

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«Финансовый университет при Правительстве
Российской Федерации»**

(Финансовый университет)

Колледж информатики и программирования

(наименование структурного подразделения)

Дипломный проект

Тема «Проектирование и разработка телеграмм - бота приемной комиссии IT-
колледжа Московского финансово-юридического университета (МФЮА)

(наименование)

Студент Старостин Дмитрий Дмитриевич

(фамилия, имя, отчество полностью)

Учебная группа 4ИСИП-520

Специальность 09.02.07 Информационные системы и программирование

(код и наименование специальности)

Руководитель

дипломного проекта

(подпись)

С.М. Володин

(инициалы, фамилия)

Председатель предметной

(цикловой) комиссии

(подпись)

Т.В. Соловьева

(инициалы, фамилия)

Москва – 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Глава 1. Аналитическая часть.....	7
1.1. Описание предметной области.....	7
1.2. Сравнительный анализ ботов-аналогов	8
1.3. Постановка задачи.....	14
1.4. Инструментальные средства реализации программы	15
Глава 2. Проектирование и реализация.....	20
2.1. Анализ требований и определение функциональности бота	20
2.2. Проектирование программного обеспечения	23
2.3. Разработка программного обеспечения	23
2.4. Тестирование телеграмм-бота	33
2.5. Пояснительная записка	35
ЗАКЛЮЧЕНИЕ	36
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	38
ПРИЛОЖЕНИЯ.....	40
Приложение 1	40

ВВЕДЕНИЕ

В эру цифровизации, когда технологии неустанно прогрессируют, каждому человеку необходимо регулярно обновлять свои знания и умения в различных сферах. Это касается не только профессионального роста на рабочем месте, но и адаптации старшего поколения к новейшим компьютерным технологиям, что стало обязательным из-за перехода многих организаций на электронный формат работы (например, клиники, пенсионные фонды и прочее). Также это относится к освоению учащимися программ для удалённого образования, что становится особенно важным во время эпидемических вспышек или для обучения детей с особыми потребностями, а также к необходимости для всех слоёв населения изучать использование интернет-магазинов и сервисов для покупки товаров и услуг.

Но обучение нужно не только для комфортного проживания в государстве и развития кругозора, обучение в любом возрасте так же способствует:

- Стимулирование умственного развития (Математика и естественные науки способствуют формированию критического мышления, обучают методам решения проблем, основам логики и фактическому анализу. Гуманитарные науки, в свою очередь, развивают навыки критического осмысления, умение убедительно излагать свои мысли и понимание социальных процессов).

- Поддержание психического благополучия (Активное участие в интеллектуальных занятиях способствует долгосрочному сохранению умственных функций. Взрослые и пожилые люди, которые не прекращают обучение и трудовую деятельность, значительно реже подвержены риску развития деменции и неврологических заболеваний, таких как болезни Альцгеймера и Паркинсона, что положительно сказывается на продолжительности и качестве их жизни.)

- Гибкость перед лицом перемен (В современном мире, где профессиональные требования меняются с невероятной скоростью,

студенты, выбирающие сегодня востребованные специальности, к моменту окончания учёбы могут обнаружить, что их навыки устарели. Однако, если они обладают способностью к обучению, они смогут эффективно адаптироваться к новым реалиям. Во времена кризиса такие люди могут легко переквалифицироваться и найти применение своим знаниям в других сферах.)

– Навык анализа информации (Студенты в процессе обучения сталкиваются с обширным потоком учебных материалов, которые им необходимо осмысленно фильтровать и извлекать ключевые аспекты для усвоения дисциплины. Эта способность критически оценивать и отбирать информацию окажется бесценной в будущем, позволяя быстро адаптироваться к различным жизненным обстоятельствам, находить нужные знания и делать обоснованные решения.)

– Расширение социального круга (С возрастом часто происходит сокращение круга общения, однако открытость новым увлечениям может привести к знакомству с людьми различного возраста и интересов, что способствует социальной интеграции и стимулирует личностный рост.)

– Прогресс в профессиональной сфере (Обогащение знаний и навыков способствует увеличению вашей ценности на рынке труда. Работодатели склонны отдавать предпочтение кандидатам с более высоким уровнем образования для заполнения вакансий с большим потенциалом.)

– Увеличение жизненных перспектив (Приобретение новых знаний раскрывает перед вами двери к неизведанным возможностям и предоставляет доступ к дополнительным ресурсам для обогащения вашей жизни.)

– Жизнь, полная смысла, без уныния (По мере взросления, жизненный круг часто ограничивается повседневностью. Непрерывное образование предотвращает погружение в будничность и способствует ощущению того, что каждый прожитый день наполнен целью.)

– Становление абитуриентом (Когда вы решаете продолжить образование в учебном заведении, вы получаете статус абитуриента. Это возможно вне зависимости от возраста или места работы, поскольку в нашей стране существует свобода обучения без всяких ограничений.)

Цель данной выпускной квалификационной работы создать телеграмм-бота для приемной комиссии ИТ- колледжа Московского финансово-юридического университета, который бы содержал основную информацию по вопросам при выборе данного учебного заведения и заинтересовал бы перспективой обучаться в нём. Используя бота, пользователь сможет получить ответы по часто возникающим вопросам при поступлении и узнать о жизни колледжа в учебный период.

Актуальность создания приложения обусловлена высокой популярностью среди населения и удобством использования мессенджера, как со стороны пользователей, так и со стороны учреждения заказчика. Телеграм-боты работают круглосуточно, семь дней в неделю, выполняя однообразную работу и этим разгружая сотрудников организации, чем способствуют улучшению решений более важных задач.

Предметом исследования являются технологии и методы разработки телеграм-бота.

Задачи, поставленные для достижения намеченной цели:

- Предпроектное исследование предметной области
- Сравнительный анализ аналогичных ботов
- Проектирование данных для бота
- Разработка программного обеспечения в виде написания кода программы
- Составление пояснительной записки

Для достижения поставленных задач будут применяться следующие инструментальные средства, языки проектирования и разработки программного обеспечения:

- Интегрированная среда разработки для языка программирования Python - PyCharm
- Язык программирования Python
- Мессенджер Телеграмм
- Сервер для проектирования диаграмм Draw.io
- Текстовый редактор Microsoft Word
- Программа подготовки и просмотра презентаций Microsoft PowerPoint.

Объектом исследования является Аккредитованное образовательное частное учреждение высшего образования «Московский финансово-юридический университет МФЮА» по направлению подготовки 09.03.02 «Информационные системы и технологии», профиль «Информационные системы и технологии в бизнесе».

Источниками для исследования являются законодательные и нормативные документы, учебная и научная литература, а также интернет-ресурсы.

Глава 1. Аналитическая часть

1.1. Описание предметной области

Сложно переоценить значение мобильного телефона в современной жизни. Он служит не только средством связи с близкими и коллегами, но и источником информации, управления финансами, а также предлагает множество приложений для творческого и здорового образа жизни. Всё это доступно в любое время и в любом месте.

В мире, где каждый день приходится делать выбор среди множества опций, легкость доступа к информации может сыграть ключевую роль в принятии решений. Современные технологии предлагают различные способы получения информации, от телефонных звонков до веб-сайтов и приложений с автоматическими ответами, в числе которых выделяются телеграм-боты. Телеграм-боты интегрированы в один из наиболее востребованных мессенджеров, что делает доступ к информации ещё более удобным, поскольку нет необходимости загружать новые приложения. Для использования ботов не требуется предоставлять личные данные, что упрощает взаимодействие и способствует защите личной информации в сети. Что касается функционала ботов, они взаимодействуют через кнопки или текстовый ввод, создавая диалог с вопросами и ответами. Кроме того, боты могут включать в себя интернет-магазины с интерактивными каталогами и возможностью добавления товаров в корзину и их оплаты, а также мобильные приложения банков или других организаций, что особенно ценно в условиях санкций. Разработка бота обходится значительно дешевле, чем создание мобильного приложения или веб-сайта. Телеграмм-бот, разрабатываемый в рамках дипломной работы, создается для упрощенного и быстрого получения информации при выборе колледжа для обучения. Исходя из этого можно сделать вывод, что бот должен включать в себя:

- Список интересующих разделов
- Информацию раздела, сгруппированную для удобства в подразделы.

- Канал для общения с другими пользователями и специалистами.
- Возможность перейти на сайт, для более углубленного изучения вопроса

1.2. Сравнительный анализ ботов-аналогов

Механизмы функционирования ботов можно классифицировать на три основных типа:

Первый тип — это боты, управляемые через кнопки, которые предоставляют информацию или направляют пользователя на веб-сайты для доступа к необходимым данным.

Второй тип — боты, похожие на интернет-магазины, включающие в себя каталоги товаров или услуг, ссылки, ценники, изображения продукции и так далее.

Третий тип — боты, основанные на технологии искусственного интеллекта, которые встречаются гораздо реже первых двух типов.

Чаще всего боты имеют смешанный тип работы, но суть остается у всех одна, боты являются программами для получения информации. Телеграмм-бот, разрабатываемый в рамках дипломной работы, будет содержать информацию об услугах колледжа и ценах на них, поэтому мы для сравнительного анализа возьмём два бота один из которых создан для помощи абитуриентам ИГУ, а второй уже непосредственно сам товар.

Бот «Абитуриент ИГУ»

Интерактивный помощник, предназначенный для облегчения процесса подачи документов в учебное заведения. Он может предоставлять информацию о различных программах обучения, условиях поступления, необходимых документах и сроках. Кроме того, чат-бот может направлять пользователей на соответствующие ресурсы или контактные данные для дополнительной помощи.. Начальная страница бота представлена на рисунке 1:

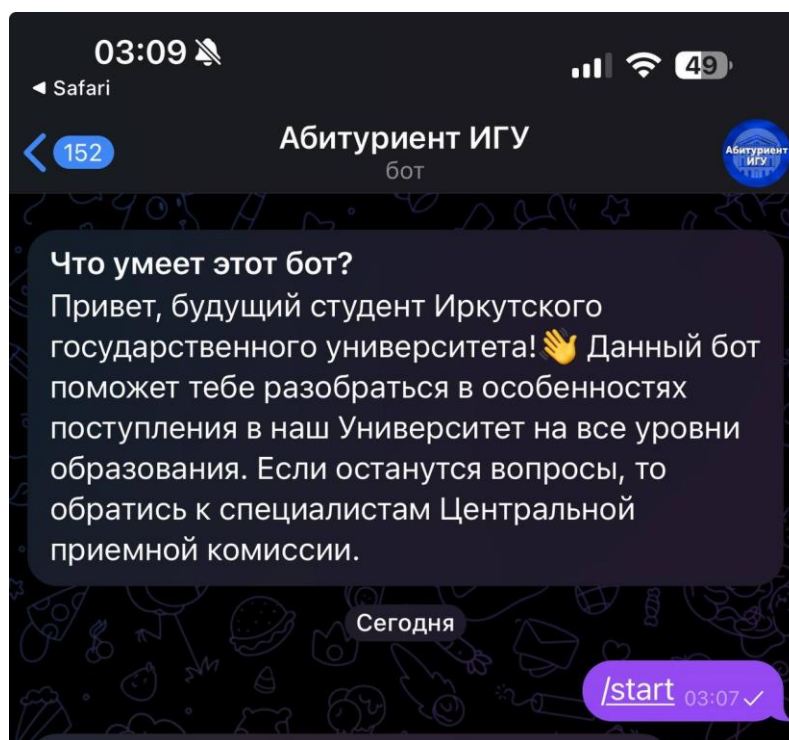


Рисунок 1. Начальная страница бота.

В данном боте нет раздела подготовки к ЕГЭ, нет возможности узнать о дне открытых дверей, ознакомиться с ценами. Бот предлагает перейти по ссылке на сайт, где можно рассчитать результаты ЕГЭ нужные для поступления. Имеется вкладка «Способы подачи документов» которая предлагает перейти в личный кабинет абитуриента

Дата начала приема абитуриентов 20 июня 2024 года.
Ознакомиться с информацией о поступлении в Иркутский
государственный университет можно на сайте
<https://isu.ru/abitur/pk2024/main/>.
Задать все интересующие вопросы можно по электронной почте
prim@isu.ru и по номерам телефонов приемной комиссии (3952) 521-
555, 521-777

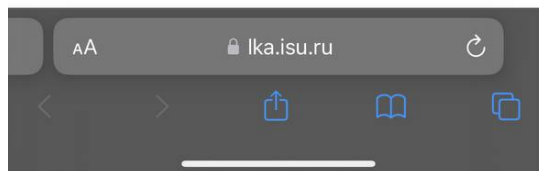


Рисунок 2. Сайт с информацией.

Достоинства:

- понятный способ пользования ботом
- есть ссылки на сайт ИГУ
- удобный интерфейс

Недостатки:

- невозможность написать администратору
- много ссылок, в которых можно запутаться

В итоге этот бот представляет собой ссылки на различные источники помогающие при поступление.

Бот «Я студент АГТУ»:

Данный бот тоже помогает абитуриентам при поступление в Университет, он представляет собой каталогом услуг, обладает возможностью получения информации о количестве бюджетных мест. После получения нужной информации предлагает перейти на сайт университета.

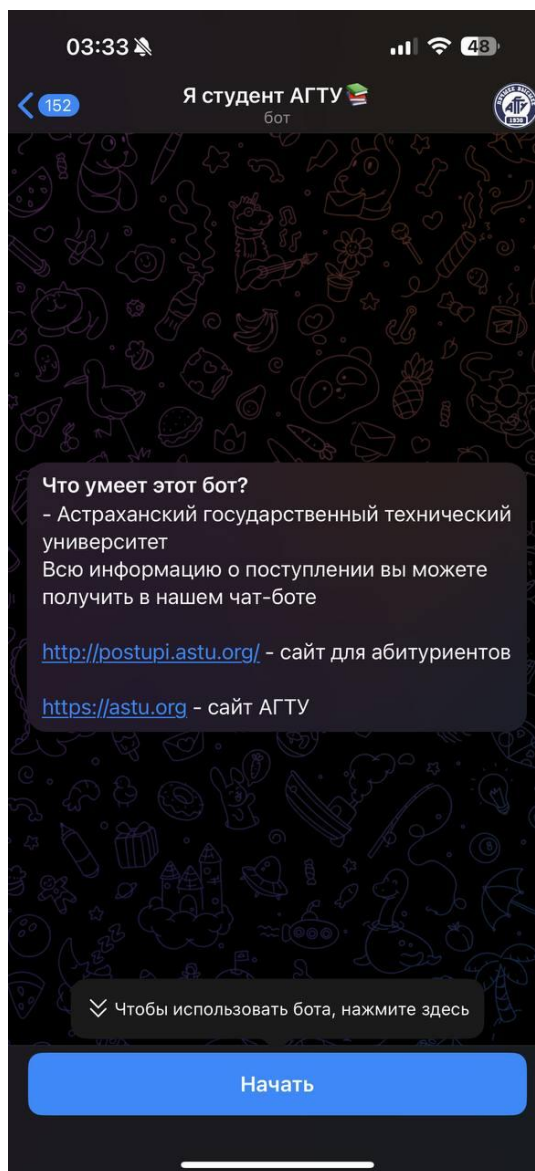


Рисунок 3. Начальная страница бота АГТУ.

На рисунке 6 представлено меню бота, в котором имеется изобилие информации объединенное по группам для облегчения поиска:

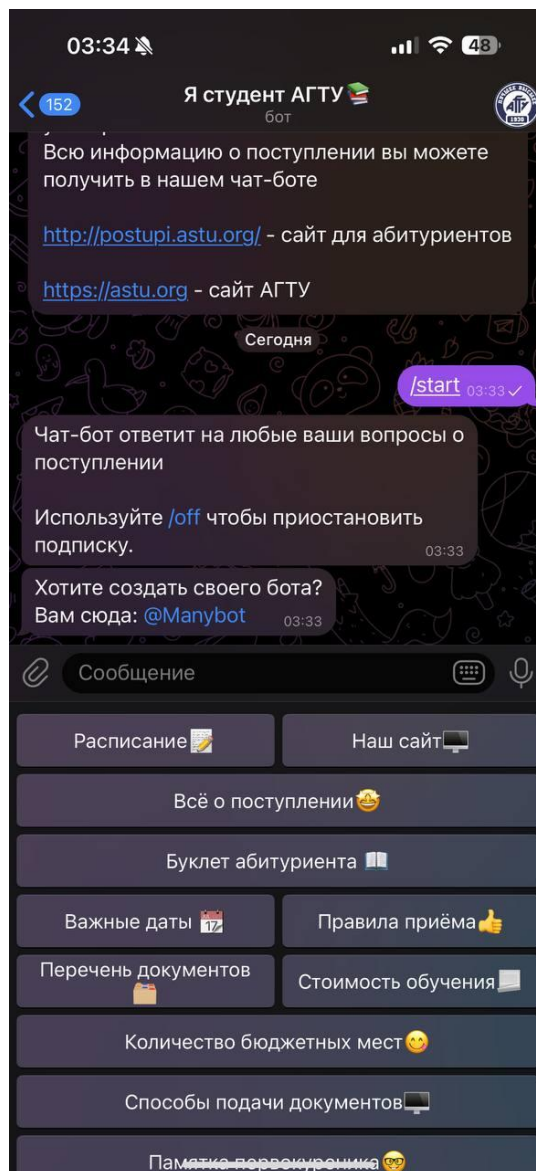


Рисунок 4. Меню бота АГТУ.

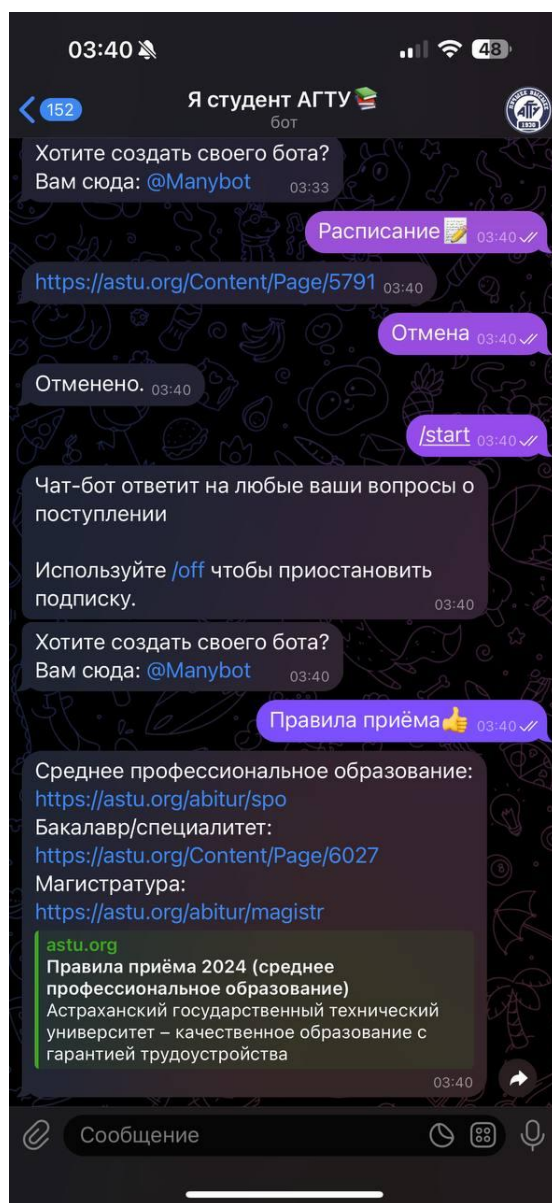


Рисунок 5. Пример поиска информации.

Плюсы:

- удобный интерфейс
- меню возможностей бота с использованием смайлов и описанием
- возможность поиска большого количества информации в боте
- есть возможность уточнения через сообщение в поддержку

Минусы:

- нет ссылок на доп. информацию

Бот представляет собой полноценный помощник для абитуриентов и студентов, но ссылки на дополнительную информацию ограничены тем, что чаще всего интересует абитуриентов.

В таблице 1 показано наглядное сравнение основных функций, по которым можно понять, какой бот будет предпочтительнее для пользователя при решении сделать заказ товара.

Таблица 1. Сравнительная таблица аналогов.

Плюс	puHerb	Доставка IHerb
Удобство интерфейса	+	+
Наличие чата	-	+
Доп. ссылки	-	+
Обильное меню	+	-
Оформление документов в боте	-	-
Высокая скорость получения информации	+	+
Стоимость	Бесплатно	Бесплатно

Главным преимуществом разрабатываемого бота является то, что в нем объединены плюсы обоих аналогов, а именно: есть и информация в самом боте, и дополнительные ссылки, и возможность сообщений, и возможность связи с менеджером.

1.3. Постановка задачи

При взаимодействии с ботом пользователи будут направлены к инструкциям на домашней странице, которые помогут им с ориентацией по следующим шагам. В меню есть опция "Изучить разделы бота", выбрав которую, можно увидеть набор кнопок, каждая из которых перенаправляет в определенный раздел. Внутри каждого раздела расположены дополнительные кнопки, предоставляющие углубленную информацию по

соответствующим темам. Также имеется функция перехода в чат-канал для взаимодействия с сообществом.

Доступные разделы:

Подготовительные курсы – информация для тех, кто собирается успешно сдать ОГЭ и поступить на бюджет.

Поступление – в разделе можно найти информацию по направлениям колледжа, о приемной комиссии, узнать контрольные цифры приема, есть ли учет индивидуальных достижений и какие есть места для перевода.

Оплата обучения – информация о стоимости и способах обучения, всё про стипендии и материальную поддержку, и немного о кредите на образование.

Договора – представлены списки документов для заключения договоров и образцы самих договоров.

Результаты и списки – представлен пофамильный список лиц, зачисленных на бюджет, так же данные по статистике приема и можно ознакомиться с приказами о зачислении.

Интересное в учебном году – самый позитивный раздел в боте, в нем можно посмотреть, как проходят дни открытых дверей, какие проводятся олимпиады и конкурсы в колледже, узнать интересные новости колледжа, позавидовать студентам просматривая список доступных им электронных библиотек.

Рекомендуемые системные требования:

- Операционная система Android 7.0 и выше
- Процессор мощностью 1,6 ГГц
- Объем оперативной памяти не менее 2ГБ
- 300 МБ свободной память

1.4. Инструментальные средства реализации программы

В процессе разработки программного обеспечения были использованы следующие инструменты:

- PyCharm интегрированная среда разработки для языка программирования Python
- Язык программирования Python
- Мессенджер Телеграмм
- Сервер для проектирования диаграмм Draw.io
- Текстовый редактор Microsoft Word
- Программа подготовки и просмотра презентаций Microsoft PowerPoint.

PyCharm — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA. PyCharm обладает рядом преимуществ:

- Помощь при написании (механизм анализа кода обеспечивает автодополнение, поиск и исправление ошибок, имеет удобную навигацию по коду)
- Повышает качество кода (делает проверки соответствия требованиям PEP8, умных перестроений кода, оказывает помощь при тестировании)
- Помимо Python, PyCharm поддерживает JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML/CSS, языки шаблонов, AngularJS, Node.js и другие технологии.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Сферы применения языка Python:

Веб-разработка – включает в себя отправку данных на серверы и с серверов, обработку и взаимодействие с базами данных, маршрутизацию URL-адресов и обеспечение безопасности.

Графический интерфейс – написанный на Python легко подстраивается под стилистику ОС, в чьей среде создается приложение.

Базы данных – на сегодняшний день есть доступ к любым базам данных, так же в языке есть SQLite встроенная в библиотеку. Она содержит практически все функции реляционной базы данных и представлена всего одним файлом.

Системное программирование – язык применяется для написания приложений, используемых системными администраторами, ускоряет поиск и открытие файлов, запуск приложений, облегчает вычисления и многое другое.

Сложные вычислительные процессы – проходя интеграцию с расширением NumPy, питон прекрасно функционирует с массивами и интерфейсами уравнений, прорисовывает объекты в среде 3D производя параллельные вычисления.

Машинное обучение – Python помогает реализовывать системы распознавания лиц и голоса, создавать нейронные сети и т. д.

Автоматизация процессов – одна из самых востребованных функций питона, при довольно простых кодах можно автоматизировать монотонные рабочие процессы, например сортировать письма по ключевому слову. Простой синтаксис позволяет легко разработать сценарий, сам код не проходит компиляцию, что облегчает процесс отладки.

Игры – чаще всего пишут скрипты для взаимодействия персонажей, запуска сцен и обработки событий.

Python так же имеет ряд библиотек, которые позволяют программистам быстрее и эффективнее писать программы. Многие специалисты выбирают Python из-за его простоты в синтаксисе, большой областью применения в решении как повседневных задач, так и конкретных

разработок внутренней части веб-сайтов и приложений. Для Python написано множество фреймворков ускоряющих работу с кодом: Django, CherryPy, Flask, TurboGears, Tornado, Pyramid, Web2py. Программы на Python гибкие, легко масштабируются, не требуют большой команды разработчиков. Python имеет мощное сообщество, которое его популяризирует и развивает.

Telegram — кроссплатформенная система мгновенного обмена сообщениями с функциями VoIP, позволяющая обмениваться текстовыми, голосовыми и видеосообщениями, стикерами и фотографиями, файлами многих форматов.

На данный момент зарегистрировано около 10млн пользователей. Имеет ряд преимуществ по сравнению с остальными мессенджерами:

- сообщения зашифрованы и есть функция самоуничтожения
- доступ с любого устройства, облачное хранение
- высокая скорость работы, серверы распределены по всему миру
- приложение бесплатное, нет рекламы
- постоянное улучшение и предложения новых функций

Telegram так же имеет функцию поиска пользователей не только по номеру телефона, но и по публичным именам, что позволяет отправить сообщение не зная номера.

Draw.io — бесплатная платформа для рисования диаграмм, интеллектуальных карт, бизнес-моделей, программных блоков и другое. Содержит открытый исходный код для возможности интеграции диаграмм, к примеру, на сайт. Имеет множество функций для редактирования, визуализации задач пользователя. Сервис не требует обязательной регистрации или авторизации на сайте, предлагает только выбрать место для экспорта созданного проекта. Взаимодействие с программой происходит в таком порядке: пользователь выбирает шаблоны фигур или элементов и перетаскивает на рабочую поверхность, далее производит редактирование, посредством изменения размера, цвета, шрифта, свойств фигуры и т. д. Более того, Вы сможете

восстанавливать изменения проектов, отслеживать их выполнение, редактирование; осуществлять экспорт/импорт файлов в PDF, XML, PNG, HTML, VSDX.

Microsoft Word – это многофункциональный программный комплекс обработки текстов. Программа предназначена для выполнения работ по созданию документов, включающих разнообразные элементы (рисунки, графики, формулы, обычные или электронные таблицы, фрагменты БД и т. д.), имеющих иерархическую организацию (главы, части, параграфы и т. п.) с обеспечением работы на уровне отдельных компонентов, документа в целом, объединяющих информацию нескольких файлов в виде главного документа.

Программа подготовки презентаций и просмотра презентаций Microsoft PowerPoint, являющаяся частью Microsoft Office и доступная в редакциях для операционных систем Microsoft Windows и macOS, а также для мобильных платформ Android и IOS. Материалы, подготовленные с помощью PowerPoint, предназначены для отображения на большом экране — через проектор, либо телевизионный экран большого размера.

Глава 2. Проектирование и реализация

2.1. Анализ требований и определение функциональности бота

Бот представляет собой программу, разработанную для того, чтобы автоматически реагировать на команды, вводимые пользователями. Спектр возможностей ботов широк: они могут выполнять всё, начиная с поиска данных и предоставления новостных обновлений, заканчивая операциями на фондовом рынке.

На рисунке 6 показано как происходит взаимодействие между пользователем и ботом. Когда вы взаимодействуете с ботом, нажимая на кнопки, этот цикл постоянно повторяется.

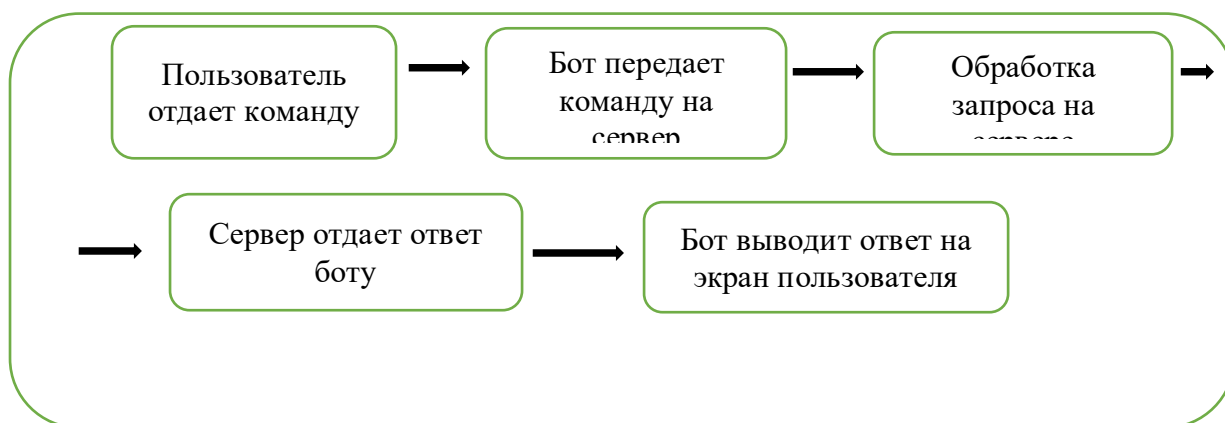


Рисунок 6. Взаимодействие между пользователем и ботом.

Современные телеграмм-боты настолько интуитивно понятны, что даже начинающие пользователи могут легко ими пользоваться. Разработка ботов охватывает широкий спектр применений: от лёгкого развлечения с юмором и эмодзи до серьёзных аналитических инструментов, предоставляющих финансовые данные и новости с мест событий.

В то время как развлекательные боты наполнены весёлыми элементами, боты для профессионального использования, такие как боты для колледжей, предназначены для предоставления полезной информации, помогающей абитуриентам с выбором и поступлением в образовательные учреждения.

Требования к телеграмм-боту для колледжа:

- Простой, строгий и удобный интерфейс
- Максимальное количество разделов, которые могут заинтересовать абитуриента
- Возможность получить дополнительную информацию с сайта колледжа
- Возможность личного общения с консультантом колледжа

Функциональность данного проекта представлена следующими диаграммами:

- Краткая функциональная диаграмма
- Детализированная функциональная диаграмма
- Диаграмма вариантов использования
- Диаграмма Ганта

На рисунке 7 представлена краткая функциональная диаграмма, на которой входными данными являются сведения, прописанные в боте, информационная база сайта колледжа и возможность получить информацию общаясь на прикрепленном к боту канале. На выходе мы получаем информацию по запросу пользователя.

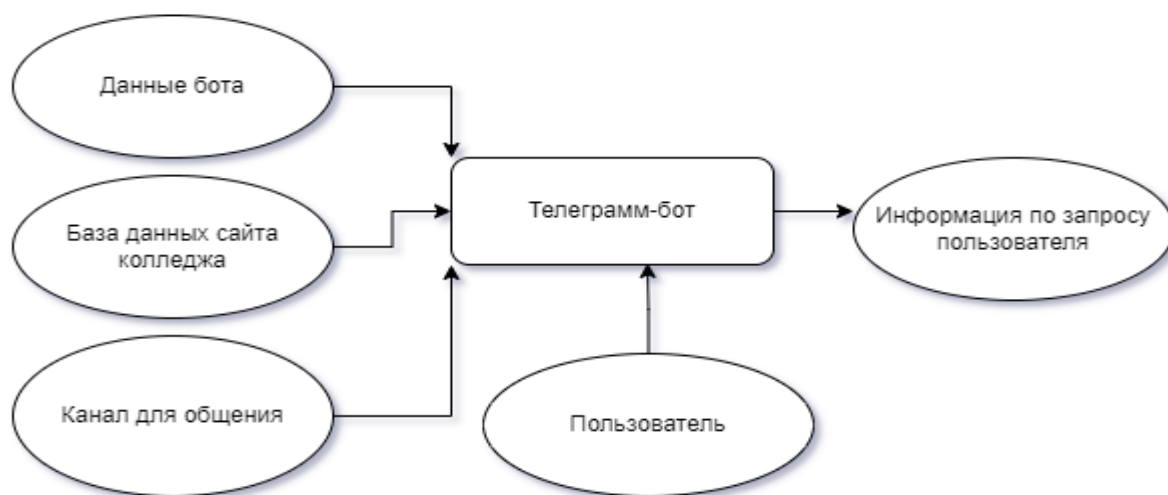


Рисунок 7. Краткая функциональная диаграмма.

Детализированная функциональная диаграмма, представленная на рисунке 8, содержит основные функции программы.



Рисунок 8. Детализированная функциональная диаграмма.

На рисунке 9 представлена диаграмма вариантов использования - диаграмма, описывающая какой функционал разрабатываемой программной системы доступен каждому пользователю.

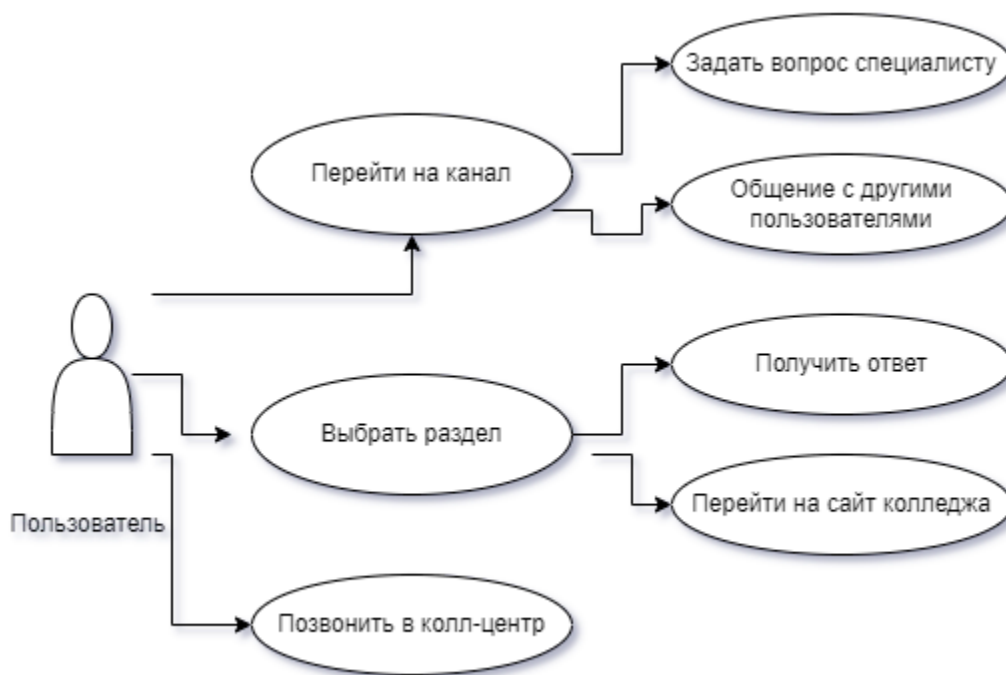


Рисунок 9. Диаграмма вариантов использования.

Диаграмма Ганта – это столбчатая диаграмма, которая используется для визуального предоставления графика работ, иллюстрирует планы или последовательность их выполнения. Диаграмма позволяет оценить все ресурсы и взаимосвязи задач, запланировать подход к их выполнению,

увидеть степень выполнения той или иной задачи, увидеть на каких этапах возникают сложности и многое другое. Чаще всего диаграмму Ганта можно встретить в методологиях, которые требуют четкой фиксации задач и порядка работ. Помимо этого, диаграмма Ганта подойдет всем, кто предпочитает планировать и работать с удобными графиками. К сожалению, у этой диаграммы есть и минусы: проекты редко полностью придерживаются плана, и диаграмма требует постоянного обновления.

На рисунке 10 представлена диаграмма Ганта показывающая уровень легкости прохождения этапов проекта на заданный период времени, что отражалось на разработке функциональности проекта.

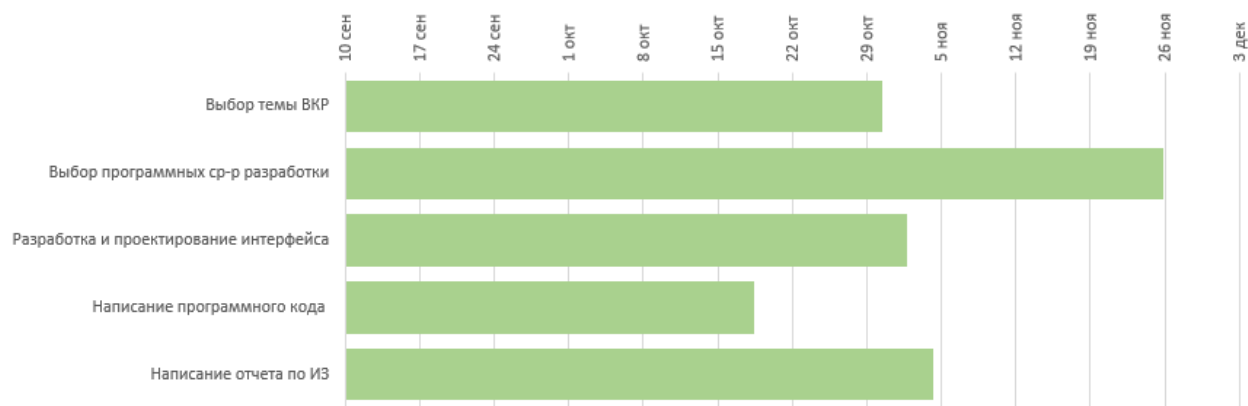


Рисунок 10. Диаграмма этапов проекта по уровню легкости.

2.2. Проектирование программного обеспечения

Под проектированием программного обеспечения подразумевается комплекс мероприятий, направленных на разработку и составление технических заданий для проекта. Проектирование включает этапы:

- Определение и ознакомление с областью работы бота
- Определение основных тем вопросов бота
- Разработка эскиза интерфейса бота

2.3. Разработка программного обеспечения

Для разработки Телеграмм-бота используется PyCharm интегрированная среда разработки для языка программирования Python. Устанавливаем на ПК PyCharm и Python. В PyCharm устанавливаем библиотеку pyTelegramBotAPI и создаем папку для написания самого кода.

Чтобы установить библиотеку telebot # в терминале вставить `pip install pyTelegramBotAPI` и нажать Enter

```
import time ;import telebot ;from telebot ; import types
```

Так же нам потребуется дать боту название, адрес и получить taken – строку идентификации бота для серверов Telegram. Далее в Telegram находим BotFather.



Рисунок 11. Стартовая страница BotFather.

Нажимая на кнопку старт, получаем список команд, выбираем команду /newbot (создаем нового бота), прописываем имя бота и адрес, получаем токен.

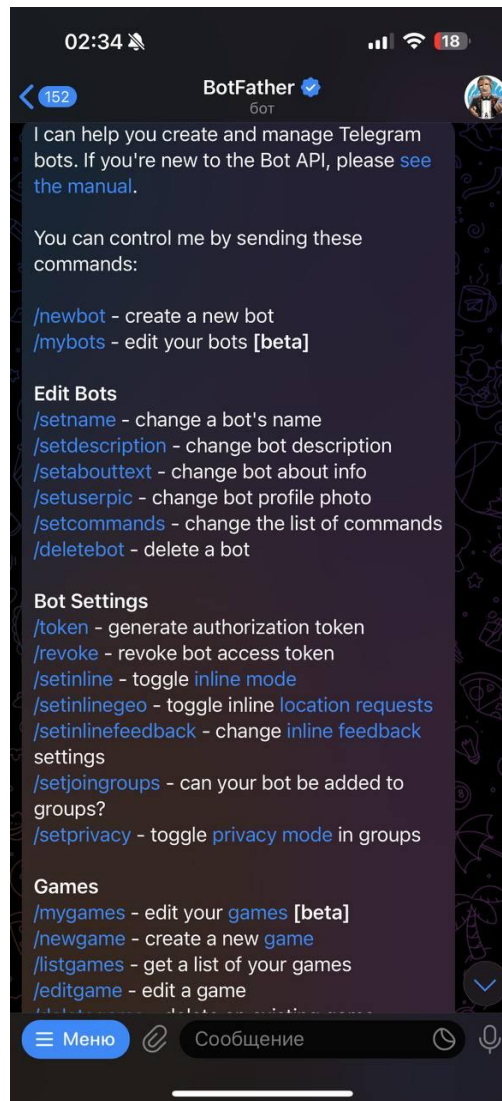


Рисунок 12. Начало работы с командами.

```
@bot.message_handler(commands=['start']) # Обработка команды /start def
start(message):
```

```
    bot.send_message(message.chat.id, ""
```

Привет! Здесь можно найти ответы на вопросы, которые могут возникнуть при выборе нашего учреждения для обучения.

```
<a href="https://t.me/ +zTpEalPP_G03N2Ey">Перейти в канал</a>
```

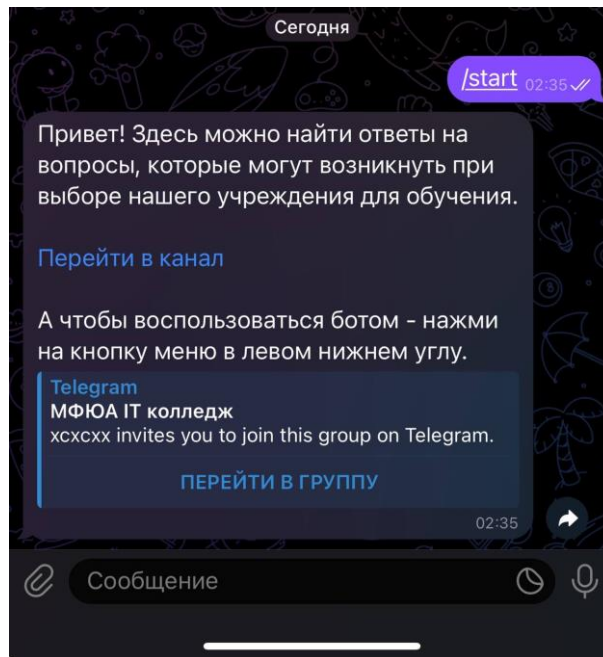


Рисунок 13. Стартовая страница нового бота.

А чтобы воспользоваться ботом - /razdel. "", parse_mode='HTML') #
Внедрен код HTML разметки для создания гиперссылки, обязательно
указывается в обработке текста 'parse_mode'

Начинаем работу по прописыванию основных кнопок:

```
@bot.message_handler(commands=['razdel']) # Обработка команды /razdel
def razdel(message):
    kb = types.InlineKeyboardMarkup(row_width=1) # добавляем клавиатуру
    # создаем поочередно кнопки
    # в callback_data задаем название кнопки для дальнейшей обработки ее
    # нажатия
    item1 = types.InlineKeyboardButton(text='Подготовительные курсы',
    callback_data='courses') # кнопка курсы
    item2 = types.InlineKeyboardButton(text='Поступление',
    callback_data='postup') # кнопка поступление
    item3 = types.InlineKeyboardButton(text='Оплата обучения',
    callback_data='pay_obuch') # кнопка оплата обучения
    item4 = types.InlineKeyboardButton(text='Договора',
    callback_data='dogovora') # кнопка договора
    item5 = types.InlineKeyboardButton(text='Результаты и списки',
    callback_data='res') # кнопка результаты и списки
    item6 = types.InlineKeyboardButton(text='Интересное в учебном процессе',
    callback_data='podacha') # кнопка интересное
    item7 = types.InlineKeyboardButton(text='Не нашли нужный раздел',
    callback_data='nenash') # кнопка не нашли нужный раздел
    item8 = types.InlineKeyboardButton(text='В начало',
```

```
callback_data='mainmenu') # главное меню
kb.add(item1, item2, item3, item4, item5, item6, item7, item8) # добавляем
кнопки в клавиатуру
bot.send_message(message.chat.id, "Выберите интересующий Вас раздел:",
reply_markup=kb) # отправляем сообщение, в reply_markup
указываем нашу клавиатуру
```

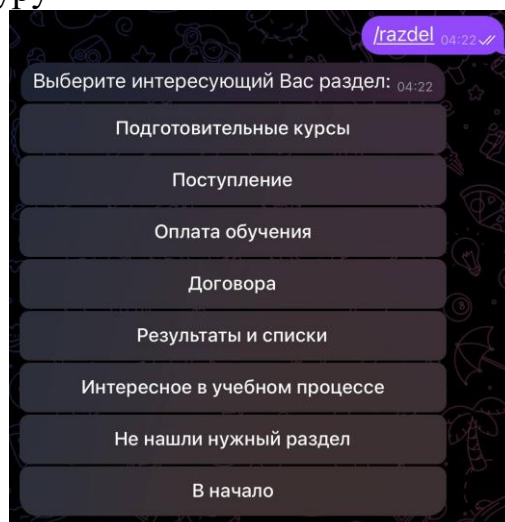


Рисунок 14. Кнопки основных разделов.

Каждый раздел содержит ряд подразделов, например на рисунке 15 изображены подразделы раздела Поступление:

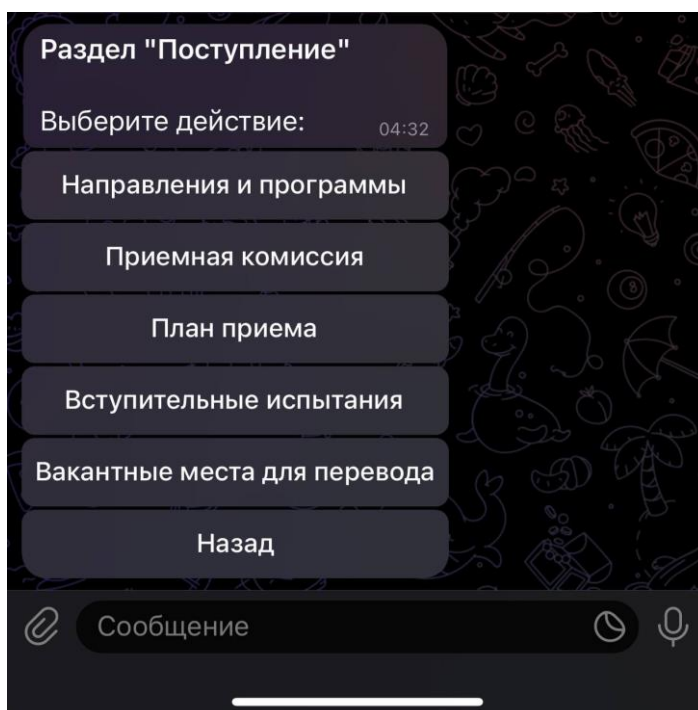


Рисунок 15. Подразделы

Для возврата в список разделов прописываем кнопку «назад»: `item1 = types.InlineKeyboardButton(text='Назад',callback_data='mainmenu')` # добавим кнопку возвращения `kb.add(item1)`

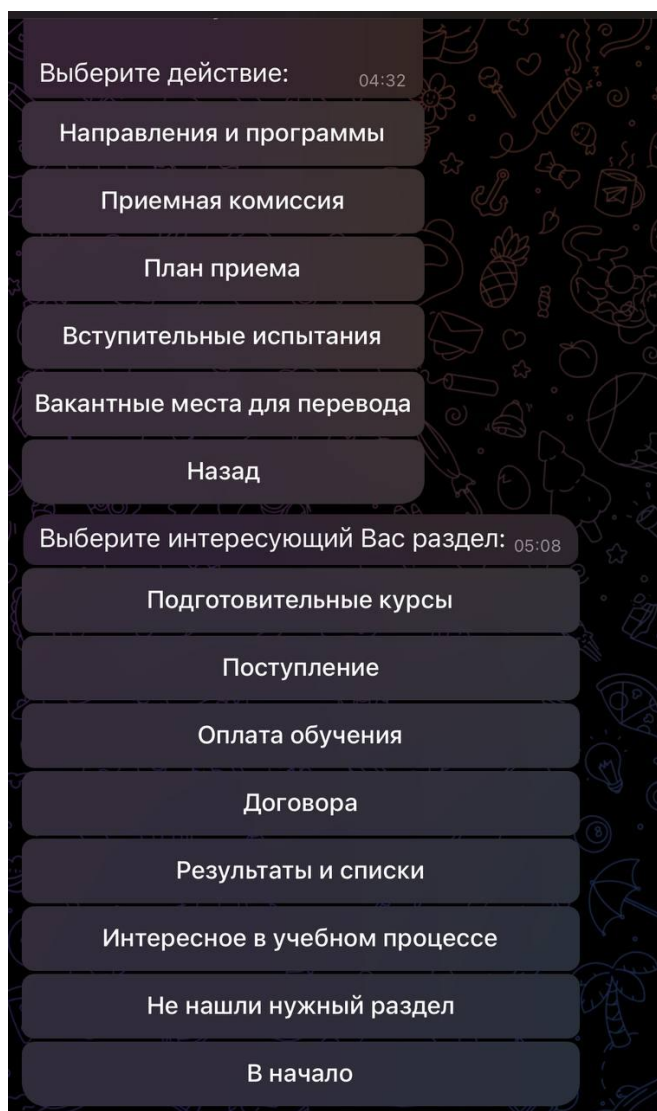


Рисунок 16. Кнопка «Назад»

Попадая в подраздел, мы получаем информацию по вопросу, сформулированному на кнопке подраздела:

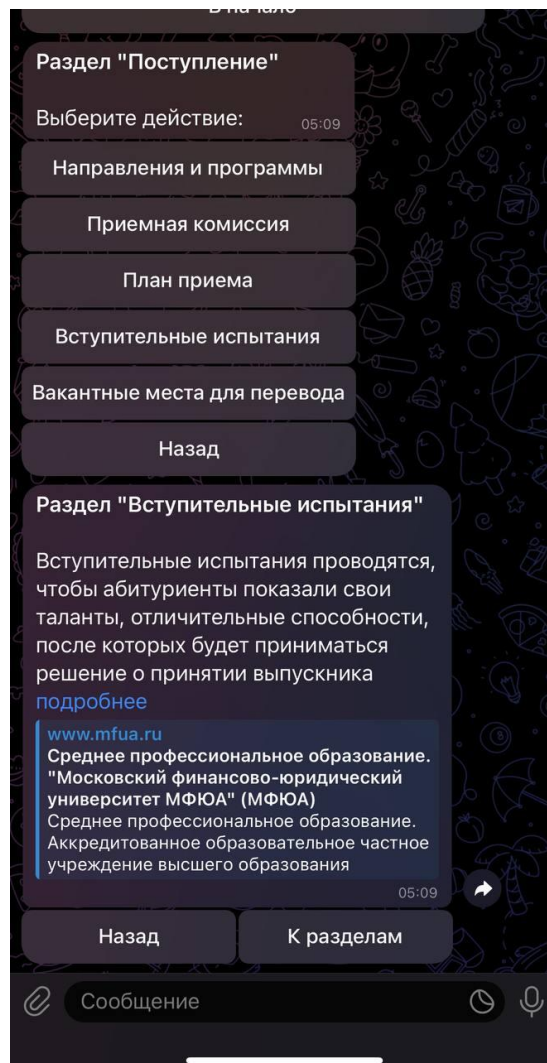


Рисунок 17. Информация по подразделу «Вступительные испытания»

```
elif call.data == 'uchet': # обрабатываем кнопку вступительные испытания
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                         callback_data='postup') # в callback_data указываем
название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел "Вступительные
испытания"</b>
Вступительные испытания проводятся, чтобы абитуриенты показали свои
таланты, отличительные способности, после которых будет приниматься
решение о принятии выпускника <a
href="https://www.mfua.ru/abitur/spo_sredn/#abitur_vstupitelnye-
ispytania">подробнее</a>
"', parse_mode='HTML', reply_markup=kb)
```

В подразделе после информации есть возможность перейти на сайт колледжа, нажав на гиперссылку «подробнее», и изучить вопрос более обширно.

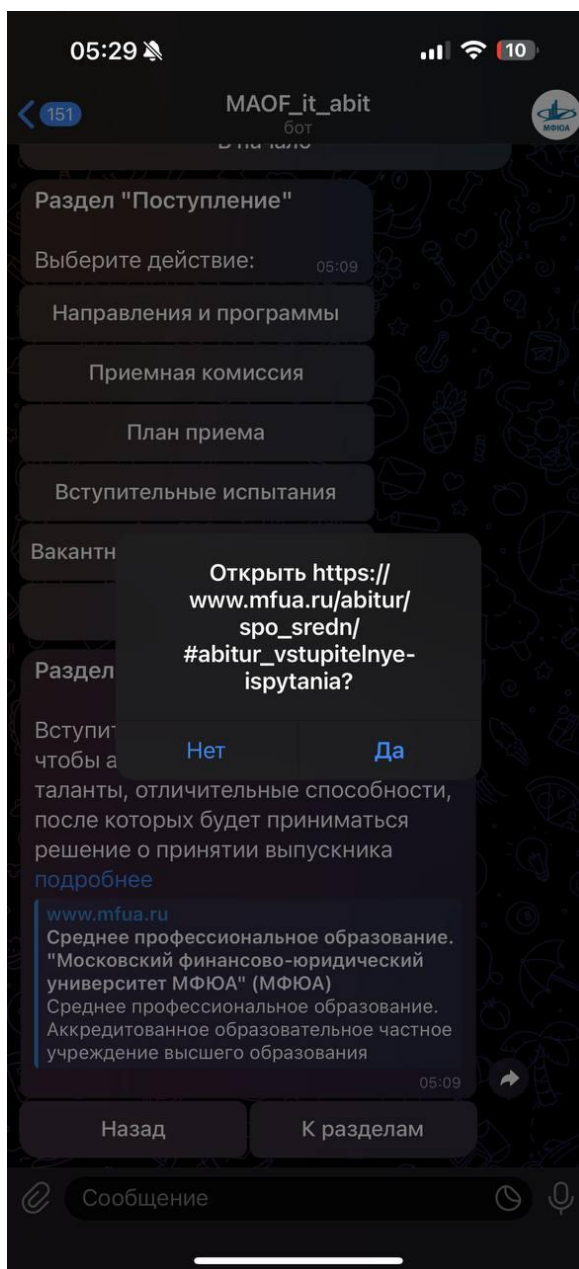


Рисунок 18. Гипер ссылка «подробнее».

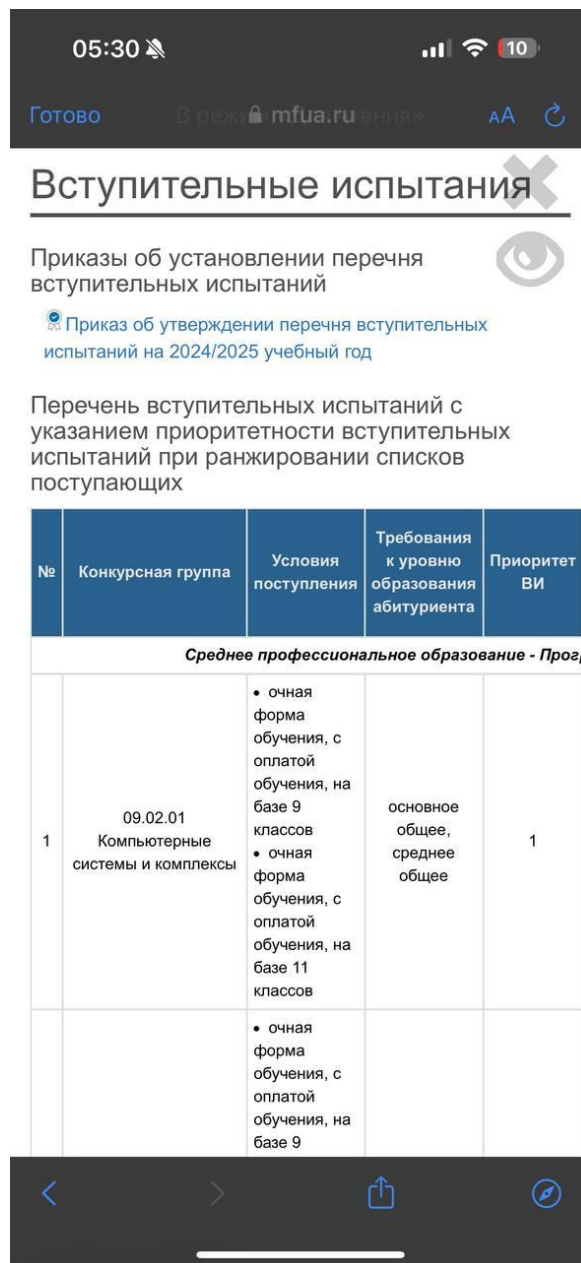


Рисунок 19. Переход по ссылке на сайт колледжа.

Так же из подраздела можно вернуться либо на предыдущую страницу, либо к основным разделам, кнопки показаны на рисунке 20.

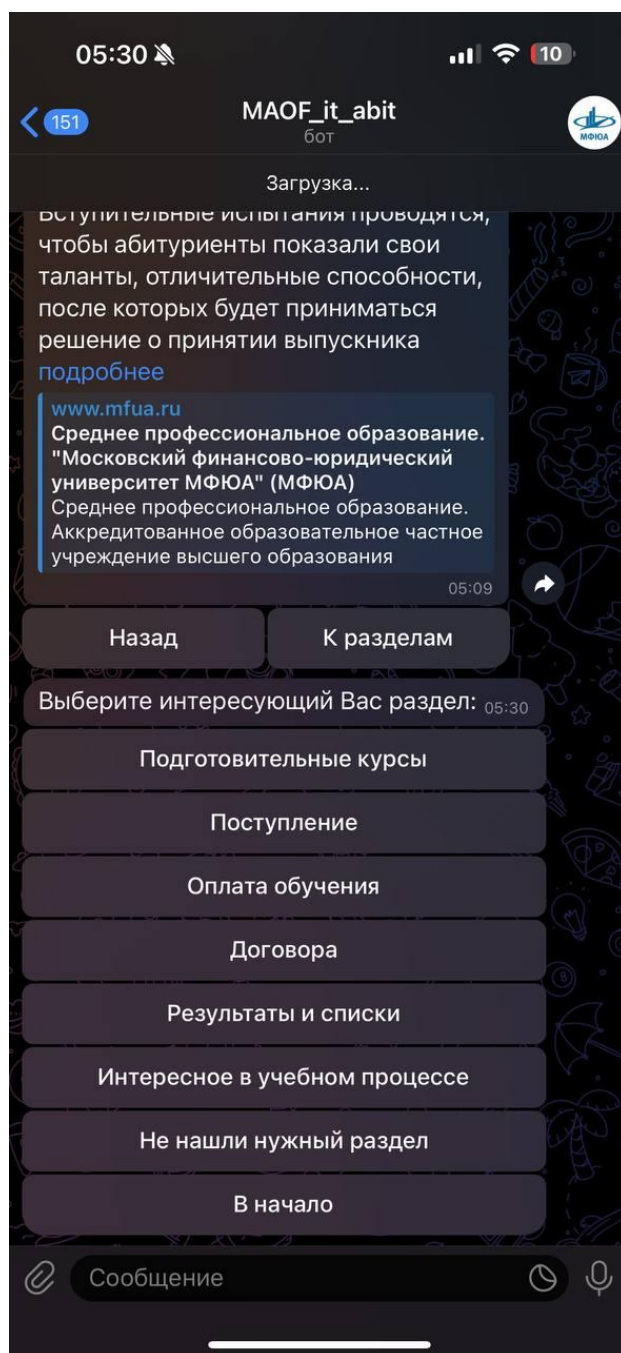


Рисунок 20. Кнопки возврата из подраздела.

Если пользователь не нашел интересующий его раздел, есть возможность перейти на канал и задать там свой вопрос, показано на рисунке 21.

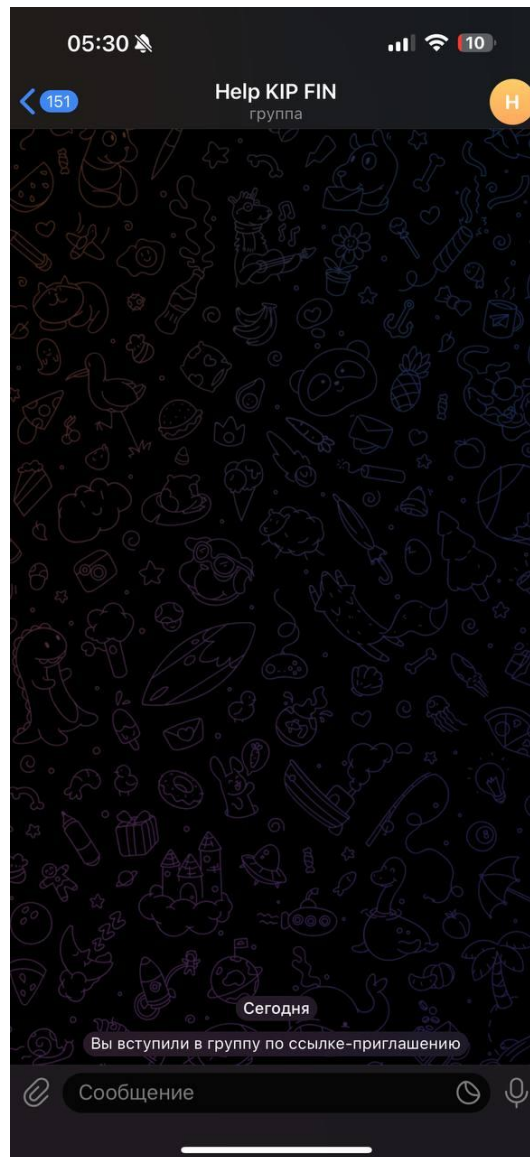


Рисунок 21. Переход на канал чата.

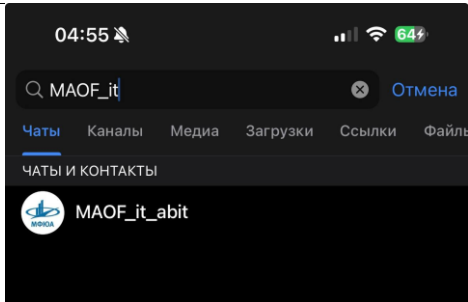
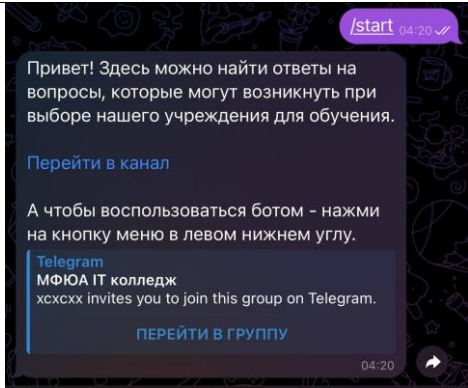
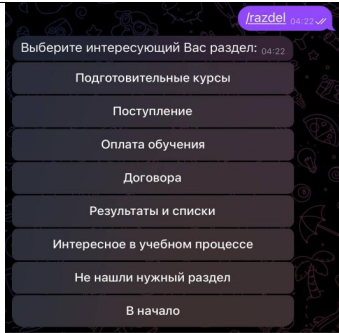
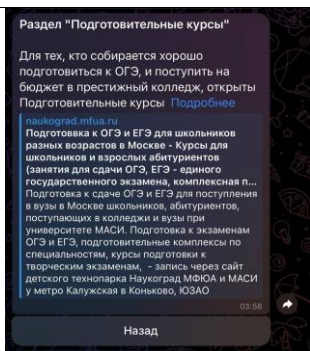
Как только интересный вопрос поступит в чат, первый же освободившийся консультант ответит на вопрос, помимо консультанта информацией в этом чате могут делиться и другие пользователи.

Полный код телеграмм-бота представлен в приложении А.

2.4. Тестирование телеграмм-бота

Для проверки реализованного телеграмм-бота необходимо сделать отладку и тестирование приложения. Результаты данного тестирования представлены в таблице 2.

Таблица 2. Тестирование телеграмм-бота.

№ Теста	Вводимый запрос	Ожидаемая реакция бота	Фактический результат	Ошибки
1	Вводим название Бота в мессенджере Телеграмм	Поиск бота в мессенджере Телеграмм		Нет
2	В боте нажимаем команду Start	Вывод основного меню бота и предложения выбора для начала работы		Нет
3	Просмотр и выбор раздела бота	Вывод информации по разделу (подразделу)		Нет
4	Выбор получения информации по ссылке на сайт	Бот переводит пользователя на сайт		Нет

2.5. Пояснительная записка

Программисту:

Назначение бота – информационный проект для абитуриентов колледжа.

Бот написан в интегрированной среде PyCharm на языке Python. Для работы с базой данных использована библиотека pyTelegramBotAPI и мессенджер Телеграмм для размещения бота.

Файл с функциями для работы с ботом лежит в основной папке проекта и называется PythonProect2, он позволяет делать запросы в базу, редактировать, удалять и добавлять данные.

Для корректной работы бота устройство должно отвечать следующим характеристикам:

- операционная система Android 7.0 и выше;
- процессор мощностью 1,6 ГГц;
- объем оперативной памяти не менее 2 ГБ;
- 300 МБ свободной памяти.

Пользователю:

Для того чтобы начать пользоваться ботом вам нужно установить на своём смартфоне мессенджер Телеграмм, после ввода всех личных данных и успешной регистрации нужно в поисковой строке написать название бота Bot Fa.ru, заходим в бота и дальнейшее использование не составит никакого труда.

Для корректной работы бота устройство должно отвечать следующим характеристикам:

- операционная система Android 7.0 и выше;
- процессор мощностью 1,6 ГГц;
- объем оперативной памяти не менее 2 ГБ;
- 300 МБ свободной памяти.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы, целью которой была реализация Телеграмм-бота «MAOF_it_abit», позволяющего быстро просмотреть основную информацию касающуюся поступления в колледж, достигнута.

Были решены все поставленные задачи, а именно:

- проанализированы требования и спецификации;
- спроектировано программное обеспечение;
- спроектирована и разработана база данных;
- разработан и написан код программы;
- проведено тестирование Телеграмм-бота.

В разработанном приложении реализованы следующие функции:

- создание разделов по основным вопросам про поступление
- создание подразделов, для получения более детального ответа
- возможность просмотреть сайт колледжа по ссылкам из бота
- возможность получения ответа лично от консультанта

В процессе работы были выявлены различные ошибки, но благодаря тестированию и отладке продукта быстро решены. Полученный опыт поможет в дальнейшей разработке.

Благодаря спроектированной базе данных можно дополнять информацию в будущем.

В результате самостоятельного изучения инструментальных средств и реализации данного проекта, мною были освоены навыки работы со специальными средами для языка Python, библиотекой pyTelegramBotAPI, куда входит подключение библиотеки к интегрированной среде, изменение и удаление элементов, добавление новых элементов, привязка различных функций управления к определенным элементам.

Так же мной был самостоятельно пройден онлайн-курс «Программирование на Python» Stepik, который рассказывает об основных

Так же мной был самостоятельно пройден онлайн-курс «Программирование типов данных, конструкциях и принципах структурного программирования, используя версию языка Python ветки 3.x.

Результатом данной дипломной работы стал информационный Телеграмм-бот.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ Р 7.0.12-2011 Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила составления. – М.: Стандартинформ, 2012. – 61 с. – Текст: непосредственный.
2. ГОСТ 7.32-2017 Отчет о научно-исследовательской работе. Структура и правила оформления – М.: Стандартинформ, 2017. – 47 с. – Текст: непосредственный.
3. ГОСТ Р 7.0.100-2018 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Стандартинформ, 2018. – 122 с. – Текст: непосредственный.
4. ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления. – Текст: непосредственный.
5. Единая система программной документации. – М.: Стандартинформ, 2005. – 128 с. – Текст: непосредственный.
6. Иванова Г.С. И201 . Технология программирования: Учебник для вузов. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. - 320 с.: ил. (Сер. Информатика в техническом университете.) ISBN 5-7038-2077-4
7. Фуфаев Э. В.Ф964 Разработка и эксплуатация удаленных баз данных : учебник для студ. учреждений сред. проф. образования / Э. В. Фуфаев, Д. Э. Фуфаев. — 3-е изд., стер. — М. : Издательский центр «Академия», 2012. — 256 с.ISBN 978-5-7695-9343-7
8. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. – Москва: ФОРУМ: ИНФРА-М, 2020. – 400 с. – (Среднее профессиональное образование). – ISBN 978-5-8199-0812-9. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1067012>– Текст: электронный.
9. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Издательство

Юрайт, 2020. – 235 с. – (Профессиональное образование). – ISBN 978-5-534-05047-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/453640>– Текст: электронный.

10. Лутц, Марк.Л86 Изучаем Python, том 2, 5-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2020.— 720 с. : ил. — Парал. тит. англ. ISBN 978-5-907144-53-8 (рус., том 2)ISBN 978-5-907144-51-4 (рус., многотом.)

11. Интернет-сервис для построения схем и диаграмм Draw.io. – [Электронный ресурс]. – URL: <https://www.draw.io/> (дата обращения: 10.12.2022) – Текст: электронный.

12. Чернышев, С. А. Основы программирования на Python : учебное пособие для среднего профессионального образования / С. А. Чернышев. — Москва : Издательство Юрайт, 2023. — 286 с. — (Профессиональное образование). — ISBN 978-5-534-15160-2. — Текст : электронный

13. Полное руководство по языку программирования Python – [Электронный ресурс]. – URL: <http://metanit.com/sharp/tutorial/> – Текст: электронный.

14. Интернет-ресурс по созданию Телеграмм-бота с нуля <https://skillbox.ru/media/code/chatboty-v-telegram-na-python-chast-1/> [Электронный ресурс]. Текст: электронный

15. Интернет-ресурс по основным командам на Python - <https://eternalhost.net/blog/razrabotka/komandy-python> [Электронный ресурс]. Текст: электронный

16. Интернет-ресурс <https://www.youtube.com/> [Электронный ресурс]. Текст: электронный

ПРИЛОЖЕНИЯ

Приложение 1

Код приложения “ Проектирование и разработка телеграмм - бота
приемной комиссии ИТ- колледжа Московского финансово-юридического
университета (МФЮА)»”

```
import time
import telebot
from telebot import types

TOKEN = '7223336335:AAFH19y29v61nkuEelTL_UP67tIUL7goqT8' # Токен
бота
bot = telebot.TeleBot(TOKEN)

@bot.message_handler(commands=['start']) # Обработка команды /start
def start(message):
    bot.send_message(message.chat.id, ""
Привет! Здесь можно найти ответы на вопросы, которые могут возникнуть
при выборе нашего учреждения для обучения.

<a href="https://t.me/+zTpEalPP_G03N2Ey">Перейти в канал</a>

А чтобы воспользоваться ботом - /razdel.

    "",
        parse_mode='HTML') # Внедрен код HTML разметки для
создания гиперссылки, обязательно указывается в обработке текста
'parse_mode'

@bot.message_handler(commands=['razdel']) # Обработка команды /razdel
def razdel(message):
    kb = types.InlineKeyboardMarkup(row_width=1) # добавляем клавиатуру
    # создаем поочередно кнопки
    # в callback_data задаем название кнопки для дальнейшей обработки ее
    нажатия
    item1 = types.InlineKeyboardButton(text='Подготовительные курсы',
callback_data='courses') # кнопка курсы
    item2 = types.InlineKeyboardButton(text='Поступление',
callback_data='postup') # кнопка поступление
```



```

    item3 = types.InlineKeyboardButton(text='Оплата обучения',
callback_data='pay_obuch') # кнопка оплата обучения
    item4 = types.InlineKeyboardButton(text='Договора',
callback_data='dogovora') # кнопка договора
    item5 = types.InlineKeyboardButton(text='Результаты и списки',
callback_data='res') # кнопка результаты и списки
    item6 = types.InlineKeyboardButton(text='Интересное в учебном процессе',
callback_data='podacha') # кнопка интересное
    item7 = types.InlineKeyboardButton(text='Не нашли нужный раздел',
callback_data='nenash') # кнопка не нашли нужный раздел
    item8 = types.InlineKeyboardButton(text='В начало',
callback_data='mainmenu') # главное меню
    kb.add(item1, item2, item3, item4, item5, item6, item7, item8) # добавляем
кнопки в клавиатуру
    bot.send_message(message.chat.id, "Выберите интересующий Вас раздел:",
reply_markup=kb) # отправляем сообщение, в reply_markup
указываем нашу клавиатуру

@bot.message_handler(commands=['calladmin']) # Обработка команды
/calladmin
def calladmin(message):
    kb = types.InlineKeyboardMarkup(row_width=1)
    item1 = types.InlineKeyboardButton(text='Назад',
callback_data='mainmenu') # добавим кнопку
возвращения на предыдущий экран
    kb.add(item1)
    bot.send_message(message.chat.id, ""
Чтобы связаться с Администратором жми <a
href="https://t.me/oxcxxxx1">СЮДА</a>", parse_mode='HTML',
reply_markup=kb)
    # Отправляем сообщение, контакт у нас в гиперссылке, также указываем
клавиатуру

# Теперь пишем обработчик для кнопок

@bot.callback_query_handler(func=lambda call: True)
def markup_answer(call):
    if call.data == 'mainmenu': # обрабатываем кнопку назад к приветствию
        bot.send_message(call.message.chat.id, ""
Привет! Здесь можно найти ответы на вопросы, которые могут возникнуть
при выборе нашего учреждения для обучения.
<a href="https://t.me/+K8-izZ36zl85OTM6">Перейти в канал</a>

```

А чтобы воспользоваться ботом - /razdel.

```
    "", parse_mode='HTML')
# по сути, просто вставляем код из обработчика команды /start

elif call.data == 'back_raz': # пишем обработчика кнопки назад в разделы
    kb = types.InlineKeyboardMarkup(row_width=1) # добавляем клавиатуру
    # создаем поочередно кнопки
    # в callback_data задаем название кнопки для дальнейшей обработки ее
    нажатия
    item1 = types.InlineKeyboardButton(text='Подготовительные курсы',
callback_data='podgotovkurs') # кнопка курсы
    item2 = types.InlineKeyboardButton(text='Поступление',
callback_data='postup') # кнопка поступление
    item3 = types.InlineKeyboardButton(text='Оплата обучения',
callback_data='pay_obuch') # кнопка оплата обучения
    item4 = types.InlineKeyboardButton(text='Договора',
callback_data='dogovora') # кнопка договора
    item5 = types.InlineKeyboardButton(text='Результаты и списки',
callback_data='res') # кнопка результаты и списки
    item6 = types.InlineKeyboardButton(text='Интересное в учебном процессе',
callback_data='podacha') # кнопка интересное
    item7 = types.InlineKeyboardButton(text='Не нашли нужный раздел',
callback_data='nenash') # кнопка не нашли нужный раздел
    item8 = types.InlineKeyboardButton(text='В начало',
callback_data='mainmenu') # главное меню
    kb.add(item1, item2, item3, item4, item5, item6, item7, item8) # добавляем
кнопки в клавиатуру
    bot.send_message(call.message.chat.id, "Выберите интересующий Вас
раздел:",
                    reply_markup=kb) # отправляем сообщение, в reply_markup
указываем нашу клавиатуру

elif call.data == 'podgotovkurs': # обрабатываем кнопку подготовительных
курсов
    kb = types.InlineKeyboardMarkup(row_width=1)
    item1 = types.InlineKeyboardButton(text='Назад', callback_data='back_raz')
    kb.add(item1)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Подготовительные
курсы\"</b>")
```

Для тех, кто собирается хорошо подготовиться к ОГЭ, и поступить на бюджет в престижный колледж, открыты Подготовительные курсы <a

```

href="https://naukograd.mfua.ru/kursy-podgotovka-k-oge-i-ege-v-
moskve/">Подробнее</a>
    "", parse_mode='HTML', reply_markup=kb)
    # Передали информацию, гиперссылку, задали кнопку назад

    elif call.data == 'postup': # обрабатываем кнопку поступление
        kb = types.InlineKeyboardMarkup(row_width=1)
        # в клавиатуру задаем кнопки меню нижнего уровня
        item1 = types.InlineKeyboardButton(text='Направления и программы',
callback_data='napravl')
        item2 = types.InlineKeyboardButton(text='Приемная комиссия',
callback_data='prikom')
        item3 = types.InlineKeyboardButton(text='План приема',
callback_data='control')
        item4 = types.InlineKeyboardButton(text='Вступительные испытания',
callback_data='uchet')
        item5 = types.InlineKeyboardButton(text='Вакантные места для перевода',
callback_data='obsche')
        item6 = types.InlineKeyboardButton(text='Назад', callback_data='back_raz')
        kb.add(item1, item2, item3, item4, item5, item6)
        bot.send_message(call.message.chat.id, "<b>Раздел \"Поступление\"</b>")

```

Выберите действие:

```

    "", reply_markup=kb, parse_mode='HTML')

    elif call.data == 'napravl': # обрабатываем кнопку направления
        kb = types.InlineKeyboardMarkup(row_width=2)
        item1 = types.InlineKeyboardButton(text='Назад',
callback_data='postup') # в callback_data указываем
название обработчика предыдущего меню
        item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
        kb.add(item1, item2)
        bot.send_message(call.message.chat.id, "<b>Раздел \"Направления и
программы\"</b>")

```

Проф. ориентация, выбор профессии — комплекс психолого-педагогических мер, направленный на профессиональное самоопределение абитуриента/студента. Колледж работает по направлениям: «Профессиональное просвещение» — обеспечение молодежи информацией о мире профессий, учебных заведениях, возможностях профессиональной карьеры; «Профессиональное воспитание» — формирование у молодежи трудолюбия, работоспособности, профессиональной ответственности, способностей и склонностей; «Профессиональное консультирование»-

вопросы выбора профессии, трудоустройства, возможностей получения профессиональной подготовки для абитуриентов можно найти тут
", parse_mode='HTML', reply_markup=kb)

```
elif call.data == 'prikom': # обрабатываем приемная комиссия
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                         callback_data='postup') # в callback_data указываем
название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Приемная
комиссия\"</b>")
```

Прием документов о поступлении осуществляется по адресу:
ЦЕНТРАЛЬНАЯ ПРИЕМНАЯ КОМИССИЯ
м. Калужская, ул. Введенского, д. 1а. подробнее
", parse_mode='HTML', reply_markup=kb)

```
elif call.data == 'control': # обрабатываем кнопку план приема
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                         callback_data='postup') # в callback_data указываем
название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"План приема\"</b>")
```

План приема на целевое обучение по образовательным программам высшего образования (программам бакалавриата, программам магистратуры, программам специалитета, программам ординатуры, программам ассистентуры-стажировки) за счет бюджетных ассигнований федерального бюджета: подробнее
>
", parse_mode='HTML', reply_markup=kb)

```
elif call.data == 'uchet': # обрабатываем кнопку вступительные испытания
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
```

```

        callback_data='postup') # в callback_data указываем
название обработчика предыдущего меню
        item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
        kb.add(item1, item2)
        bot.send_message(call.message.chat.id, "<b>Раздел \"Вступительные
испытания\"</b>")

```

Вступительные испытания проводятся, чтобы абитуриенты показали свои таланты, отличительные способности, после которых будет приниматься решение о принятии выпускника <a

href="https://www.mfua.ru/abitur/spo_sredn/#abitur_vstupitelnye-
ispytania">подробнее

```

        "", parse_mode='HTML', reply_markup=kb)

```

```

elif call.data == 'obsche': # обрабатываем кнопку вакантные места
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
        callback_data='postup') # в callback_data указываем
название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Вакантные места для
перевода\"</b>")

```

Информация о количестве вакантных мест для приема (перевода) обучающихся по каждой реализуемой образовательной программе за счет бюджетных ассигнований федерального бюджета и за счет средств физических и (или) юридических лиц подробнее

```

        "", parse_mode='HTML', reply_markup=kb)

```

```

elif call.data == 'pay_obuch': # обрабатываем кнопку оплаты обучения
    kb = types.InlineKeyboardMarkup(row_width=1)
    # в клавиатуру задаем кнопки меню нижнего уровня
    item1 = types.InlineKeyboardButton(text='Стоимость обучения',
        callback_data='stoimost') # кнопка стоимости
обучения
    item2 = types.InlineKeyboardButton(text='Стипендии',
callback_data='stipend') # кнопка стипендии
    item3 = types.InlineKeyboardButton(text='Порядок оказания услуг',
        callback_data='oplata') # кнопка Порядок оказания
услуг

```

```

        item4 = types.InlineKeyboardButton(text='Образовательный кредит',
callback_data='credit') # кнопка кредит
        item5 = types.InlineKeyboardButton(text='Назад', callback_data='back_raz')
        kb.add(item1, item2, item3, item4, item5)
        bot.send_message(call.message.chat.id, "<b>Раздел \"Оплата обучения\"</b>")

```

Выберите действие:

```

        "", reply_markup=kb, parse_mode='HTML')

```

```

elif call.data == 'stoimost': # обрабатываем кнопку стоимости обучения
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
callback_data='pay_obuch') # в callback_data
указываем название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Стоимость
обучения\"</b>")

```

Прейскурант стоимости образовательных услуг по основным профессиональным образовательным программам среднего профессионального образования - программам подготовки специалистов среднего звена на 2024/2025 учебный год подробнее

```

        "", parse_mode='HTML', reply_markup=kb)

```

```

elif call.data == 'stipend': # обрабатываем кнопку стипендии
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
callback_data='pay_obuch') # в callback_data
указываем название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Стипендии\"</b>")

```

Для студентов, обучающихся на коммерческой основе, также предусмотрены стипендии, учрежденные Попечительским советом МФЮА. Они назначаются студентам с момента поступления на платное отделение для следующих направлений: «Журналистика», «Юриспруденция», «Информатика и вычислительная техника», «Менеджмент», «Экономика», «Таможенное дело», «Правовое обеспечение национальной безопасности».

Первую стипендию первокурсникам начисляют за высокие баллы ЕГЭ. А уже после сдачи первой сессии студенты могут рассчитывать на стипендию в случае всех оценок «хорошо» — 2 тысячи рублей, «хорошо» и «отлично» — 3 тысячи рублей, все «отлично» — 4 тысячи рублей. Выплаты осуществляются каждый месяц.

```
        "", parse_mode='HTML',
reply_markup=kb)

elif call.data == 'oplata': # обрабатываем кнопку Порядок оказания услуг
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                         callback_data='pay_obuch') # в callback_data
указываем название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Порядок оказания услуг</b>")
```

Порядок оказания платных образовательных услуг по образовательным программам в МФЮА. подробнее

```
        "", parse_mode='HTML',
reply_markup=kb)
```

```
elif call.data == 'credit': # обрабатываем кнопку кредита
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                         callback_data='pay_obuch') # в callback_data
указываем название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Образовательный кредит\"</b>")
```

Во многих банках предоставляют кредит на образование. подробнее

```
        "", parse_mode='HTML',
reply_markup=kb)
```

```
elif call.data == 'dogovora': # обрабатываем кнопку договора
```

```

kb = types.InlineKeyboardMarkup(row_width=1)
# в клавиатуру задаем кнопки меню нижнего уровня
item1 = types.InlineKeyboardButton(text='Заключение договоров',
                                   callback_data='zakl') # кнопка заключения договора
item2 = types.InlineKeyboardButton(text='Образцы договоров',
                                   callback_data='obraz') # кнопка образцы документов
item3 = types.InlineKeyboardButton(text='Назад', callback_data='back_raz')
# кнопка назад
kb.add(item1, item2, item3)
bot.send_message(call.message.chat.id, "<b>Раздел \"Договора\"</b>")

```

Выберите действие:

```

", reply_markup=kb, parse_mode='HTML')

```

```

elif call.data == 'zakl': # обрабатываем кнопку заключение договоров
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                       callback_data='dogovora') # в callback_data
указываем название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
                                       callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Заключение
договоров\"</b>")

```

Для заключения договора необходимы документы подробнее

```

", parse_mode='HTML', reply_markup=kb)

```

```

elif call.data == 'obraz': # обрабатываем кнопку образцов документов
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                       callback_data='dogovora') # в callback_data
указываем название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
                                       callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Образцы
договоров\"</b>")

```

Договора об оказании платных образовательных услуг по программам среднего профессионального образования заключаются между: колледж - обучающийся; колледж – (родитель + обучающийся); колледж –(юр. лицо +

обучающийся). Формы договоров: подробнее
", parse_mode='HTML', reply_markup=kb)

```
elif call.data == 'res': # обрабатываем кнопку результаты и списки
    kb = types.InlineKeyboardMarkup(row_width=1)
    # в клавиатуру задаем кнопки меню нижнего уровня
    item1 = types.InlineKeyboardButton(text='Список зачисленных',
                                         callback_data='spispostup') # кнопка по фамильного
списка
    item2 = types.InlineKeyboardButton(text='Условия поступления и
зачисления',
                                         callback_data='yslovpostup') # кнопка Условия
поступления и зачисления
    item3 = types.InlineKeyboardButton(text='Назад', callback_data='back_raz')
# кнопка назад
    kb.add(item1, item2, item3 )
    bot.send_message(call.message.chat.id, "<b>Раздел \"Результаты и
списки\"</b>")
```

Выберите действие:

", reply_markup=kb, parse_mode='HTML')

```
elif call.data == 'spispostup': # обрабатываем кнопку списка зачисленных
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                         callback_data='res') # в callback_data указываем
название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
                                         callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Список
зачисленных\"</b>")
```

Список лиц, зачисленных на обучение по программе подготовке специалистов среднего звена на места за счет бюджетных ассигнований федерального бюджета подробнее
", parse_mode='HTML', reply_markup=kb)

```
elif call.data == 'yslovpostup': # обрабатываем кнопку Условия поступления
и зачисления
```

```

kb = types.InlineKeyboardMarkup(row_width=2)
item1 = types.InlineKeyboardButton(text='Назад',
                                     callback_data='res') # в callback_data указываем
название обработчика предыдущего меню
item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
kb.add(item1, item2)
bot.send_message(call.message.chat.id, "<b>Раздел \"Условия поступления
и зачисления\"</b>")

```

Условия поступления и зачисления на обучения ежегодно формируют сами абитуриенты в рамках приёмной кампании текущего года. подробнее

```

", parse_mode='HTML', reply_markup=kb)

```

```

elif call.data == 'podacha': # обработка кнопки подача документов и
вступительные экзамены

```

```

kb = types.InlineKeyboardMarkup(row_width=1)
# в клавиатуру задаем кнопки меню нижнего уровня
item1 = types.InlineKeyboardButton(text='День открытых дверей',
callback_data='docpod') # кнопка день открытых дверей
item2 = types.InlineKeyboardButton(text='Олимпиады и конкурсы',
callback_data='mesta') # кнопка олимпиады и конкурсы
item3 = types.InlineKeyboardButton(text='Новости', callback_data='news') #
кнопка успешная социализация
item4 = types.InlineKeyboardButton(text='Назад', callback_data='back_raz')
# кнопка назад
kb.add(item1, item2, item3, item4 )
bot.send_message(call.message.chat.id, "<b>Раздел \"Интересное в учебном
процессе\"</b>")

```

Выберите действие:

```

", reply_markup=kb, parse_mode='HTML')

```

```

elif call.data == 'docpod': # обрабатываем кнопку день открытых дверей
kb = types.InlineKeyboardMarkup(row_width=2)
item1 = types.InlineKeyboardButton(text='Назад',
                                     callback_data='podacha') # в callback_data указываем
название обработчика предыдущего меню
item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
kb.add(item1, item2)
bot.send_message(call.message.chat.id, "<b>Раздел \"День открытых

```

дверей"

Почему выбирают МФЮА: поступление без вступительных испытаний и экзаменов, высококвалифицированные преподаватели, только актуальные знания, прохождение практик под руководством профессионалов в ведущих компаниях, поступление в МФЮА без ЕГЭ. Видео материал о днях открытых дверей подробнее

```
", parse_mode='HTML', reply_markup=kb)
```

```
elif call.data == 'mesta': # обрабатываем кнопку олимпиады и конкурсы
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                         callback_data='podacha') # в callback_data указываем
название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Олимпиады и
конкурсы\"</b>")
```

В МФЮА проводятся различные олимпиады и конкурсы по результатам которых участники получают сертификаты. Такой сертификат станет отличным дополнением вашего портфолио. Участвовать могут и студенты, и абитуриенты подробнее

```
", parse_mode='HTML', reply_markup=kb)
```

```
elif call.data == 'news': # обрабатываем кнопку новости
    kb = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton(text='Назад',
                                         callback_data='podacha') # в callback_data указываем
название обработчика предыдущего меню
    item2 = types.InlineKeyboardButton(text='К разделам',
callback_data='back_raz')
    kb.add(item1, item2)
    bot.send_message(call.message.chat.id, "<b>Раздел \"Новости\"</b>")
```

Жизнь колледжа проходит очень насыщенно, постоянно проводятся различные мероприятия, такие как международные форумы, тренинги, праздники отмечаются концертами, проходят спортивные соревнования, дни здоровья и многое другое. Обо всем этом можно узнать из новостей в группе колледжа подробнее

```
", parse_mode='HTML', reply_markup=kb)
```

```
elif call.data == 'nenash': # обрабатываем кнопку не нашли раздел
    kb = types.InlineKeyboardMarkup(row_width=1)
    item1 = types.InlineKeyboardButton(text='Назад', callback_data='back_raz')
    kb.add(item1)
    bot.send_message(call.message.chat.id, "Контакт центр учреждения +7
(499) 348-83-62", reply_markup=kb)

bot.polling()

if __name__ == "__main__":
    try:
        bot.polling(none_stop=True)
    except:
        print('bolt')
        time.sleep(1)
```

Данная работа выполнена мною самостоятельно

подпись

« ____ » _____ 20 ____ г.

дата сдачи работы