МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное бюджетное учреждение

"Слободской колледж педагогики и социальных отношений"

**КУРСОВОЙ ПРОЕКТ**

по ПМ 01 «Разработка программных модулей» на тему:

**РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ "ИНФОРМАЦИОННЫЙ ЧАТ-БОТ ДЛЯ ПРИЕМНОЙ КОМИССИИ КОЛЛЕДЖА"**

Выполнил: Смагин Егор Максимович

Специальность 09.02.07

Информационные системы и программирование

Группа 21П-1

Форма обучения: очная

Руководитель: Махнев Александр Анатольевич

Дата защиты курсового проекта:

Председатель ПЦК:

Оценка за защиту курсового проекта:

Слободской

2024

ОГЛАВЛЕНИЕ

[Введение 3](#_Toc184849913)

[1. Анализ предметной области 5](#_Toc184849914)

[2. Разработка технического задания 1](#_Toc184849915)1

[3. Описание алгоритмов и схема функционирования программного модуля 17](#_Toc184849916)

[4. Тестирование программного модуля 21](#_Toc184849917)

[Заключение 25](#_Toc184849918)

[Список литературы 26](#_Toc184849919)

[Приложения 28](#_Toc184849920)

# **ВВЕДЕНИЕ**

# Современный образовательный ландшафт претерпевает значительные трансформации под влиянием стремительного развития информационных технологий. В условиях высокой конкуренции между образовательными учреждениями и растущих требований абитуриентов, эффективная коммуникация и доступность информации играют решающую роль в процессе привлечения новых студентов. Традиционные методы информирования, такие как дни открытых дверей, информационные буклеты и телефонные консультации, зачастую оказываются недостаточно оперативными и не способны удовлетворить потребности современной аудитории, привыкшей к мгновенному доступу к информации.

# В этом контексте разработка информационного чат-бота для приемной комиссии колледжа представляет собой актуальное и перспективное направление. Чат-боты, основанные на технологиях искусственного интеллекта и машинного обучения, способны автоматизировать процесс коммуникации с абитуриентами, предоставляя им круглосуточный доступ к необходимой информации, отвечая на часто задаваемые вопросы и осуществляя первичную консультацию. Такой подход не только оптимизирует работу приемной комиссии, освобождая сотрудников от рутинных задач, но и существенно повышает качество обслуживания абитуриентов, обеспечивая персонализированный подход и оперативное реагирование на их запросы.

# Актуальность данного курсового проекта обусловлена следующими факторами: Повышение эффективности работы приемной комиссии: Автоматизация ответов на типовые вопросы, сбор предварительных данных от абитуриентов и другие рутинные операции позволяют сотрудникам приемной комиссии сосредоточиться на более сложных и важных задачах.

# Улучшение качества обслуживания абитуриентов: Чат-бот обеспечивает мгновенный доступ к информации, персонализированные рекомендации и круглосуточную поддержку, что способствует повышению удовлетворенности абитуриентов.

# Расширение охвата аудитории: Чат-бот доступен в любое время и в любом месте, что позволяет привлечь большее количество потенциальных студентов, независимо от их географического положения и временных ограничений.

# Сокращение расходов на информирование: Автоматизация коммуникации снижает затраты на печатные материалы, телефонные звонки и другие традиционные методы информирования.

# Современный и инновационный подход: Использование чат-бота демонстрирует современный подход колледжа к взаимодействию с абитуриентами и повышает его привлекательность в глазах молодой аудитории.

# Целью данного курсового проекта является разработка информационного чат-бота для приемной комиссии колледжа, который позволит:

# Предоставлять абитуриентам актуальную информацию о специальностях, правилах приема, необходимых документах и сроках подачи заявлений.

# Отвечать на часто задаваемые вопросы, связанные с поступлением и обучением в колледже.

# Осуществлять первичную консультацию абитуриентов, помогая им выбрать подходящую специальность и определиться с процессом поступления.

# Собирать контактную информацию от потенциальных студентов для дальнейшей работы приемной комиссии.

# В рамках данного проекта будут рассмотрены различные аспекты разработки чат-бота, включая выбор платформы, проектирование диалоговой системы, интеграцию с информационными системами колледжа и тестирование функциональности. Результатом проекта станет готовый к использованию чат-бот, способствующий оптимизации работы приемной комиссии и повышению качества обслуживания абитуриентов.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметная область охватывает процессы взаимодействия приемной комиссии колледжа с абитуриентами, а также автоматизацию этих процессов с помощью чат-бота. Ключевыми аспектами предметной области являются:

**Информационные потребности абитуриентов:**

**Информация о специальностях:** Описание специальностей, учебные планы, перспективы трудоустройства, квалификация выпускников.

**Правила приема:** Условия поступления, необходимые документы, сроки подачи заявлений, вступительные испытания.

**Стоимость обучения:** Информация о стоимости обучения на различных специальностях, возможности получения льгот и стипендий.

**Проживание:** Информация о наличии общежития, условиях проживания и стоимости.

**Внеучебная деятельность:** Информация о спортивных секциях, кружках, клубах и других видах внеучебной деятельности.

**Контакты:** Контактная информация приемной комиссии, деканата и других подразделений колледжа.

**Процессы работы приемной комиссии:**

**Консультирование абитуриентов:** Ответы на вопросы, связанные с поступлением и обучением в колледже.

**Прием документов:** Проверка поданных документов на соответствие требованиям.

**Проведение вступительных испытаний:** Организация и проведение экзаменов или собеседований.

**Ведение базы данных абитуриентов:** Хранение и обработка информации о поступающих.

**Функциональные требования к чат-боту:**

**Предоставление информации:** Чат-бот должен предоставлять абитуриентам доступ к всей необходимой информации.

**Ответы на вопросы:** Чат-бот должен уметь отвечать на часто задаваемые вопросы, используя базу знаний.

**Сбор контактной информации:** Чат-бот должен иметь возможность собирать контактную информацию от абитуриентов для дальнейшей связи с ними.

**Технические аспекты:**

**Выбор платформы разработки:** Определение платформы для создания и размещения чат-бота (Telegram, Viber, WhatsApp, веб-сайт колледжа и т.д.).

**База знаний:** Создание и поддержание базы знаний с ответами на часто задаваемые вопросы.

**Ограничения и риски:**

**Сложность обработки нестандартных запросов:** Чат-бот может испытывать трудности с пониманием сложных или нестандартных вопросов.

**Необходимость постоянного обновления информации:** База знаний и информация, предоставляемая чат-ботом, должны регулярно обновляться.

**Защита персональных данных:** Необходимо обеспечить безопасность и конфиденциальность данных, собираемых чат-ботом.

Анализ предметной области позволяет определить основные требования к разрабатываемому чат-боту и сформировать план дальнейшей работы над проектом. Учет всех перечисленных аспектов позволит создать эффективный инструмент для коммуникации с абитуриентами и оптимизации работы приемной комиссии колледжа.

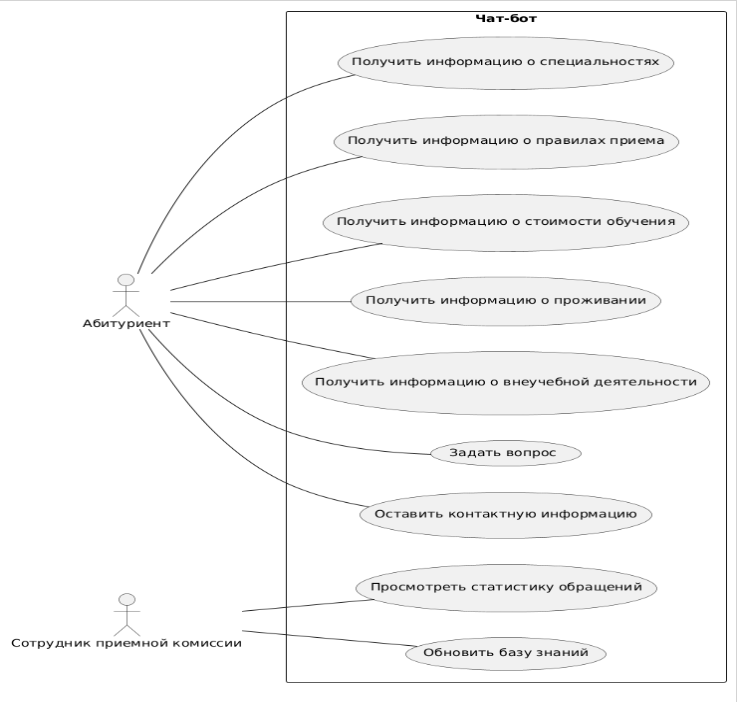


Рисунок 1 - диаграмма вариантов использования

**Программы аналоги:**

Чат-бот приёмной комиссии ГГУ — это обратной связи для поступления в Государственный Университет имени Ф.Скорины (Рисунок 2).

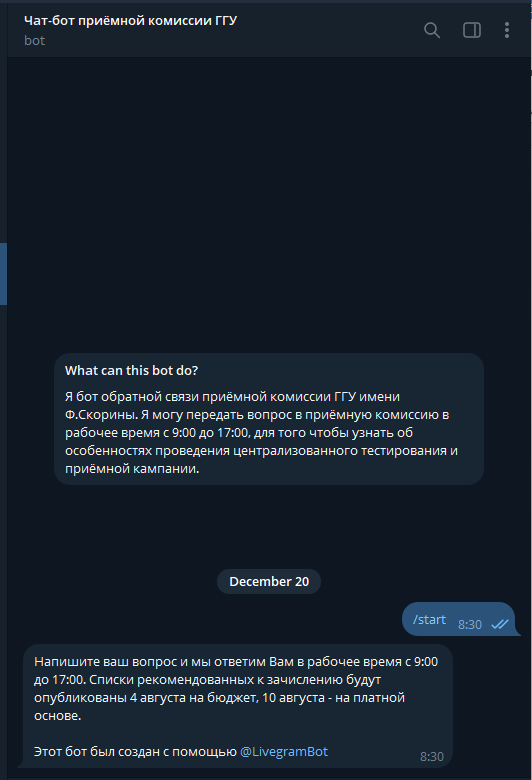


Рисунок 2 – Чат-бот приёмной комиссии ГГУ

**Преимущества:**

* гибкая настройка приложении позволяет адаптировать систему под специфику Университета
* стабильная работа;
* быстрая и качественная техническая поддержка.

**Недостатки:**

* ограниченное количество функций.
* бот не может самостоятельно отвечать на задаваемые вопросы

TalapkerENU - помощник приемной комиссии НАО «Евразийский национальный университет имени Л.Н.Гумилева» (Рисунок 3).

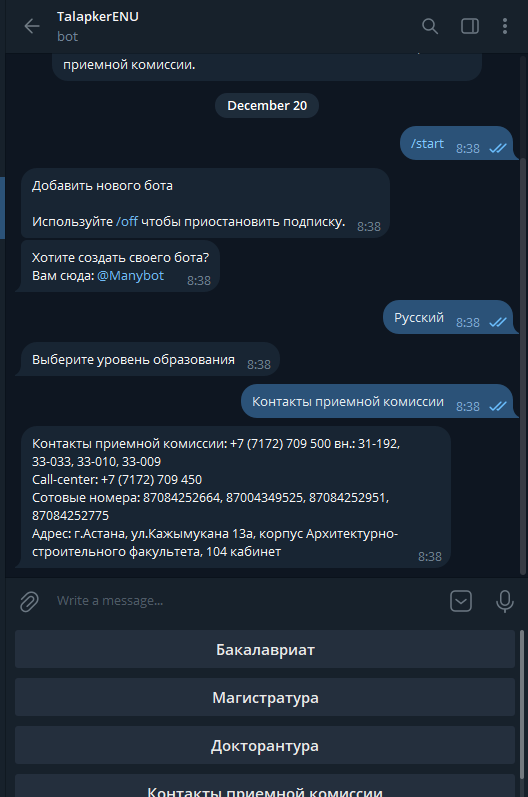


Рисунок 3 - TalapkerENU

**Преимущества:**

* простой интерфейс;
* возможность автономного ответа бота на популярные запросы

**Недостатки:**

* ограниченные возможности настройки;
* платная поддержка бота
* техническая поддержка своим качеством может варьироваться в зависимости от купленного плана.

# РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ

При разработке технического задания мы руководствовались требованиями ГОСТ [1; 2].

**Наименование программы**: "Бот для приемной комиссии Слободского колледжа педагогики и социальных отношений".

**Заказчик**: КОГПОБУ Слободской колледж педагогики и социальных отношений.

**Исполнитель**: Смагин Егор Максимович

**Основание для разработки**: заявка на разработку информационного чат-бота для приемной комиссии от Директора КОГПОБУ СКПиСО Шеренцовой Ольги Михайловны.

**Функциональное назначение**: автоматизация информирования абитуриентов и облегчение работы приемной комиссии колледжа.

**Цели и задачи проекта**

**Цель проекта**: разработка чат-бота, предоставляющего абитуриентам быстрый и удобный доступ к информации о поступлении и обучении в колледже.

**Задачи проекта**:

\*Реализация функционала предоставления информации о специальностях, правилах приема, стоимости обучения, проживании и внеучебной деятельности.

\* Реализация механизма ответов на часто задаваемые вопросы.

\* Реализация функции сбора контактной информации абитуриентов.

\* Реализация механизма сбора статистики обращений к чат-боту.

Требования к функциональности

Функции для абитуриентов:

\*Управление информацией о специальностях (получение описания, учебного плана, перспектив трудоустройства).

\*Управление информацией о правилах приема (получение информации о необходимых документах, сроках подачи, вступительных испытаниях).

\*Управление информацией о стоимости обучения (получение информации о стоимости).

\*Управление информацией о проживании (получение информации о наличии общежития, условиях и стоимости).

\*Управление информацией о внеучебной деятельности (получение информации о кружках, секциях и т.д.).

\* Отправка вопросов в свободной форме.

\* Оставление контактной информации для обратной связи.

Функции для сотрудников приемной комиссии:

\* Расчет количества обращений к чат-боту по различным темам.

\* Обновление базы знаний чат-бота (добавление, изменение, удаление информации).

**Требования к надежности**

Время восстановления после сбоя электропитания или некритического сбоя ОС: не более 5 минут.

Время восстановления после критического сбоя ОС или аппаратной неисправности: время, необходимое для устранения неисправности и восстановления системы.

Меры по предотвращению отказов: работа пользователей без административных привилегий.

**Требования к техническим средствам**

IBM-совместимый ПК:

Процессор: 1 ГГц или выше.

Оперативная память: 512 Мб или выше.

Свободное место на жестком диске: 500 Мб или выше.

Монитор: разрешение 1366x768 или выше.

Оптический привод.

Клавиатура.

Мышь.

Бесперебойное питание.

Лицензионное ПО.

Антивирусная программа.

**Требования к программному обеспечению**

Язык программирования: Python.

Библиотеки: python-telegram-bot и другие необходимые библиотеки.

Среда разработки: Visual Studio Code.

**Требования к интерфейсу**

Графический пользовательский интерфейс (GUI) в Telegram.

Интуитивно понятная навигация.

**Требования к программной документации**

Техническое задание.

Руководство пользователя.

**Стадии и этапы разработки**

Анализ требований.

Проектирование:

Разработка программной документации (техническое задание).

Разработка алгоритма программы.

Кодирование.

Тестирование и отладка.

**Приемо-сдаточные испытания**: проверка работоспособности с реальными или демонстрационными данными.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 7/8/10/11.

Программное обеспечение поставляется в виде изделия на CD диске либо USB накопителя.

Упаковка программного изделия должна осуществляться в упаковочную тару предприятия-изготовителя компакт диска или USB накопителя.

Требования к транспортировке и хранению должны соответствовать условиям эксплуатации носителей, на которых находится программный продукт.

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса.

Предварительный состав программной документации включает в себя следующие документы:

* + техническое задание;
  + руководство оператора.

Разработка должна быть проведена в следующие стадии и этапы:

**Анализ требований:**

На стадии анализ требований формулируются цели и задачи проекта. Создается основа для дальнейшего проектирования

**Проектирование:**

На стадии проектирование должны быть выполнены перечисленные ниже этапы работ:

**Разработка программной документации;**

На этапе разработка программной документации должна быть выполнена разработка технического задания.

При разработке технического задания должны быть выполнены перечисленные работы: постановка задачи, определение и уточнение требований к техническим средствам, определение требований к программе, определение стадий, этапов и сроков разработки программы и документации на нее, выбор языков программирования.

**Разработка алгоритма программы;**

На этапе разработки алгоритма программы должен быть разработан алгоритм работы программы.

**Кодирование;**

На стадии кодирования происходит реализация алгоритмов в среде программирования.

**Тестирование и отладка.**

На стадии тестирование и отладка происходит проверка алгоритмов, реализованных в программе на работоспособность в различных ситуациях. Исправление выявленных ошибок, повторное тестирование.

Приемо-сдаточные испытания должны проводиться при использовании технических средств. Приемка программы заключается в проверке работоспособности программы путем ввода реальных или демонстрационных данных.

Во время приемки работы разработчик предоставляет программу и документацию, которая к ней прилагается. Проводятся испытания программы, при успешных испытаниях программа вводится в эксплуатацию. При ошибках, недопустимых для успешной работы программного продукта – отправляется на доработку.

Было описано техническое задание, содержащее в себе информацию о программном продукте, его функциях, эксплуатации и требования, которые должны учитываться при создании программы и документации к ней.

# ОПИСАНИЕ АЛГОРИТМОВ И СХЕМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОГО МОДУЛЯ

Данный программный модуль предназначен для автоматизации процесса информирования абитуриентов о поступлении в колледж. Он реализован как чат-бот в Telegram с использованием библиотеки telebot. Бот предоставляет информацию о колледже, отвечает на часто задаваемые вопросы и помогает абитуриентам сориентироваться в процессе поступления.

**Схема функционирования:**

1. **Инициализация и загрузка данных:**
   * При запуске модуля происходит инициализация бота с указанным токеном.
   * Из файла или другого источника данных загружается информация о колледже, которая хранится в структурированном виде (например, в словаре college\_info). Это включает в себя общую информацию о колледже, список специальностей, контакты, FAQ и другую необходимую информацию.
2. **Ожидание запросов:**
   * После инициализации бот переходит в режим ожидания запросов от пользователей Telegram. Библиотека telebot предоставляет механизмы для прослушивания входящих сообщений.
3. **Обработка команд:**
   * При получении команды (сообщение, начинающееся с символа /) бот определяет тип команды и вызывает соответствующую функцию-обработчик.
   * Каждая команда реализует определенную функцию, например,

предоставление информации о специальностях, контактных данных или графика работы приемной комиссии.

Пример: команда /Специальности вызывает функцию, которая формирует и отправляет пользователю список доступных специальностей и их описание.

1. **Обработка текстовых сообщений (NLP и FAQ):**
   * Если полученное сообщение не является командой, модуль переходит к обработке текста.
   * **Поиск по ключевым словам:** с помощью регулярных выражений производится поиск ключевых слов в тексте сообщения. Это позволяет определить тематику вопроса пользователя, даже если вопрос сформулирован нестандартно. Например, паттерн r"\b(проживани[ея]?|студент(ов|а|ы)?|общежити[ея]?)\b(?:\s+\b\w+\b){0,5}" позволяет определить вопросы об общежитии, независимо от точной формулировки.
   * **Поиск по FAQ:** если поиск по ключевым словам не дал результатов, производится поиск совпадения с вопросами из FAQ. Если вопрос пользователя похож на один из вопросов в FAQ, бот отправляет соответствующий ответ.
   * **Нераспознанные вопросы:** если ни один из вышеперечисленных методов не дал результата, бот отправляет сообщение о том, что не понял вопрос и предлагает воспользоваться командами или обратиться в приемную комиссию напрямую.
2. **Формирование и отправка ответа:**
   * После обработки запроса бот формирует ответ для пользователя. Ответ может содержать текст, ссылки, изображения или другие медиафайлы.
   * Библиотека telebot предоставляет функции для отправки сообщений различных форматов. Использование Markdown позволяет форматировать текст для лучшей читаемости.

**Алгоритмы:**

**Алгоритм обработки команд:** алгоритм основан на сопоставлении полученной команды с предопределенным списком команд и вызове соответствующей функции-обработчика.

**Алгоритм обработки текста:** алгоритм использует комбинацию регулярных выражений и поиска по FAQ для определения тематики вопроса и формирования ответа.

**Доступные функции:**

/start: начало работы с ботом, вывод приветственного сообщения и клавиатуры с основными командами.

/Информация: получение общей информации о колледже (название, адрес, контакты, сайт).

/График: получение информации о графике работы приемной комиссии.

/Специальности: просмотр списка доступных специальностей и их описание (срок обучения, квалификация).

/Вопросы: просмотр часто задаваемых вопросов (FAQ) и ответов на них.

/Контакты: получение контактной информации приемной комиссии, администрации и других отделов колледжа.

Обработка текстовых запросов, связанных с поступлением, проживанием, документами и другими вопросами.

**Дальнейшее развитие:**

**Интеграция с CRM или базой данных:** для хранения и обработки информации о заявках абитуриентов, автоматизации процесса приема документов.

**Расширенная обработка естественного языка (NLP):** использование более сложных NLP моделей для более точного понимания вопросов пользователей и предоставления более релевантных ответов. Возможность обучения модели на реальных вопросах абитуриентов.

**Персонализированные рекомендации:** на основе данных о пользователе (например, выбранные специальности) предоставлять персонализированные рекомендации и информацию.

**Многоязычность:** поддержка нескольких языков для общения с абитуриентами из разных стран.

**Интеграция с другими платформами:** расширение функциональности бота для работы в других мессенджерах или на сайте колледжа.

Это расширенное описание предоставляет более детальную информацию о функционировании чат-бота и его возможностях. Оно также включает в себя перспективы развития и улучшения функциональности бота в будущем.

# ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

Для проведения тестирования программы мною было произведено базовое тестирование во время разработки программы. При тестировании был выявлен ряд ошибок, которые возникли в ходе выполнения программы.

**Попытка использования слов не добавленных в паттерны базы знаний.**

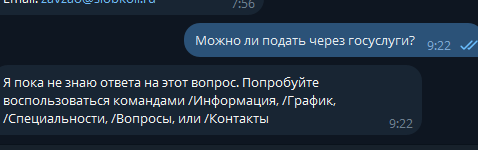


Рисунок 4 – Последствие использования не ключевых слов

Ожидаемый результат: Сообщение о невозможности ответить на данный вопрос.

Фактический результат: Сообщение о невозможности ответить на данный вопрос. (Рисунок 4).

Решение проблемы: добавление новых ключевых слов в базу знаний бота, для дальнейшего расширения его возможностей в диалоге.

**Попытка использования кнопок контекстного меню**

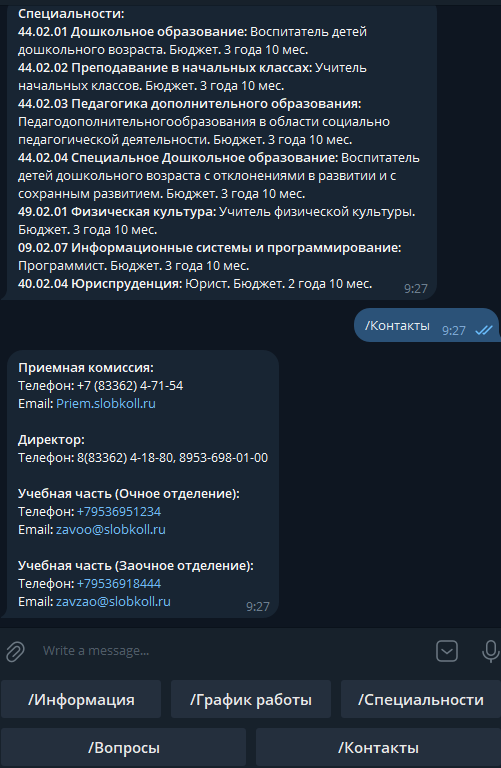


Рисунок 5 – Использование кнопок контекстного меню

Ожидаемый результат: при использовании кнопок контекстного меню вся информация отображается подробно и свободна для понимания абитуриента.

Фактический результат: при использовании кнопок контекстного меню вся информация отображается подробно и свободна для понимания абитуриента (Рисунок 5).

Решение проблемы: программирование кнопок с заранее прописанными данными для максимальной информативности и удобства абитуриента.

**Попытка использования кнопок контекстного меню с выключенным сервером**



Рисунок 6 – Использование кнопок контекстного меню при отключенном сервере

Ожидаемый результат: при использовании кнопок контекстного меню никакого ответа не последовало.

Фактический результат: при использовании кнопок контекстного меню никакого ответа не последовало. (Рисунок 6).

Примечание: после начало работы сервера, на все сообщения отправленные бот, незамедлительно произойдут ответы от чат-бота (Рисунок 7).

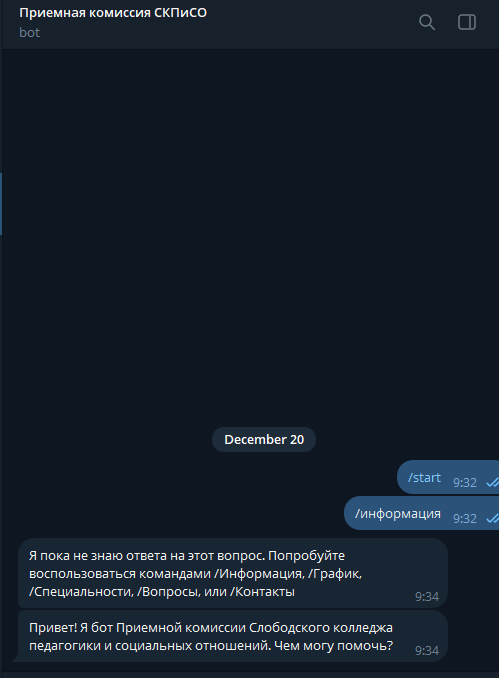


Рисунок 7 – Мгновенный ответ бота при включении сервера.

Решение проблемы: Качественная поддержка сервера для регулярной и качественной работы информационного чат-бота.

# ЗАКЛЮЧЕНИЕ

# В заключение, разработанный информационный чат-бот для приемной комиссии колледжа представляет собой эффективное решение для автоматизации процесса информирования абитуриентов. Данный проект успешно реализует основные функции, необходимые для предоставления информации о колледже, специальностях, процессе поступления и ответах на часто задаваемые вопросы. Использование Telegram в качестве платформы для бота обеспечивает широкий охват аудитории и удобный доступ к информации для абитуриентов.

# В ходе разработки были применены современные технологии, такие как библиотека telebot для взаимодействия с Telegram API и регулярные выражения для обработки текстовых запросов. Это позволило создать гибкий и функциональный инструмент, способный обрабатывать различные формулировки вопросов и предоставлять релевантную информацию.

# Тестирование программного модуля подтвердило корректность работы всех реализованных функций и алгоритмов. Бот успешно отвечает на команды, обрабатывает текстовые запросы и предоставляет информацию в соответствии с заданными сценариями.

# В целом, разработанный чат-бот является перспективным инструментом для повышения эффективности работы приемной комиссии колледжа и улучшения качества обслуживания абитуриентов. Дальнейшее развитие проекта позволит расширить его функциональность и сделать его еще более полезным для пользователей. Опыт, полученный в ходе разработки данного проекта, может быть использован для создания аналогичных решений для других образовательных учреждений.

# СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77. Единая система программной документации. Виды программ и программных документов, введ. 01.01.1978. – г. Москва: Изд-во стандартов, 1980. – 4 с.
2. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению, введ. 01.01.1980. – М.: Изд-во стандартов, 1988. – 3 с.
3. Документация по библиотеке telebot [Элекетронный ресурс] URL: <https://github.com/eternnoir/pyTelegramBotAPI> (дата обращения: 10.07.2024).
4. Python Tutorial (Официальное руководство по Python) [Элетронный ресурс] URL: <https://docs.python.org/3/tutorial/> (дата обращения: 10.07.2024).

5. Daemen J., Rijmen V. AES Proposal: Rijndael, 1999. [Электронный ресурс] URL: http://www.eng.tau.ac.il/~yash/crypto-netsec/Rijndael.pdf (дата обращения: 10.07.2024).

6. Lu, C.-S. Multimedia security: Steganography and bot sending techniques for protection of intellectual property – Hershey: Idea Group Publishing, 2005. – 255 p.

7. Microsoft Portable Executable and Common Object File Format Specification, 2017. [Электронный ресурс] URL: http://www.microsoft.com/whdc/system/platform/firmware/PECOFF.mspx (дата обращения: 10.07.2024).

8. National Policy on the Use of the Advanced Encryption Standard (AES). [Электронный ресурс] URL: http://cryptome.org/aes-natsec.htm (дата обращения: 10.07.2024).

9. Parallel AES Encryption Engines for Many-Core Processors Arrays, 2014. [Электронный ресурс] URL: http://www.rroij.com/open-access/parallelaes-encryption-engines-for-manycoreprocessor-arrays.pdf (дата обращения: 10.07.2024).

10. Pietrek M. An In-Depth Look into the PY, 2002. [Электронный ресурс] URL: https://msdn.microsoft.com/ruru/magazine/bb985992(en-us).aspx (дата обращения: 10.07.2024).

11. Pietrek M. An In-Depth Look into the PY, Part 2, 2002. [Электронный ресурс] URL: https://msdn.microsoft.com/ruru/magazine/bb985994(en-us).aspx (дата обращения: 10.07.2024).

12. Рietrek M. Peering Inside the PE: A Tour of the PY. 1994. [Электронный ресурс] URL: https://msdn.microsoft.com/en-us/library/ms809762.aspx (дата обращения: 10.07.2024).

13. Recommendation for Block Chiper Mods of Operation, 2001. [Электронный ресурс] URL: http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf (дата обращения: 10.07.2024).

14. Shin D., Kim Y., Byun K., Lee S. Data Hiding in Windows Executable Files, 2008. [Электронный ресурс] URL: http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1050&context=adf (дата обращения: 10.07.2024).

15. Specification for the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197, 2001. [Электронный ресурс] URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (дата обращения: 10.07.2024).

# 

# ПРИЛОЖЕНИЯ

**Приложение 1**

**РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ "ИНФОРМАЦИОННЫЙ ЧАТ-БОТ ДЛЯ ПРИЕМНОЙ КОМИССИИ КОЛЛЕДЖА"**

import telebot

import logging

import re

BOT\_TOKEN = '7318292371:AAFR19d8hjgSwzW-XMxip\_xZ2pDYQCPH0YQ'

bot = telebot.TeleBot(BOT\_TOKEN)

*# Настройка логирования*

logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')

*# Информация о колледже (словарь)*

college\_info = {

    "name": "Приемной комиссии Слободского колледжа педагогики и социальных отношений",

    "address": "Российская Федерация, Кировская область, г. Слободской, ул. Рождественская, д.69",

    "phone": "+7 (83362) 4-71-54",

    "website": "https://slobkoll.ru",

    "email" : "Priem@slobkoll.ru",

    "specialities": {

        "Специальности":"",

        "44.02.01 Дошкольное образование": "Воспитатель детей дошкольного возраста. Бюджет. 3 года 10 мес.",

        "44.02.02 Преподавание в начальных классах": "Учитель начальных классов. Бюджет. 3 года 10 мес.",

        "44.02.03 Педагогика дополнительного образования": "Педагодополнительногообразования в области социально педагогической деятельности. Бюджет. 3 года 10 мес.",

        "44.02.04 Специальное Дошкольное образование": "Воспитатель детей дошкольного возраста с отклонениями в развитии и с сохранным развитием. Бюджет. 3 года 10 мес.",

        "49.02.01 Физическая культура": "Учитель физической культуры. Бюджет. 3 года 10 мес.",

        "09.02.07 Информационные системы и программирование": "Программист. Бюджет. 3 года 10 мес.",

        "40.02.04 Юриспруденция": "Юрист. Бюджет. 2 года 10 мес.",

    },

    "admission\_info": "График работы приемной комиссии: \n ПН–ПТ с 9:00 до 16:00 \n СБ, ВС – выходные дни",

    "faq": {

        "\nДолжен ли родитель присутствовать при подаче докуменов абитуриентом?": "Да, конечно. Если на момент подачи документов, абитуренту нет 18 лет",

        "\nНужно ли оставлять оригинал аттестата сразу?": "Нет, но если к концу отбора его не будет, то абитуриент автоматически не проходит",

        "\nНужно ли будет проходить вступительные испытания?": "Для педагогических специальностей, да!. Для физической культуры также обязательно, но при наличии значков ГТО абитуриент освобождается от испытаний. Для программистов вступительные испытания не предусмотрены.",

        "\nКакие документы нужны для поступления?": "Паспорт, аттестат, медицинская справка, снилс."

    },

    "contacts": {

        "Приемная комиссия": {"phone": "+7 (83362) 4-71-54", "email": "Priem.slobkoll.ru"},

        "Директор": {"phone": "8(83362) 4-18-80, 8953-698-01-00"},

        "Очное отделение" :{"phone": "+79536951234", "email":"zavoo@slobkoll.ru"},

        "Заочное отделение" :{"phone": "+79536918444", "email":"zavzao@slobkoll.ru"},

    }

}

*# --- Обработчики команд ---*

@bot.message\_handler(commands=['start'])

**def** start\_command(message):

    keyboard = telebot.types.ReplyKeyboardMarkup(resize\_keyboard=True)

    buttons = ['/Информация', '/График работы', '/Специальности', '/Вопросы', '/Контакты']

    keyboard.add(\*[telebot.types.KeyboardButton(button) for button in buttons])

    bot.send\_message(message.chat.id, **f**"Привет! Я бот {college\_info['name']}. Чем могу помочь?", reply\_markup=keyboard)

@bot.message\_handler(commands=['Информация'])

**def** info\_command(message):

    info\_text = **f**"\*{college\_info['name']}\*\n" \

**f**"Адрес: {college\_info['address']}\n" \

**f**"Телефон: {college\_info['phone']}\n" \

**f**"Почта: {college\_info['email']}\n"  \

**f**"Сайт: [{college\_info['website']}]({college\_info['website']})\n"

    bot.send\_message(message.chat.id, info\_text, parse\_mode="Markdown")

@bot.message\_handler(commands=['График'])

**def** admission\_command(message):

    bot.send\_message(message.chat.id, college\_info['admission\_info'], parse\_mode="Markdown")

@bot.message\_handler(commands=['Специальности'])

**def** specialities\_command(message):

    specialities\_text = "\n".join([**f**"\*{name}\*: {description}" for name, description in college\_info['specialities'].items()])

    bot.send\_message(message.chat.id, specialities\_text, parse\_mode="Markdown")

@bot.message\_handler(commands=['Вопросы'])

**def** faq\_command(message):

    faq\_text = "\n".join([**f**"\*{question}\*: {answer}" for question, answer in college\_info['faq'].items()])

    bot.send\_message(message.chat.id, faq\_text, parse\_mode="Markdown")

@bot.message\_handler(commands=['Контакты'])

**def** contacts\_command(message):

    contacts\_text = **f**"\*Приемная комиссия:\*\nТелефон: {college\_info['contacts']['Приемная комиссия']['phone']}\nEmail: {college\_info['contacts']['Приемная комиссия']['email']}\n\n\*Директор:\*\nТелефон: {college\_info['contacts']['Директор']['phone']}\n\n\*Учебная часть (Очное отделение):\*\nТелефон: {college\_info['contacts']['Очное отделение']['phone']}\nEmail: {college\_info['contacts']['Очное отделение']['email']}\n\n\*Учебная часть (Заочное отделение):\*\nТелефон: {college\_info['contacts']['Заочное отделение']['phone']}\nEmail: {college\_info['contacts']['Заочное отделение']['email']}"

    bot.send\_message(message.chat.id, contacts\_text, parse\_mode="Markdown")

*# Обработчик текстовых сообщений*

@bot.message\_handler(func=**lambda** message: True)

**def** handle\_text(message):

    question = message.text.lower()

*# Регулярные выражения для поиска ключевых слов*

    patterns = {

**r**"\b(привет)": "Привет! Я бот Приемной комиссии от Слободского коллелджа педагогики и социальных отношений, чем могу помочь?",

**r**"\b(специальност(и|ей|ями)?|специальность)\b(?:\s+\b\w+\b){0,5}": "\n".join([**f**"\*{name}\*: {description}" for name, description in college\_info['specialities'].items()]),

**r**"\b(адрес(а|у)?|адрес)\b(?:\s+\b\w+\b){0,5}": college\_info['address'],

**r**"\b(телефон(а|у)?|телефон|номер)\b(?:\s+\b\w+\b){0,5}": college\_info['phone'],

**r**"\b(сайт(а)?|сайт|веб-сайт)\b(?:\s+\b\w+\b){0,5}": **f**"[{college\_info['website']}]({college\_info['website']})",

**r**"\b(процесс|поступлени[еяь]|поступа[тьт]|поступить|какие|экзамен(ы|а|ов)?)\b(?:\s+\b\w+\b){0,5}": "Подать документы можно несколькими способами.\n Самый простой способ, прийти очно в приемную комиссию и подать документы лично.\n Также предоставить документы возможно и через Портал Госуслуги, но в любом случае оригиналы документов нужно будет предоставить до конца работы приемной комиссии, такое возможно через МФЦ, Почту России или принести лично",

**r**"\b(проживани[ея]?|студент(ов|а|ы)?|общежити[ея]?)\b(?:\s+\b\w+\b){0,5}": "Нуждающимся иногородним студентам очного отделения предоставляется общежитие. Общежитие расчитано на 165 мест, 5 этажей",

**r**"\b(документ(ы|а|ов)?|зачислен(ия|ий|ию)?|необходимые\s+документы)\b(?:\s+\b\w+\b){0,5}": "Список документов для зачисления: Паспорт РФ, аттестат (2 страницы), медицинская справка(086-У), снилс.",

**r**"\b(спортивн(ые|ой|ого|ыми|ым)?|секци(и|й|ей|ями|ям)|кружк(и|а|ов|е|ой|у)?|спорт|секции|кружки)\b(?:\s+\b\w+\b){0,5}": "Информация о спортивных секциях и кружках: Волейбол, Баскетбол, Теннис(настольный) Танцы, Фитнес.",

**r**"\b(трудоустройств(о|а|у)|выпускник(ов|а|и|е|ой|у)?|работа|трудоустройство|карьера)\b(?:\s+\b\w+\b){0,5}": "Колледжом учрежден Центр карьеры (Центр содействия трудоустройству выпускников).",

**r**"\b(перевод|из|друг(ого|их|ое|ую|их)|учебн(ого|ых|ое|ую|ыми|ым)?|заведени[еяь]?|перевестись)\b(?:\s+\b\w+\b){0,5}": "При наличии вакантных бюджетных мест зачисление осуществляется на первый курс на базе основного общего образования и переводом на Индивидуальный учебный план (второй курс) на основании документов, подтверждающих соответствующий уровень образования.",

**r**"\b(инфраструктур(а|ы|е|ой|у)|библиотека|лаборатор(ия|ии|ий|иям|иями)|аудитор(ия|ии|ий|иям|иями)?|здание|кампус|оборудование)\b(?:\s+\b\w+\b){0,5}": "3 учебных здания, библиотека, столовая, спортзал. 4 мастерские",

**r**"\b(языков(ые|ого|ой|ыми|ым)?|курс(ы|а|ов)?|язык|английский|немецкий)\b(?:\s+\b\w+\b){0,5}": "В колледже предусмотрено изучение 2 языков: английский и немецкий",

}

    for pattern, answer in patterns.items():

        if re.search(pattern, question):

            bot.send\_message(message.chat.id, answer, parse\_mode="Markdown")

            return

*# Проверка вопросов из FAQ*

    for q, a in college\_info['faq'].items():

        if q.lower() in question:

            bot.send\_message(message.chat.id, a, parse\_mode="Markdown")

            return

*# Если ни один паттерн не совпал*

    bot.send\_message(message.chat.id, "Я пока не знаю ответа на этот вопрос. Попробуйте воспользоваться командами /Информация, /График, /Специальности, /Вопросы, или /Контакты")

*# Запуск бота*

bot.polling()

Полный исходный код программного продукта поставляется на DVD накопителе в Word файле (Приложение 3).

**Приложение 2**

**РУКОВОДСТВО ОПЕРАТОРА**

Данное руководство предназначено для операторов приемной комиссии, использующих чат-бота для взаимодействия с абитуриентами. Оно описывает основные функции бота, порядок работы с ним и способы решения возможных проблем.



Рисунок 8 – QR-код, ссылка на чат-бота

**1. Общие сведения:**

Чат-бот – это программа, которая автоматически отвечает на вопросы абитуриентов в Telegram. Он предоставляет информацию о колледже, специальностях, процессе поступления и других важных аспектах. Бот работает круглосуточно и позволяет операторам приемной комиссии сосредоточиться на более сложных задачах.

**2. Функциональные возможности бота:**

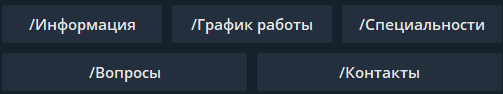


Рисунок 9 – Функциональные кнопки в контекстном меню

* **Ответ на команды:** Бот реагирует на определенные команды, предоставляя информацию по запросу:
  + /start – начало работы с ботом, отображение приветствия и основных команд.
  + /Информация – общая информация о колледже (адрес, контакты, сайт).
  + /Специальности – список доступных специальностей и их описание.
  + /График работы – график работы приемной комиссии.
  + /Вопросы – ответы на часто задаваемые вопросы (FAQ).
* /Контакты – контактная информация приемной комиссии и других отделов колледжа (Рисунок 9).
* **Обработка текстовых запросов:** Бот анализирует текстовые сообщения от пользователей и пытается найти подходящий ответ, используя ключевые слова и базу знаний FAQ (Рисунок 10).

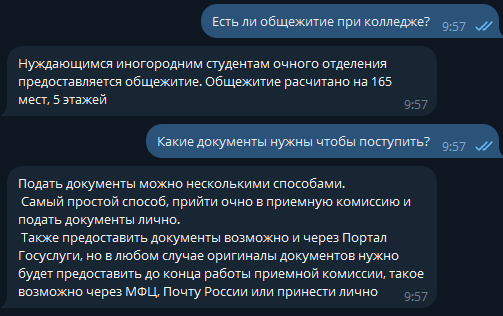


Рисунок 10 - Обработка текстовых запросов

Команды, которые на данный момент включены в базу данных бота приведены в блоке «Patterns» (Рисунок 11).

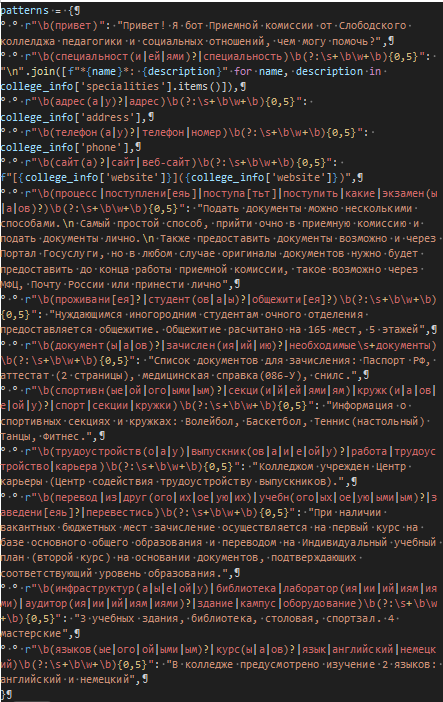


Рисунок 11 - Блок ключевых слов «Patterns»

**3. Порядок работы с ботом:**

* **Мониторинг работы бота:** Периодически проверяйте работу бота, убедитесь, что он отвечает на запросы пользователей корректно.
* **Обработка нестандартных ситуаций:** Если бот не может ответить на вопрос пользователя, он уведомит об этом оператора (если реализована такая функция). В этом случае оператор должен подключиться к разговору и предоставить необходимую информацию.
* **Обновление информации:** Следите за актуальностью информации, предоставляемой ботом. При необходимости обновляйте базу знаний FAQ, информацию о специальностях и другие данные. (Опишите, как это делается, если есть специальный интерфейс или процедура).
* **(Опционально) Работа с CRM/базой данных:** Если бот интегрирован с CRM/базой данных, операторы могут использовать его для управления заявками абитуриентов, отслеживания статусов и коммуникации с абитуриентами. (Опишите конкретные действия)

**4. Решение возможных проблем:**

* **Бот не отвечает:** Проверьте подключение к интернету. Убедитесь, что сервер, на котором работает бот, функционирует нормально. Обратитесь к администратору системы, если проблема не устранена.
* **Бот отвечает некорректно:** Проверьте базу знаний FAQ и информацию о колледже на наличие ошибок. Обратитесь к разработчику бота для исправления ошибок в коде.
* **Проблемы с CRM/базой данных:** Если возникли проблемы с интеграцией с CRM/базой данных, обратитесь к администратору системы или разработчику бота.

Это руководство поможет операторам приемной комиссии эффективно использовать чат-бота для взаимодействия с абитуриентами и оптимизировать свою работу. Регулярно обновляйте руководство по мере развития функциональности бота.

**Приложение 3**

Программный продукт поставляется на DVD накопителе.

На DVD накопителя находиться:

* Файл курсового проекта в формате Microsoft Word (.docx)
* Файл исходного кода программного продукта в формате Microsoft Word (.docx)
* Файл с презентацией курсового проекта
* Файл исходного кода программного продукта в формате Python(.py)

Также программный продукт можно получить, введя ссылку:

* GitHub: https://github.com/Koni1ak/KURSOVAYA\_RABOTA\_SM