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](#_Toc7682)

[1 ОПИСАНИЕ МЕТОДОВ РАЗРАБОТКИ 4](#_Toc23744)

[1.1 Мессенджеры 4](#_Toc3672)

[1.2 Чат-боты 7](#_Toc29388)

[1.4 Выбор среды разработки 8](#_Toc20904)

[1.5 Telegram Bot API 11](#_Toc26780)

[2 РАЗРАБОТКА ТЕЛЕГРАММ-БОТА 13](#_Toc16538)

[2.1 Регистрация чат-бота 13](#_Toc31680)

[2.2 Разработка функционала чат-бота. 14](#_Toc29662)

[2.3 Подготовка платформы к работе. 16](#_Toc30213)

[2.4 Проектировка чат-бота 17](#_Toc14975)

[ЗАКЛЮЧЕНИЕ 25](#_Toc11502)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](#_Toc29685)

ВВЕДЕНИЕ

Мессенджеры и социальные сети – это уникальные среды для всех сфер деятельности. Будь то общение, развлечения или обучение. В данное время в мире есть огромное 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 телеграм-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. Объектом исследования является процесс создания телеграм-бот. Предметом исследования является разработанный телеграм-бот “EFTbotGUIDE”. Цель проекта является - создание бота для оказания помощи пользователям онлайн-игры “Escape from Tarkov” в мессенджере Телеграм.

Задачей курсовой работы является:

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

1 ОПИСАНИЕ МЕТОДОВ РАЗРАБОТКИ

* 1. Мессенджеры

Своё начало общение 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. Не малая часть людей предпочитает голосовые сообщения обычным текстовым сообщениям [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 всего лишь 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. Например МТС предоставляет пакет с безлимитным трафиком на мессенджеры. Широкое распространение смартфонов любых ценовых категорий, высокие скорости и низкие цены относительно прошлого. Все мессенджеры имеют повторяющийся набор инструментов, таких как отправка сообщений, медиафайлов, аудио и видео общение.

В данный момент существует большое разнообразие мессенджеров, среди которых нельзя обойти стороной некоторые их них, а также их 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:

– приватность – все чаты зашифровываются, а если выбрать специальную приватность чата, то сообщения уничтожатся через выбранное время;

– быстрота – скорость отправления сообщений выше, чем у одноклассников;

– распределение – сервера Telegram находятся по всему земному шару, что повышает отказоустойчивость;

– открытость – использование открытого протокола MTProto и API, бесплатных для всех;

– отсутствие подписок и рекламы, (рекламу можно встретить непосредственно в чатах от самих пользователей);

– отсутствие ограничений на размер сообщений и вложенных файлов .

Отличительные качества Telegram.

Кросс-платформенность, надежность доставки сообщений и скорость. Факт того, что приложение работает на всех платформах, сейчас не удивляет ни кого, но вопрос в том, насколько связанно разные приложения взаимодействуют друг с другом.

WhatsApp в общем и целом хороший мессенджер, но иногда доставка сообщений тормозит, особенно если они были отправлены при отсутствии связи или ее низком качестве (например, в метро или подвальном помещении). К тому же, хоть и у него появилось декстопное приложение, но из-за ситуации в стране скачать его нельзя, а веб-версия – не совсем то, что хотелось бы видеть на своем устройстве в плане удобства.

Telegram выглядит гораздо лучше во всех этих аспектах, и это не беря во внимание ключевой аспект безопасности. Клиент для Windows и приложение Android работают крайне быстро, а прочитанные сообщения синхронизируются буквально моментально. Telegram хорошо кэширует сообщения и картинки (настраивается), что значительно

Боты в мессенджерах сейчас везде, и даже Microsoft включилась в игру на Build 2016. 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. Чат-боты делятся на две категории – обычные и встраиваемые (inline).

С обыкновенными ботами пользователь работает напрямую, отправляя им текст (например, свой вопрос или запрос в поисковик) или команды, которые учёл разработчик.

У большого количества чат-ботов команды продублированы в графическом интерфейсе (меню), поэтому вовсе необязательно вводить их с клавиатуры. Хотя на смартфонах зачастую в поле ввода встраивают кнопку для удобства использования.

Если попробовать пообщаться с парочкой ботов, то можно понять, как это работает: ya (бот сервисов Яндекса), Meduza (новостной сайт), @truemafiabot (бот для игры в мафию), vkmusic\_bot (бот для скачивания музыки из ВКонтакте).

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]

Актуальность выбора Telegram.

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 сою 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. Это неоднократно подтвердили различные статистические статьи.

Главной причиной активного введения в свой бизнес работу с мессенджерами и непосредственно чат-ботов является то, что у людей на сегодняшний день, в среднем, на мобильном устройстве стоит до 4 различных мессенджеров. Этот факт, как никакой другой, даёт понять, что данная сфера популярна и этим фактом обязательно нужно пользоваться в целях продвижения, рекламы и улучшения своего бизнеса в целом. Так же удастся автоматизировать большое множество процессов, которые раньше выполнялись людьми. А вышеупомянутые чат-боты послужат нам средством для достижения этой цели.

1.2 Чат-боты

Чат-ботов можно разделить на некоторые 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, нет возможности текстового ввода. Текстовый чат-бот. Данный вид уже более прогрессивный и функциональный: возникает возможность текстового ввода, происходит анализ запроса и выдается на его базу ответ. Следовательно, мы можем иметь диалог между ботом и пользователем [5].

Если углубиться в статистику использования чат-ботов в наше время, то можно сделать ряд выводов, что боты это некие универсальные средства решения большого множества проблем и жизненных задач, таких как: общение с людьми, развлечение (игры, музыка, фильмы и сериалы), здоровье (например, 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, разработка различных приложений. [8].

Плюсы Python. Python это интерпретируемый язык. Он не компилируется как обычный язык. Это значит, что пайтон до запуска кода является просто текстом. Следовательно, на нем можно кодить почти везде, на всех платформах. Сам язык отлично спроектирован и логичен.

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

Минусы языка программирования Python. Несмотря на множество плюсов (простота, лаконичность, универсальность), у языка есть и минусы.

Скорость слабая сторона языка. Например язык Swift, который используется, для разработки программ на iphone OC, может выполнить работу в 8 раз быстрее, чем конкурент. Пайтон не очень хороший вариант для выполнения задач, требующих больших объёмов памяти. Такого типа задачи лучше выполнять при помощи языка Си или Си++.

Глобальный блок интерпретатора (GIL) не позволяет одновременно выполнить несколько потоков Пайтон в реализации СПайтон, Но Global Interpreter lock можно отключить на время, как например это сделано в пакете NumPy для математики.

Вывод: можно уверенно сказать, 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. Java придерживается принципа: «Написал один раз, а потом используешь где хочешь». Язык используется в различных сферах, будь то мобильные приложения или декстопные, сетевые или корпоративные.

Плюсы языка.

ООП (Объектно-ориентированное программирование). Данный язык программирования имеет концепцию ООП. Таким образом мы можем не просто указать тип данных и структуру, а также определить некий набор функций присущий его типу. Тем самым наша структура данных стала объектом, с которым можно взаимодействовать для создания связей с другими объектами.

Java – высокоуровневый язык с довольно простым синтаксисом и хорошим подходом к освоению. Язык похож на наш 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.

Многопоточность. Самой малой единицей обработки данных в программировании является поток. Для того чтобы максимизировать КПД процессора, люди придумали запускать несколько потоков одновременно. Их количество будет завесить от мощности процессора и количества его ядер. Java это умеет, а значит ещё один плюс в копилку.

Автоматическое управление памятью (АММ). Чтобы писать код для управления памятью на Java людям не нужно делать это руками, как например в Си. АММ (Automatic Memory Management) также используется в языке Swift и при очистке памяти приложениями, которые этим занимаются.

Минусы языка.

Многословный и сложный код. Код очень большой, это так. Длинные и чрезвычайно сложные операторы затрудняют изучение. Многие считают, что так проще будет разобраться в коде, но для меня это не так. Как и С++, Java высокоуровневый язык программирования. Им свойственно иметь большой развернутый код. Так его лучше будет понять неспециалистом, ведь они

смогут детальней разобрать операторы. Но это делает код громоздким и менее компактным.

Низкая производительность. Как и у своих сверстников Java имеет довольно низкую производительность. Все языки высокого уровня имеет 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 поток получить доступ к одному ресурсу. Здесь появится ошибка из-за нехватки памяти, но это можно решить если умело планировать работу.

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 нам и нужен сервер, принимающий http запросы и выдающий ответы.[8]

Есть всего 2 способа получать обновления с сервера:

– long pulling – телеграмм 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 сериализованный 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 [11].

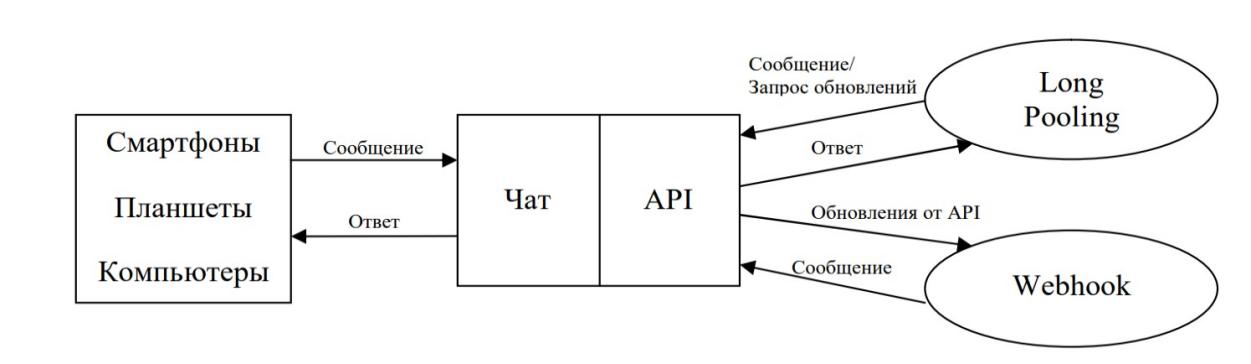


Рисунок 9 – Принцип работы чат-бота на платформе Telegram

Чтобы получить для нашего бота ключ доступа, он же токен, необходимо написать botfather боту. Он создан для того, чтобы создавать ботов. Ниже приведены некоторые примеры методов:

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

– sendMessage – метод осуществляет обычную отправку сообщения пользователю.

Вывод: В данном разделе были рассмотрены методы разработки. Данное исследование позволило сделать ряд выводов. Были изучены мессенджеры и их актуальность. Был проведен анализ полезности чат-ботов.

2 РАЗРАБОТКА ТЕЛЕГРАММ-БОТА

2.1 Регистрация чат-бота

Для начала потребуется зарегистрировать бота в телеграм, то есть нужно получить токен бота, он же ID уникальный идентификатор. В этом может помочь специальный бот @BotFather.

Как это сделать:

1. Пишем в бота команду /start.
2. Далее пишем команду /newbot (рисунок 10).

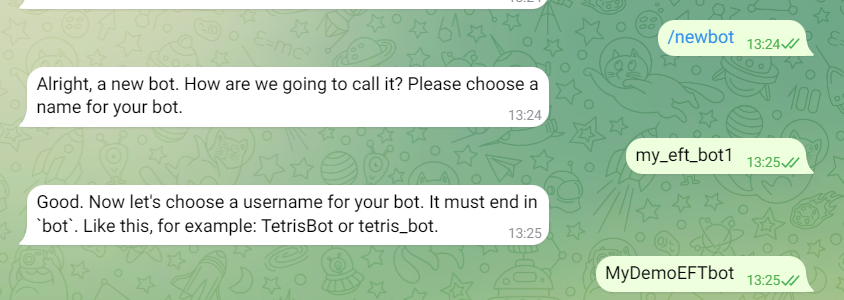


Рисунок 10 – Регистрация бота

3. Придумываем имя для чат-бота.

4. Помощник пришлёт поздравления и выдаст токен (рисунок 11).

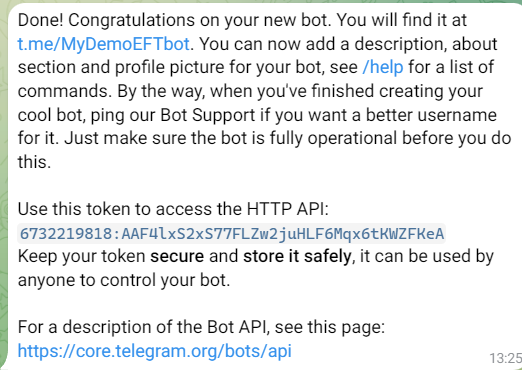


Рисунок 11 – Токен бота

2.2 Разработка функционала чат-бота.

Чат-бот в Telegram будет создаваться в качестве персонального компаньона для оперативного доступа к необходимой информации, хранящейся в собственной базе данных бота. Основываться разработка будет на  [многопользовательско](https://ru.wikipedia.org/wiki/%D0%9C%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%B8%D0%B3%D1%80%D0%B0" \o "Многопользовательская игра)й [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" \o "Онлайн-игра) 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" \o "Английский язык) — «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" \o "Тактический шутер) [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" \o "Шутер от первого лица) 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" \o "Компьютерная ролевая игра) 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" \o "Массовая многопользовательская онлайн-игра), 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. Однако, это не всегда удобно. 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, предлагается перенести весь процесс в удобное мобильное приложение. А так же, важно сохранить возможность использования этого приложения с компьютера. Помимо этого проект доложен отвечать другим задачам, таким как:

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

На все поставленные задачи в качестве платформы для разработки подходят чат-боты создаваемые в мессенджере Телеграм.

Для создания такого чат-бота который поможет игрокам быстро получать информацию об игровых предметах и заданиях нужно определить основной функционал: пользователи хотят просто и быстро получать ответы на свои запросы. Ответы должны быть ёмкие и не содержать ничего лишнего.

Разработка базы данных: Создать базу данных, в которой будет храниться информация об игровых предметах, заданиях и других важных данных.

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

Синхронизация данных: обеспечьте возможность синхронизации данных между мобильным устройством и компьютером, чтобы пользователь мог начать поиск информации на одном устройстве и продолжить на другом без потери данных или состояния поиска.

Тестирование: тестирование будем проводить по мере наполнения новым функционалом чтобы иметь возможность сразу вносить коррективы в код программы.

Такое приложение 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 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, поддержки веб- и корпоративной разработки 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 парадигмой или 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 необходимо установить его. После установки необходимо в настройках подключить модуль «MySQL-8.0» и открыть автоматически установленное приложение PhpMyAdmin. Далее нужно создать новую базу данных ”tg-bots”. Её нужно оставить пустой, так как программа будет сама создавать нужные таблицы и наполнять их.

2.3.2 Установка и настройка IntelliJ IDEA.

IntelliJ IDEA Community Edition скачивается с официального сайта и после установки настраивается под конкретную версию JAVA, выбирается SDK.

В дальнейшем после запись каких либо библиотек в файл зависимостей, компилятор автоматически загрузит необходимые библиотеки из сети Интернет.

2.3.3 Подключение фреймворка Spring

На сайте Spring initializer нужно указать необходимые параметры:

1. Выбрать версию фремворка
2. Выбрать язык разработки
3. Выбрать версию Spring
4. Вести данные (Группу, название и т.д.)
5. Выбрать версию SDK и сгенерировать проект

После этого этапа проект готов к началу разработки программы.

2.4 Проектировка чат-бота

Первым этапом мы заполняем ресурсный файл из которого программа будет брать необходимые ей параметры для подключения к базе данных и бэкенду телеграма:

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 необходимо добавить 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>  
<!--Подключение библиотеки 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>

Далее следует приступить к созданию файла конфигурациии который будет получать токен бота, имя бота и код чата администратора:

public class BotConfig {

//сюда будет обращаться главный класс TelegramBot чтобы получить токен и имя

//бота для предоставления бэкенду телеграмма

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

String botName;

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

String token;

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

Long ownerID;

}

После этого нужно прописать файл инициализации чат-бота для подключения бота к Telegram API:

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

После выполнения этих действий можно приступать к написанию основного класса. Первое что нам нужно добавить - это кнопки, которые будут расположены в нижнем правом углу интерфейса (рисунок 12).

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

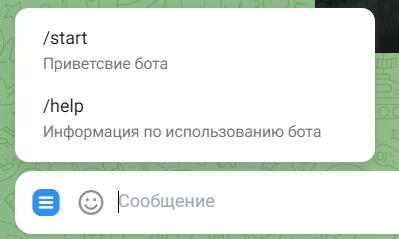


Рисунок 12 – Функциональные кнопки

В дальнейшем при добавлении функций настройки или функций управления ботом, есть место для их удобного размещения.

Следующая функция для реализации это - считывание основных команд.

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 (Рисунок 13).

//Создание экранной клавиатуры бота которая отправляет готовые сообщения  
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);

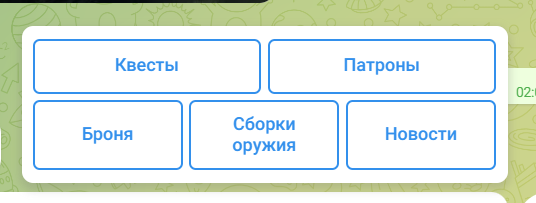


Рисунок 13 – Экранная клавиатура

Для реализации рассылки, ведения статистики создаётся метод который регистрирует пользователей при нажатии ими на кнопку «Старт» (Рисунок 14).

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

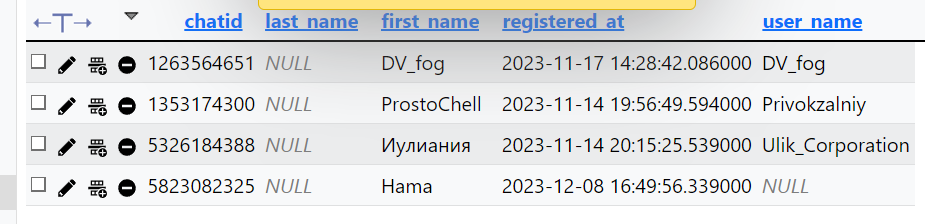


Рисунок 14 – Зарегистрированные пользователи

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

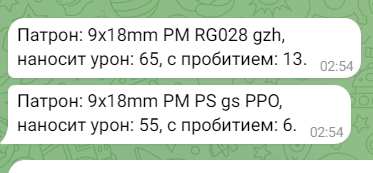


Рисунок 16 – Вывод информации

Этап создания чат-бота завершён. 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

Исходный код системы защиты данных