**Введение**

Данный проект посвящен разработке программного обеспечения в форме Telegram-бота "ProffIT", предназначенного для решения задач профессиональной ориентации и развития компетенций в сфере информационных технологий. Актуальность разработки обусловлена стремительным развитием IT-индустрии и возрастающей потребностью в квалифицированных специалистах, а также сложностью самостоятельного выбора направления развития среди многообразия IT-профессий.

Программный продукт представляет собой интеллектуальную систему, которая комплексно подходит к вопросам профориентации и профессионального роста. Бот не только помогает определить наиболее подходящее IT-направление на основе многофакторного анализа, но и обеспечивает непрерывное развитие через систему персонализированных рекомендаций, практических заданий и подготовки к собеседованиям. Интеграция с образовательными платформами и адаптивный подход к обучению делают систему эффективным инструментом для начинающих и практикующих IT-специалистов.

Далее приведем краткое описание разделов пояснительной записки.

В разделе, посвященном анализу задачи, рассматривается постановка проблемы, исследуется предметная область и определяются функциональные требования к системе. Здесь же представлена диаграмма вариантов использования, иллюстрирующая взаимодействие между пользователями, администратором и ботом. Описываются выбранные инструменты разработки, включая язык Python, библиотеку pyTelegramBotAPI и СУБД SQLite, а также обосновывается план работы и модель жизненного цикла проекта.

Раздел проектирования задачи фокусируется на архитектурных и интерфейсных решениях. В нем представлена диаграмма последовательности, детализирующая сценарий прохождения теста, а также описаны принципы построения системы меню и UX-проектирования пользовательского интерфейса с учетом специфики платформы Telegram.

Далее следует руководство программиста, содержащее техническую документацию по реализации системы: описание алгоритмов, структуры базы данных, ключевых компонентов и точек расширения функционала.

Раздел тестирования описывает стратегию и результаты проверки работоспособности бота, включая тестирование команд, пользовательских сценариев и административных функций.

Заключительная часть документации представляет собой руководство пользователя, которое включает общие сведения о программном продукте, инструкции по установке для системного администратора, а также подробное описание работы с ботом: от запуска и использования команд до завершения работы и обращения к справочной системе.

Список использованных источников систематизирует библиографическую базу проекта. В нем приведен перечень нормативных документов, учебной и технической литературы, а также электронных ресурсов, включая официальную документацию по языку Python, СУБД SQLite и Telegram API, к которым осуществлялось обращение в процессе проектирования и разработки программного продукта.

В раздел «Приложение» вынесены вспомогательные материалы справочного характера, которые дополняют основной текст пояснительной записки. В частности, здесь представлен листинг программного кода, демонстрирующий техническую реализацию описанных алгоритмов, а также диаграмма вариантов использования и разработанные тест-кейсы.

Данный проект направлен на создание не просто функционального Telegram-бота, а комплексной системы профессионального развития, где каждый элемент тщательно проработан с точки зрения пользовательского опыта и образовательной эффективности. Разработка и интеграция всех компонентов - от интеллектуального тестирования до системы ежедневных заданий - позволяет создать по-настоящему персонализированный подход к карьерному росту в IT-сфере.

**1 Анализ задачи**

**1.1 Постановка задачи**

**1. Организационно-экономическая сущность задачи**

Наименование задачи: разработка Telegram-бота «ProffIT».

Цель разработки: создание интеллектуальной системы профориентации и непрерывного развития в IT-направлениях, предоставляющей пользователям комплекс услуг: определение наиболее подходящих IT-специализаций на основе тестирования, персонализированные рекомендации по развитию профессиональных навыков, практику через микро-задания, подготовку к собеседованиям в IT-компании и доступ к актуальным образовательным ресурсам.

Назначение: программный продукт предназначен для широкого круга пользователей, включая студентов технических и нетехнических специальностей, выпускников вузов, определяющихся с IT-специализацией, специалистов, желающих сменить профессию на IT, начинающих IT-специалистов для углубления знаний, а также карьерных консультантов и HR-специалистов. Среди подразделений-пользователей выделяются учебные заведения (кафедры, центры карьеры), IT-компании (отделы обучения и развития) и карьерные центры с хедхантинговыми агентствами.

Периодичность использования: по мере необходимости.

Источники и способы получения данных:обрыазовательные платформы Skillbox, Stepik, Udemy, IT-Academy, базы вакансий и требований работодателей, а также актуальные исследования рынка IT-профессий, видеоплатформа YouTube. Внутренние источники составляют результаты пользовательского тестирования, статистика выполнения заданий и обратная связь от пользователей.

Обзор существующих аналогичных ПП: аналогичные решения включают "Профессию в IT" с ограниченным функционалом тестирования и отсутствием заданий, "IT Mentor" с фокусом на менторстве без системного подхода к развитию, "Career Hub" как общий карьерный консультант без IT-специализации и "Codecademy" с практическими заданиями без профориентационной составляющей. Конкурентные преимущества разрабатываемого продукта заключаются в комплексном подходе, объединяющем тестирование, развитие и подготовку, персонализации на основе обучения, интеграции практики и фокусе исключительно на IT-сфере.

**2. Функциональные требования**

Бот проводит тестирование, состоящее из 25 вопросов, которое анализирует интересы, навыки и личностные качества пользователя. На основе результатов тестирования система рассчитывает весовые коэффициенты для различных IT-направлений и формирует персональный профиль с рекомендациями по выбору специализации.

Бот предоставляет задания для выполнения. Сложность заданий адаптируется в зависимости от прогресса пользователя.

Для подготовки к трудоустройству бот предлагает доступ к базе вопросов по различным IT-специализациям. Пользователи могут решать практические задачи, проходить симуляцию технических собеседований.

Бот обеспечивает доступ к актуальным образовательным ресурсам.

В боте присутствует административная панель, с помощью которой можно управлять данными бота, тестами, вопросами и курсами.

Завершает функционал система аналитики и отчетности, которая визуализирует прогресс обучения, анализирует сильные и слабые стороны пользователя, предоставляет рекомендации по коррекции плана развития и сравнивает навыки пользователя с текущими рыночными требованиями. Обеспечена возможность экспорта данных для использования в резюме и портфолио.

Таким образом, функционал бота выстраивает целостную траекторию профессионального развития пользователя в IT: от диагностики потенциала и выбора специализации через практику к подготовке к трудоустройству и доступу к образовательным ресурсам.

**3. Описание исходной (входной) информации**

Перечень исходной информации: данные пользователя включают идентификационные данные (Telegram ID, имя) и результаты профориентационного тестирования. Контентные данные состоят из базы вопросов тестирования с весовыми коэффициентами, библиотеки учебных материалов и заданий, базы вопросов для собеседований, каталога образовательных курсов.

Формы представления: текстовые сообщения.

Перечень пользователей исходной информации: пользователи бота предоставляют идентификационные данные и ответы на тесты, администратор системы наполняет базы вопросов и материалов, обновляет образовательный контент.

**4. Описание результатной (выходной) информации**

Перечень результатной информации: включают результаты профориентационного тестирования, учебные задания, подборки вопросов для собеседований и рекомендации образовательных курсов. Агрегированные данные содержат аналитику популярности направлений.

Формы представления: текстовые сообщения и ссылки на курсы.

Периодичность и сроки представления: мгновенно предоставляются результаты тестирования и ответы на запросы, по надобности выдаются материалы для собеседований, задания и курсы.

Перечень пользователей результатной информации: конечные пользователи получают персональные рекомендации и задания, администратор образовательных программ анализирует эффективность.

**5. Описание используемой условно-постоянной информации**

Перечень условно-постоянной информации: классификаторы и справочники включают справочник IT-специализаций с атрибутами, классификатор навыков и компетенций, справочник технологий и инструментов, уровни сложности заданий (Junior/Middle/Senior) и типы учебных материалов и ресурсов. Нормативные данные содержат весовые коэффициенты вопросов тестирования, матрицы соответствия навыков и специализаций. Контентные базы включают базу вопросов профориентационного тестирования, библиотеку учебных заданий по специализациям, коллекцию вопросов для технических собеседований, каталог образовательных курсов и ресурсов.

Формы представления: вопросы теста (текстовые сообщения и изображения, кнопки).

**6. Нефункциональные (эксплуатационные) требования**

Требования к применению: качество пользовательского интерфейса обеспечивается интуитивно понятным интерфейсом на основе кнопок и меню, минимальным количеством шагов для выполнения ключевых операций, единым стилем оформления всех элементов бота, поддержкой как текстовых команд, так и интерактивных элементов и адаптацией под мобильные устройства различных диагоналей.

Требования к производительности: эффективность использования ресурсов обеспечивается временем отклика на команды пользователя не более 2 секунд, обработкой до 100 одновременных пользователей, оптимизацией использования памяти и процессорного времени, эффективным кэшированием часто используемых данных и минимизацией трафика за счет компактных сообщений. Пропускная способность системы включает обработку до 10 000 сообщений в час, оптимальное использование API Telegram без превышения лимитов и балансировку нагрузки в пиковые часы активности. Время реакции системы предусматривает мгновенную обработку простых запросов (менее 1 секунды), обработку сложных расчетов (тестирование, аналитика) до 5 секунд.

Требования к реализации: технические стандарты включают язык программирования Python 3.14, основную библиотеку python-telegram-bot, базу данных SQLite 3 и систему контроля версий Git. Операционная среда состоит из операционной системы Windows 11, веб-сервера Render, среды выполнения PyCharm 2025.2.1. Архитектурные требования включают модульную архитектуру с четким разделением ответственности, возможность горизонтального масштабирования, использование асинхронного программирования, разделение на микросервисы при необходимости и API-first подход для будущих интеграций.

Требования к надёжности: допустимая частота сбоев включает общую доступность системы 95% в течение месяца, максимальное время простоя не более 36 часов в месяц, коэффициент готовности не менее 0.95 и допустимое количество критических ошибок не более 3 в месяц. Воздействие сбоев минимизируется через сохранение целостности данных пользователей при сбоях, минимизацию потери прогресса выполнения заданий, автоматическое восстановление работоспособности после перезапуска сервиса и резервное копирование данных не реже 1 раза в сутки. Возможности восстановления включают автоматический перезапуск приложения при сбоях, восстановление пользовательских сессий из последних доступных данных, мониторинг состояния системы и уведомления администратора о критических сбоях.

Требования к интерфейсу: внешние сущности и взаимодействие включают пользователей Telegram как основной интерфейс через мессенджер, административную панель как веб-интерфейс для управления контентом, внешние API для интеграции с образовательными платформами и систему мониторинга для отслеживания производительности и ошибок. Регламент взаимодействия предусматривает авторизацию пользователей через Telegram ID, разграничение прав доступа (пользователь/администратор), шифрование передаваемых данных и соблюдение политики конфиденциальности Telegram. Интерфейс пользователя обеспечивает единообразие элементов управления во всех разделах, адаптацию под различные версии Telegram.

**1.2 Диаграмма вариантов использования**

Диаграмма вариантов использования системы «ProffIT» включает трех основных актёров: пользователя, администратора и Telegram-бота как автономного актёра системы. Пользователь взаимодействует с ботом через платформу Telegram, получая доступ к профориентационному тестированию, персональным рекомендациям по развитию, адаптивным микро-заданиям, примерам вопросов с собеседований и ссылкам на образовательные курсы. Администратор системы управляет контентом через специализированный интерфейс, осуществляя добавление, редактирование и удаление тестовых вопросов, рекомендаций, заданий и учебных материалов, а также получая аналитику использования системы. Telegram-бот выступает как активный участник взаимодействия, проводя тестирование, вычисляя результаты, выдавая персональные рекомендации на основе ответов пользователя, отправляет адаптивные задания и предоставляет соответствующие образовательные ресурсы. Диаграмма наглядно демонстрирует комплексный подход системы к профессиональному развитию пользователей, где бот обеспечивает автоматизацию персонального подхода, а администратор поддерживает актуальность и качество контента.

Разработанную диаграмму вариантов использования можно увидеть в приложении А, на рисунке А.1.

**1.3 Инструменты разработки**

Для реализации проекта Telegram-бот «ProffIT» будет выбран язык Python и среда разработки PyCharm – это интегрированная среда разработки (IDE), разработанная компанией JetBrains для языка программирования Python.

UX дизайн будет разработан в Figma. Интерактивные элементы интерфейса, такие как кнопки и меню, создаются непосредственно в среде Telegram с использованием Telegram Bot API и не требуют отдельной визуальной разработки.

Иные инструменты, используемые при разработке и написании сопутствующей документации:

* + WEB-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм;
  + Microsoft Office Word – для написания документации к программному продукту;
  + Microsoft Office Excel – для создания диаграммы Ганта.
  + PythonAnywhere – платформа будет использоваться для развёртывания бота.

Разработка будет производиться на ноутбуке Asus с характеристиками:

* процессор: AMD Ryzen 7;
* ОЗУ: 16 GB RAM;
* графика: NVIDIA GeForce RTX 4050;
* ОС: Windows 11;
* разрешение экрана: 1920 × 1080.

**1.4 Разработка плана работы над проектом**

Для разработки плана над проектом было использовано приложение Microsoft Оffice Excel.

В данном программном обеспечении можно проанализировать объём работ. Показывает в какие сроки может быть выполнена та или иная задача.

В данном случае Microsoft Оffice Excel позволил нам быстро составить план разработки проекта, с возможностью отмечать степень его готовности. Таким образом это должно увеличить скорость разработки. План разработки представлен на рисунке 1.1:

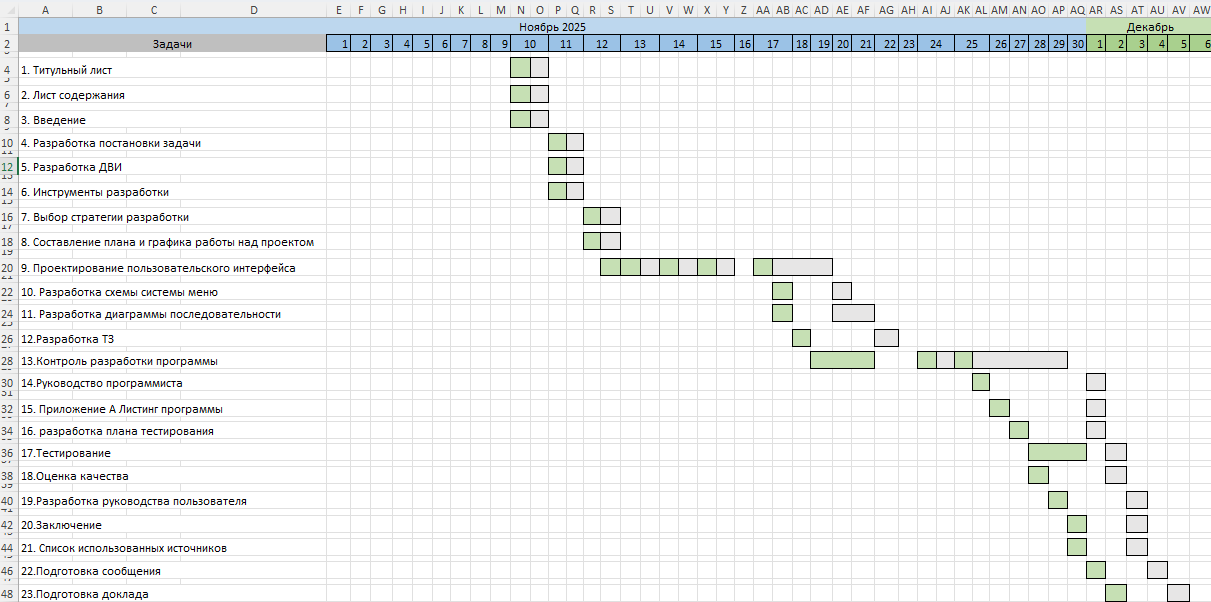


Рисунок 1.1 – Диаграмма Ганта

**1.5 Выбор стратегии разработки и модели жизненного цикла**

Для разработки Telegram-бот «ProffIT» следует выбрать стратегию разработки и модель жизненного цикла. Осуществляем выбор посредством составления таблиц:

Таблица 1.1 – Выбор модели жизненного цикла на основе характеристик требований.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | **Да** | **Да** | **Да** | Нет | Нет | Нет |

Продолжение таблицы 1.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | **Да** | **Да** | **Да** | **Да** | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | **Нет** | **Нет** | **Нет** | **Нет** | Да | Да |
| 4. | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | **Да** | Нет | **Да** | **Да** |
| 5. | Требуется ли проверка концепции программного средства или системы? | **Нет** | **Нет** | Да | **Нет** | Да | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы  (программного средства) в ЖЦ? | Нет | Нет | Нет | **Да** | **Да** | **Да** |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | **Да** | **Да** | **Да** | **Да** |
| **Итоги** | | **4** | **4** | **5** | **5** | **3** | **3** |

Итог: на основе результатов заполнения табл. 1.1 подходящей является RAD модель и инкрементная модель.

Таблица 1.2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | **Да** | **Да** |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | **Да** | **Да** | Нет | Нет | Нет | **Да** |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | **Нет** | **Нет** | **Нет** | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | **Да** | **Да** | Нет | **Да** | Нет | Нет |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | **Да** | **Да** | **Да** | **Да** | Нет | Нет |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | **Да** | **Да** | Нет | **Да** | **Да** | **Да** |
| **Итоги** | | **5** | **5** | **2** | **3** | **2** | **3** |

Итог: на основе результатов заполнения табл. 1.2 подходящими являются каскадная и V-образная модели.

Таблица 1.3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | **Нет** | Да | **Нет** | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | **Да** | **Да** | **Да** |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | **Нет** | **Нет** | Да | **Нет** | Да | **Нет** |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | **Нет** | **Нет** | **Нет** | **Нет** | Да | Да |
| **Итоги** | | **2** | **2** | **2** | **3** | **2** | **2** |

Итог: на основе результатов заполнения табл. 1.3 подходящей является инкрементная модель.

Таблица 1.4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. | Разрабатывается ли в проекте  продукт нового для организации направления? | Нет | Нет | Нет | **Да** | **Да** | **Да** |
| 2. | Будет ли проект являться  расширением существующей  системы? | Да | Да | Да | Да | **Нет** | **Нет** |
| 3. | Будет ли проект крупно- или  среднемасштабным? | **Нет** | **Нет** | **Нет** | Да | Да | Да |

Продолжение таблицы 1.4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4. | Ожидается ли длительная  эксплуатация продукта? | **Да** | **Да** | Нет | **Да** | Нет | **Да** |
| 5. | Необходим ли высокий уровень  надежности продукта проекта? | Нет | **Да** | Нет | **Да** | Нет | **Да** |
| 6. | Предполагается ли эволюция  продукта проекта в течение ЖЦ? | Нет | Нет | Нет | **Да** | **Да** | **Да** |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | **Да** | **Да** | **Да** |
| 8. | Является ли график сжатым? | Нет | Нет | **Да** | **Да** | **Да** | **Да** |
| 9. | Предполагается ли повторное  использование компонентов? | Нет | Нет | **Да** | **Да** | **Да** | **Да** |
| 10. | Являются ли достаточными ресурсы (время, деньги,  инструменты, персонал)? | Нет | Нет | Нет | Нет | **Да** | **Да** |
| **Итоги** | | **2** | **3** | **3** | **7** | **7** | **9** |

Итог: на основе результатов заполнения табл. 1.4 подходящей является эволюционная модели.

Таблица 1.5 – Выбор итоговой модели жизненного цикла.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| **Итоги** | **13** | **14** | **12** | **18** | **14** | **17** |

Общий итог: в итоге заполнения табл. 1.1 – 1.4 наиболее подходящей является **инкрементная** модель, но так как возможно в будущем я захочу добавить какие-нибудь новые функции в проект, то я выберу **эволюционную**.

**2 Проектирование задачи**

**2.1 Диаграмма последовательности**

Диаграмма последовательности системы «ProffIT» иллюстрирует автоматизированный процесс взаимодействия пользователя с Telegram-ботом при выполнении профориентационного тестирования. На диаграмме отражены три ключевых участника: пользователь, сам бот и серверная часть системы, включающая базу данных. Процесс начинается с инициации теста через команду /start и проходит через этапы загрузки вопросов, выбора ответов, расчёта результатов и получения рекомендаций. Взаимодействие строится таким образом, что бот выступает активным посредником: он запрашивает данные у системы, отображает контент пользователю, передаёт его ответы для обработки и в итоге выдаёт персонализированный результат. Диаграмма наглядно показывает, как система обеспечивает сквозную автоматизацию тестирования — от первого сообщения до выдачи индивидуальных рекомендаций, с чётким разделением ролей между интерфейсом бота и внутренней логикой обработки данных. Диаграмму можно увидеть на рисунке 2.1:

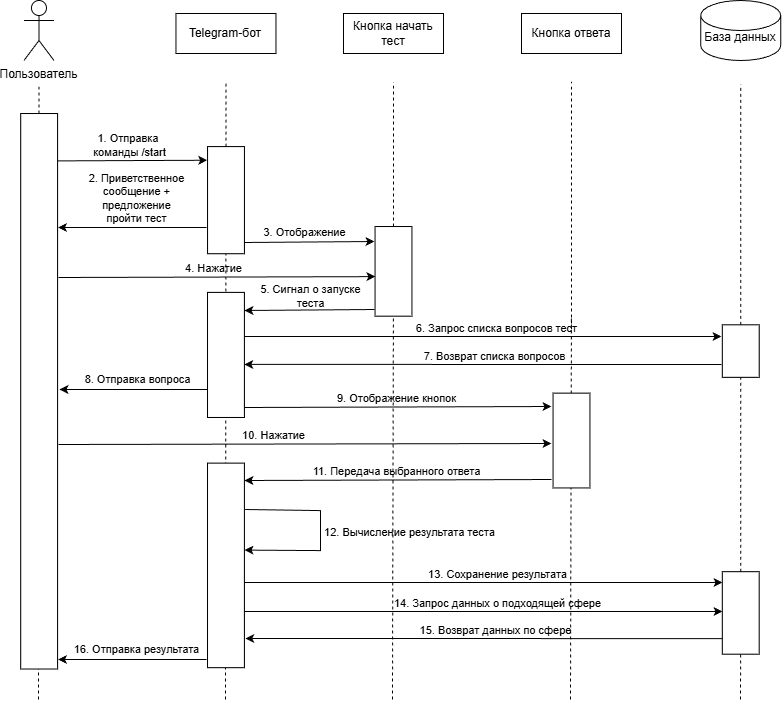


Рисунок 2.1 – Диаграмма последовательности

**2.2 Проектирование системы меню для чат-бота**

Главное меню пользователя сформировано для обеспечения простого и интуитивного доступа ко всем ключевым функциям бота. Оно включает четыре основных раздела. Диаграмму можно увидеть на рисунке 2.2:



Рисунок 2.2 – Структура системы меню

Административная панель предназначена для управления всеми аспектами контента и настройки бота. Она организована в виде иерархического меню с основными разделами, каждый из которых содержит вложенные опции. Такая структура позволит администратору гибко обновлять и настраивать все элементы системы, обеспечивая её актуальность и персонализацию для конечных пользователей. Диаграмма представлена на рисунке 2.3:



Рисунок 2.3 – Структура админ-панели

**2.3 UX-проектирование пользовательского интерфейса**

В проекте Telegram-бота «ProffIT» UX (User Experience) отвечает за проектирование и организацию логики взаимодействия между пользователем и системой. Он определяет, как пользователь будет двигаться по сценарию: от начала теста к результатам, затем к рекомендациям, заданиям и курсам. UX продумывает последовательность шагов, структуру меню, расположение кнопок, текст сообщений и реакции бота на действия пользователя. Именно UX делает работу бота интуитивной и предсказуемой.[7, с. 29]

Ниже можно наблюдать изображения прототипов:

1. На рисунке 2.4 показано приветствие бота и сообщение, которое видит пользователь после команды «/start».
2. Рисунок 2.5 иллюстрирует процесс начала теста, отображения его результатов и предложения получить вопросы.
3. Примеры вопросов, интерфейс для предложения и получения заданий, а также ссылка на бесплатный курс представлены на рисунке 2.6.
4. Раздел с платным курсом, который открывается при выборе кнопки «да» и вспомогательные команды, можно увидеть на рисунке 2.7.
5. Рисунок 2.8 демонстрирует основное меню с командами, а также ответ бота на неизвестную команду.
6. Админ-панель и система управления меню показаны на рисунке 2.9.

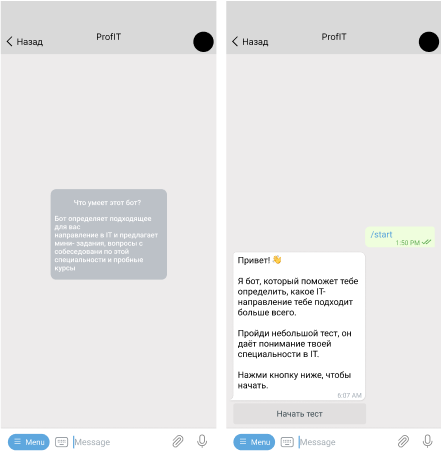


Рисунок 2.4 – Приветствие бота, сообщение после команды

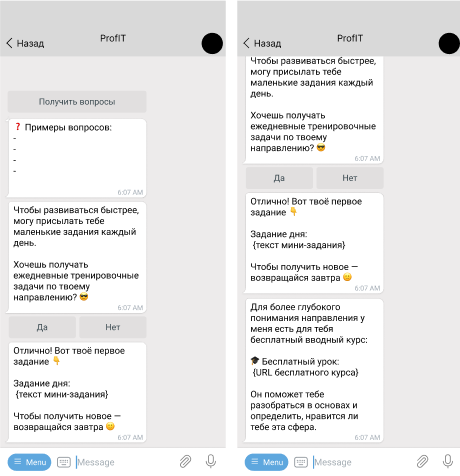


Рисунок 2.6 – Примеры вопросов, предложение и получение заданий, ссылка на бесплатный курс

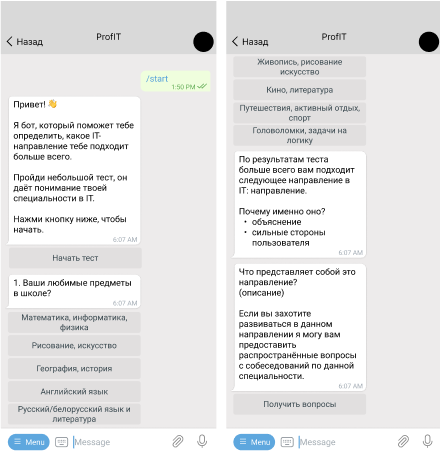


Рисунок 2.5 – Начало теста, результаты, предложение получить вопросы

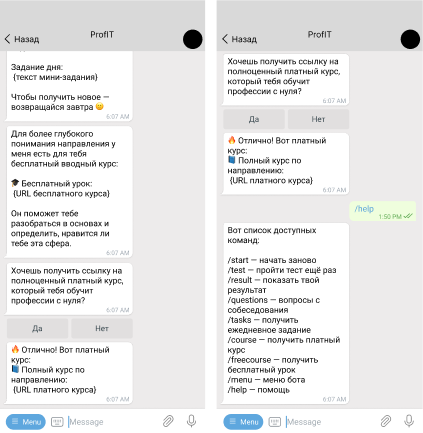


Рисунок 2.7 – Платный курс, при выборе кнопки «да» и команды

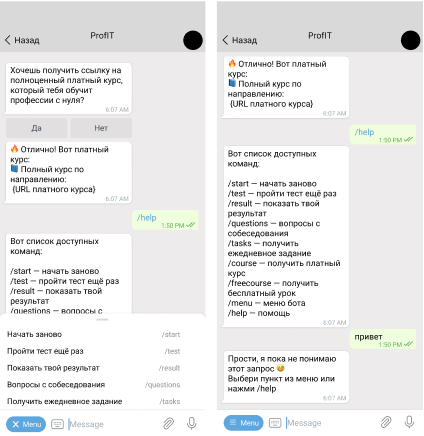


Рисунок 2.8 – Меню с командами, ответ при неизвестной команде

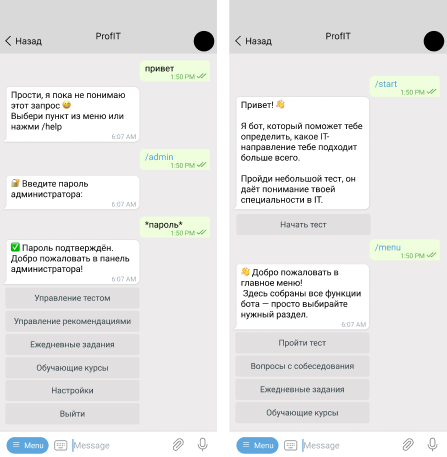


Рисунок 2.9 – Админ-панель и система меню

**3. Реализация**

**3.1 Руководство программиста**

Телеграм-бот «ProffIT» представляет собой программный комплекс для профессиональной ориентации в IT-сфере, реализованный на языке Python с использованием библиотеки pyTelegramBotAPI. Система построена по клиент-серверной архитектуре, где серверная часть выполняет функции обработки запросов, управления данными и бизнес-логикой, а клиентская часть реализована в рамках интерфейса мессенджера Telegram. Для взаимодействия с Telegram API используются основные компоненты библиотеки pyTelegramBotAPI: класс TeleBot для управления ботом, types.InlineKeyboardMarkup и types.InlineKeyboardButton для создания интерактивных клавиатур, декораторы @bot.message\_handler и @bot.callback\_query\_handler для обработки команд и callback-запросов, а также методы bot.send\_message(), bot.send\_photo() и bot.edit\_message\_text() для отправки и редактирования сообщений.[3],[4]

Для хранения данных используется встроенная база данных SQLite, обеспечивающая сохранность пользовательских результатов, контента тестов и административных настроек. База данных включает шесть основных таблиц: users (данные пользователей и прогресс теста), test\_questions (вопросы теста с вариантами ответов и весами), directions (IT-направления с описаниями и ссылками на курсы), interview\_questions (вопросы для собеседований), daily\_tasks (ежедневные задания) и bot\_texts (настраиваемые текстовые шаблоны). Все таблицы связаны через внешние ключи, а для хранения структурированных данных (массивов вариантов ответов и весов) используются поля типа JSON.[5]

Ключевой алгоритм системы — алгоритм определения профессиональной предрасположенности — основан на взвешенной сумме баллов, где каждый ответ пользователя в тесте увеличивает счет определенных IT-направлений согласно предустановленным коэффициентам, после чего выбирается направление с максимальным количеством баллов.

Проект включает два основных модуля: пользовательский и административный. Пользовательский модуль обеспечивает прохождение профориентационного теста, получение персонализированных результатов, доступ к вопросам для собеседований, ежедневным заданиям и обучающим курсам. Административный модуль реализован как многоуровневая система управления контентом с использованием конечного автомата (FSM) для обработки многошаговых операций, таких как добавление вопросов, редактирование направлений и обновление учебных материалов. Оба модуля используют единую базу данных, структура которой включает таблицы пользователей, вопросов теста, направлений, вопросов для собеседований, ежедневных заданий и настраиваемых текстов бота.

Система реализована в виде одного исполняемого файла main.py, который содержит инициализацию базы данных, регистрацию обработчиков команд и callback-запросов, а также бизнес-логику взаимодействия с пользователем. Для реализации функционала периодических действий (ежедневная рассылка) в систему интегрирован модуль планировщика задач. Его работа обеспечивается параллельным потоком исполнения (threading), что позволяет выполнять автоматические действия по расписанию независимо от основного цикла обработки сообщений от пользователей (polling). Для обеспечения отказоустойчивости реализован механизм повторных попыток при сетевых сбоях и глобальный перехват исключений с автоматическим перезапуском. Развертывание системы требует установки Python 3.8+, библиотек pyTelegramBotAPI и requests, а также настройки переменной окружения с токеном бота. Документация обеспечивает возможность модификации кода новыми разработчиками за счет четкого разделения ответственности между функциями и модулями.

**3.1.1** **Структура и описание процедур и функций пользователя**

Основные пользовательские функции проекта сгруппированы по функциональным модулям и представлены в таблице 3.1.

Таблица 3.1 – Основные функции и процедуры

|  |  |  |
| --- | --- | --- |
| Имя функции | Модуль | Назначение |
| 1 | 2 | 3 |
| init\_db() | База данных | Инициализация БД, создание таблиц, заполнение начальными данными |
| get\_user(user\_id) | База данных | Получение информации о пользователе по ID |
| create\_or\_update\_user() | База данных | Создание или обновление записи пользователя |
| get\_test\_questions() | База данных | Получение всех вопросов теста с порядком сортировки |
| calculate\_result(answers, questions) | Тестирование | Алгоритм определения направления по ответам |
| start\_test(chat\_id, user, message\_id) | Тестирование | Начало тестирования для пользователя |
| send\_question(chat\_id, user\_id, question\_index, message\_id) | Тестирование | Отправка вопроса с вариантами ответов |
| process\_answer(chat\_id, user\_id, answer\_index, message\_id) | Тестирование | Обработка ответа и переход к следующему вопросу |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| send\_daily\_tasks\_to\_all() | Рассылка | Выполняет выборку пользователей с определенным направлением и инициирует массовую отправку заданий. |
| show\_test\_result(chat\_id, user) | Результаты | Отображение итогов теста с рекомендацией |
| show\_interview\_questions(chat\_id, direction, user\_id) | Результаты | Показ вопросов для собеседования по направлению |
| schedule\_checker() | Планировщик | Бесконечный цикл, работающий в фоновом потоке, проверяющий наступление времени для выполнения отложенных задач. |
| force\_tasks\_message(message) | Администрирование | Обработчик скрытой команды для принудительного (ручного) запуска ежедневной рассылки администратором. |
| offer\_daily\_tasks\_after\_test(chat\_id, user, user\_id) | Результаты | Предложение тренировочных заданий |
| send\_daily\_task(chat\_id, direction, user\_id) | Результаты | Генерация и отправка задания |
| show\_free\_course\_after\_test(chat\_id, direction, user\_id) | Результаты | Рекомендация бесплатного обучающего курса |
| show\_paid\_course(chat\_id, direction) | Результаты | Предложение платного курса по профессии |
| show\_admin\_panel(chat\_id) | Администрирование | Отображение главной панели администратора |
| show\_test\_management\_menu(chat\_id) | Администрирование | Меню управления вопросами теста |
| show\_recommendations\_management\_menu(chat\_id) | Администрирование | Меню управления рекомендациями |
| show\_tasks\_management\_menu(chat\_id) | Администрирование | Меню управления ежедневными заданиями |
| show\_courses\_management\_menu(chat\_id) | Администрирование | Меню управления обучающими курсами |
| show\_settings\_menu(chat\_id) | Администрирование | Меню настройки текстов бота |
| handle\_admin\_callback(callback) | Администрирование | Обработчик callback-запросов администратора |
| handle\_admin\_states(message) | Администрирование | Обработчик состояний администратора |
| get\_bot\_text(text\_type) | Настройки | Получение текстового шаблона по типу |
| update\_bot\_text(text\_type, new\_content) | Настройки | Обновление текстового шаблона |
| main() | Главный модуль | Основная функция запуска и работы бота |

В приложении Б можно увидеть листинг представленных в таблице 3.1 функций.

**3.1.2** **Описание использованных компонентов и библиотек**

Для реализации Telegram-бота «ProffIT» использованы следующие компоненты и библиотеки, описанные в таблице 3.2.

Таблица 3.2 – Использованные библиотеки и компоненты

|  |  |
| --- | --- |
| Компонент/библиотека | Назначение |
| telebot.types.InlineKeyboardButton | Элементы кнопок с callback-данными |
| telebot.apihelper | Настройка HTTP-сессии для повторных запросов |
| telebot.TeleBot | Основной класс для создания и управления Telegram-ботом |
| telebot.types.InlineKeyboardMarkup | Создание интерактивных inline-кнопок |
| Декоратор @bot.message\_handler | Обработка текстовых сообщений и команд пользователя |
| Декоратор @bot.callback\_query\_handler | Обработка нажатий на inline-кнопки |
| Декоратор++ @bot.message\_handler(content\_types=['photo']) | Обработка загрузки изображений |
| sqlite3 | Работа с встроенной базой данных SQLite |
| json | Сериализация и десериализация структур данных (варианты ответов, веса) |
| datetime, time | Управление временными метками, задержками, расписанием |
| requests.Session() | Настройка HTTP-сессии с поддержкой повторных запросов |
| urllib3.util.retry.Retry | Конфигурация политики повторных попыток соединения |
| requests.adapters.HTTPAdapter | Адаптер для HTTP-сессии с настройкой ретраев |
| schedule | Библиотека для планирования периодических задач и запуска функций по заданному времени (human-friendly синтаксис). |
| threading | Стандартный модуль Python для реализации многопоточности (обеспечивает параллельную работу бота и планировщика). |
| os.environ | Загрузка конфигурационных параметров из переменных окружения |
| sys | Системные функции для управления выполнением программы |

**3.2 Спецификация проекта**

Проект Telegram-бота «ProffIT» имеет следующую файловую структуру, представленную в таблице 3.3.

Таблица 3.3 – Спецификация программы

|  |  |
| --- | --- |
| Имя файла/компонента | Назначение и содержание |
| main.py | Главный исполняемый файл, содержащий весь исходный код бота. |
| bot\_database.db | Файл базы данных SQLite (создаётся автоматически при первом запуске) |
| файл зависимостей requirements.txt | Список необходимых Python-пакетов. |
| .venv/ | Папка виртуального окружения Python, содержащая изолированную среду выполнения с установленными зависимостями (pyTelegramBotAPI, requests и др.). Содержит все необходимые библиотеки для работы проекта |
| .idea/ | Служебная папка среды разработки PyCharm/IntelliJ IDEA, содержащая конфигурации проекта, настройки IDE, метаданные и файлы индексации |

**4. Тестирование**

В период с 28 по 30 ноября происходил процесс тестирования Telegram-бота «ProffIT». Успешно было пройдено 100% тест-кейсов крайне высокого приоритета, 0% тест-кейсов высокого приоритета, 100% тест-кейсов среднего приоритета и 50% тест-кейсов низкого приоритета.

Команда тестировщиков описана в таблице 4.1.

Таблица 4.1 – Команда тестировщиков

|  |  |  |
| --- | --- | --- |
| ФИО | Должность | Роль |
| Наумовец Анастасия Юрьевна | Разработчик, тестировщик | Ответственный за тестирование приложения |

Тестирование производилось под ОС Windows 11 x64. Все тест-кейсы выполнялись вручную. Тестирование показало среднюю стабильность функциональности (обнаружено два дефекта с важностью «критическая» и четыре – «высокая»). Повторное тестирование показало прирост качества (исправлено 78% обнаруженных ранее дефектов). Расписание проведения тестирования представлено в таблице 4.2.

Таблица 4.2 – Расписание проведения тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Дата | Деятельность | Продолжительность, ч |
| Наумовец Анастасия Юрьевна | 28.11.2025 | Разработка тест-кейсов | 3 |
| Наумовец Анастасия Юрьевна | 29.11.2025 | Тестирование приложения | 4 |
| Наумовец Анастасия Юрьевна | 29.11.2025 | Анализ выполнения тест-кейсов | 1 |
| Наумовец Анастасия Юрьевна | 30.11.2025 | Тестирование приложение. Проведение регрессионного тестирования | 3 |
| Наумовец Анастасия Юрьевна | 30.11.2025 | Составление отчёта о результатах тестирование | 2 |

Разработанные тест-кейсы и результаты тестирования представлены в приложении В в таблицах В.1 для пользователя и В.2 для администратора. Статистика по разработанным тест-кейсам представлена в таблице 4.3.

Таблица 4.3 – Статистика по разработанным тест-кейсам

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Статус | Количество тест-кейсов | Приоритет | | | | |
| Крайне низкий | Низкий | Средний | Высокий | Крайне высокий |
| Пройдено | 14 | 0 | 1 | 2 | 0 | 11 |
| Не пройдено | 2 | 0 | 1 | 0 | 1 | 0 |
| Заблокировано | 0 | 0 | 0 | 0 | 0 | 0 |
| Не протестировано | 0 | 0 | 0 | 0 | 0 | 0 |

В таблице 4.4 представлена статистика по всем дефектам.

Таблица 4.4 – Статистика по выявленным ошибкам

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Статус | Количество ошибок | Важность | | | |
| Низкая | Средняя | Высокая | Критическая |
| Найдено | 7 | 1 | 3 | 2 | 1 |
| Исправлено | 5 | 0 | 3 | 1 | 1 |
| Проверено | 5 | 0 | 3 | 1 | 1 |
| Открыто заново | 0 | 0 | 0 | 0 | 0 |
| Отклонено | 0 | 0 | 0 | 0 | 0 |

Список найденных дефектов представлен в таблице 4.5.

Таблица 4.5 – Список найденных дефектов

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Важность | Описание | Статус |
| 1 | 2 | 3 | 4 |
| D\_01 | Средняя | Вопросы с собеседованиями отображаются без форматирования | Исправлено |
| D\_02 | Высокая | В админ-панели отсутствует валидация при добавлении вопроса: система позволяет сохранить вопрос с пустым текстом, что нарушает целостность данных | Не исправлено |
| D\_03 | Низкая | Бот некорректно обрабатывает и сохраняет имена пользователей, содержащие эмодзи и спецсимволы (имя обрезается после первого спецсимвола) | Не исправлено |
| D\_04 | Средняя | Бот не сохраняет ответы пользователя во время прохождения теста | Исправлено |
| D\_05 | Высокая | Админка не требует пароля при прямом доступе через /admin | Исправлено |
| D\_06 | Критическая | После ответа на вопрос теста бот не переходит к следующему вопросу, а сразу выдает вопросы для собеседования | Исправлено |
| D\_07 | Средняя | Изображение с вопросом и ответами не отображается при начале теста | Исправлено |

**5. Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Telegram-бот «ProffIT» представляет собой программную систему для автоматизированной профессиональной ориентации в IT-сфере. Назначение программы — определение профессиональной предрасположенности пользователей к различным IT-специальностям посредством прохождения адаптивного тестирования и предоставление персонализированных рекомендаций по развитию. Основные возможности включают: прохождение профориентационного теста (25 вопросов), получение детализированных результатов с обоснованием выбора направления, доступ к базе вопросов с собеседований, получение практических заданий и рекомендаций обучающих курсов. Программа функционирует в среде мессенджера Telegram и не предъявляет специфических требований к аппаратному обеспечению пользователя — достаточно устройства с установленным приложением Telegram и доступом в интернет. Использование системы носит эпизодический характер: тестирование выполняется по необходимости, задания — по желанию пользователя.

**5.2 Руководство системного программиста**

Установка программного продукта требует предварительной подготовки: наличия Python версии 3.8+, аккаунта в Telegram с созданным ботом через @BotFather, и доступа к серверу с постоянным интернет-соединением. Процесс инсталляции включает: установку зависимостей через pip install pyTelegramBotAPI requests, настройку переменных окружения (токен бота и пароль администратора в файле .env), запуск инициализации командой python main.py. При успешной установке система выводит сообщения: «Бот запускается...». Возможные сообщения об ошибках включают проблемы с подключением к базе данных (требуют проверки прав доступа) или ошибки аутентификации (неверный токен). После установки необходимо проверить работоспособность, отправив боту команду /start и убедившись в получении приветственного сообщения.

**5.3 Руководство пользователя**

**5.3.1 Запуск программы**

Запуск программы осуществляется через мессенджер Telegram тремя способами: через поиск по username бота, по прямой ссылке формата https://t.me/ @PR0FIT0\_bot, или путем ввода команды /start@PR0FIT0\_bot в любом чате. После открытия диалога с ботом необходимо нажать кнопку «Запустить» или «Start». Индикацией успешного запуска служит приветственное сообщение с именем пользователя и появление кнопки «Начать тест», можно посмотреть на рисунке 5.1:

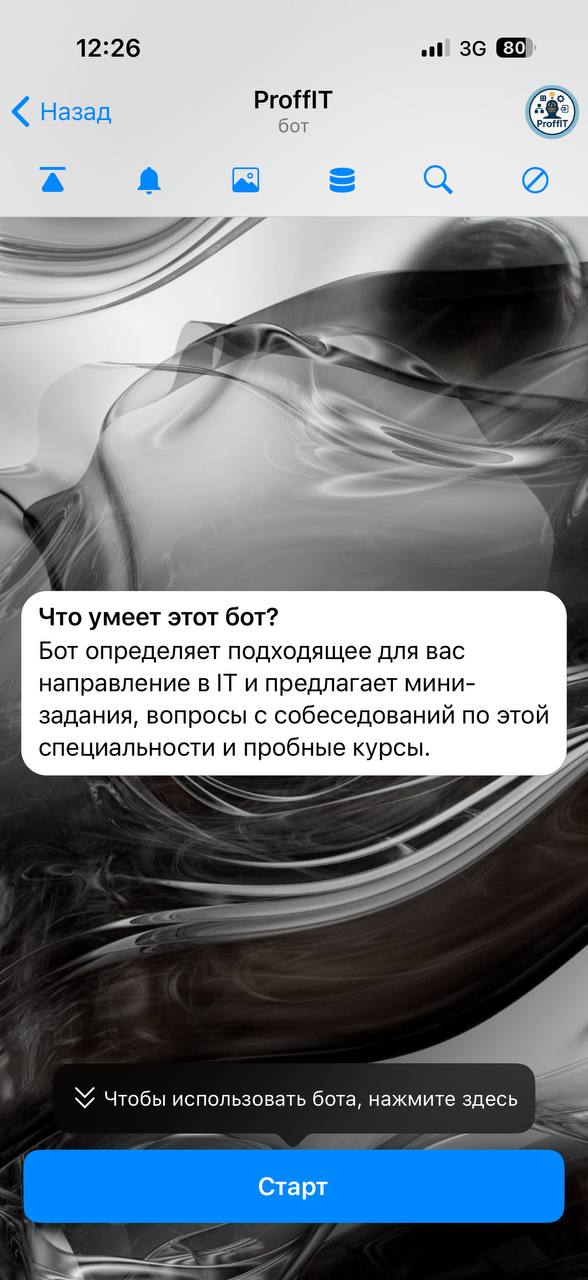


Рисунок 5.1 – Запуск бота

**5.3.2 Инструкции по работе с программой**

**5.3.2.1** **Инструкции по работе с программой для пользователя**

Управление программой осуществляется через текстовые команды и интерактивные кнопки. Основные команды можно увидеть на рисунке 5.2:

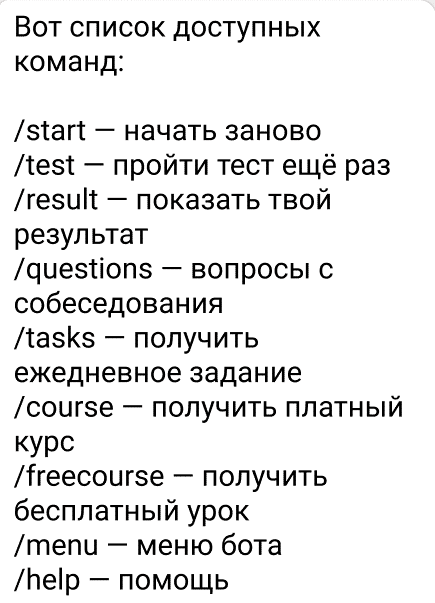


Рисунок 5.2 – Команды для управления ботом

Процесс тестирования состоит из 25 вопросов с пятью вариантами ответов каждый, на рисунке 5.3 можно увидеть, что он начинается после нажатия кнопки старт либо же отправки команды /start и выбора кнопки «начать тест»; выбор варианта осуществляется нажатием на соответствующую инлайн-кнопку, представлено на рисунке 5.4. На рисунке 5.5 видно, что после завершения теста система автоматически вычисляет результат и выводит рекомендации по подходящему IT-направлению с обоснованием и предложение получить вопросы с помощью нажатия на соответствующую кнопку. Далее можно посмотреть, что пользователю последовательно предлагается: на рисунке 5.6 получить вопросы для собеседования, на рисунке 5.7 получить задание, на рисунке 5.8 ознакомиться с обучающими курсами. Все взаимодействия сохраняются, позволяя продолжить работу с любого этапа.

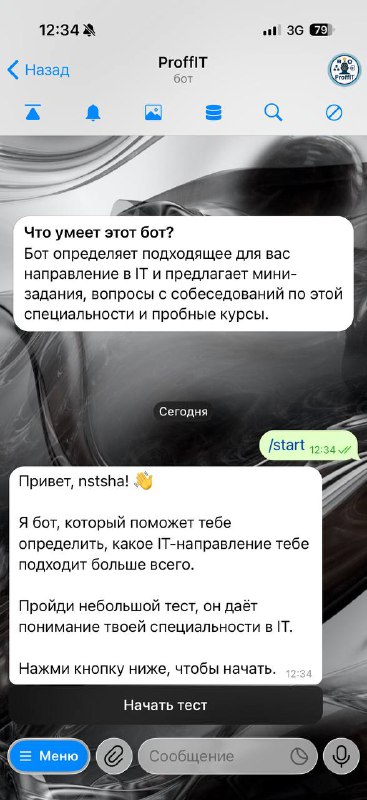


Рисунок 5.3 – Начать тест

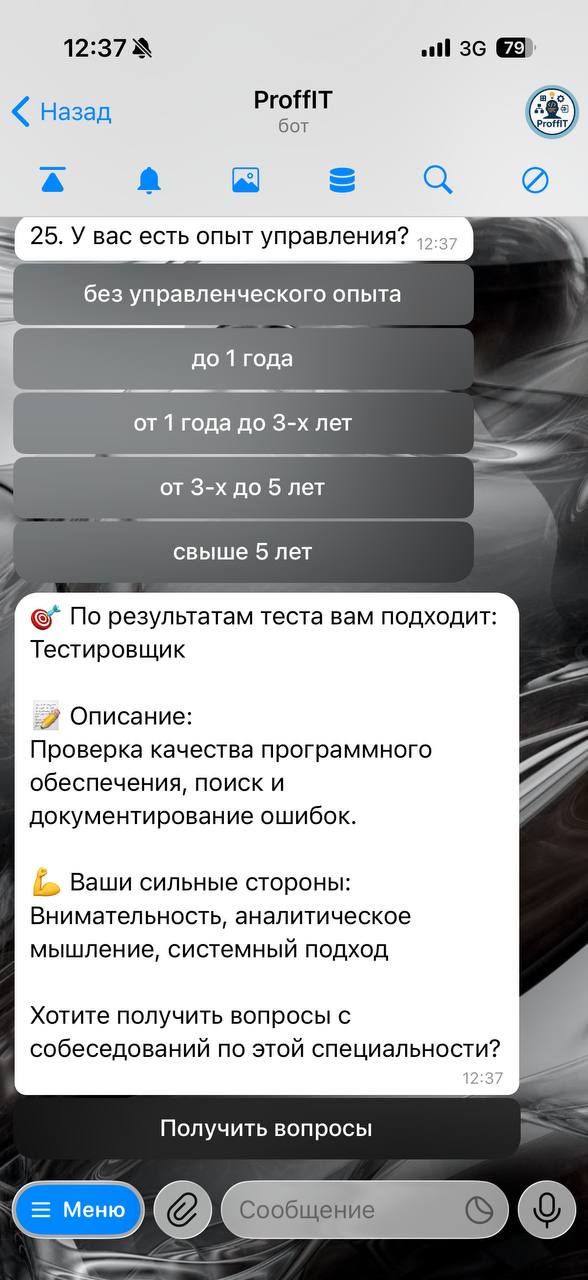


Рисунок 5.5 – Результаты и кнопка «получить вопросы»

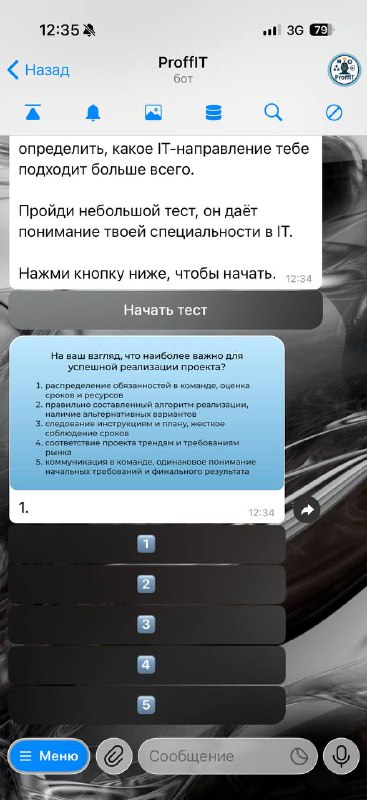


Рисунок 5.4 – Выбор ответа

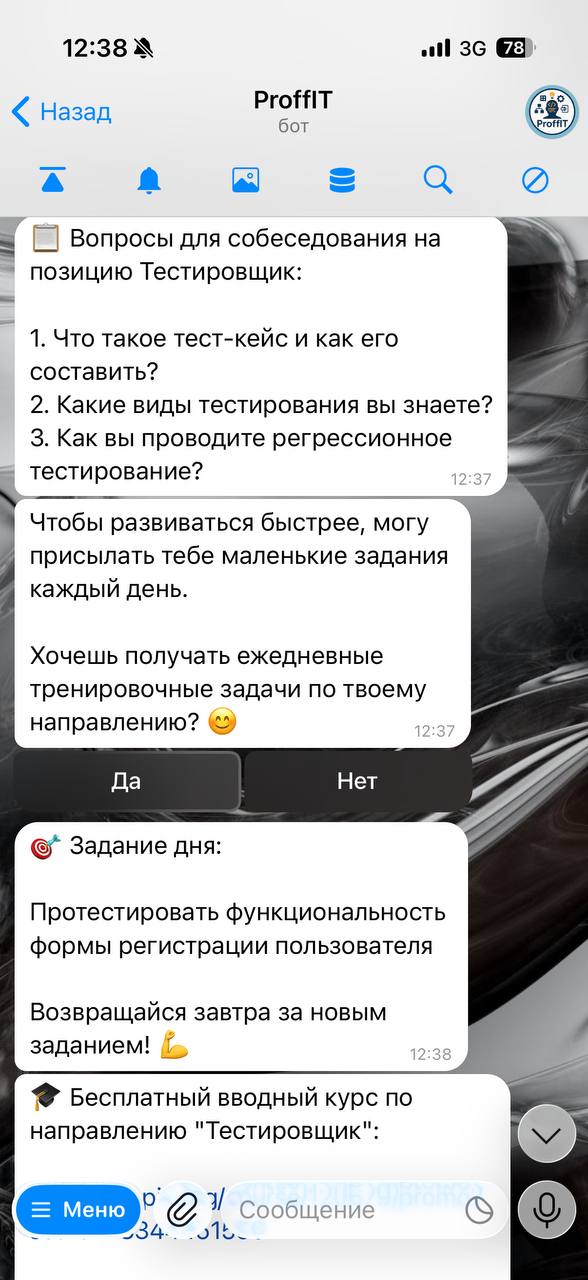


Рисунок 5.7 – Получение задания

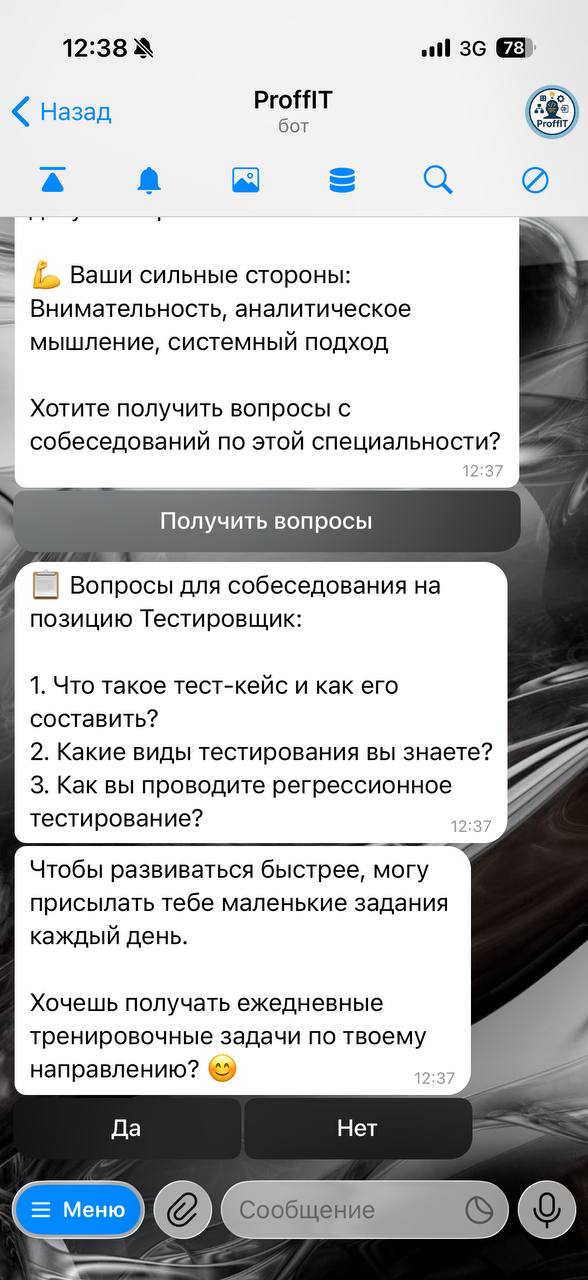


Рисунок 5.6 – Вопросы

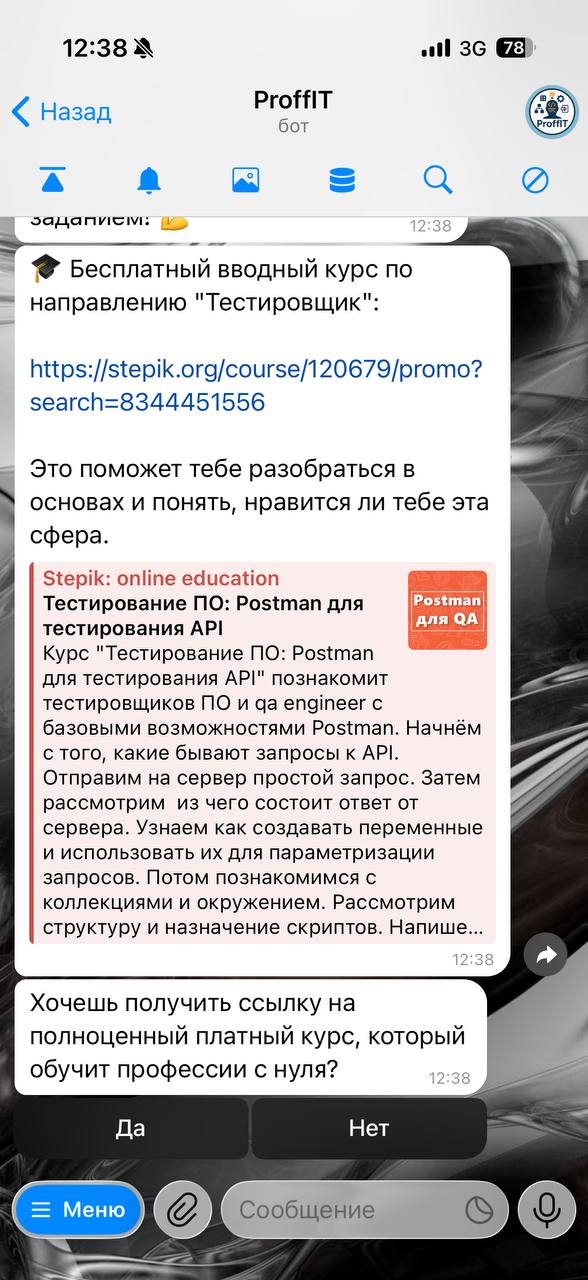


Рисунок 5.8 – Обучающие курсы

**5.3.2.2** **Инструкции по работе с программой для администратора**

Как войти в режим администратора показано на рисунке 5.9: нужно написать боту команду /admin и ввести правильный пароль. После этого будет доступна админ-панель со статистикой бота, которую можно увидеть на рисунке 5.10. Для управления контентом бота можно выбирать соответствующие кнопки меню.

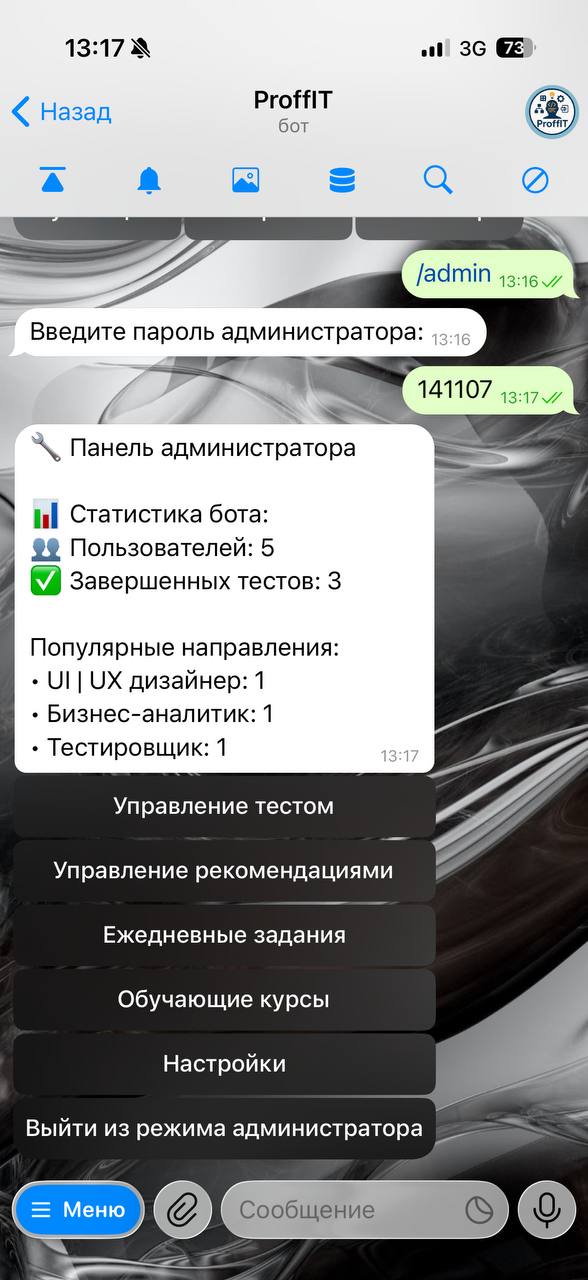


Рисунок 5.10 – Админ-панель

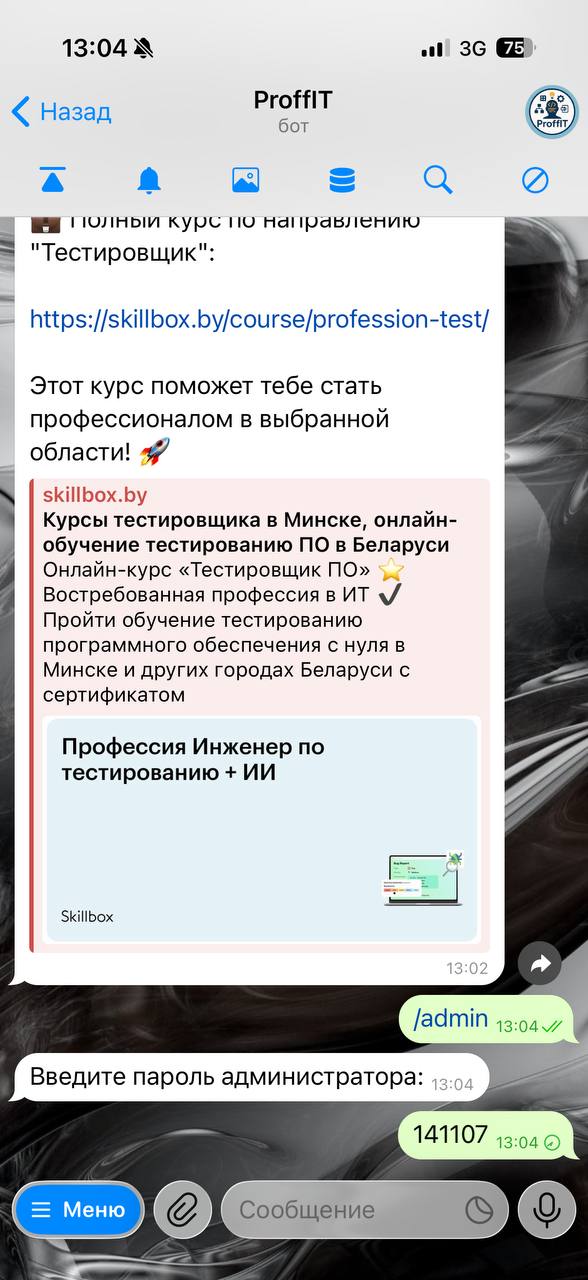


Рисунок 5.9 – Вход в админ-панель

На рисунках 5.11 и 5.12 представлен функционал кнопок админ панели.

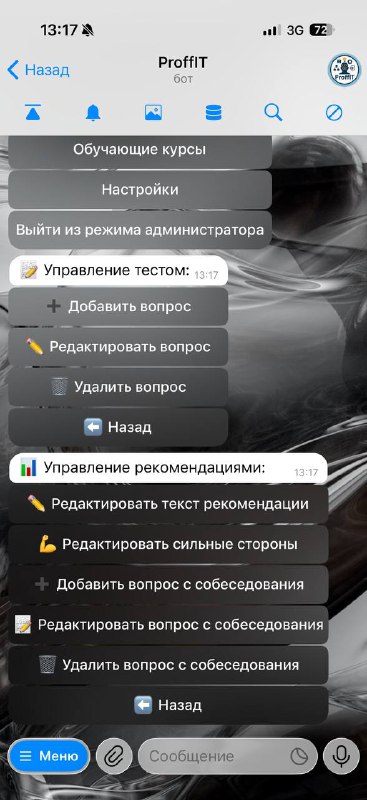


Рисунок 5.11 – Функционал кнопок

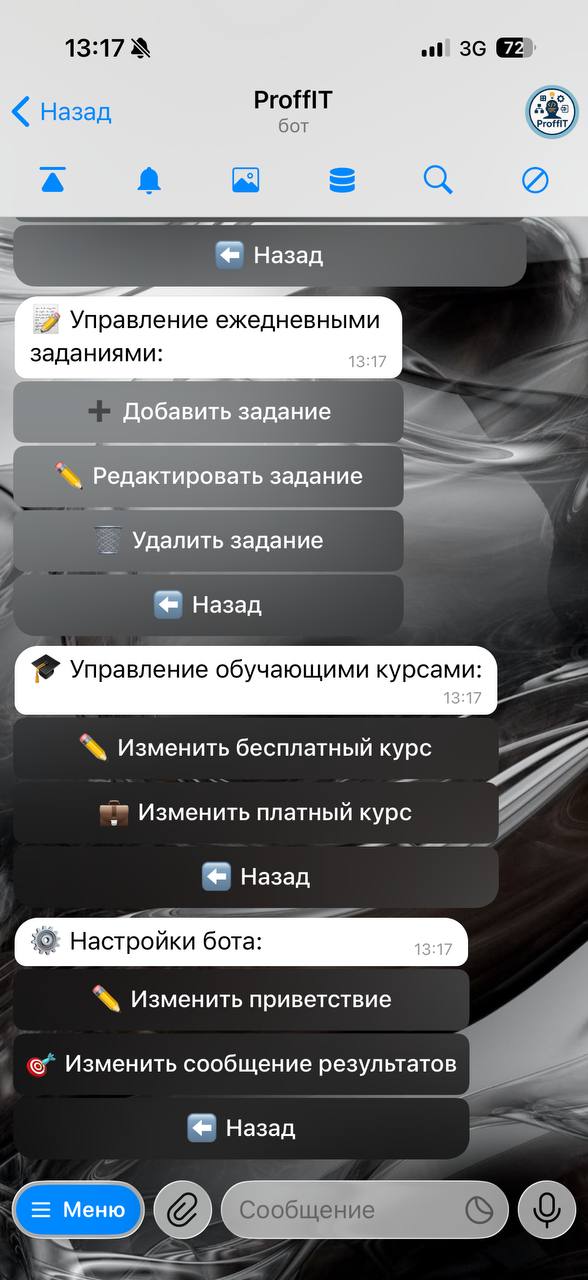


Рисунок 5.12 – Функционал кнопок

**5.3.3 Завершение работы с программой**

Программа не требует явного завершения работы. Пользователь может просто закрыть чат с ботом — все данные автоматически сохраняются. При неактивности более 5 минут во время теста сессия прерывается с сохранением прогресса. Для полного сброса данных следует отправить команду /start. Работа на нескольких устройствах одновременно поддерживается, синхронизация данных осуществляется через облачную базу данных.

**5.3.4 Использование системы справочной информации**

Система справочной информации активируется командой /help, выводящей полный список доступных команд с краткими описаниями. Контекстная помощь предоставляется автоматически: при вводе неизвестной команды бот предлагает использовать /help.

**Заключение**

Целью данного проекта является разработка программного продукта «ProffIT» — интеллектуального Telegram-бота для профессиональной ориентации в IT-сфере. В ходе работы создан полнофункциональный программный комплекс с комплексной логикой тестирования, персонализации и администрирования.

Приложение рассчитано на широкий круг пользователей: студентов, выпускников, специалистов, рассматривающих смену профессионального направления, а также всех, интересующихся возможностями развития в IT-индустрии. Система позволяет пользователям не только определить наиболее подходящее IT-направление, но и получить практические инструменты для профессионального роста: подготовку к собеседованиям, практические задания и подборку обучающих курсов.

В ходе выполнения курсовой работы разработан и протестирован Telegram-бот с многоуровневой архитектурой, включающей: алгоритм взвешенного определения профессиональной предрасположенности, систему персонализированных рекомендаций, механизм отправления заданий, базу вопросов для собеседований, административную панель управления контентом. Реализована гибкая система навигации с двумя типами пользовательских путей — линейным (последовательное прохождение этапов) и модульным (самостоятельный выбор разделов).

В качестве дальнейшего развития проекта возможны следующие улучшения: расширение базы направлений и вопросов теста, интеграция с внешними API образовательных платформ, реализация системы прогресса и достижений пользователей, добавление функции сравнения результатов с группой, внедрение искуственного интеллекта для генерации заданий или вопросов теста.

При разработке использована технология Python с библиотекой pyTelegramBotAPI и SQLite в качестве базы данных, что позволило создать масштабируемую систему с четким разделением ответственности между компонентами. В процессе работы освоены принципы проектирования чат-ботов, работы с конечными автоматами (FSM) для управления сложными операциями, интеграции медиаконтента и реализации устойчивых к сбоям систем.

В заключение можно констатировать, что разработанный программный продукт является законченным, полнофункциональным решением, готовым к использованию целевой аудиторией. Бот успешно решает поставленные задачи профориентации и предоставляет пользователям комплексный инструмент для профессионального развития в IT-сфере.

**Список использованных источников**

1. Лутц М. Изучаем Python / М. Лутц. – 5-е изд. – СПб.: Символ-Плюс, 2022. – 1648 с.

2. Ромальо Л. Python. Книга рецептов / Л. Ромальо. – 3-е изд. – М.: ДМК Пресс, 2023. – 768 с.

3. Telegram Bot API. Официальная документация [Электронный ресурс]. – URL: https://core.telegram.org/bots/api – дата доступа: 21.11.2025.

4. PyTelegramBotAPI Documentation [Электронный ресурс]. – URL: https://pytba.readthedocs.io/en/latest/ – дата доступа: 21.11.2025.

5. SQLite Documentation [Электронный ресурс]. – URL: https://www.sqlite.org/docs.html – дата доступа: 22.11.2025.

6. Python Documentation [Электронный ресурс]. – URL: https://docs.python.org/3/ – дата доступа: 24.11.2025.

7. Чекрыжова О.И. Проектирование пользовательских интерфейсов / О.И. Чекрыжова. – М.: ДМК Пресс, 2021. – 304 с.

8. Кузнецов С.Д. Базы данных / С.Д. Кузнецов. – М.: Логос, 2022. – 488 с.