МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Благовещенский государственный педагогический университет»

Физико-математический факультет

Кафедра информатики и методики преподавания информатики

КУРСОВАЯ РАБОТА

на тему: Разработка чат-бота компаньона на платформе Telegram

Исполнитель:

студент ФМФ, группы 3 «Б»

дата подпись

Д.А. Елякин

Руководитель:

ст. преподаватель \_\_\_А.А. Ягелло

дата подпись

|  |  |  |
| --- | --- | --- |
| Защита состоялась | « » 202 г. | Оценка |

Благовещенск 2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc25302)

[1 ОПИСАНИЕ МЕТОДОВ РАЗРАБОТКИ 4](#_Toc13573)

[1.1 Мессенджеры 4](#_Toc25830)

[1.2 Чат-боты 8](#_Toc22053)

[1.4 Выбор среды разработки 9](#_Toc25395)

[1.5 Telegram Bot API 12](#_Toc12123)

[2 РАЗРАБОТКА ТЕЛЕГРАМ-БОТА 14](#_Toc28634)

[2.1 Регистрация чат-бота 14](#_Toc679)

[2.2 Разработка функциональных возможностей чат-бота 15](#_Toc17361)

[2.3 Подготовка платформы к работе 17](#_Toc28756)

[2.4 Проектировка чат-бота 18](#_Toc17590)

[ЗАКЛЮЧЕНИЕ 26](#_Toc31744)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#_Toc12866)

ВВЕДЕНИЕ

Мессенджеры и социальные сети – это уникальные среды для всех сфер деятельности. В данное время в мире есть множество возможностей, вариантов и средств общения, и огромная часть так или иначе связана с современными техническими возможностями, которые, в частности, представлены использованием глобальной компьютерной сети.

В данной работе рассматривается процесс создания телеграмм-бота и приводится его реализация. Эта тема считается актуальной, поскольку мессенджеры, и в особенности чат-боты, очень популярны среди пользователей интернета. Чат-боты могут выполнять различные задачи, например, предоставлять информацию о погоде или последних новостях. Основным преимуществом чат-ботов перед обычными приложениями является то, что они могут выполнять все функции в рамках одного мессенджера.

Цель курсовой работы - создание бота для оказания помощи пользователям онлайн-игры “Escape from Tarkov” в мессенджере Telegram.

Задачи курсовой работы:

1. Провести сравнительный анализ решений в этой области.
2. Спроектировать работу телеграмм-бота.
3. Разработать рабочую версию бота.
4. Провести тестирование программного продукта.

1 ОПИСАНИЕ МЕТОДОВ РАЗРАБОТКИ

* 1. Мессенджеры

Своё начало общение людей в сети интернет берет с чатов, потом приоритет отдали мессенджерам, после на первое место встали социальные сети, но не так давно мессенджеры снова возглавили этот список, став самым перспективным сервисом для общения. Это было подтверждено исследованиями компании «Yota» в 2022 году: количество пользователей, использующих мессенджеры в Российской Федерации, увеличилось на более чем 15%. А также, по средствам опроса пользователей, они определили, как формируется новый цифровой этикет [1].

Мессенджеры пользуются популярностью в основном у людей среднего возраста, так на долю лиц младше 18 лет будет приходится всего 8% от всего количества аудитории, пользователи в возрасте до 25 лет составляют 10%, от 25 до 35 лет 32%, от 35 до 45 лет – 26%. При этом 13% людей от 45 до 55 лет, а 8% - от 55 до 64 лет. И 3% пользователей находится в возрасте старше 64-х лет (рисунок 1).

Рисунок 1 – Диаграмма возрастных категорий пользователей.

Ещё одной весомой причиной резкой популярности мессенджеров является изменение тарифов мобильного интернета. МТС предоставляет пакет с безлимитным трафиком на мессенджеры. Широкое распространение смартфонов любых ценовых категорий, высокие скорости и низкие цены.

В данный момент существует большое разнообразие мессенджеров, имеющие свои плюсы и минусы. Особого внимания достоин мессенджер Telegram.

Telegram – это кроссплатформенное приложение, разработанное на языке программирования C++, оно позволяет обмениваться сообщениями и файлами большинства форматов. Мессенджер использует специально разработанную серверную часть с закрытым кодом, работающим на серверах Германии и США.

Telegram имеет ряд преимуществ:

1. Приватность – все чаты зашифровываются, а если выбрать специальную приватность чата, то сообщения уничтожатся через выбранное время;
2. Быстрота – скорость отправления сообщений выше, чем у одноклассников;
3. Распределение – сервера Telegram находятся в большом количестве регионов мира, что повышает отказоустойчивость;
4. Открытость – использование открытого протокола MTProto и API, бесплатных для всех;

Отличительными качествами Telegram является кроссплатформенность, надежность доставки сообщений и скорость.

Мессенджер WhatsApp, обладает хорошей производительностью в целом, однако возникают трудности с доставкой сообщений в случае отсутствия или низкого качества связи (например, в метро или подвальных помещениях). Кроме того, несмотря на наличие десктопного приложения, невозможность его скачивания из-за ограничений в некоторых регионах, а также функциональные ограничения в веб-версии, создают неудобства для пользователей на их устройствах.

Telegram выглядит гораздо лучше во всех этих аспектах, а также клиент для Windows и приложение Android работают крайне быстро.

Telegram разрабатывает свой API для ботов, и с каждым днем их становится все больше. У каждого бота также есть имя пользователя, поэтому его можно найти в глобальном поиске.

На сегодняшний день, приложение Telegram является одним из самых востребованных мессенджеров во всем мире. Простота использования, лёгкость установки, доступность на множестве платформ и операционных систем и бесплатная модель распространения позволили мессенджеру уверенно закрепиться на рынке. Одной из главных и уникальных функций данного приложения является возможность создания пользователем чат-бота, которого можно спроектировать практически под любые нужды.

Все компании активно начинают внедрять в свою работу чат-ботов и использование мессенджеров. Это происходит во всех отраслях. Все это из-за того, что там сконцентрирована самая большая и живая аудитория. Они же и являются потенциальными потребителями.

1.2 Чат-боты

Чат-ботов можно разделить на категории, в зависимости от форматов взаимодействия с ним.

Кнопочный чат-бот. Такой бот взаимодействует с пользователем посредством кнопок (уже готовых запросов), другими словами, связь весьма ограничена группой созданных предварительно фраз, без возможности текстового ввода.

Текстовый чат-бот. Данный вид уже более прогрессивный и функциональный: возникает возможность текстового ввода, происходит анализ запроса и выдается ответ. Следовательно, мы можем вести диалог между ботом и пользователем [5].

Некоторые боты, естественно, не будут поддерживать взаимодействие, так как они предназначены под узкую задачу вроде автосообщений в новостном канале.

Встраиваемые (inline) боты умеют принимать команды из других чатов. Например, gif или pic можно вызвать из любого чата для поиска подходящей к теме картинки [4].

Если углубиться в статистику использования чат-ботов в наше время, то можно сделать ряд выводов. Боты — это некие универсальные средства решения множества проблем и жизненных задач, таких как: общение с людьми, развлечение (игры, музыка, фильмы и сериалы), здоровье (бот для записи в платную клинику), заказ продуктов или товаров.

Чат-бот является прикладной программой, которая логически преобразует информацию, полученную от пользователя, в нужный ему ответ. Предоставляет человеку возможность без участия живого человека по ту сторону, получить информацию, которая ему нужна.

1.4 Выбор среды разработки

1.4.1 Python

Python или же «пайтон» стоит на высоком уровне среди языков программирования. Он является полноценным скриптовым языком программирования. Его уникальность в том, что он универсален и подходит для решения различных сложных задач. Пайтон применяется во многих IT сферах, например, машинное обучение или разработка различных приложений [8].

Плюсы Python. Python это интерпретируемый язык. Он не компилируется как обычный язык. Сам язык отлично спроектирован и логичен.

Примечателен язык также в том, что строчки кода сокращены до минимума. Простота языка — это хорошее качество, так как разработка программы ведется гораздо быстрее.

Минусами языка программирования Python является слабая скорость языка. Пайтон не очень хороший вариант для выполнения задач, требующих больших объёмов памяти. Глобальный блок интерпретатора (GIL) не позволяет одновременно выполнить несколько потоков.

Вывод: Пайтон – популярный и универсальный язык программирования. Он востребован во многих сферах. Высокоуровневый язык программирования со своими достоинствами и недостатками.

1.4.2 Java

Java является языком ООП подхода, то есть следует объектно-ориентированной парадигме. Это делает его языком общего назначения. Язык используется в различных сферах, будь то мобильные приложения или десктопные, сетевые или корпоративные.

Плюсы языка включают в себя: данный язык программирования имеет концепцию ООП. Можно указать тип данных и структуру, а также определить некий набор функций присущий его типу. Структура данных становится объектом, с которым можно взаимодействовать для создания связей с другими объектами. Помимо этого, Java умеет использовать многопоточность процессора что ускоряет работу приложения. А также присутствует автоматическое управление памятью (АММ).

Java – высокоуровневый язык с довольно простым синтаксисом и хорошим подходом к освоению. Язык схож с привычный разговорный. Такие языки преобразуются с помощью компиляторов и интерпретаторов, что делает Java языком высокого уровня. Это значительно позволяет упростить разработку, делая её простой для писания кода, а также его чтения и модерирования.

К минусам языка относится:

многословный и сложный код. Длинные и сложные операторы затрудняют изучение.

Низкая производительность. Как и у своих сверстников Java имеет довольно низкую производительность. Это связано с компиляцией при помощи виртуальной машины. Но это не единственная проблема. Если плохо настроить процедуру хеширования это приведет к чрезмерному использованию памяти. Может произойти взаимная блокировка потоков. Виной этому послужит попытка потоков получить доступ к одному ресурсу.

1.5 Telegram Bot API

Это обычный HTTP-interface для взаимодействия с чат-ботами в мессенджере Telegram. Каждый созданный бот должен уметь автоматически отправлять сообщения и принимать входящие. Для этого нужен сервер, принимающий http запросы и выдающий ответы.[8]

Есть всего 2 способа получать обновления с сервера:

– long pulling – Telegram сам будет раз в какое-то время, например в 1 секунду, спрашивать сервер о новых уведомлениях;

– webhook – или сервера сами будут присылать оповещение, сразу как появится новое уведомление [4].

Все оповещения будут сохранены на сервере без обработки, но не долго, максимум до 24 часов. В ответ будет отправлен объект «Апдейт», заранее се риализованный в JSON.

Все запросы к Telegram Bot API должны осуществляться через HTTPS в следующем виде: [https://api.telegram.org/bot<token>/НАЗВАНИЕ\_МЕТОДА.](https://api.telegram.org/bot%3ctoken%3e/ÐšÐ’ÐŠÐ™Ð’ÐšÐŸÐŁ_ÐœÐŁÐ¢ÐžÐﬂÐ’) Принцип работы взаимодействия чат-бота и пользователя изображен на рисунке 9.

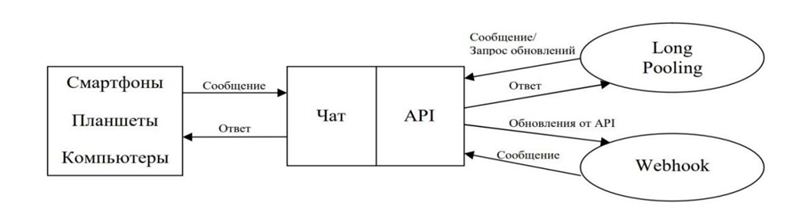


Рисунок 9 – Принцип работы чат-бота на платформе Telegram.

Вывод: В данном разделе были рассмотрены методы разработки. Данное исследование позволило сделать ряд выводов. Были изучены мессенджеры и их актуальность. Был проведён анализ полезности чат-ботов. Выявлено что самый лучший вариант платформы для разработки чат-бота является мессенджер Telegram. Для разработки программы выбран язык программирования Java.

2 РАЗРАБОТКА ТЕЛЕГРАМ-БОТА

2.1 Регистрация чат-бота

Перед началом создания телеграм-бота, была проведена регистрация бота в Telegram с помощью официального чат-бота «BotFather». Новому чат-боту было дано имя и тег (Рисунок 10)/

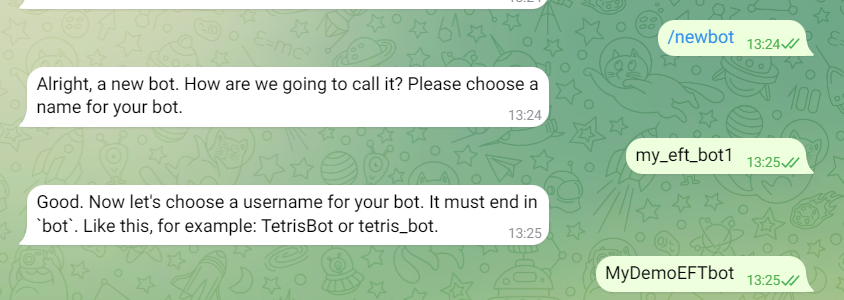


Рисунок 10 – Регистрация бота.

После чего «BotFather» сгенерировал личный ID разрабатываемого бота и зарегистрировал его в мессенджере Telegram (Рисунок 11).

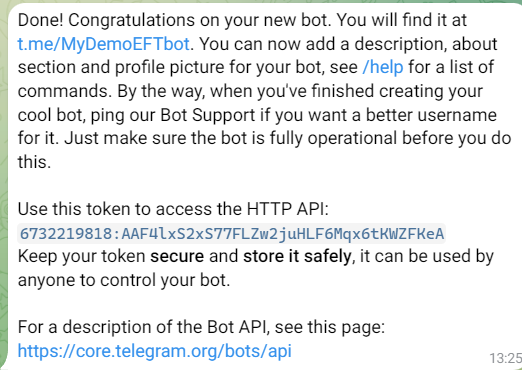


Рисунок 11 – Токен бота.

2.2 Разработка функциональных возможностей чат-бота

Чат-бот в Telegram создан в качестве персонального компаньона для оперативного доступа к необходимой информации, хранящейся в собственной базе данных бота. Основывается разработка на   [онлайн-игре](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%B8%D0%B3%D1%80%D0%B0) Escape from Tarkov (с [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) — «Побег из Тáркова») в жанре [тактического шутера](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%88%D1%83%D1%82%D0%B5%D1%80) [от первого лица](https://ru.wikipedia.org/wiki/%D0%A8%D1%83%D1%82%D0%B5%D1%80_%D0%BE%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BB%D0%B8%D1%86%D0%B0) с элементами [RPG](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%80%D0%BE%D0%BB%D0%B5%D0%B2%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) и [MMO](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%81%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B0%D1%8F_%D0%BE%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%B8%D0%B3%D1%80%D0%B0), разрабатываемая и издаваемая Русской компанией Battlestate Games.

В игре присутствует значительное количество внутриигровых предметов, с которыми необходимо взаимодействовать. Кроме того, в игре представлено множество заданий различной сложности. Все это стимулирует постоянное обращение к сторонним ресурсам для получения информации по различным аспектам игры. В дополнение, игра требует от пользователя высокой концентрации и внимания, что создаёт определённые условия, в которых важно длительное сохранение контроля над ситуацией, минимизируя отвлечение на запуск браузера, поиск сайта и извлечение необходимой информации.

Для оптимизации времени между возникновением проблемы и ее решением через поиск информации, необходимо перенести весь процесс в удобное мобильное приложение. А также, важно сохранить возможность использования этого приложения с компьютера. Помимо этого, проект доложен отвечать другим требованиям:

1. Рассылать сообщения пользователям об обновлениях.
2. Вести логи ошибок и логи использования бота.
3. Сохранять информацию пользователей.
4. Управление ботом должно быть максимально эргономичным и удобным для использования.

Для создания такого чат-бота, который поможет игрокам быстро получать информацию об игровых предметах и заданиях были поставлены следующие залачи:

Разработать базу данных, в которой будет храниться информация об игровых предметах, заданиях и других важных данных.

Разработать удобный интерфейс для мобильного приложения или веб-приложения, чтобы пользователи могли легко искать информацию. Интерфейс бота должен быть интуитивно понятным и удобным для использования на смартфонах и компьютерах.

Протестировать и отладить ошибки, устранить недочёты работы.

Такое решение может значительно упростить процесс получения информации об игровых аспектах, сократив время между возникновением проблемы и её решением.

2.3 Подготовка платформы к работе

Для реализации бота выбрано приложение Open Server Panel и PhpMyAdmin для создания базы данных бота, IntelliJ IDEA Community Edition для создания самого программного продукта.

Open Server Panel — это портативная программная среда, позволяющая запустить веб-приложение PhpMyAdmin представляющее собой веб-интерфейс для администрирования СУБД MySQL.

IntelliJ IDEA Community Edition — ведущая IDE для разработки программ на Java и Kotlin. Она помогает работать продуктивнее за счёт интеллектуальной помощи в написании кода, надёжных рефакторингов, быстрой навигации по коду, широкого набора встроенных инструментов разработчика и многих других полезных возможностей.

Помимо этого, для уменьшения и автоматизации процессов разработки приложения на языке Java было принято решение использовать фреймворк: Spring.

Spring или Spring Framework — фреймворк для языка программирования Java. Он нужен, чтобы разработчикам было легче проектировать и создавать приложения. Spring не связан с конкретной моделью программирования, поэтому его могут использовать как каркас для разных видов приложений.

2.3.1 Установка и настройка Open SP

Скачав и установив клиент Open SP с официального сайта разработчика, был подключен модуль «MySQL-8.0». С помощью автоматически установленного приложения PhpMyAdmin было создано новая база данных «tg-bots» и оставлена пустой, так как программа будет сама создавать нужные таблицы и наполнять их.

2.3.2 Установка и настройка IntelliJ IDEA

IntelliJ IDEA Community Edition скачен с официального сайта и после установки настроена 21 версия Java. В дальнейших настройках компилятор не нуждается так как в дальнейшем после запись библиотек в файл зависимостей, компилятор автоматически загрузит необходимые библиотеки из сети Интернет.

2.3.3 Подключение фреймворка Spring

На сайте Spring initializer есть возможность указав необходимые параметры сгенерировать проект, с которого начнется разработка.

Был выбран фреймворк maven, язык разработки Java. Версия Spring 3.0.12, версия SDK 21.

2.4 Проектировка чат-бота

Создание программы происходит в проекте, сгенерированном на сайте Spring initializer. Первым этапом является заполнение ресурсного файла, из которого программа будет брать необходимые ей параметры для подключения к базе данных и бэкенду Telegram (Листинг 1).

Листинг 1 – Содержание ресурсного файла.

bot.name=EFTbotGUIDE

bot.token=6732219818:AAEsH3IWRn8hMNHCM5npEMwXnPaPD6Mhhec

bot.owner=1263564651

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/tg-bot

spring.datasource.username=root

spring.datasource.password=Parol1/5

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect

spring.jpa.show-sql=true

Далее добавлены все зависимости для загрузки библиотек.

<!--Подключение библиотеки телеграма-->  
<dependency>  
 <groupId>org.telegram</groupId>  
 <artifactId>telegrambots</artifactId>  
 <version>${telegram.version}</version>  
</dependency>  
<!--Подключение библиотек java позволяющая сократить шаблонный код-->  
<dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 </dependency>  
<!--Библиотека для подключения бота к БД -->  
<dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.33</version>  
</dependency>  
<!--Подключение библиотеки спринга для того чтобы бот проходил проверку в БД-->  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-validation</artifactId>  
</dependency>  
<!--Подключение библиотеки эмоджи-->  
<dependency>  
 <groupId>com.vdurmont</groupId>  
 <artifactId>emoji-java</artifactId>  
 <version>5.1.1</version>  
</dependency>

После необходимо было приступить к созданию файла конфигурации, который будет получать токен бота, имя бота и код чата администратора (Листинг 3).

Листинг 3 – Класс конфигурации.

public class BotConfig {

//Сюда обращается главный класс TelegramBot чтобы получить токен и имя

//бота для предоставления бэкенду телеграма

@Value("&{bot.name}")

String botName;

@Value("${bot.token}")

String token;

@Value("${bot.owner}")

Long ownerID;

}

Для подключения бота к Telegram API важно прописать класс инициализации. Без этого бот не сможет подключиться к бекэнду телеграма (Листинг 4).

Листинг 4 – Класс инициализации.

public void init () throws TelegramApiException {  
 TelegramBotsApi telegramBotsApi = new TelegramBotsApi(DefaultBotSession.class);  
 try {  
 telegramBotsApi.registerBot(bot);  
 }  
 catch (TelegramApiException e){  
 log.error("Error: " + e.getMessage());  
 }  
 }  
}

Подготовительные действия завершены, в следующем этапе разработки нужно наполнить главный класс и сразу провести тестирование новых возможностей бота. Первое что добавляется — это кнопки, которые будут расположены в нижнем левом углу интерфейса (Листинг 5).

Листинг 5 – Реализация кнопок.

List<BotCommand> listofCommand = new ArrayList<>();  
listofCommand.add(new BotCommand("/start", "Приветсвие бота"));  
listofCommand.add(new BotCommand("/help", "Информация по использованию бота"));

Кнопки, расположенные в этом месте, отвечают за управление, настройку и администрирования бота. Проверка показывает, что название команд и описание выводится корректно (Рисунок 12).

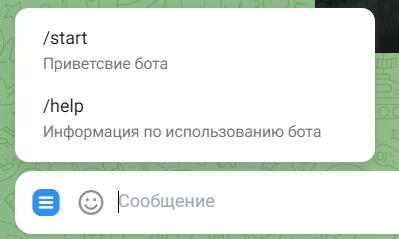


Рисунок 12 – Кнопки управления.

Следующая функция для реализации это - считывание основных команд (Листинг 6).

Листинг 6 – Прописанные команды бота.

switch (messageText) {  
 //При вызове пользователем этой команды, бот вызывает метод проверки регистрации пользователя в бд.  
 case "/start", "старт", "start", "Привет", "Hi", "привет":  
 registerUser(update.getMessage());  
 startCommandReceived(chatID, update.getMessage().getChat().getFirstName());  
 break;  
 case "/help", "помощь", "help":  
 sendMessage(chatID, HELP\_TEXT);  
 break;  
 case "/quests", "квесты", "задания", "Квесты", "Quests", "quests":  
 questschoice(chatID);  
 break;  
 case "/ammo", "патроны", "Ammo", "Патроны", "ammo":  
 ammochoice(chatID);  
 break;

default: //ответ бота на не определённые комнады  
 commandNotFound(chatID);  
 break;  
}

Чат-бот считывает и проверяет относится ли текст, отправленный пользователем к командам, если нет, то вызывает метод, который указывает пользователю на неверный ввод команды.

Для быстрой навигации пользователя по возможностям предлагаемым ему чат-ботом необходимо создать клавиатуру с перечнем аспектов игры (Листинг 7).

Листинг 7 – Создание экранной клавиатуры с заготовленными командами.

ReplyKeyboardMarkup keyboardMarkup = new ReplyKeyboardMarkup();  
List<KeyboardRow> keyboardRows = new ArrayList<>();  
KeyboardRow row = new KeyboardRow();  
row.add("Квесты");  
row.add("Патроны");  
keyboardRows.add(row);  
row = new KeyboardRow();  
row.add("Броня");  
row.add("Сборки оружия");  
row.add("Новости");  
keyboardRows.add(row);  
keyboardMarkup.setKeyboard(keyboardRows);  
message.setReplyMarkup(keyboardMarkup);  
executeMessage(message);

Тестирование клавиатуры не выявило ошибок и недочетов, команды исполняются без задержек. Визуальное представление клавиатуры на рисунке 13.

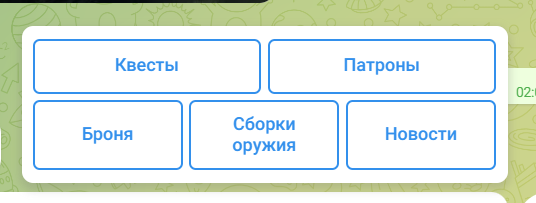


Рисунок 13 – Экранная клавиатура.

Для реализации рассылки, ведения статистики, и сохранения пользователей в базу данных создан метод, который регистрирует пользователей при нажатии ими на кнопку «Старт» (Листинг 8).

private void registerUser(Message msg) {  
 if(userRepository.findById(msg.getChatId()).isEmpty()){  
 var chatId = msg.getChatId();  
 var chat = msg.getChat();  
 User user = new User();  
 user.setChatID(chatId);  
 user.setFirstName(chat.getFirstName());  
 user.setLastName(chat.getLastName());  
 user.setUserName(chat.getUserName());  
 user.setRegisteredAt(new Timestamp(System.currentTimeMillis()));  
 userRepository.save(user);  
 log.info("registered a user: " + user, chatId); //отправляет запись в лог файл об регистрации пользователя

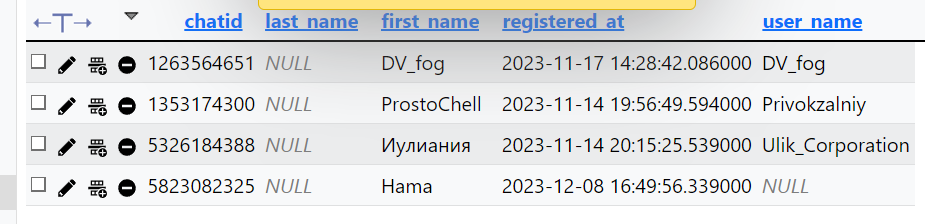


Рисунок 14 – Зарегистрированные пользователи.

Сохранение новых пользователей происходит в заранее созданную таблицу User.

Самый главный функционал чат-бота заключается в том что он должен обращаться в базу данных за конкретными данными. Пример реализации этой функции будет представлять собой запрос на вывод перечня патронов одного калибра. Первым этапом пользователь вводит команду которая вызывает метод который предлагает пользователю выбрать интересующей его калибр (Рисунок 15). После чего генерируется запрос и выводится ответ (Рисунок 16).

// Метот создания экранных кнопок под сообщениями бота, которые при нажатии отправляют вместо сообщений данные

private void ammochoice (Long chatID) {  
 SendMessage message = new SendMessage();  
 message.setChatId(String.valueOf(chatID));  
 message.setText("Выбери калибр:");  
 InlineKeyboardMarkup markupInline = new InlineKeyboardMarkup(); //создание экранных кнопок под сообщением  
 List<List<InlineKeyboardButton>> rowsInLine = new ArrayList<>();  
 List<InlineKeyboardButton> rowInLine = new ArrayList<>();  
 var pmButton = new InlineKeyboardButton();  
 pmButton.setText("9x18PM"); // название кнопки  
 pmButton.setCallbackData("Caliber9x18PM"); //данные которые она возвращает  
 var arButton = new InlineKeyboardButton();  
 arButton.setText("7,62x51");  
 arButton.setCallbackData("Caliber762x51");  
 rowInLine.add(pmButton);  
 rowInLine.add(arButton);  
 rowsInLine.add(rowInLine);  
 markupInline.setKeyboard(rowsInLine);  
 message.setReplyMarkup(markupInline);  
 executeMessage(message);  
}

//проверка если вместо сообщения прислали какое либо значение(нажали кнопку)

String ammoType = null;  
// обработка данных  
if (callbackData.equals("Caliber9x18PM")) { //если обратно пришли эти данные бот вызывает метод с запросом из бд и подставляет указанные данные в запрос.  
 ammoType = "Caliber9x18PM";  
} else if (callbackData.equals("Caliber762x51")) {  
 ammoType = "Caliber762x51";

// Метод при вызове которого бот обращается в базу данных и выполняет указанный запрос, после чего выводит его пользователю  
// Метод составляет необходимый запрос, путем нажатия пользователем кнопки которая возвращает данные  
// для этого метода в которых указан необходимое значение для подстановки в запрос  
private void ammos (long chatID, String ammo) throws SQLException{  
 Connection connection = DriverManager.getConnection(url, username, password);  
 String query = "SELECT \* FROM ammo where caliber like ";  
 query = query + "'" + ammo + "'" + ";";  
 System.out.println(query);  
 Statement statement = connection.createStatement();  
 ResultSet resultSet = statement.executeQuery(query);  
 while (resultSet.next()) { // Отправка информации в чат  
 String name = resultSet.getString("name");  
 String damage = resultSet.getString("damage");  
 String penetrationPower = resultSet.getString("penetration\_power");  
 sendMessage (chatID,"Патрон: " + name + ", \nнаносит урон: " + damage + ", с пробитием: " +penetrationPower+ ".");  
 SendMessage message = new SendMessage();  
 message.setChatId(chatID);  
 }  
}



Рисунок 15 – Кнопки выбора.

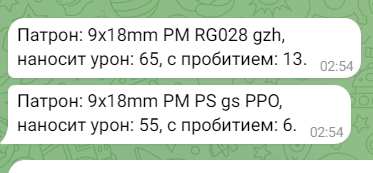


Рисунок 16 – Вывод информации по запросу пользователя.

Этап создания чат-бота завершён. Проведение тестирования по мере написания кода программы, позволило выявить ошибки и недочёты а так же внести коррективы в работу бота. Поставленные задачи по реализации выполнены, бот соответствует установленным требованиям.

ЗАКЛЮЧЕНИЕ

В наше время мессенджеры находятся на своём пике популярности среди всех возрастных категорий людей. Ежедневно физические и юридические лица используют их в своих целях. От отправки смешных картинок до учётных записей крупных складов. Обмен сообщениями на огромном расстоянии далеко не новая технология, но именно мессенджеры прямо подошли к задаче, без излишеств. Это и цепляет людей. Скорость, надёжность и приватность это самое главное для такого вида работы.

Чат-боты стали неотъемлемой частью этой системы. Если потребуется скачать видео с интернет ресурса, то для этого найдётся чат-бот. Отпала необходимость скачивания сотни приложений для каких-либо нужд. Достаточно всего лишь скачать мессенджер и начать работу с нужным ботом.

В рамках курсовой работы были выполнены поставленные задачи. Были изучены мессенджеры и их устройство. Было проведено сравнение мессенджеров друг с другом. При сравнивании был выявлен лучший в плане удобства в работе и наличия документации Telegram Bot API.

Было проведено сравнение средств разработки, в следствии которого была выбрана платформа JAVA.

Таким образом результатом курсовой работы является реализованный чат-бот для платформы Telegram для с поддержки пользователей которые имеют необходимость получить помощь в решении проблем возникающих в процессе игры в онлайн-проект «Escpe from Tarkov».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Резкий рост популярности российских мессенджеров: анализ интернет-трафика пользователей Yota 2022 год [Электронный ресурс]: URL: <https://www.cnews.ru/news/line/2022-0317_rezkij_rost_populyarnosti>

2. Рейтинг языков программирования 2020 [Электронный ресурс]: URL: <https://techrocks.ru/2020/02/08/programming-languages-rank-2020>

3. Телеграм-бот как простой и удобный способ получения информации [Электронный ресурс]: URL: [https://cyberleninka.ru/article/v/teleg](https://cyberleninka.ru/article/v/telegram-bot-kak-prostoy-i-udobnyy-sposobpolucheniya-informatsii) [ram-bot-kak-prostoy-i-udobnyy-sposobpolucheniya-informatsii](https://cyberleninka.ru/article/v/telegram-bot-kak-prostoy-i-udobnyy-sposobpolucheniya-informatsii)

4. Чат боты и разговорный искусственный интеллект [Электронный ресурс]: URL:

5. Чат-боты – кто они и что умеют [Электронный ресурс]: URL: <https://efsol.ru/articles/messendzhery-i-chat-boty-dlya-biznesadostavki.html>

6. Чат-боты: история, современность и перспективы [Электронный ресурс]: URL: http://w7phone.ru/chat-boty-istoriya-sovremennost-i-perspektivy-132460/

7. Data Serialization [Электронный ресурс]: URL: <https://link.springer.com/chapter/10.1007/978-1-4842-2692-6_4>

8. HTTP [Электронный ресурс]: URL:  [https://link.springer.com/chapte](file:///D:/Users/Downloads/Telegram%20Desktop/Â€https:/link.springer.com/chapter/10.1007/978-1-4302-5861-2_11) [r/10.1007/978-1-4302-5861-2\_11](file:///D:/Users/Downloads/Telegram%20Desktop/Â€https:/link.springer.com/chapter/10.1007/978-1-4302-5861-2_11)

9. Katarzyna Musiał & Przemysław Kazienko «Social networks on the

Internet» [Электронный ресурс]: URL: <https://link.springer.com/article/10.1007/s11280-011-0155-z#Bib1>

10. Make a Difference in a Different Way: Twitter Bot Creators and Wikipedia Transparency [Электронный ресурс]: Режим доступа: <https://link.springer.com/article/10.1007/s10606-021-09411-6>

11. Telegram Bot API [Электронный ресурс] : URL: <https://core.telegram.org/bots/api>

ПРИЛОЖЕНИЕ