был представлен первый браузер Chromium в публичный релиз которого входил новый движок. Помимо браузеров на основе Chromium, например, Chrome, Comodo Dragon, Opera, Амиго и другие, существует ряд других продуктов, использующих V8:

- Maxthon – веб-браузер со встроенным блокиратором рекламы, использующий два движка рендеринга: WebKit и Trident;

- стандартный браузер Android;
- операционная система Android;
- HP webOS – операционная система от Hewlett-Packard для коммуникаторов, нетбуков и планшетов;
- Google Chrome OS – операционная система от Google на базе проекта Chromium, ориентированная на облачные сервисы.

В 2009 году программист Rayn Dahl, после двух лет экспериментов над созданием серверных веб-компонентов, представляет Node.js. В ходе исследований разработчик пришел к выводу, что вместо традиционной модели параллелизма на основе потоков следует обратиться к событийно-ориентированным системам из-за их простоты, низких расходов и быстродействия. Сам Node.js предлагает «легкий способ построения масштабируемых сетевых серверов». Всю разработку финансирует компания Joyent.

В качестве примера на рисунке 8 изображен фрагмент кода, который реализует простой сервер.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Рисунок 8 – Реализация простого сервера на Node.js

## 2.1.2 Telegram Bot API

Bot API представляет собой HTTP-интерфейс для работы с ботами в Telegram. Каждый бот – это специальный аккаунт, созданный для автоматического обрабатывания и отправления сообщений. Существует два противоположных по логике способа получения обновлений от бота:

- long polling – приложение автоматически опрашивает сервера Telegram на наличие каких-либо обновлений для бота. По умолчанию 100мс;
- webhook – сервера Telegram сами оповещают приложение на сервере как только появятся какие-либо обновления.

Входящие обновления будут храниться на сервере до тех пор, пока их не обработают, но не дольше 24 часов. Независимо от способа получения обновлений, в ответ отправляется объект Update, сериализованный в JSON.

Все запросы к Telegram Bot API должны осуществляться через HTTPS в следующем виде: `https://api.telegram.org/bot<token>/НАЗВАНИЕ_МЕТОДА`. Принцип работы взаимодействия чат-бота и пользователя изображен на рисунке 9.

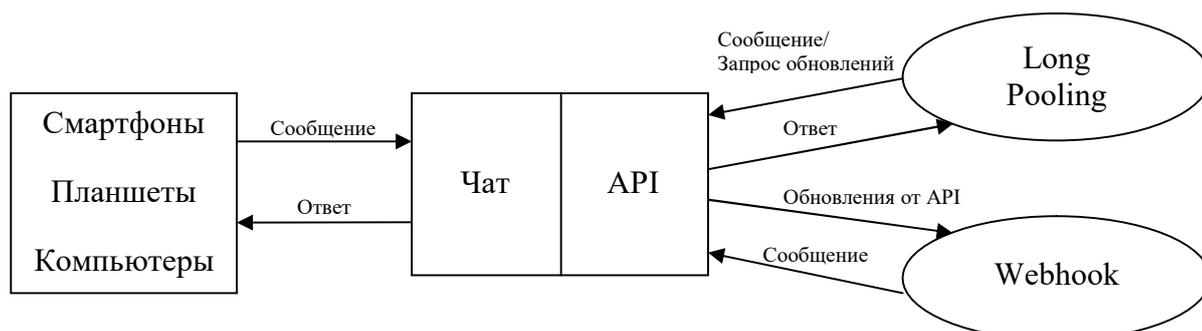


Рисунок 9 – Принцип работы чат-бота на платформе Telegram

Для того чтобы получить token необходимо написать специальному боту @BotFather. Примеры доступных для API методов описаны ниже:

- getUpdates – этот метод используется для получения обновлений по технологии long polling;

- `setWebhook` – метод привязывает к боту url домена, где содержится запущенный бот;
- `sendMessage` – метод отправляет текстовое сообщение в клиент Telegram;
- `sendLocation` – метод отправляет сообщение с координатами в клиент Telegram;
- `getFile` – метод возвращает загруженный файл по его имени и др.

Допускаются POST и GET запросы. Для передачи параметров в Bot API есть 4 способа:

- Запрос в URL
- `application/x-www-form-urlencoded`
- `application/json` (не приемлем для загрузки файлов)
- `multipart/form-data` (для загрузки файлов)

### 2.1.3 Heroku

**Heroku** – облачная PaaS-платформа, поддерживающая ряд языков программирования. С 2010 года является дочерней компанией Salesforce. Heroku, одна из первых облачных платформ, появилась в июне 2007 года и изначально поддерживала только язык программирования Ruby, но на данный момент список поддерживаемых языков также включает в себя Java, Node.js, Scala, Clojure, Python, Go и PHP. На серверах Heroku используются операционные системы Debian или Ubuntu.

Heroku основали в 2007 году Джеймс Линденбаун, Адам Уиггинс и Орион Генри для поддержки проектов, основанных на Rack. 8 декабря 2010 года Heroku была куплена компанией Salesforce, сделав её своей дочерней компанией. В 2011 году Ю. Мацумото, создатель языка программирования Ruby, пришёл в компанию в роли ведущего инженера. В этом же месяце была внедрена поддержка Node.js и Clojure. Также платформа предоставляет

поддержку таких СУБД, как Cloudant, Membase, MongoDB и Redis, помимо PostgreSQL.

Приложения, работающие на платформе Heroku, используют также DNS-сервер (обычно приложения имеют доменное имя вида «имя\_приложения.herokuapp.com»). Для всех приложений выделяются несколько независимых виртуальных процессов, названные «dynos». Они распределены по специальной виртуальной сетке, которая состоит из нескольких серверов. На Heroku имеется собственную систему контроля версий.

#### **2.1.4 MongoDB**

MongoDB – документо-ориентированная база данных с открытым исходным кодом, которая не требует описания схемы таблиц. Основные возможности:

- поддержка отказоустойчивости и масштабируемости: асинхронная репликация, набор реплик и распределения базы данных на узлы;
- документо-ориентированное хранение (JSON-подобная схема данных);
- эффективное хранение больших объектов, административный интерфейс, серверные функции, Map/Reduce и другое;
- использование Javascript в качестве языка для формирования запросов;
- профилирование запросов;
- широкий набор атомарных операций над данными (условный поиск, сложная вставка/обновление и тому подобное);
- разные типы данных (в том числе поддержка массивов);
- поддержка индексов (B-Tree);

- полнотекстовый поиск, в том числе на русском языке, с поддержкой морфологии;
- журналирование операций, изменяющих данные в базе данных.

В отличие от реляционных баз данных MongoDB предлагает документо-ориентированную модель данных, благодаря чему она работает быстрее, обладает лучшей масштабируемостью, ее легче использовать.

Но, даже учитывая все недостатки традиционных баз данных и достоинства MongoDB, важно понимать, что задачи бывают разные и методы их решения бывают разные. В какой-то ситуации MongoDB действительно улучшит производительность вашего приложения, например, если надо хранить сложные по структуре данные. В другой же ситуации лучше будет использовать традиционные реляционные базы данных. Кроме того, можно использовать смешанный подход: хранить один тип данных в MongoDB, а другой тип данных – в традиционных БД.

Вся система MongoDB может представлять не только одну базу данных, находящуюся на одном физическом сервере. Функциональность MongoDB позволяет расположить несколько баз данных на нескольких физических серверах, и эти базы данных смогут легко обмениваться данными и сохранять целостность.

Рассмотрим несколько свойств этой базы данных:

#### 1) Формат данных в MongoDB.

Одним из популярных стандартов обмена данными и их хранения является JSON (JavaScript Object Notation). JSON эффективно описывает сложные по структуре данные. Способ хранения данных в MongoDB в этом плане похож на JSON, хотя формально JSON не используется. Для хранения в MongoDB применяется формат, который называется BSON или сокращение от binary JSON.

BSON позволяет работать с данными быстрее: быстрее выполняется поиск и обработка. Хотя надо отметить, что BSON в отличие от хранения

данных в формате JSON имеет небольшой недостаток: в целом данные в JSON-формате занимают меньше места, чем в формате BSON, с другой стороны, данный недостаток с лихвой окупается скоростью.

## 2) Кроссплатформенность.

MongoDB написана на C++, поэтому ее легко портировать на самые разные платформы. MongoDB может быть развернута на платформах Windows, Linux, MacOS, Solaris. Можно также загрузить исходный код и самому скомпилировать MongoDB, но рекомендуется использовать библиотеки с официального сайта.

## 3) Коллекции.

Если в традиционном мире SQL есть таблицы, то в мире MongoDB есть коллекции. И если в реляционных БД таблицы хранят однотипные жестко структурированные объекты, то в коллекции могут содержать самые разные объекты, имеющие различную структуру и различный набор свойств.

## 4) Репликация.

Система хранения данных в MongoDB представляет набор реплик. В этом наборе есть основной узел, а также может быть набор вторичных узлов. Все вторичные узлы сохраняют целостность и автоматически обновляются вместе с обновлением главного узла. И если основной узел по каким-то причинам выходит из строя, то один из вторичных узлов становится главным.

## 5) Простота в использовании.

Отсутствие жесткой схемы базы данных и в связи с этим потребности при малейшем изменении концепции хранения данных пересоздавать эту схему значительно облегчают работу с базами данных MongoDB и дальнейшим их масштабированием. Кроме того, экономится время разработчиков. Им больше не надо думать о пересоздании базы данных и тратить время на построение сложных запросов.

## 6) GridFS.

Одной из проблем при работе с любыми системами баз данных является сохранение данных большого размера. Можно сохранять данные в файлах,

используя различные языки программирования. Некоторые СУБД предлагают специальные типы данных для хранения бинарных данных в БД (например, BLOB в MySQL).

В отличие от реляционных СУБД MongoDB позволяет сохранять различные документы с различным набором данных, однако при этом размер документа ограничивается 16 Мб. Но MongoDB предлагает решение - специальную технологию GridFS, которая позволяет хранить данные по размеру больше, чем 16 Мб.

Система GridFS состоит из двух коллекций. В первой коллекции, которая называется files, хранятся имена файлов, а также их метаданные, например, размер. А в другой коллекции, которая называется chunks, в виде небольших сегментов хранятся данные файлов, обычно сегментами по 256 Кб.

## **2.2 Среда разработки**

### **2.2.1 Visual Studio Code**

Visual Studio Code - это легкий, но мощный редактор исходного кода, который совместим с Windows, MacOS и Linux.

IDE поддерживает большое количество популярных языков программирования. Поддерживаемые расширения и среды выполнения представлены ниже:

- JavaScript;
- TypeScript;
- Node.js;
- C ++;
- C #;
- Java;
- Python;

- PHP;
- Go;
- .NET;
- Unity.

Для удобства разработки программных продуктов VS Code включает в себя встроенный отладчик, инструменты для работы с Git-репозиториями, подсветку синтаксиса, средства для рефакторинга и IntelliSense (технология автодополнения, которая предлагает команду по первым буквам).

IDE VS Code была анонсирована 29 апреля 2015 года компанией Microsoft в рамках конференции Blink. 18 ноября 2015 года редактор был выпущен в релиз под лицензией Массачусетского технологического института.

Visual Studio Code основан на Electron – фреймворк, позволяющий используя Node.js разрабатывать настольные приложения, работающие на движке Blink. Несмотря на то, что редактор основан на Electron, он не использует редактор Atom. Вместо него был реализован веб-редактор Monaco, разработанный для Visual Studio Online.

Интерфейс IDE Visual Studio Code представлен на рисунке 10.

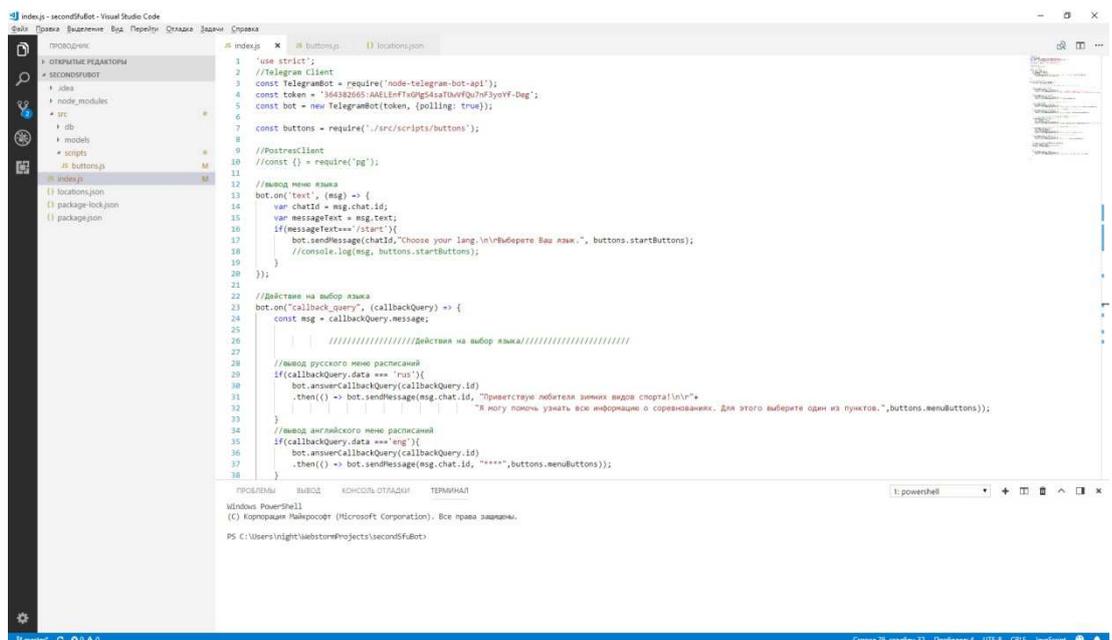


Рисунок 10 – Интерфейс IDE Visual Studio Code

## 3 Разработка информационной системы

### 3.1 Регистрация чат-бота для Telegram Bot API

Первым шагом разработки приложения является регистрация у специального чат-бота «BotFather». Регистрация начинается с команды «/newbot», после чего предлагается ввести название чат-бота с обязательным условием: в конце название должно быть указано «Bot» или «\_bot». Если все условия были удовлетворены, то «BotFather» выдает токен (специальный набор символов для доступа к HTTP API Telegram Bot) и URL-адрес для доступа к чат-боту. Пример регистрации чат-бота представлен на рисунке 11.

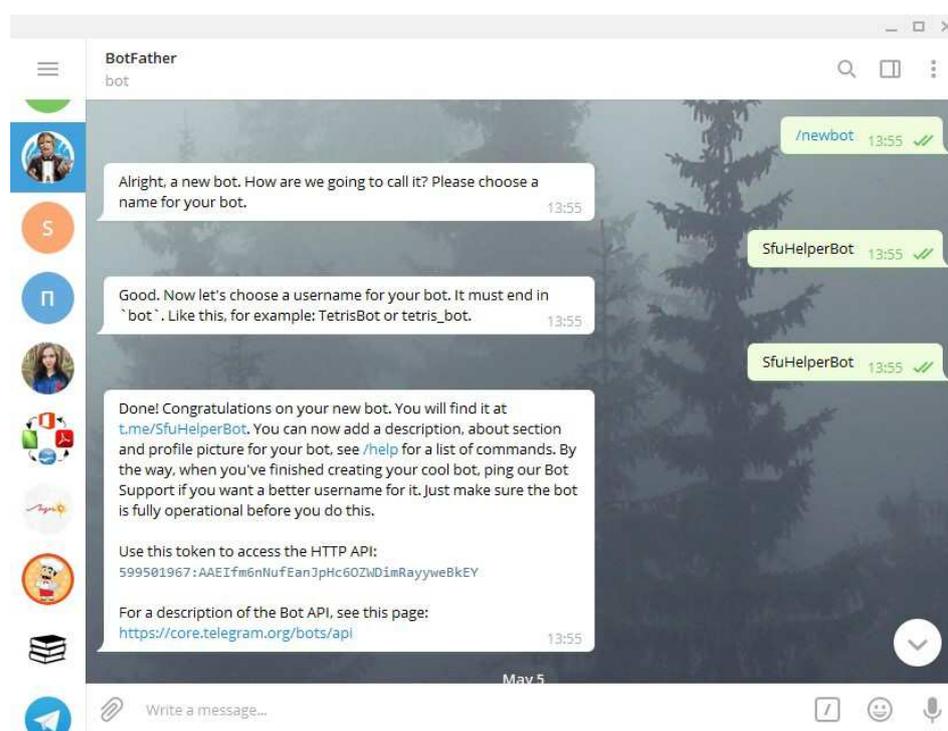


Рисунок 11 – Пример регистрации чат-бота

Для установки дополнительных параметров, таких как иконка чат-бота, приветственное сообщение, описание чат-бота, а так же удаление имеющихся чат-боты, существуют следующие команды (таблица 1).

Таблица 1 – Доступные команды для изменения чат-ботов

Команда	Описание
/setname	Изменяет существующее имя
/setdescription	Присваивает текст, который будет отображаться при первом открытии чат-бота
/setabouttext	Присваивает текст в поле «О чат-боте»
/setuserpic	Присваивает выбранную картинку
/setcommands	Позволяет создать список доступных команд
/deletebot	Удаляет выбранного чат-бота

Помимо команд для изменения основных параметров чат-бота существует ряд команд, которые позволяют выводить неизменяемые параметры (токен), а так же присваивать значения, которые представлены в таблице 2.

Таблица 2. – Доступные команды для дополнительной настройки чат-бота

Команда	Описание
/token	Возвращает полученный ранее токен у выбранного бота
/revoke	Аннулирует токен доступа к боту
/setinline	Включает или выключает возможность вызывать бота из других чатов
/setinlinegeo	Включает или выключает возможность передавать местоположение бота из другого чата
/setinlinefeedback	Позволяет получать информацию о количестве выбранных пользователями команд
/setjoingroup	Определяет может ли быть добавлен в групповые диалоги бот
/setprivacy	Включает режим конфиденциальности. В этом режиме бот получает, обрабатывает и отправляет обратно информацию отдельно для каждого пользователя в чате

После настроек на стороне Telegram и получении токена можно приступить к разработке программной части чат-бота.

## 3.2 Реализация клиентской части чат-бота

Пользователи Telegram могут взаимодействовать с чат-ботами 2 способами: команды («/start», «/new», «/location» и другие) с параметрами, либо встроенные клавиатуры (inline keyboards). Для удобства пользователей было принято решение сделать интерфейс со встроенной клавиатурой.

Для реализации поставленных целей потребуется пять различных меню:

- меню выбора языка;
- основное меню;
- меню выбора по дате;
- меню выбора по виду спорта;
- меню выбора расположений спортивных объектов

В мессенджере Telegram каждая клавиатура реализована как объект, а ее кнопки как JSON строка. В документации Telegram Bot API сказано, что каждая клавиатура должна иметь один обязательный параметр – имя кнопки (text), и шесть необязательных – ссылка (url), обратные данные (callback\_data), возможность встроенного запроса (switch\_inline\_query), возможность вывода клавиатуры из другого чата (switch\_inline\_query\_current\_chat), вызов описания запущенной игры (callback\_game) и кнопка с возможностью покупки (pay).

Пример реализации клавиатуры одного из меню чат-бота изображен на рисунке 12.

```
var menuButtons = {
  parse_mode: 'markdown',
  disable_web_page_preview: false,
  reply_markup: JSON.stringify({
    inline_keyboard:
      [
        [{text: `Сегодня`, callback_data:'today'}],
        [{text: `Расписание по дате`, callback_data:'date'}],
        [{text: `Расписание по виду спорта`, callback_data:'discipline'}],
        [{text: `Расположение спортивных объектов`, callback_data:'location'}]
      ]
  })
};
```

Рисунок 12 – Реализованная клавиатура основного меню

При первом запуске чат-бота пользователю предлагается выбрать между русским и английским языком (рисунок 13).

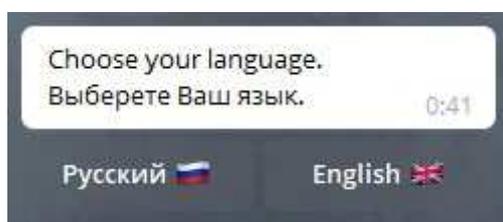


Рисунок 13 – Меню выбора языка

Следующей задачей была реализацию основного меню. Оно должно позволять пользователю выбрать, какую именно интересующую его информацию он хочет получить. Среди предоставляемой информации есть вывод всех спортивных событий на текущий день, а также возможность перехода в другие меню для получения более конкретной информации (рисунок 14).

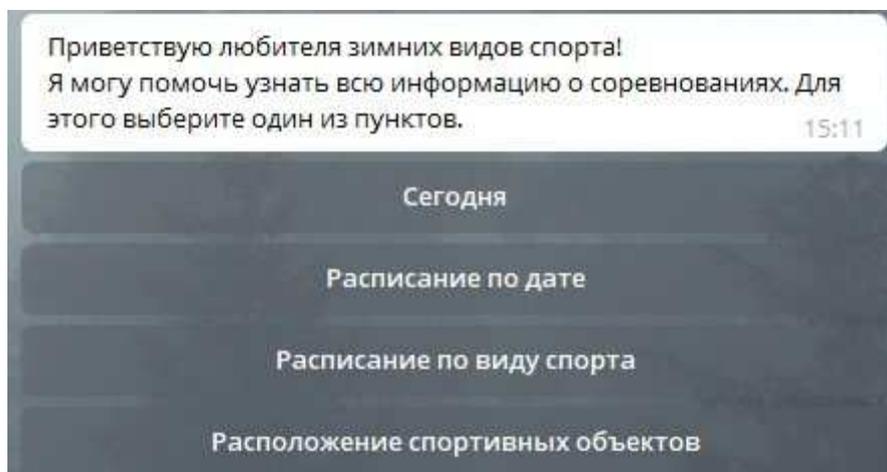


Рисунок 14 – Основное меню чат-бота

Для реализации одной из главных задач, а именно, отображение списка спортивных событий, которые произойдут в выбранный пользователем день. Зимняя Универсиада 2019 будет проходить в Красноярске в период со 2 по 12 марта 2019 года. Реализованное меню изображено на рисунке 15.

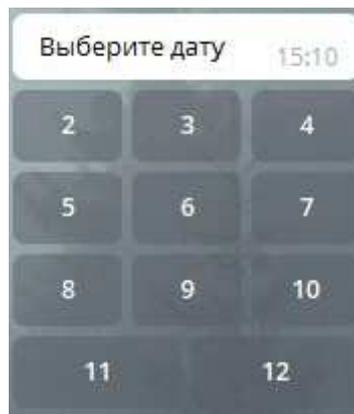


Рисунок 15 – Меню вывода событий по дате

Вторая реализуемая функция представляет собой меню из набора видов спорта, и в зависимости от выбора будет отображена информация обо всех датах проведения соревнований. Программа Зимней Универсиады 2019 состоит из одиннадцати видов спорта, которые в свою очередь делятся на дисциплины. Результат реализации представлен на рисунке 16.



Рисунок 16 – Меню выбора по видам спорта

Последним этапом реализации клиентской стороны чат-бота, было создание меню с возможностью выбора спортивного объекта с последующей навигацией. Все события будет проходить на одиннадцати спортивных сооружениях. Итог реализации данной задачи изображен на рисунке 17.

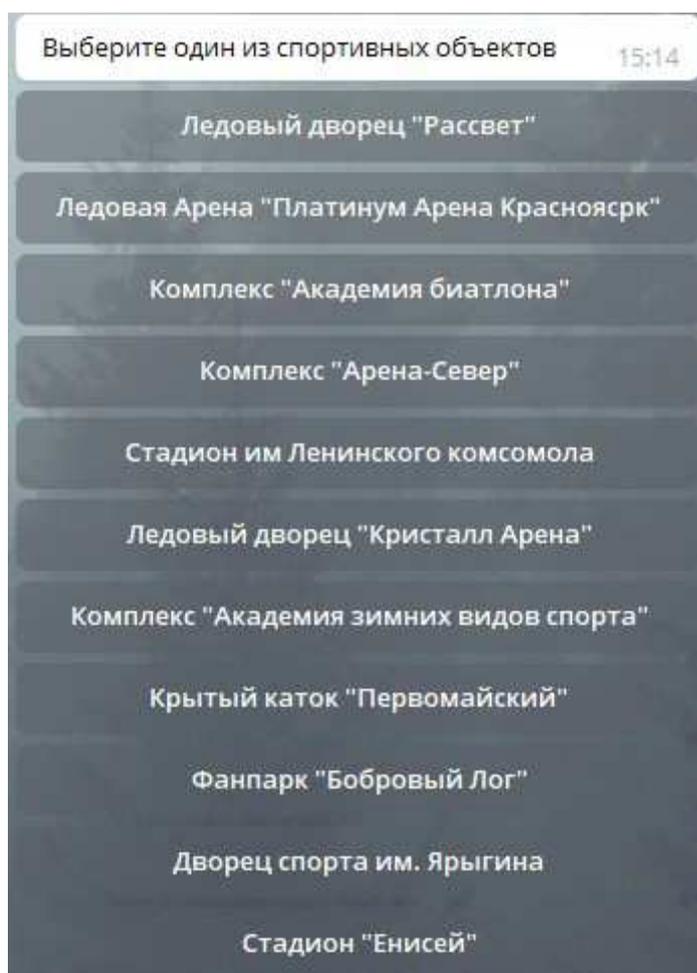


Рисунок 17 – Меню выбора спортивных объектов

Таким образом, была реализована клиентская часть чат-бота для Зимней Универсиады 2019 в городе Красноярске.

### **3.2 Проектирование и реализация базы данных чат-бота**

Для реализации поставленных задач появилась необходимость в базе данных (БД). В проекте используется noSQL БД MongoDB. База данных чат-бота представляет собой три таблицы. Диаграмма структуры БД представлена на рисунке 18.

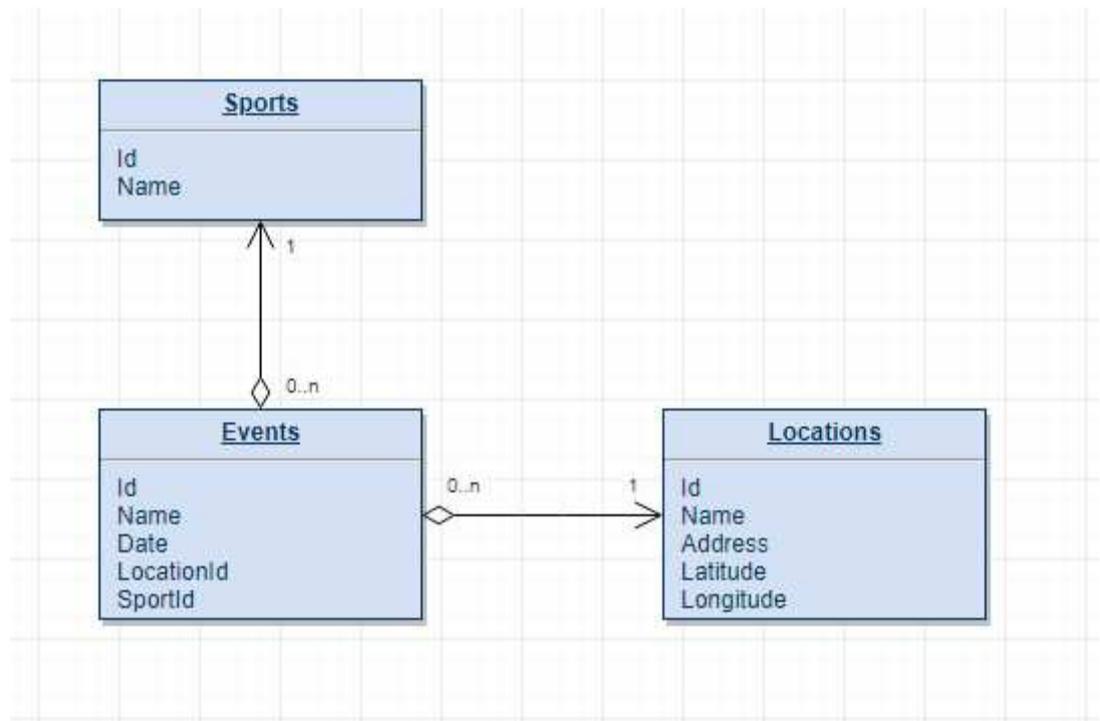


Рисунок 18 – Диаграмма структуры базы данных

Каждая таблица выполняет свою функцию:

- Sports – описывает все виды спорта, который будет представлен в рамках Зимней Универсиады 2019;
- Events – описывает все события, а именно их название, дату, место проведения и вид спорта;
- Location – описывает спортивные объекты на которых будут проходить события.

Таким образом, была реализована база данных для чат-бота на платформе Telegram.

### 3.3 Реализация серверной части чат-бота

Серверная часть чат-бота на платформе Telegram должна решать следующие задачи:

- получать и распознавать сообщение отправленное пользователем;

- извлекать и сравнивать полученную информацию для составления корректного ответа;
- извлекать информацию из базы данных путем выполнения запросов с полученными параметрами;
- составлять и отправлять ответ в виде текста, сформированного на основе информации пользователя.

Для того чтобы чат-бот имел возможность распознавать сообщение пользователя каждая кнопка inline-клавиатуры имеет так называемую обратную информацию (англ. callback data). После нажатия кнопки пользователем серверная часть сравнивает полученный ответ со значением или массивом значений.

В первую очередь была реализована функция навигации между спортивными объектами. Результат разработки изображен на рисунке 19.

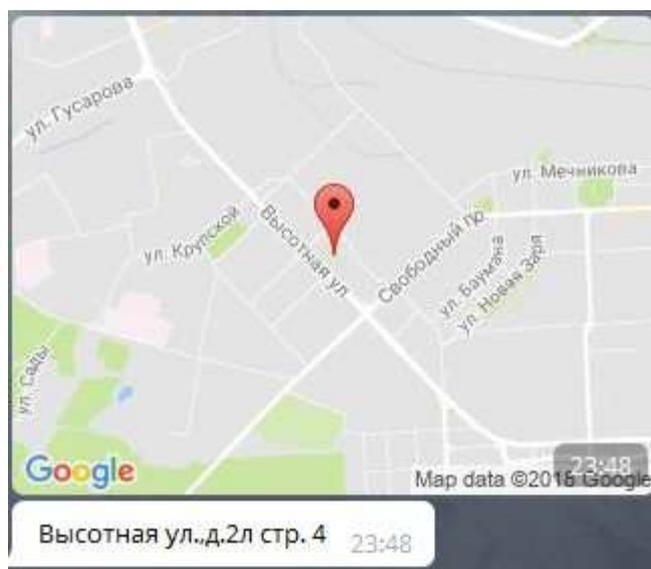


Рисунок 19 – Результат выполнения функции навигации между спортивными объектами

Для построения маршрута от текущего местоположения до спортивного объекта, необходимо нажать на карту, после чего откроется приложение с

построенным маршрутом. Пример построенного маршрута в приложении «2ГИС» для iOS 11 представлен на рисунке 20.

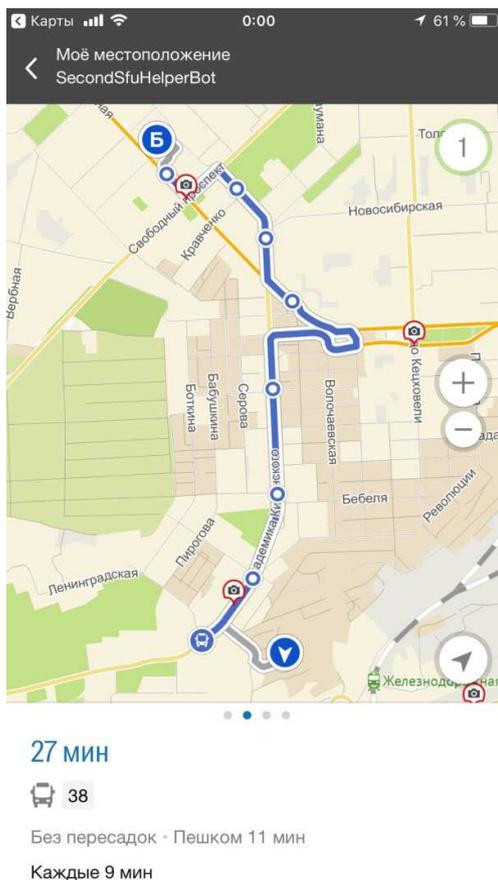


Рисунок 20 – Построенный маршрут по запрошенным данным

Второй реализованной функцией чат-бота является отображение событий на текущую или выбранную пользователем дату. Все расписание событий на день выводится одним сообщением (рисунок 21). Каждое событие характеризуется видом спорта, временем и местом проведения.



Рисунок 21 – Список событий на третье марта

Последней была реализована функция вывода расписания событий на все время проведения Зимней Универсиады 2019 по выбранному пользователем виду спорта. Каждое событие характеризуется временем, датой и местом проведения. Все события группируются и отправляются пользователю в виде одного сообщения (рисунок 22).



Рисунок 22 – Список событий по горнолыжному спорту

Таким образом, были реализованы следующие функции чат-бота на платформе Telegram:

- навигация между спортивными сооружениями Зимней Универсиады 2019;
- формирование списка спортивных событий на текущую или выбранную пользователем дату;
- формирование списка спортивных событий по выбранному пользователем виду спорта.

### 3.4 Размещение чат-бота на удаленном сервере Heroku

Для постоянного получения доступа к чат-боту, после реализации его необходимо разместить на удаленном сервере. В качестве платформы был выбран Heroku.

Размещение на облачной PaaS-платформе Heroku схоже с работой распределенной системы контроля версий (Git). Существует три способа развертывания:

- Heroku Git и Container Registry используя Heroku CLI;
- подключение аккаунта GitHub с автоматическим развертыванием;
- подключение аккаунта Dropbox с автоматическим развертыванием.

В выпускной квалификационной работе был выбран способ и использованием Heroku CLI. Для развертывания чат-бота были выполнены следующие действия:

- регистрация на облачной SaaS-платформе Heroku;
- скачен и установлен Heroku CLI;
- в терминале операционной системы или встроенном в IDE выполнен вход в аккаунт Heroku командо «heroku login»;
- клонирован репозиторий с удаленного Git-сервера Heroku на локальную машину при помощи команды «heroku git:clone -a <APP\_NAME>»;
- зафиксированы изменения в коде при помощи команд «git add .» и «git commit -am <COMMIT\_NAME>»;
- все зафиксированные изменения были отправлены на удаленный сервер облачных вычислений Heroku командой «git push heroku master».

После выполнения всех указанных команд в правильной последовательности начинается передача данных на удаленные сервера. Если развертывание прошло успешно, то на терминале отобразится справочная информация о состоянии приложения и режиме доступа к нему (рисунок 23).

```
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:      Procfile declares types      -> (none)
remote:      Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:      Done: 19.4M
remote: -----> Launching...
remote:      Released v3
remote:      https://sfutelegrambotdip.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sfutelegrambotdip.git
* [new branch]      master -> master
```

Рисунок 23 – Результат развертывания чат-бота

Таким образом, чат-бот был развернут на удаленном сервере облачных вычислений Heroku, что позволит пользователям получать доступ бесперебойно.

## ЗАКЛЮЧЕНИЕ

В настоящее время популярность мессенджеров как средств общения неизменно растет. Компании, семьи, друзья ежедневно пользуются возможностями обмена сообщениями и медиаконтентом на расстоянии. Так же стоит отметить рост популярности такого вида программных продуктов как чат-боты, которые работают на платформах мессенджеров.

Круглосуточная служба поддержки пользователей, конвертирование документов и медиафайлов, заказ такси, поиск необходимых данных и многое другое в настоящее время может быть реализовано в рамках лишь одного мессенджера. Пользователям не придется скачивать множество приложений для решения узконаправленных задач, потому что достаточно иметь лишь мессенджер и необходимый набор чат-ботов, которые не занимают место в памяти смартфона.

В рамках выпускной квалификационной работы были выполнены поставленные задачи. Во-первых, были изучены мессенджеры. Было проведено сравнение и анализ достоинств и недостатков, вследствие чего был выбран мессенджер Telegram как самый удобный и доступный в плане документации Telegram Bot API.

Во-вторых, были изучены имеющиеся аналоги чат-бота на платформе Telegram, а также выявлены их достоинства, недостатки и интересные решения. На основе этого были выявлены требования для разработки авторского чат-бота посвященного Зимней Универсиаде 2019 в городе Красноярске.

В рамках последней выполненной задачей были выбраны технологии и среда для разработки чат-бота среди которых Node.JS, MongoDB, SaaS Heroku и VS Code.

Таким образом, результатом выпускной квалификационной работы является полностью реализованный чат-бот для помощи гостям и жителям города во время проведения Зимней Универсиады 2019 в Красноярске.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Messina, C. 2016 will be the year of conversational commerce [Электронный ресурс] / C. Messina // Medium. – 2016. – Режим доступа: <https://medium.com/chris-messina/2016-will-be-the-year-of-conversational-commerce-1586e85e3991>
- 2) Telegram Bot API [Электронный ресурс] : Telegram Documents. – Режим доступа: <https://core.telegram.org/bots/api>
- 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](https://elibrary.ru/download/elibrary_29653726_34734017.pdf)
- 4) Агальцов, В. Базы данных: В 2-х книгах. Книга 2: Распределенные и удаленные базы данных : Учебник / В. Агальцов – Москва : Форум, 2018. – 271 с.
- 5) Архаков, Д. NodeJS: Делаем кнопки в Telegram API (inline-keyboards) [Электронный ресурс] / Д. Архаков // Блог о программировании. – 2016. – Режим доступа: <https://archakov.im/post/nodejs-make-buttons-on-telegram-api.html>
- 6) 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\\_34058358.pdf](https://elibrary.ru/download/elibrary_26673675_34058358.pdf)
- 7) 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>
- 8) Матвеева Н. Ю., Технологии создания и применения чат-ботов [Электронный ресурс] / Н. Ю. Матвеева, А. В. Золотарюк. // Научные записки

молодых исследователей. – 2018. – №1. – с. 28-30. – Режим доступа: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chat-botov>

9) Официальный сайт Node.JS [Электронный ресурс]: About – Режим доступа: <https://www.nodejs.org/>.

10) Официальный сайт Telegtam [Электронный ресурс]: API – Режим доступа: <https://core.telegram.org/api>.

11) 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>

12) Холмс, С. Стек MEAN. Mongo, Express, Angular, Node. : Учебное пособие / С. Холмс – Санкт-Петербург : Питер Пресс, 2017. – 496 с.

13) Хэррон, Д. Node.js. Разработка серверных веб-приложений на JavaScript. : Учебное пособие / Д. Хэррон – Москва : ДМК Пресс, 2014. – 144 с.

14) 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\\_whatsapp\\_stal\\_samym\\_populyarnym\\_messendzheram\\_v](http://www.cnews.ru/news/top/2018-02-28_whatsapp_stal_samym_populyarnym_messendzheram_v)

15) Янг, А. Node.js в действии. : Учебное пособие / А Янг., Б Мек., М Кантелон. – Санкт-Петербург : Питер СПб, 2018. – 432 с.

16) СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 9.01.2014. – Красноярск: ИПК СФУ, 2014. – 60 с.

17) Прамодкумар, Д. NoSQL: новая методология разработки нереляционных баз данных.: пер с англ. / Д. Прамодкумар, Садаладж, М. Фаулер; под ред. С.Н. Тригуб. – Москва.: Вильямс, 2013 - 192 с.

18) Советов, Б.Я. Информационные технологии.: учебное пособие / Б. Я. Советов, В. В. Цехановский. – Москва.: Высш. шк., 2003. - 263 с.

19) Флэнаган, Д. JavaScript: карманный справочник, 3-е изд.: пер с англ. / Д.Флэнаган; под ред. С.Н. Тригуб – Москва.: Вильямс, 2013. – 230 с.

- 20) Шаши, Ш. Основы построения баз данных.: пер с англ. / Ш. Шаши. – Москва.: КУДИЦ-ОБРАЗ, 2004. – 336с.
- 21) Пауэрс, Ш. Изучаем Node.js: пер с англ./ Ш. Пауэрс. - Санкт-Петербург: Питер, 2014. – 400 с.
- 22) Официальный сайт Heroku [Электронный ресурс]: Облачная PaaS-платформа. – Режим доступа: <https://www.heroku.com/>.
- 23) Рейтинг систем управления базами данных 2016 [Электронный ресурс] // Tagline – рейтинги сервисов и технологий - Режим доступа: <http://tagline.ru/database-management-systems-rating>.
- 24) Хабрахабр [Электронный ресурс] : Node.js, Express с MongoDB: API за полчаса. – Режим доступа: <https://habrahabr.ru/company/ruvds/blog/321104/>
- 25) Бэнкер, К. MongoDB в действии : учебное пособие / К. Бэнкер – Москва : ДМК Пресс, 2012. – 394 с.

## ПРИЛОЖЕНИЕ А

### Преимущества и недостатки наиболее популярных мессенджеров

Название	Достоинства	Недостатки
WhatsApp	Доступ по протоколу HTTPS	Существенное снижение качества передаваемых медиафайлов
	Групповые чаты до 256 человек	
	Доступ к чатам в режиме оффлайн	Большая задержка в видеочате
	Удобные голосовые и видеозвонки	
Viber	Push-уведомления в оффлайн-режиме	Наличие большого количества рекламы
	Встроенные службы для оплаты в сети Интернет	Наличие встроенных покупок
	Наличие публичных аккаунтов	
Skype	Демонстрация экрана во время звонка или видеотрансляции	Нестабильность соединений
	Удобные групповые звонки и видеоконференции	Сильная зависимость от скорости Интернет соединения
		Отсутствие шифрования
iMessage	Большой набор стандартных анимаций чата	Большая задержка доставки сообщений
	Высокая степень интеграции со сторонними приложениями Apple	Работает только на системе iOS
		Невозможность создания чат-ботов
Hangouts	Особый протокол передачи голоса Google Voice	Не интуитивно понятный интерфейс
	История сообщений хранится на удаленном сервере	Сильно сжатие медиафайлов
	Интегрированные сервисы Google+ и Gmail	
Facebook Messenger	Поддержка ботов	Сообщения нельзя удалять либо редактировать
	Шифрование сообщений	
	Возможность подтверждения прочтения из Push-уведомления	Нет поддержки десктопных ОС
	Функция «Секретный чат»	
Самоуничтожение выбранных сообщений через указанное время	Конфиденциальность данных	
Telegram	Самая удобная платформа для чат-ботов	Отсутствуют видеозвонки
	Открытый исходный код	
	Встроенный редактор фото	
	Специально созданный протокол шифрования MTProto	Десктопные версии не поддерживают секретные чаты
	Секретные чаты и самоуничтожающиеся сообщения	

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и Информационных Технологий  
институт

Информационные системы  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

  
подпись

Л.С. Троценко  
инициалы, фамилия

«13» июня 2018 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 Информационные системы и технологии

Бот-помощник для мессенджера Telegram

Руководитель

  
подпись, дата

13.06.18  
доцент, к.т.н  
должность, ученая степень

И.А. Легалов  
инициалы, фамилия

Выпускник

  
подпись, дата

13.06.18

М.С. Русин  
инициалы, фамилия

Нормоконтролер

  
подпись, дата

13.06.18  
ст. преподаватель  
должность, ученая степень

Ю.В. Шмагрис  
инициалы, фамилия

Красноярск 2018