НПОУ «ЯКУТСКИЙ КОЛЛЕДЖ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ» (НПОУ «ЯКИТ»)

Отделение экономики, менеджмента и информационных технологий

Специальность 09.02.07 «Информационные системы и программирования»

КУРСОВАЯ РАБОТА

по дисциплине: Технические методы и средства, технологии защиты информации

по теме:

РАЗРАБОТКА TELEGRAM-БОТА ЭЛЕКТРОННОГО КУРАТОРА.

Исполнитель студент гр. **КИСП-9-19** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **И.И. Иванов**

подпись, дата

Руководитель,

**преподаватель** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **И.И. Иванов**

подпись, дата

Якутск, 2022

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# ВВЕДЕНИЕ

Социальные сети, с момента их появления все больше проникают в нашу жизнь. Это происходит потому, что они позволяют быстро получать и давать информацию. В связи с этим стали развиваться чат-боты, которые позволяют сэкономить время на поиске нужного запроса.

Одним из видов чат-бота является TelegramBot. Он помогает выполнять разные действия: переводить и комментировать(@ytranslatebot), обучать и тестировать (@InMindTechBot), искать и находить (@fmusbot), спрашивать и отвечать(@AccentsBot), играть и развлекать(@daenerys\_cash\_game\_bot), транслировать и агрегировать(@videollamada\_bot), встраиваться в другие сервисы и платформы(@airtrack\_bot), взаимодействовать с датчиками и вещами, подключенными к интернету. И это далеко не все возможности, которые представляет платформа для их создания. Чат-боты на платформе Telegram активно развиваются и набирают все больше популярность.

Для бизнеса Telegram применяется достаточно широко. Это обуславливается тем, что возможности мессенджера позволяют оперативно делиться новостями, акциями и рассылками с большим количеством пользователей. Боты уменьшают нагрузку на менеджеров, беря на себя выполнение монотонных заданий.

Также Telegram боты применяются и в учёбе, достаточно много примеров разных ботов помогающих с подготовкой к Основному и Единому государственным экзаменам, либо же продающих или предоставляющих курсы по различным предметам и специальностям, а также облегчающих взаимодействие с сервисами.

Цель работы: создание чат-бота на платформе Telegram с помощью библиотеки PyTelegramBotAPI Python для реализации электронного куратора.

Задачи:

1. Анализ платформы Telegram;
2. Разработка структуры проекта;
3. Реализация проекта;
4. АНАЛИЗ ПЛАТФОРМЫ И ПРОГРАМНЫХ СРЕДСТВ.

# 1.1. Возможности и примеры применение чат-ботов

Чат-бот – это робот-помощник, который позволяет выполнять определенные функции, не привлекая людей, а работа будет выполнена моментально.

Они делают жизнь людей намного проще, – заказать пиццу или такси, забронировать билеты или купить платье можно теперь не выходя из мессенджера.

Их использование, с одной стороны, полезно для клиентов, т.к. они могут получить нужную информацию или совершать какие-либо действия более простым и удобным способом, и, с другой стороны, это выгодно для компаний, которые с помощью чатботов могут продвинуть свой бренд, собрать от ЦА необходимые сведения и повысить ее лояльность.

Об эффективности использования чатботов рассказывал Марк Цукерберг на конференции F8 в 2016 году, – на его взгляд, они существенно сокращают время обслуживания клиентов и являются одним из самых актуальных трендов на сегодняшний день [1].

Вся переписка с ним ведётся через обычный чат Telegram. Бот получает в виде сообщения команды, которые он готов выполнять круглосуточно. Его основная задача ответить на вопрос пользователя, согласно своей программе. Боты помогают, экономят время и управлять ими очень просто.

На сегодня роботы Telegram могут:

* проводить обучение;
* развлекать и играть с пользователем;
* работать поисковиками в интернете;
* скачивать текстовую информацию, видео или аудио;
* быть обычной напоминалкой;
* участвовать в групповых чатах, допустим, для согласования времени встречи, оптимальной для всех участников;
* комментировать нужные статьи;
* использоваться для управления умным домом и др.;
* делать покупки.

Существует множество ботов, которыми пользуются ежедневно. Одними из популярных являются:

1. @TGhost\_bot – бот-файлообменник;
2. @InMindTechBot – позволяет изучать английский язык с возможностью практиковаться в общении со случайным собеседником;
3. @maksobot – озвучивает сообщения пользователя;
4. @flibustafreebookbot - Бот для поиска и скачивания книг;
5. @ytranslatebot - Яндекс Переводчик - с помощью этого бота можно перевести свои сообщения на любой из 12 языков.

Боты являются посредниками между человеком и многочисленными веб-службами. Одно из главных преимуществ – это общая оболочка, так как Telegram позволяет на своей платформе реализовать вышеперечисленное.

Несомненный плюс в экономии времени за счёт уменьшения количества приложений в гаджете. Люди всегда хотят получать всё, что им нужно, в одном окне. Теперь нет необходимости в установке лишних приложений для погоды, новостей и т. п., что было особенно критично для маломощного смартфона. Теперь всё это заменяет Telegram с возможностью установки необходимых роботов.

Принцип работы: выбирается бот, отправляется ему текстовое сообщение (команду) и через доли секунды приходит ответ.

Достоинства:

* Круглосуточная помощь — по сути, их работу остановит только авария на сервере, что случается крайне редко.
* Удобство использования — большинство команд находится в меню бота.
* Ответы приходят за несколько секунд.
* Для работы используются ресурсы сторонних серверов, так что мощности вашего устройства не задействованы.
* Безопасность. Ботам не доступны персональные данные пользователя. Они обрабатывают только команды из чата.
* Установка дополнительных программ не требуется [2].

# **1.2. Платформа для разработки чат-бота Telegram**

Telegram в своей сущности – из популярнейших текстовых мессенджеров, имеющий клиенты для всех современных платформ : Android, iOS (iPhone/iPad), Windows Phone, Windows, macOS, Linux, а также веб-версия, доступная на всех веб-браузерах, выпущенных после 2012 года. Рисунок 1 демонстрирует график роста активных пользователей.

С точки построения чат-ботов Telegram обладает инструментарием под названием Botfather. Он позволяет быстро настроить внешнее представление бота для пользователей (его никнейм, имя пользователя, аватар), а также сгенерировать ключ для Telegram API, посредством которого будет взаимодействие пользователя с сущностью бота, реализующей исполняемый код. Таким образом, разработка программы, выполняющей алгоритмы работы бота, не имеет жесткой привязки к платформе.

Статистика аудитории Telegram на март 2023:

* Аудитория Telegram составляет 700 миллионов пользователей в месяц
* Мессенджером Telegram пользуются 55.2 миллиона человек ежедневно
* Чаще всего Telegram загружают в Индии и России
* Объем трафика Telegram в России составляет 68%
* Telegram используют в основном мужчины – 58%, женщины – 42%
* Суммарная аудитория пятидесяти самых крупных телеграмм-каналов составляет более 100 млн подписчиков
* Самые популярные темы в Telegram – новости (82%), на втором месте – развлечение и политика (59%), на третьем месте – образование (55%)
* Люди проводят в мессенджере в среднем около 4 часов в месяц
* Стоимость компании Telegram составляет от 2.2 до 124 млрд долларов
* Самый дорогой никнейм в Telegram (@news) был продан за 994 тыс. долларов
* В 2023 году Telegram монетизирует свой сервис с помощью рекламных сообщений и платной подписки Telegram Premium
* 62% администраторов каналов Telegram зарабатывают на рекламе, 45% получают доход от продажи своих товаров или услуг, 23% зарабатывают на партнерских программах
* С 2020 года спрос на накрутку подписчиков в Telegram вырос в 23 раза
* 12.6% администраторов каналов Telegram зарабатывают более 100 тысяч рублей
* 36% рекламодателей расходуют на рекламу в
* Telegram до 25 тысяч рублей
* Telegram занимает 9-е место в рейтинге самых популярных социальных сетей
* Telegram заблокирован в Китае, Пакистане и Иране Telegram входит в ТОП-5 популярных мессенджеров в мире по аудитории. [3].

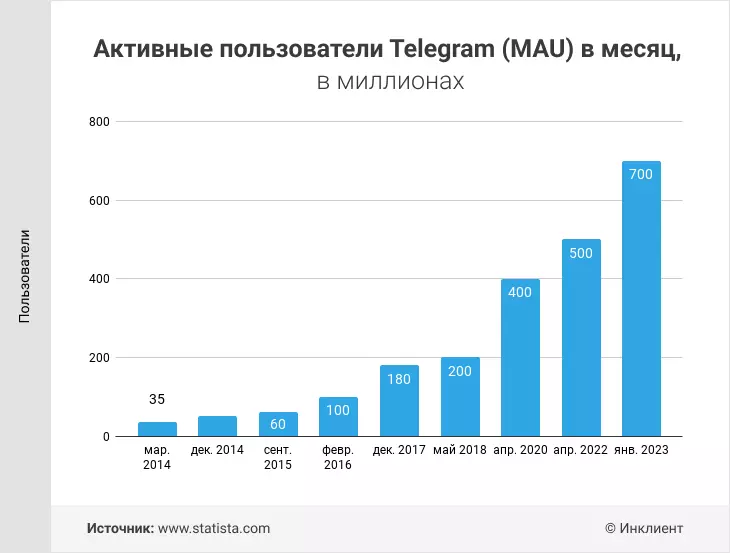


Рисунок 1 - Общая аудитория

**1.3. Средства и методы разработки**

**1.3.1. Язык программирования**

Для реализации проекта был выбран язык программирования C#.

Язык Python универсален , с его помощью можно создавать самые разнообразные проекты в том числе и реализовать Telegram-бота.

Для достижения цели мы использовали PyCharm IDE Community Edition (PyCharm), так как он предоставляет обширный функционал на бесплатной основе, позволяя нам разрабатывать Telegram-бота без материальных вложений, сохраняя эффективность и простоту среды. Внутри среды PyCharm мы сможем сконцентрироваться непосредственно на выполнении следующих поставленных задач без дополнительных усилий:

* Быстро реализовать Telegram-бота. Программировать на Python очень просто. Это мощный, легко осваиваемый язык, а PyCharm позволяет автоматизировать большинство процессов.
* Фокусироваться на решении реальных задач**.**За конечный результат работы, разумеется, отвечает разработчик. Но IDE позволяет концентрироваться на глобальных вещах, взяв на себя:
* слежение за всеми проектами;
* упрощенное редактирование кода;
* отслеживание графики, аудиофайлов, значков и прочих ресурсов;
* управление данными.

**1.3.2. СУБД**

Для хранения данных о пользователях, группах, кураторах и другой информации необходимо использовать базу данных. В качестве системы управления базой данных была выбрана СУБД SQLite.

SQLite - это система управления реляционными базами данных (RDBMS), которая содержится в библиотеке C. В отличие от большинства других систем управления базами данных, SQLite не является клиент-серверным ядром базы данных, а встроен в конечную программу.

Для работы с SQLite мы использовали DB Browser for SQLite (DB Browser) и встроенную в Python библиотеку SQLite3.

DB Browser for SQLite - это высококачественный инструмент с открытым исходным кодом для проектирования, создания и редактирования файлов базы данных, совместимый с SQLite. Она обладает огромными функциональными возможностями и подходит для решения самых разных задач.

Преимущества SQLite:

* Улучшенная производительность:
  + Чтение и запись из базы данных SQLite выполняется быстрее, чем чтение и запись файлов непосредственно с диска.
  + Приложение загружает только те данные, которые ему нужны, вместо того, чтобы читать весь файл и сохранять полный синтаксический анализ в памяти.
  + При внесении небольших изменений перезаписываются только те части файла, которые изменяются, что сокращает время записи и износ SSD-накопителей.
* Снижение стоимости и сложности приложения:
  + Нет кода ввода-вывода файла приложения для записи и отладки.
  + Доступ к содержимому и его обновление могут осуществляться с помощью кратких SQL-запросов вместо длительных процедурных процедур.
  + Формат файла может быть просто расширен в будущих версиях путем добавления новых таблиц или столбцов.
  + Проблемы с производительностью часто можно решить, даже на более поздних этапах цикла разработки, с помощью CREATE INDEX, что помогает избежать дорогостоящих усилий по редизайну, перезаписи и повторному тестированию.
* Переносимость
  + Файл приложения переносим во всех операционных системах.
* Надежность
  + Содержимое может постоянно обновляться, так что при сбое питания теряется мало работы или ее вообще нет.
  + Sql-запросы во много раз меньше, чем эквивалентные процедурные коды, и, следовательно, количество ошибок в строке кода примерно постоянно, что означает меньшее количество ошибок в целом.[4].

**1.3.3. Telegram Bot API**

Telegram Bot API – это интерфейс, основанный на HTTP, созданный для разработчиков, заинтересованных в создании ботов для Telegram. Чтобы воспользоваться функциями Telegram Bot API необходимо получить специальный ключ аутентификации. Каждому боту при создании присваивается свой уникальный ключ.

Запросы к Telegram Bot API выглядят следующим образом: https://api.telegram.org/bot<token>/команда, где <token> – это ключ аутентификации, полученный при регистрации бота, команда – обращения к Telegram Bot API [5].

2. РЕАЛИЗАЦИЯ ПРОЕКТА.

2.1. Регистрация бота в Telegram

Прежде чем начинать разработку, бота необходимо зарегистрировать и получить его уникальный id, являющийся одновременно и токеном. Для этого в Telegram существует специальный бот —@BotFather.

Команды @BotFather:

1. */start* – выдает список доступных команд.
2. */newbot* – команда, которая позволит создать нового бота, выбрать ему ник (name) и название аккаунта (username) — тут есть ограничение, название должно обязательно оканчиваться на слово bot.

После успешного создания бота, получим *token*, необходимый для дальнейшей работы с ботом, а именно подключения скрипта к нему.

В нашем случае мы выбрали боту имя YAKIT Curator и дали название yakitcuratorbot, в ответ получили токен “6173750891:AAHh6JY7OWCKZWoOhsR6Xtod0RqDCLkqqVA”, который и будет неким ключом доступа к боту.

На этом регистрация завершена.

2.2. Объявление команд

Мы заранее обдумали функционал бота, и на основе обдуманного составили следующие команды:

* Базовая команда /start - начало работы с нашим ботом.
* Базовая команда /help - помощь с использованием бота.
* Созданная команда /note - создать заметку.
* Созданная команда /shownote - отобразить заметки.
* Созданная команда /direct - отправить сообщение куратору.
* Созданная команда /schedule - получить расписание
* Созданная команда /reply - ответ на сообщение.
* Созданная команда /toall - рассылка для студентов.

2.3. Работа бота

Сперва импортируем все нужные модули, а именно: PyTelegramBotAPI, XML, SQLite3 и Requests.

Далее объявляем экземпляр класса TeleBot, вписываем ему токен для подключения к боту (токен получили из п. 2.1, когда создали и зарегистрировали бота).

Следующим шагом начинаем писать функции-хэндлеры, которые будут обрабатывать все события в чате с пользователями. Для этого используем декораторы с аргументами commands для обработки команд и func для обработки различных событий, например - отправки обычного текста.

Обработчик команд */start* и */help*:

Когда бот получит такую команду, он отправит запрос в базу данных, чтобы проверить, зарегистрирован ли пользователь. Если да - отправляет список доступных команд для группы пользователя, иначе начинает процесс регистрации (сторона группы пользователей-студентов).

2.3.1. Сторона группы пользователей-студентов

Если пользователь не находится заранее в базе данных, то он автоматически причисляется к группе пользователей-студентов, и имеет доступ к командам */start-help, /note, /shownote, /schedule, /direct*. Подробнее:

Команда */start*: Начало работы. Краткий принцип работы описан в п. 2.3. Если пользователь не зарегистрирован, то начинается процесс регистрации из 3 шагов, а именно: указание группы, в которой числится студент, его ФИО и его даты рождения. В случае опечатки (некорректного ввода) бот откатит шаг и потребует корректной записи, но защита от ввода ненастоящих данных не предусмотрена. При окончании процесса регистрации бот создаёт запись в таблице users, где будет сохранен его ID из Telegram (tid), который будет являться первичным ключом, ФИО (name), дата рождения (birth) и группа (course).

Команда */help*: Помощь. При отправке данной команды пользователь получит сообщение со всем функционалом бота и его описанием, доступный его группе пользователей.

Команда */note*: Создание заметки. При отправке данной команды пользователь переходит в процесс создания заметки, состоящей из 1 шага, а именно: отправка текста заметки. После получения текстового сообщения бот сохраняет заметку в таблице notes с указанием уникального ID владельца, равного его ID в Telegram, а также номер личной заметки для корректного вывода и чтения.

Команда */shownote:* Получить заметки. При отправке данной команды пользователь получает все свои заметки в виде пронумерованного списка, где каждая его заметка имеет уникальный номер в зависимости от его уникального номера, который был присвоен во время его создания, а также имеет возможность выборочно удалить конкретные заметки по номеру благодаря keyboard-кнопке, т. е. кнопке, расположенной под клавиатурой, после нажатия на которую пользователь отправляет номер удаляемой заметки, бот получает его и удаляет заметку из таблицы по фильтру текста, при этом номер заметки в таблице не меняется, не требуется.

Команда */schedule:* Получить расписание на текущую неделю. При отправке данной команды пользователь получает расписание на текущую неделю в виде списка с указанием дат, номеров занятий, их временем и платформой/кабинетом проведения (Moodle, Zoom либо кабинет). Бот, после получения данной команды, получает текст xml-файла расписания со специально созданного адреса <https://www.yakit.ru/rasp/rasp.xml>, далее перекодирует полученный текст в кодировку ISO-8859-1 чтобы все символы (кириллица) перенеслись в responce правильно, а затем снова перекодирует в UTF-8, чтобы все символы стали правильно отображаться и чтобы их можно было использовать в коде, далее сохранить в файл с расширением xml для корректной записи и чтения. После нужно создать экземпляр класса xml.ElementTree, который мы импортируем как ET, затем открыть xml-файл с расписанием, созданный ранее, и начать чтение. Выборка из всех элементов расписания происходит по тегу “Group”, благодаря которому можно отправлять расписание пользователям для их конкретной группы.

Команда */direct:* Отправить сообщение своему куратору. При отправке данной команды пользователь должен отправить второе сообщение, которое будет полностью скопировано и отправлено куратору, включая все приложения (медиафайлы, документы и т.д.).

# **Список литературы**

1. Почему чат-боты захватывают мир +25 примеров для вдохновения [Электронный ресурс]. – URL: <https://texterra.ru/blog/pochemu-chatboty-zakhvatyvayut-mir-25-primerov-dlya-vdokhnoveniya.html> (Дата обращения 25.03.2018).
2. Боты в Telegram [Электронный ресурс]. – URL: <https://iklife.ru/socialnye-seti/telegram/boty-v-telegram-chto-eto-takoe.html> (Дата обращения 11.03.2018).
3. Статистика аудитории Telegram на март 2023 [Электронный ресурс]. – URL:

<https://inclient.ru/telegram-stats/> (Дата обращения 15.04.2023).

1. Что такое SQLite browser и как им пользоваться?[Электронный ресурс]. – URL:

<https://translated.turbopages.org/proxy_u/en-ru.ru.0e687b2b-643a5267-9ce08034-74722d776562/https/www.edureka.co/blog/sqlite-browser/#advantages> (Дата обращения 15.04.2023).

1. Telegram API Bot [Электронный ресурс]. – URL: <http://telegramzy.ru/api-bot/> (Дата обращения 21.03.2018).