

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
подпись
«_____» _____ 2022 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка Telegram-бота для информирования управляющих и
ресурсоснабжающих организациях

Руководитель _____ доцент, канд. техн. наук Е. Н. Скуратенко
подпись, дата

Выпускник _____ А. А. Чаптыков
подпись, дата

Консультанты
по разделам:

Экономический _____ Е. Н. Скуратенко
подпись, дата

Нормоконтролер _____ В. И. Кокова
подпись, дата

Абакан 2022

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка Telegram-бота для информирования о управляющих и ресурсоснабжающих организациях» содержит 65 страниц текстового документа, 57 иллюстраций, 11 таблиц, 1 использованный источник, 7 формул.

ЧАТ-БОТ, ГИС ЖКХ, ТЕЛЕГРАММ-БОТ, МЕССЕНДЖЕР, TELEGRAM, ПРОГРАММНЫЙ ПРОДУКТ, PYTHON, PYSNARM, ЗАТРАТЫ, РИСКИ, ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ.

Целью выпускной квалификационной работы является создание Telegram-бота для информирования жителей Республики Хакасия об управляющих и ресурсоснабжающих организациях.

Объект выпускной квалификационной работы: разработка Telegram-бота.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области.
2. Разработать актуальное проектное решение.
3. Создать Telegram-бот.
4. Создать информационную систему.
5. Сдать информационную систему заказчику для внедрения.
6. Провести расчет экономических показателей реализации проекта.

В результате был разработан Telegram-бот для информирования жителей Республики Хакасия об управляющих и ресурсоснабжающих организациях.

SUMMARY

The theme of the graduation thesis is «Telegram-Bot Development for Public Regulatory Organization and Utility Companies». It contains 65 pages, 57 figures, 11 tables, 11 reference items, 7 formulae.

CHATBOT, STATE IT SYSTEM OF HOUSING AND PUBLIC UTILITIES, TELEGRAM-BOT, MESSENGER, TELEGRAM, SOFTWARE PRODUCT, PYTHON, PYCHARM, COSTS, RISKS, ECONOMIC EFFICIENCY.

The purpose of the graduation thesis is to create a Telegram-bot to inform residents of Republic of Khakassia on Public Regulatory Organization and Utility Companies.

The object of the thesis is to develop a Telegram-bot.

Objectives:

1. to conduct the analysis of the subject area;
2. to develop an up-to-date design solution;
3. to create a Telegram-bot;
4. to design an IT system;
5. to hand over the IT system to the customer for implementation;
6. to calculate the economic costs of the project.

English language supervisor
signature, date

_____ full name

N.V. Chezybaeva

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Анализ предметной области.	7
1.1 Описание основной деятельности Центра управления регионом Республики Хакасия.....	7
1.2 Обоснование актуальности разработки программного продукта.....	8
1.3 Анализ аналогов и обоснование выбора средств разработки	10
1.4 Архитектура программного продукта.....	18
1.5 Описание диаграмм бизнес-процессов	19
1.6 Выводы по разделу «Анализ предметной области»	22
2 Разработка Telegram-бота поиска управляющих и ресурсоснабжающих организаций	23
2.1 Установка необходимого ПО	23
2.2 Описание программного кода.....	25
2.3. Тестирование программного продукта	42
2.4. Выводы по разделу «Разработка Telegram-бота поиска управляющих и ресурсоснабжающих организаций»	46
3 Расчёт затрат и оценка экономической эффективности реализации проекта «Разработка Telegram-бота поиска управляющих и ресурсоснабжающих организаций»	47
3.1 Анализ состава и стоимости ресурсов, необходимых для реализации проекта.....	47
3.2 Расчет проектных затрат	50
3.3 Расчет капитальных затрат	52
3.4 Расчет эксплуатационных затрат	55
3.5 Расчет совокупной стоимости владения системой.....	56
3.6 Оценка рисков реализации проекта	58
3.7 Экономическая эффективность реализации проекта	59
3.8 Выводы по разделу «Расчёт затрат и оценка экономической эффективности реализации проекта «Разработка Telegram-бота для информирования о управляющих и ресурсоснабжающих организаций»»	60
ЗАКЛЮЧЕНИЕ	61
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	62

ВВЕДЕНИЕ

На сегодняшний день интернет является неотъемлемой частью современного мира, т.к. предоставляет широчайшие возможности свободного получения и распространения научной, деловой, познавательной и развлекательной информации. Средствами для предоставления информации могут являться разнообразные сайты, социальные сети, мессенджеры. Популярными средствами общения и получения информации становятся мессенджеры.

Работа выполнена по заказу Центра управления регионом Республики Хакасия, основными задачами которого являются сбор сообщений, жалоб, предложений, поступающих из разных источников с целью оперативного рассмотрения и принятия решений правительством республики. Так же одной из задач Центра управления регионом Республики Хакасия является информирование граждан по жизненно важным вопросам, в том числе предоставление информации об управляющих и ресурсоснабжающих организациях.

На данный момент получить информацию об управляющих и ресурсоснабжающих организациях можно в информационном портале ГИС ЖКХ и при личном обращении в Министерство строительства и жилищно-коммунального хозяйства Республики Хакасия.

Одним из самых популярных мессенджеров является Telegram. Telegram позволяет общаться с другими пользователями, обмениваться информацией и файлами, а также выполнять много полезных задач. Также через мессенджер можно узнавать о погоде, играть в игры, переводить тексты, просматривать последние новости и так далее. А выполнять все эти действия помогают специальные роботы – Telegram-боты. Telegram-бот – это специальный аккаунт, позволяющий совершать различные действия через сам мессенджер. Основным достоинством ботов является совмещение всех возможностей на платформе одного мессенджера. Также такой бот способен

упростить получение информации об организациях, которые обслуживают дома по их адресу.

Автоматизация этого действия и послужила поводом для разработки бота в мессенджере Telegram.

Целью выпускной квалификационной работы является создание Telegram-бота для информирования жителей Республики Хакасия об управляющих и ресурсоснабжающих организациях.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области.
2. Разработать актуальное проектное решение.
3. Создать Telegram-бот.
4. Создать информационную систему.
5. Сдать информационную систему заказчику для внедрения.
6. Провести расчет экономических показателей реализации проекта.

1 Анализ предметной области.

1.1 Описание основной деятельности Центра управления регионом Республики Хакасия

Центр управления регионом Республики Хакасия – специализированный пункт управления в формате постоянно действующей межведомственной рабочей группы, предназначенный для обеспечения деятельности Главы Республики Хакасия – Председателя Правительства Республики Хакасия, Правительства Республики Хакасия, при реализации функций мониторинга и координации обработки обращений и сообщений граждан, объединений граждан или юридических лиц, оценки, анализа и прогнозирования ситуации, стратегического, текущего и оперативного планирования, мониторинга и контроля исполнения управленческих решений.

В своей деятельности ЦУР руководствуется Конституцией Российской Федерации, национальной программой «Цифровая экономика Российской Федерации», утвержденной протоколом заседания президиума Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам от 04.06.2019 № 7, постановлением Правительства Российской Федерации от 31.10.2018 № 1288 «Об организации проектной деятельности в Российской Федерации», иными нормативными правовыми актами Российской Федерации и Республики Хакасия.

Месторасположение ЦУР: Республика Хакасия, г. Абакан, просп. Ленина, д. 67.

В структуру ЦУР входят сотрудники исполнительных органов государственной власти/подведомственных учреждений Республики Хакасия по тематикам: жилищно-коммунальное хозяйство, твердые коммунальные отходы, энергетика, образование, здравоохранение, дороги, транспорт, социальная защита [1].

Одной из задач Центра управления регионом Республики Хакасия является информирование граждан по жизненно важным вопросам, в том числе предоставление информации об управляющих и ресурсоснабжающих организациях.

На данный момент существуют проблемные моменты предоставления такой информации в республике, в этой связи была поставлена задача: создание чат-бота поиска управляющих и ресурсоснабжающих организаций.

1.2 Обоснование актуальности разработки программного продукта

С развитием социальных сетей и различных мобильных приложений люди все чаще обращаются за информацией, используя данные ресурсы. Владельцу мобильного телефона гораздо удобнее воспользоваться приложением, нежели заходить на сайты и искать требуемую информацию.

В наши дни с целью информирования граждан по различным вопросам деятельности государственных органов развиваются группы социальных сетей различных государственных учреждений. Достичь поставленной цели – информирования граждан своевременно, невозможно без применения информационных технологий, например, таких как чат-бот. Чат-бот, иными словами виртуальный собеседник, программа, имитирующая реальный разговор с пользователем. Чат-боты позволяют общаться с помощью текстовых или аудио сообщений на сайтах, в мессенджерах, мобильных приложениях или по телефону.

Весной 2022 года в России одним из самых популярных мессенджеров стал Telegram. Многие государственные деятели и государственные организации создали там свои аккаунты. Учитывая вышесказанное, многие специализированные группы государственных органов в Telegramканале стали пользоваться технологией чат-ботов для удобного информирования граждан по разным вопросам.

Один из вопросов интересующих граждан республики Хакасия- где узнать перечень организаций, занимающихся предоставлением жилищно-коммунальных услуг по определенному адресу. Для упрощения получения информации об организациях, обслуживающих дома, было бы удобно создать чат-бот в популярном мессенджере Telegram.

Для полноты ответа на запрос пользователя о перечне организаций, обслуживающих определенный дом, необходимо выводить следующую информацию:

- 1) вид организации;
- 2) фактический адрес;
- 3) сайт в сети Интернет;
- 4) контактный телефон.

На сегодняшний день узнать данный перечень возможно путем получения квитанций, приходящих на почту по адресу или с помощью сайта ГИС ЖКХ. И тот и другой способ имеют затруднения, например, потеря квитанции и сложность поиска информации на всероссийском сайте. Необходимый раздел с управляющими и ресурсоснабжающими компаниями на сайте ГИС ЖКХ называется «Реестры», так что часто пользователи не связывают это с перечнем организаций, далее необходимо сделать несколько переходов чтобы попасть в поле ввода адреса и затем получить информацию, что еще раз подтверждает необходимость дополнительного ресурса получения информации о перечне организаций, занимающихся предоставлением жилищно-коммунальных услуг по определенному адресу.

Все вышесказанное подтверждает актуальность темы работы: Разработка Telegram-бота поиска управляющих и ресурсоснабжающих организаций.

1.3 Анализ аналогов и обоснование выбора средств разработки

Чат-бот – это программа, выполняющая различные действия, автоматически или по определенной команде.

В Telegram используется один общий вид ботов, которых от обычных пользователей отличает только наличие приставки «bot» в имени. Сами же боты делятся на несколько направлений:

1. Чат-боты. Представляют собой простейший чат, имитирующий общение на заданную пользователем тематику.

2. Боты-информаторы. Отдельный вид ботов, главная цель которых — информирование пользователя о тех или иных событиях (новости, мероприятия, публикации и т.п.).

3. Игровые боты. Боты, в которых можно поиграть в различные игры. По большей части, это текстовые версии разных игр.

4. Боты-ассистенты. Боты, разработанные различными онлайн-сервисами, как дополнение к основной веб-версии.

На самом деле, четкого деления нет, т.к. некоторые боты содержат сразу несколько механик и успешно выполняют множество пользовательских задач. С их помощью можно переводить, обучаться, тестировать, искать информацию, играть в игры и даже пользоваться другими сервисами и взаимодействовать с вещами, имеющими выход в глобальную сеть. Все боты в Telegram являются бесплатными, но в 2017 году Павлом Дуровым была анонсирована возможность настроить и платное пользование такими программами [2].

Они предоставляют возможность решать элементарные задачи при помощи мгновенных команд, причем все эти программы не нуждаются в установке и не занимают отдельное место в памяти девайса. Первым типом бота с заранее определенными действиями является разрабатываемый чат-бот «Бот, определяющий управляющие компании с контактами».

Аналогами разрабатываемого бота могут являться следующие представители:

а) Бот для проверки утечек паролей (@MailSearchBot). Этот Telegram-бот позволяет проверить аккаунты в социальных сетях на утечку паролей. Для работы необходимо ему написать никнейм аккаунта, далее в базе данных утечек паролей Telegram-бот ищет данный аккаунт и выдает информацию о безопасности аккаунта.

На рисунке 1 представлен интерфейс Telegram-бота – @MailSearchBot.

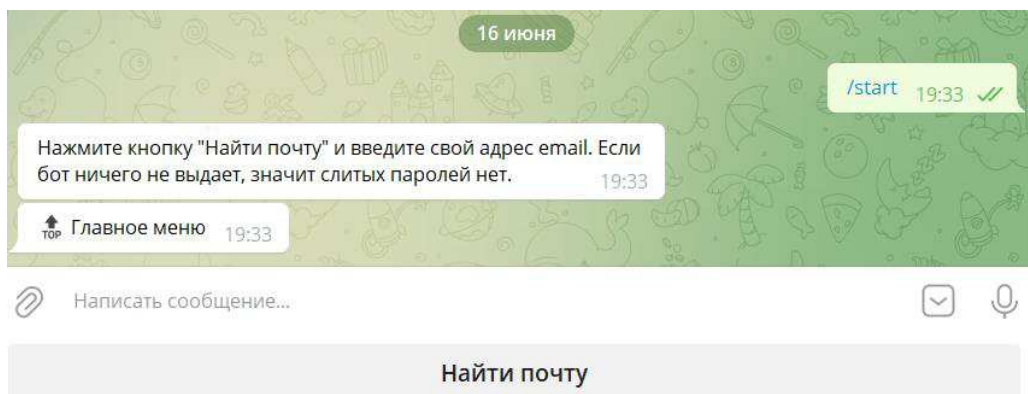


Рисунок 1 – Интерфейс Telegram-бота – @MailSearchBot

б) Штрафы ГИБДД (@youshtraf_bot). Данный Telegram-бот работает аналогичным образом. Для начала работы необходимо написать государственный номер автомобиля, и после этого он находит штрафы и другую информацию в открытых источниках, таких как базой данных ГИБДД и системой «Платон».

Необходимо отметить, что в данном боте применяются только команды, которые можно изучить при начале работы с ним, а также их можно посмотреть и применить в выдвижном меню бота.

На рисунке 2 представлена инструкция Telegram-бота – @youshtraf_bot, список команд представлен на рисунке 3.

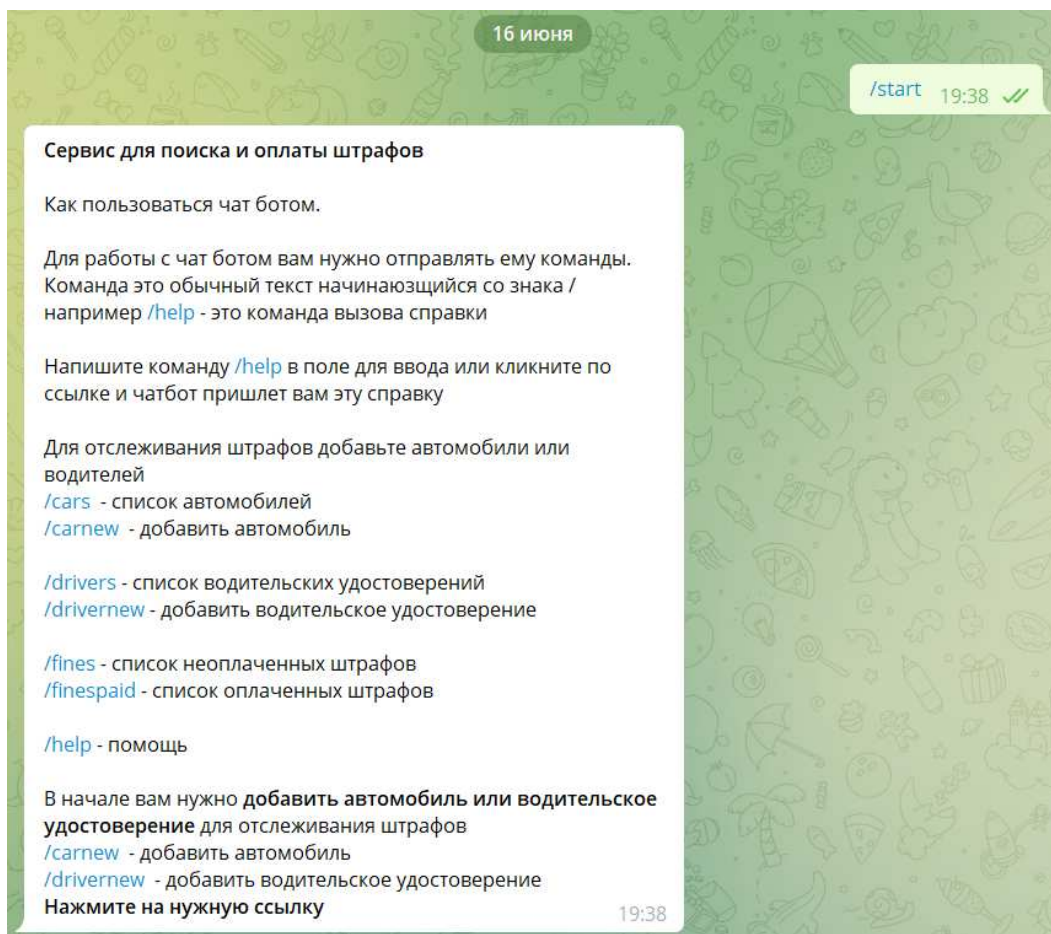


Рисунок 2 – Интерфейс Telegram-бота – @youshtraf_bot

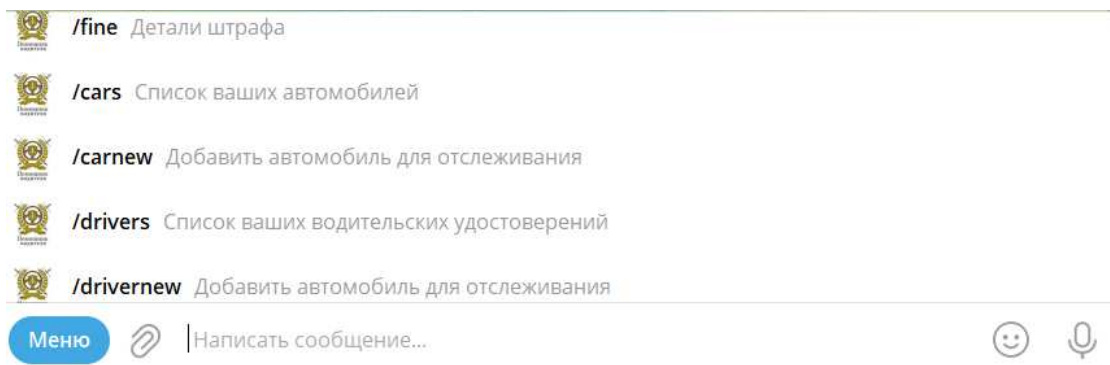


Рисунок 3 – Список команд Telegram-бота – @youshtraf_bot

в) Бот-бухгалтер (@iloveipbot). Помощник для индивидуальных предпринимателей. Умеет рассчитывать страховые взносы ИП, помогает уменьшить налог по УСН, напоминает о предстоящих платежах и сдаче отчетности. В будущем разработки обещают реализовать интеграцию бота с

банковскими системами, добавить функции загрузки выписок и отправки платёжных поручений, подготовки и сдачи отчётности через интернет.

На рисунке 4 представлен интерфейс Telegram-бота – @iloveipbot.

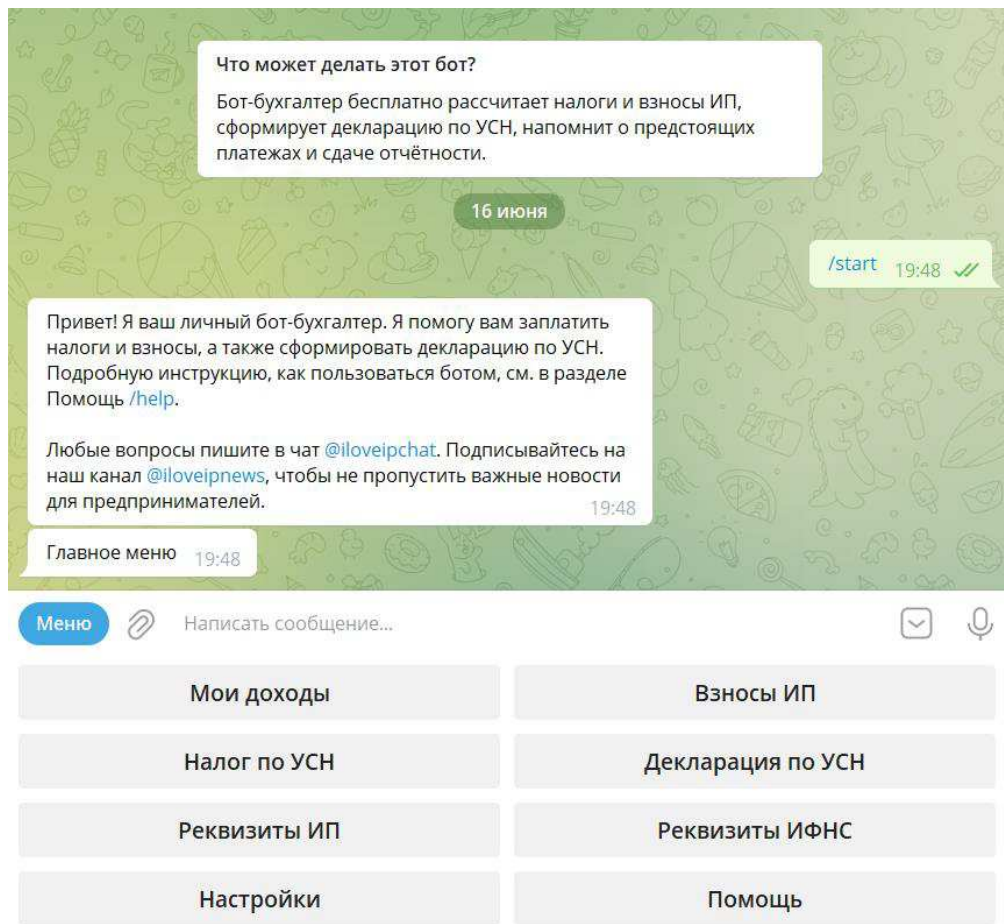


Рисунок 4 – Интерфейс Telegram-бота – @iloveipbot

г) Словарный бот (@dictsbot). Сервис, предоставляющий доступ к внушительной коллекции толковых и тематических словарей на русском, украинском, английском и немецком языках. Поддерживается взаимодействие с «Википедией», а также с системами перевода «Яндекса» и Google[3].

На рисунке 5 представлен интерфейс Telegram-бота – @dictsbot.

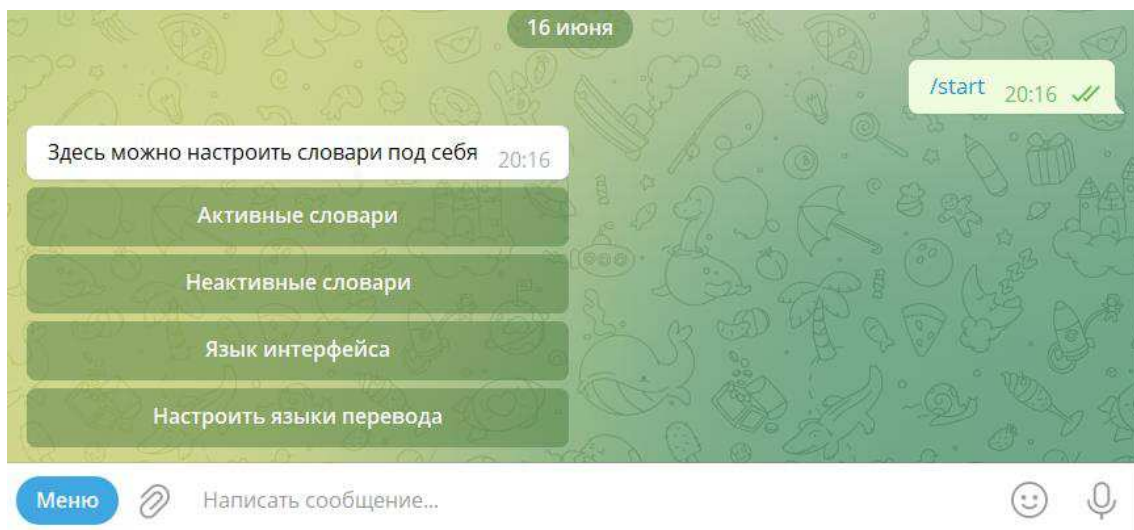


Рисунок 5 – Интерфейс Telegram-бота – @dictsbot

д) Выборы 2021. Хакасия (@tsur19_izbirkom_bot). Бот позволяет получить всю необходимую информацию о голосовании 2021, в том числе узнать адрес участковой избирательной комиссии.

На рисунке 6 представлен интерфейс Telegram-бота – @tsur19_izbirkom_bot.

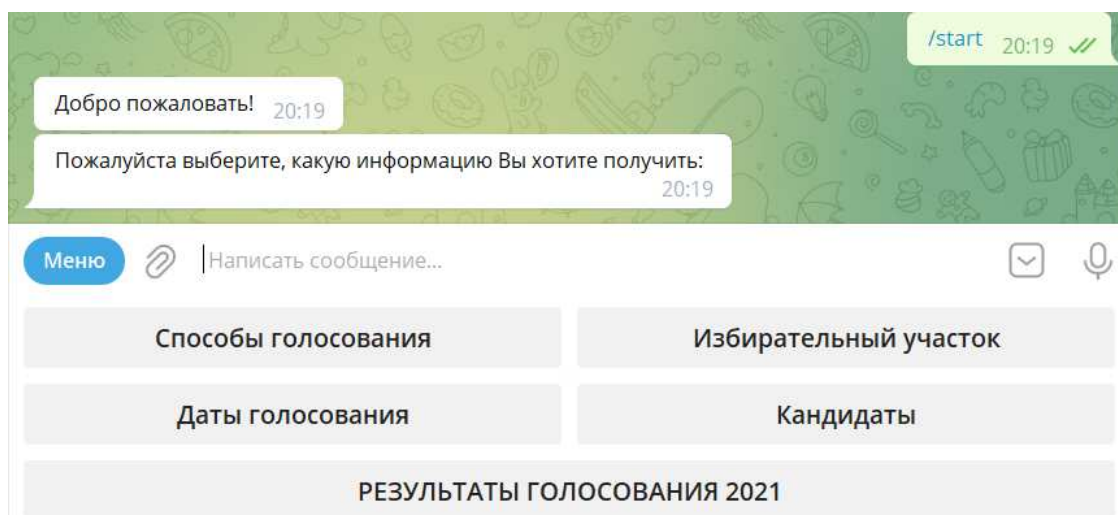


Рисунок 6 – Интерфейс Telegram-бота – @tsur19_izbirkom_bot

Является самым приближенным аналогом создаваемого бота, т.к. действия аналогичные, но на выходе получаем данные участковой избирательной комиссии.

На основе данных о представленных Telegram-ботах составим таблицу для сравнения по следующим критериям: интерфейс, скорость ответа, полезность, где:

5 – отлично;

4 – хорошо(таблица 1).

Таблица 1 – Сравнительная таблица Telegram-ботов

Критерий	@MailSearch Bot	@youshtraf_bot	@iloveipbot	@dictsbot
Интерфейс	5	5	5	5
Скорость ответа	4	5	5	4
Полезность	5	5	5	5
В целом	5	5	5	5

Если же разбирать по отдельности критерии, то необходимо отметить, что интерфейс существующих Telegram-ботов в основном всегда удобен, так как именно в Telegram-ботах интерфейс состоит из кнопок выдвижного меню, и у разработчика не возникает особых трудностей. Скорость ответа Telegram-бота зависит от: архитектуры решения, алгоритмов, мощности сервера, скорости сети, стабильности управляющего сервиса, и загруженности Telegram-бота. Полезность же подразумевает, насколько же этот Telegram-бот действительно необходим и полезен разным категориям граждан. В нашем случае были приведены глобальные проекты, пользующиеся высоким спросом, поэтому и полезность высокая. Но если же рассматривать отдельные категории Telegram-ботов, то можно сказать, что существуют и боты личного пользования, а также боты, используемые определенной группой лиц, которые несут пользу только им.

Для реализации чат-бота в Telegram подходят языки программирования Python, php, C# и другие. Для того чтобы реализовать чат-бот нужно выбрать

среду для разработки.

Если в качестве языка программирования выбрать C# или Node.js в среде разработки Microsoft Visual Studio, то можно воспользоваться готовым набором шаблонов, таких как Bot Application, Bot Framework Emulator, Bot Dialog и Bot Controller.

Также подойдет язык php. У него имеются уже готовые библиотеки для работы с Телеграмм, позволяющие свести усилия разработчика к минимуму.

Как показывает статистика, самым распространенным языком программирования для создания Telegram-ботов является Python. В нем также содержится множество библиотек, которые работают именно с Telegram. Помимо этого, по данному языку больше всего документаций и бесплатных курсов для разработки Telegram-ботов.

Из списка рассмотренных языков программирования самым подходящим является Python, в виду своей популярности и множеством реализованных библиотек.

После выбора языка программирования, необходимо определиться со средой разработки, в которой будет разрабатываться Telegram-бот. Самыми распространенными средами разработки являются: MCVisualStudio, SublimeText и PyCharm. Сравним эти три среды и выберем наиболее подходящую из них. Сравнимые характеристики сред разработки представлены в таблице 2.

Таблица 2 – Среда разработки

Функции	Visual Studio Code	PyCharm	Sublime Text
Стоимость	Бесплатная среда разработки.	Professional версия платная. Community версия для разработки только на python бесплатная.	Разработчик позволяет бесплатно ознакомиться с продуктом, однако программа уведомляет о необходимости приобретения

			лицензии.
--	--	--	-----------

Продолжение таблицы 2

Интерфейс	Удобный интерфейс для пользователя.		
Поддержка языков программирования	Поддерживает большое количество языков программирования.	Только полная поддержка языка программирования Python и всех связанных с ним библиотек.	Поддерживает большое количество языков программирования.
Функционал	Большой функционал, так же при необходимости можно загрузить дополнительный функционал.	Базовый функционал, который расширяется при помощи установки библиотек или подключения уже имеющихся.	Базовый функционал, который расширяется при помощи различных плагинов и расширений.

Visual Studio Code распространяется в бесплатном доступе и поддерживается всеми актуальными операционными системами, также эта среда разработки поддерживает все языки программирования. Данная среда разработки – обычный текстовый редактор, с возможностью подключения различных плагинов, в котором возможно работать со всеми языками программирования для разработки различных программных продуктов, в том

Среда разработки PyCharm имеет две версии – Professional и Community. Community версия является бесплатной. Данная версия среды разработки PyCharm работает исключительно с языком программирования Python и всеми связанными с ним библиотеками. Интерфейс PyCharm очень удобен, также имеются функции автодополнения и анализа кода, подсвечивания ошибок, а автоматические рефакторинги помогают эффективно редактировать код.

Sublime Text является кроссплатформенным редактором кода. Имеет базовый функционал, который расширяется при помощи различных плагинов и расширений.

Для разработки чат-бота будут использоваться язык программирования Python и среда разработки PyCharm. Данная среда разработки выбрана в связи с тем, что она полностью поддерживает работу с языком программирования Python и обладает терминалом, необходимым для подключения дополнительных библиотек, необходимых для работы с Telegram-ботами.

1.4 Архитектура программного продукта

Архитектура программного продукта представляет собой описание структуры программного продукта, сюда включаются программные компоненты, их свойства и отношения между ними. Архитектура чат-бота даст примерно понять, какую среду разработки лучше использовать для реализации программного продукта.

Архитектура программного продукта – «Бот, определяющий управляющие компании с контактами» будет включать в себя следующие составные части:

- сервер Telegram;
- сервер приложения;
- клиентская часть (аккаунт бота в Telegram);
- API сайта ГИС ЖКХ.

На рисунке 7 показана связь компонентов программного продукта.

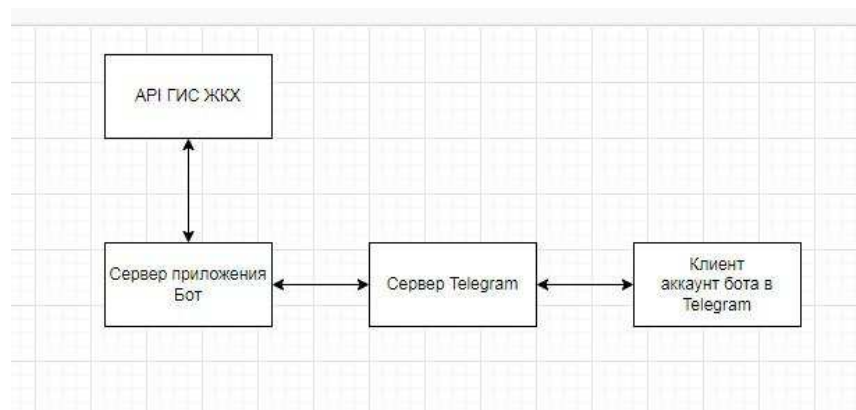


Рисунок 7– Концептуальная архитектура Telegram-бота

Архитектура чат-бота достаточно простая, сервер приложения связывается с сервером Telegram, а тот в свою очередь проверяет аккаунт бота по заданному токену. Для получения информации с сайта ГИС ЖКХ, в этот момент сервер связывается с сайтом, используя его API, и получает данные из формы, далее в аккаунт бота Telegram выводится текстовая надпись.

1.5 Описание диаграмм бизнес-процессов

Перейдём к моделированию диаграммы «Как будет» в нотации IDEF0. На рисунке 8 изображён функциональный блок «Получение информации об организациях, предоставляющих услуги ЖКХ по адресу».

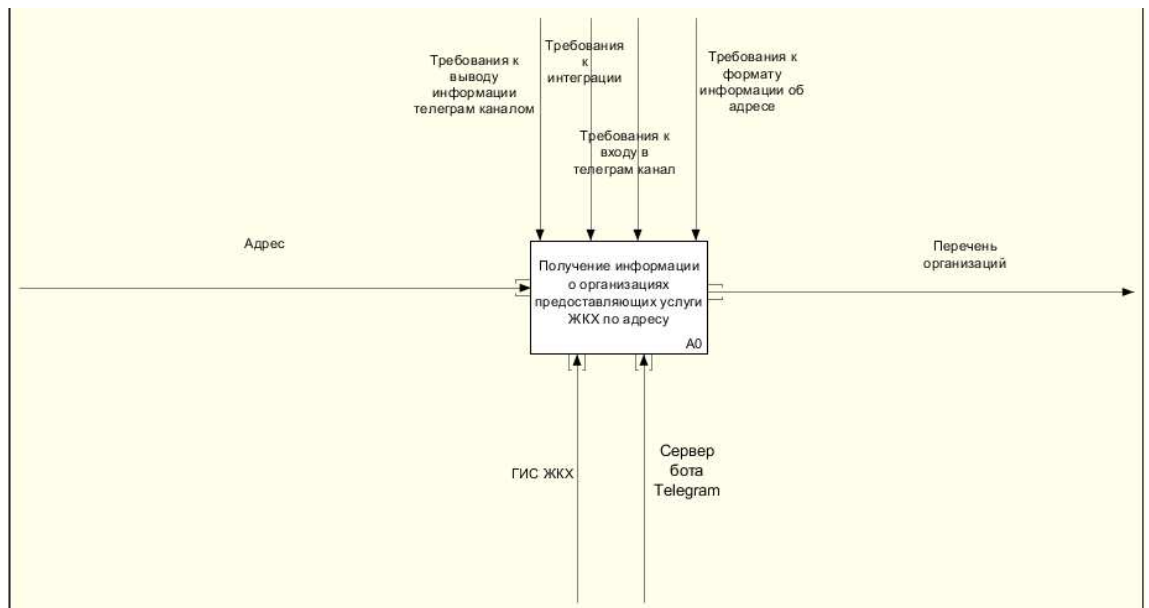


Рисунок 8 – Функциональный блок

На входе имеется адрес, он необходим для поиска организаций в ГИС ЖКХ.

В механизмах имеются сервер бота Telegram и ГИС ЖКХ.

На выходе пользователь получает данные об организациях, предоставляющих услуги ЖКХ.

На рисунке 9 показана декомпозиция функционального блока.

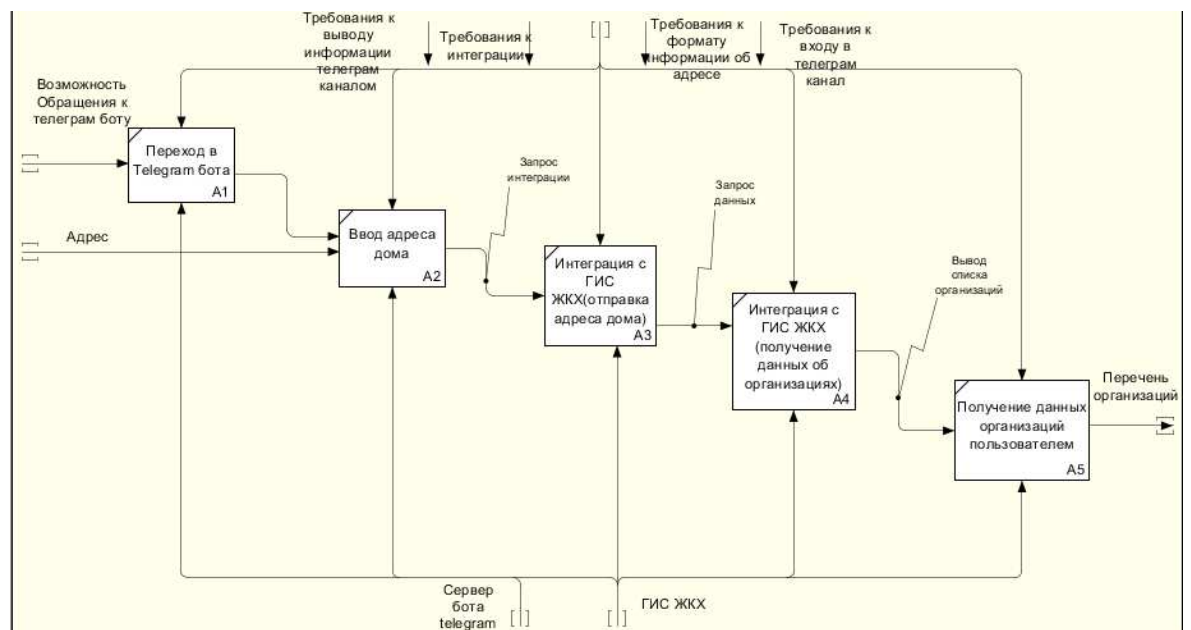


Рисунок 9 – Декомпозиция функционального блока

Разберём и опишем течение информации в пределах системы, то есть поток данных.

На рисунке 10 показан функциональный процесс A0 и 2 сущности.

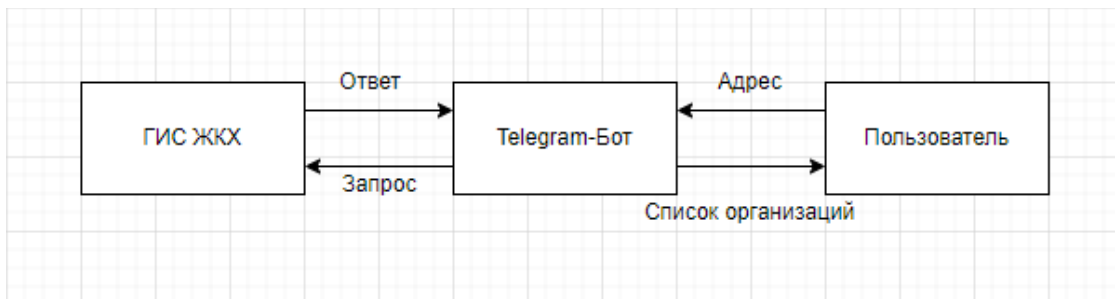


Рисунок 10 – Диаграмма системы «Как будет» в нотации DFD

Внешняя сущность «Пользователь», чтобы получить данные об организациях, должен будет предоставить свой адрес, а внешняя сущность «ГИС ЖКХ» предоставит данные об организациях по запросу.

Далее опишем рисунок 11.

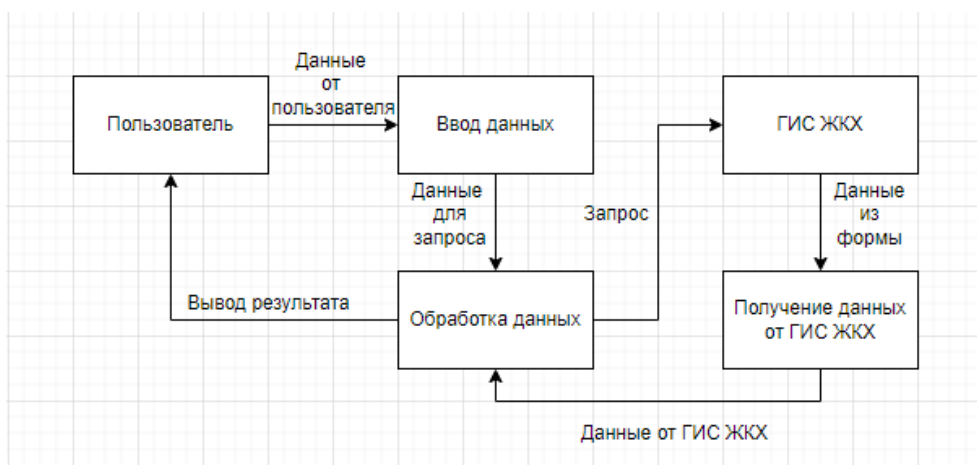


Рисунок 11– Декомпозиция диаграммы системы «Как будет» в нотации DFD

Пользователь передаёт данные для запроса, а именно адрес. Далее данные из формы ГИС ЖКХ обрабатываются и выводятся пользователю.

Пользователь, который хочет получить информацию о

ресурсоснабжающих организациях, заходит в Telegram. Далее находит этот чат-бот. После вводит /start, то есть запускает работу бота.

Бот приветствует пользователя и выдает краткое описание работы своих функций.

После чего человек может продолжить работу с ботом или выйти.

Далее человек в определенной последовательности вводит (выбирает) адрес.

Диаграмма пошаговой работы с системой указана на рисунках 12-13 и представлена в нотации IDEF3.

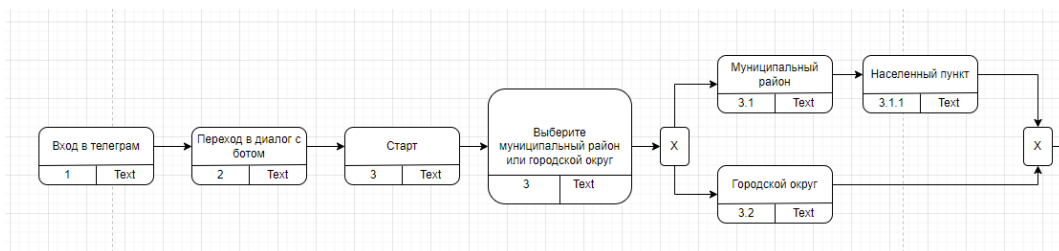


Рисунок 12 – Часть первая диаграммы IDEF3 системы «Как будет»

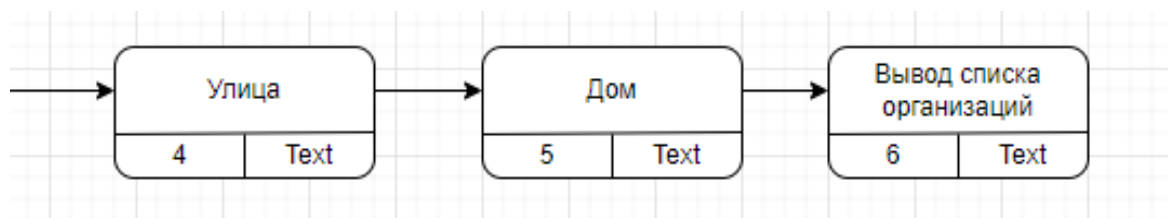


Рисунок 13 – Часть вторая диаграммы IDEF3 системы «Как будет»

1.6 Выводы по разделу «Анализ предметной области»

По результатам аналитического раздела была описана деятельность предприятия – заказчика, обоснована актуальность разработки, проведен анализ аналогов разрабатываемого Telegram-бота. Также были проанализированы и выбраны средства разработки и описаны диаграммы бизнес-процессов.

2 Разработка Telegram-бота поиска управляющих и ресурсоснабжающих организаций

2.1 Установка необходимого ПО

Перед этапом разработки необходимо установить программное обеспечение. Сперва требуется установить язык программирования Python. Для этого переходим на официальный сайт Python во вкладку «Загрузка языка программирования» (рисунок 14) [4]. Выбираем версию Python и нажимаем загрузить. В данном проекте будет использоваться версия языка 3.9.10 от января 2022 года.



Release version	Release date		Click for more
Python 3.10.4	March 24, 2022	Download	Release Notes
Python 3.9.12	March 23, 2022	Download	Release Notes
Python 3.10.3	March 16, 2022	Download	Release Notes
Python 3.9.11	March 16, 2022	Download	Release Notes
Python 3.8.13	March 16, 2022	Download	Release Notes
Python 3.7.13	March 16, 2022	Download	Release Notes
Python 3.9.10	Jan. 14, 2022	Download	Release Notes
Python 3.10.2	Jan. 14, 2022	Download	Release Notes

[View older releases](#)

Рисунок 14 – Версии языка программирования Python

Далее необходимо загрузить среду разработки — PyCharm. Перейдем на сайт JetBrains — международной компании, которая разрабатывает инструменты для разработки на языках Java, Kotlin, C#, F#, C++, Ruby, Python, PHP, JavaScript и многих других [5]. На сайте можно скачать две версии: Professional и Community. В нашем случае необходима разработка только на Python, поэтому скачиваем версию Community (рисунок 15).



Версия: 2022.1
Сборка: 221.5080.212
13 апреля 2022 г.

[Системные требования](#)

[Инструкция по установке](#)

[Другие версии](#)

[Стороннее ПО](#)

Скачать PyCharm

[Windows](#) [macOS](#) [Linux](#)

Professional

Для научной и веб-разработки на Python. Поддерживает HTML, JS и SQL.

Скачать

Доступна бесплатная пробная версия на 30 дней

Community

Для разработки только на Python

Скачать

Бесплатная, на базе открытого исходного кода

Рисунок 15 – Версии PyCharm

После скачивания необходимо начать процесс установки, а далее создать проект, в котором будет разрабатываться Telegram-бот.

Переходим в «File», затем в «NewProject». Проект создан, далее зададим место, в котором будет храниться проект. Какой-либо предварительной настройки перед написанием кода PyCharm не требует. Окно создания проекта представлено на рисунке 16.

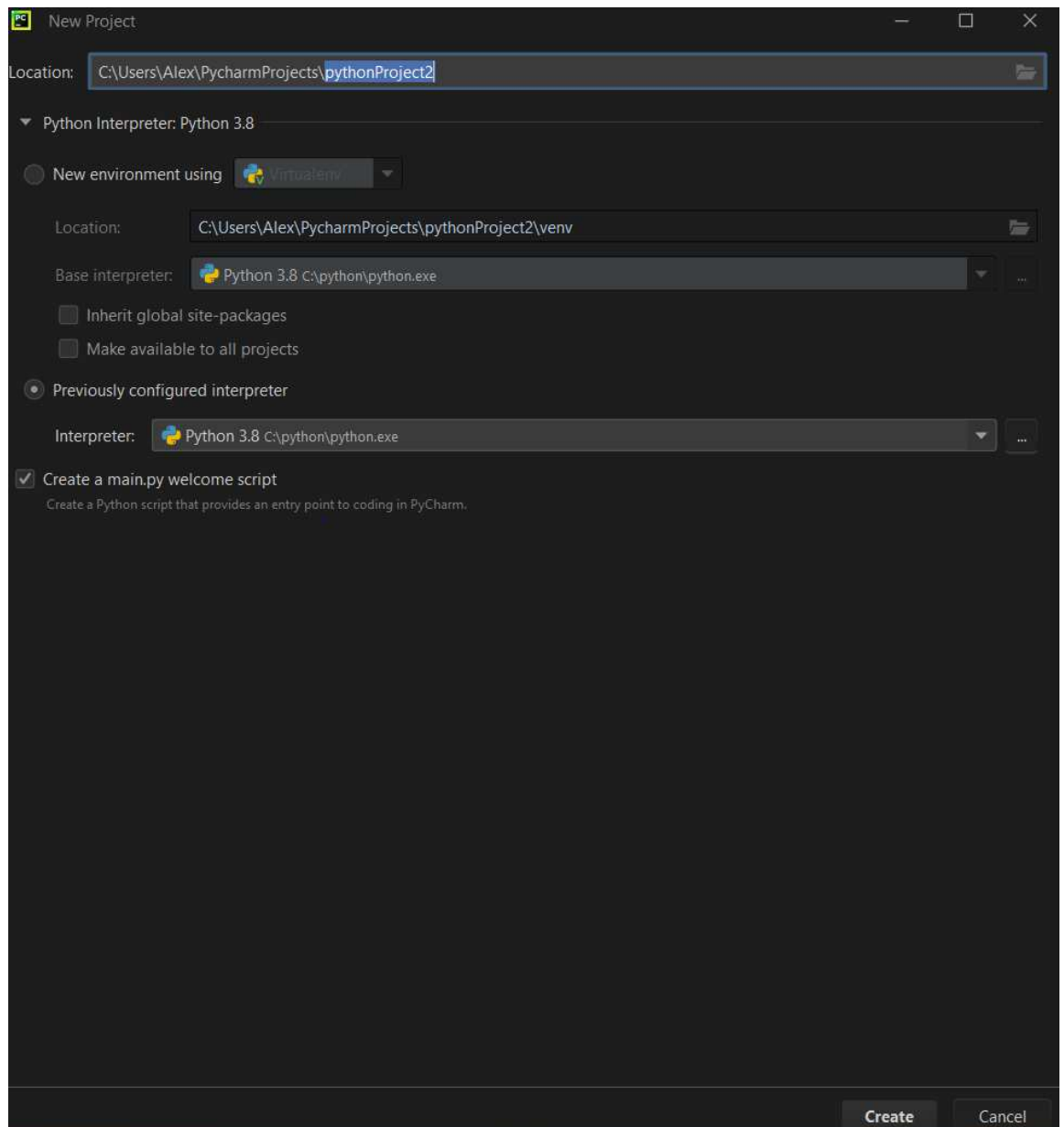


Рисунок 16 – Окно создания проекта

Для начало работы над проектом необходимо установить библиотеки, с которыми нужно будет работать.

2.2 Описание программного кода

Используемые библиотеки представлены на рисунке 17.

```
import json
import requests
import math
import difflib
import numpy
import telebot
from telebot import types
```

Рисунок 17 – Используемые библиотеки

Библиотека telebot нужна для создания бота в Telegram. Модуль json-простой формат обмена данными, в нашем случае используется для работы с json-файлами. Работа с post/getзапросами для получения данных с сайта реализует библиотека requests. Методы get и post используются для отправки данных на сервер. Это необходимо для решения любых задач, связанных с передачей информации от пользователя к серверу и обратно. Math, difflib, numpy – это вспомогательные библиотеки для ускорения работы.

Создадим и импортируем файлы, в которых будут находиться некоторые функции (рисунок 18). В getdata будут находиться функции, отвечающие за интеграцию с сайтом, changewrite и changenext будут отвечать за создание кнопок. Так же необходимо отметить, что создан файл, в котором хранится весь текст, который будет выводить бот.

```
import pyo.getdata as gd
import pyo.change_write as cw
import pyo.change_next as cn
```

Рисунок 18 – Импорт функций

Теперь зарегистрируем бота и получим токен. Токен — это специальный ключ от бота, с помощью которого его можно подключать к сторонним сервисам.

В Telegram предусмотрен отдельный бот-аккаунт под названием «BotFather», в котором регистрируются и настраиваются все боты Telegram.

Для создания нового бота необходимо отправить команду «/newbot» чат-боту «BotFather» (рисунок 19).

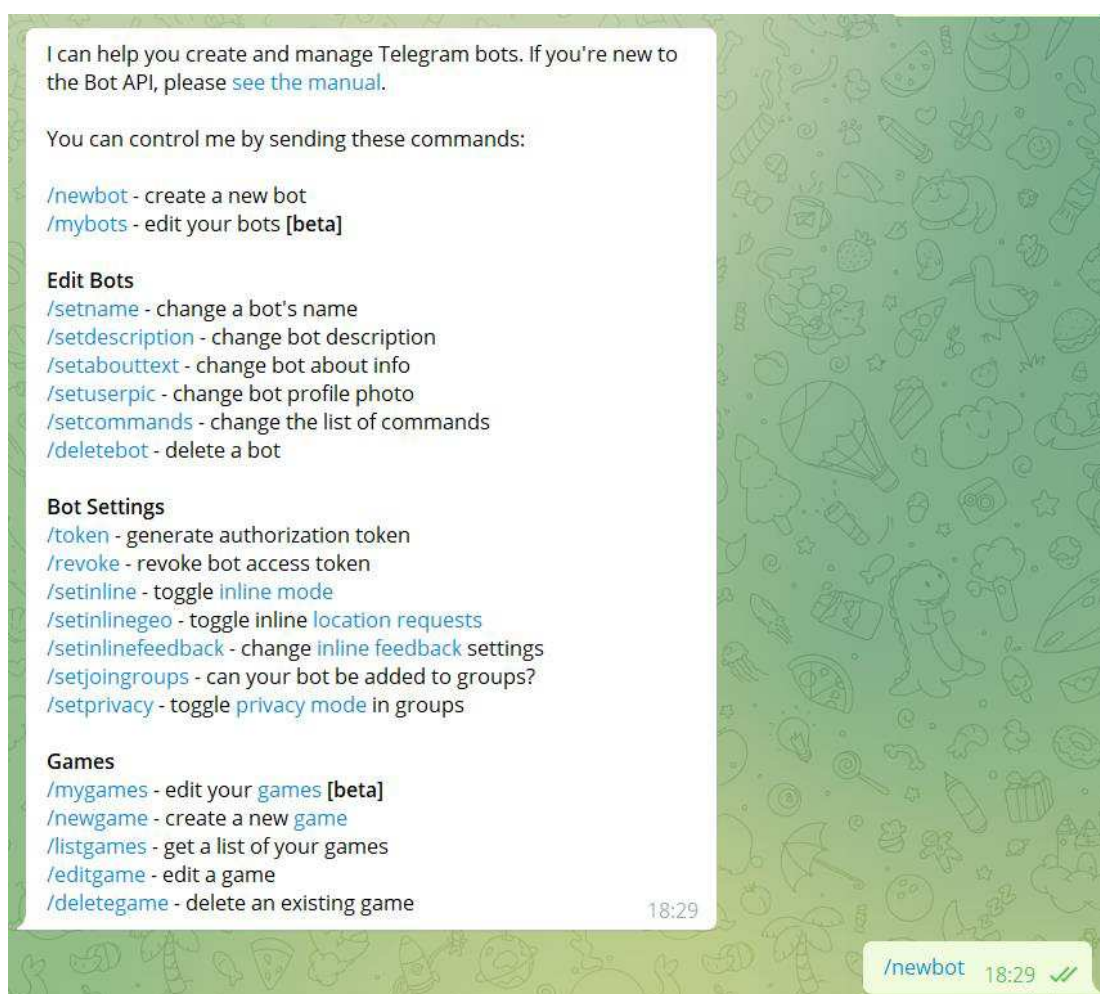


Рисунок 19 – Процесс регистрации чат-бота в «BotFather»

Далее необходимо ввести название Telegram-бота и придумать адрес в Telegram. Далее BotFather выдает токен созданного бота (рисунок 20).

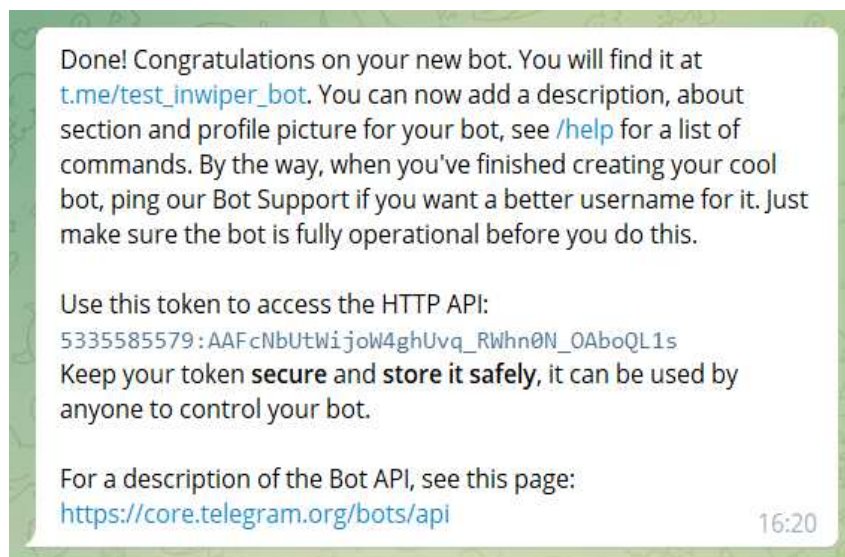


Рисунок 20 – Получение токена

Присвоим боту токен (рисунок 21).

```
bot = telebot.TeleBot('5335585579:AAFcNbUtWijow4ghUvq_RWhn0N_0AboQL1s')
```

Рисунок 21– Присвоение токена

Для настройки параметров, таких как картинка бота, описание бота, а также удаление существующих ботов есть определенные команды. Полный список команд представлен в таблице 3.

Таблица 3 – Доступные команды для настройки Telegram-ботов

Команда	Описание
/mybots	Вывод списка имеющихся ботов
/setname	Изменение описание бота
/setabouttext	Изменение информации о боте

Продолжение таблицы 3

/setuserpic	Изменение фото профиля бота.
/setcommands	Изменение списка команд, поддерживаемых ботом

/deletebot	Удаление бота
------------	---------------

Для того чтобы информация о городах, районах, селах и т.д. генерировалась в кнопки необходим файл, в котором будет храниться информация о них. Иначе это файл аналогичный кэш, который записывается на каждого пользователя отдельно, т.е. при выборе города записывается информация, и выводятся кнопки улиц, затем эта информация о городе и улице посылается на сайт и выводятся кнопки домов. Ниже на рисунке 22 представлено открытие этого файла.

```
with open('file/database_user.json', 'r', encoding='utf-8') as fp:  
    database_user = json.load(fp)
```

Рисунок 22 – Открытие файла database_user

При начале работы с любым ботом необходимо написать /start. Далее напишем код для обработки этой команды (рисунок 23). Также создается запись пользователя для того чтобы сохранять туда кэш.

```
@bot.message_handler(commands=["start"])  
def start(m, res=False):  
    bot.send_message(m.chat.id, 'Напишите "начать" для поиска управляющих и ресурсоснабжающих организаций по адресу')
```

Рисунок 23 – Программный код обработки команды /start

Создадим функцию для обработки команды «Начать». Когда человек вводит эту команду, бот должен выводить список регионов и кнопки «назад» и «далее» для перелистывания страниц с регионами (рисунок 24).

```

@bot.message_handler(content_types=["text"])

def handle_text(message):
    #print(database_user)
    if (str(message.chat.id) in database_user) == False:
        database_user[str(message.chat.id)] = {
            "regions_numb":0, #номер страницы
            "citi_numb":0,
            "adres":""
        }

        #print(database_user)

        with open('file/database_user.json', "w", encoding='utf-8') as file:
            json.dump(database_user, file, indent=4, ensure_ascii=False)

    if message.text == "Начать":
        markup=types.InlineKeyboardMarkup()
        a=regions_list(regions,0)

        for i in a:
            item1=types.InlineKeyboardButton(i[0],callback_data="regionCode " + i[1])
            markup.add(item1)

            item1=types.InlineKeyboardButton("назад",callback_data='regions -')
            item3=types.InlineKeyboardButton("[0/"+str(int((len(regions)/8))+"]",callback_data='chet')
            item2=types.InlineKeyboardButton("далее",callback_data='regions +')

            markup.add(item1,item3,item2)
            markup.add(item_start)

        bot.send_message(message.chat.id, 'Выберите субъект', reply_markup=markup)

```

Рисунок 24 – Функция для обработки команды «Начать»

Переменной «starturl»присвоена ссылка, куда Telegram-бот отправляет запросы (рисунок 25).

```
start_url = 'https://dom.qosusluqi.ru/nsi/api/rest/services/nsi/fias/v4'
```

Рисунок 25 – Переменная «starturl»

Бот каждый раз отправляет get запрос, когда обновляет кэш, на каждом следующем шаге (при выборе) добавляет в исходную ссылку специальные ключи, взятые из прошлого запроса. Эта функция показана на рисунке 26.

```

def get_area(regionCode):
    url = start_url+' /areas?actual=true&itemsPerPage=100&page=1&regionCode='+regionCode+'&searchString='

    s = requests.Session()
    logging = s.get(url)
    return json.loads(logging.text)

```

Рисунок 26 – Get запрос

На рисунке 26 показана функция для регионов, аналогичные функции так же созданы для районов, городов, деревень, улиц и домов.

Теперь напишем функции, которые будут отправлять postзапросы, они так же добавляют в исходную ссылку специальные ключи, взятые из прошлого запроса, но без обновления кэша (рисунок 27).

```
def get_compuni(regionCode,cityCode,streetCode,homeCode):
    url = 'https://dom.gosuslugi.ru/ppa/api/rest/services/ppa/public/organizations/'

    s = requests.Session()

    with open('file/5.json', 'r',encoding='utf-8') as fp:
        datas = json.load(fp)

    datas["regionCode"]=regionCode
    datas["cityCode"]=cityCode
    datas["streetCode"]=streetCode
    datas["houseCode"]=homeCode

    logging = s.post(url,json=datas)
```

Рисунок 27 – Postзапрос

В «file/5.json» хранится шаблон ключей, которые можно найти на сайте ГИС ЖКХ.

После считывания файла с шаблона, происходит преобразование в формат jsonи подставляются новые значения, введенные пользователем.

Приведенный выше программный код — это функция для поселков, для городов создается аналогичная функция.

После этого происходит формирование сообщения на основе полученных данных со списком организаций (рисунок 28).

```

if json.loads(logging.text)["organizationSummaryWithNsilist"] == 0:
    return "нет компаний"
else:
    text = 'Список организаций:'
    g = 0
    for i in json.loads(logging.text)["organizationSummaryWithNsilist"]:
        g = g + 1
        text = text + "\n"+str(g)+" " +i["shortName"]+"\n"
        if i["phone"] != None:
            text = text + "\n" + telephon_number(str(i["phone"]))+"\n"

        if i["url"] != None:
            text = text + "\n" + str(i["url"])+"\n"
        else:
            text = text + "\n"
    return text

```

Рисунок 28 – Программный код формирования сообщения со списком организаций

На рисунке 29 представлена функция, которая берет из кэша данные и создает список, состоящий из имени кнопки и ключа для следующего шага.

```

def regions_list(list,numb):

    a = []

    b = 0
    c = []
    for i in list:

        if b<8:
            c.append([i["offName"] + ' ' + i["shortName"], i["aoGuid"]])
            b=b+1

        else:
            a.append(c)
            c = []
            c.append([i["offName"] + ' ' + i["shortName"], i["aoGuid"]])
            b = 1
    if len(c)!=0:
        a.append(c)

```

Рисунок 29 – Функция-создание списка

Далее создадим функцию, которая отвечает за переход между списками регионов, городов и других элементов. (рисунок 30). «Area» выдает первое меню, «area +» отвечает за кнопку «далее» и «area-» отвечает за кнопку «назад». Если страниц будет больше, то перебрасывает на следующую страницу, если меньше перебросит на первую страницу.

```
def area_next(bot, call, database_user):  
  
    if call.data == "area":  
        database_user[str(call.message.chat.id)]["cashe"] = gd.get_area(database_user[str(call.message.chat.id)]["regions"])  
        database_user[str(call.message.chat.id)]["area_num"] = 0  
        a=area_list(database_user[str(call.message.chat.id)]["cashe"],0)  
        button_gen(database_user,bot, call, area_next_text, a, "area", "area")  
    if call.data == "area +":  
        if int((int((len(database_user[str(call.message.chat.id)]["cashe"])/4))) > (database_user[str(call.message.chat.id)]["area_num"] +1):  
  
            database_user[str(call.message.chat.id)]["area_num"] = database_user[str(call.message.chat.id)]["area_num"] +1  
            a=area_list(database_user[str(call.message.chat.id)]["cashe"],database_user[str(call.message.chat.id)]["area_num"])  
            button_gen(database_user,bot, call, area_next_text, a, "area", "area")  
        else:  
            database_user[str(call.message.chat.id)]["area_num"] = 0  
            markup=types.InlineKeyboardMarkup()  
            a=area_list(database_user[str(call.message.chat.id)]["cashe"],0)  
            button_gen(database_user,bot, call, area_next_text, a, "area", "area")  
    if call.data == "area -":  
        if 0<= (database_user[str(call.message.chat.id)]["area_num"] -1):  
  
            database_user[str(call.message.chat.id)]["area_num"] = database_user[str(call.message.chat.id)]["area_num"] -1  
            markup=types.InlineKeyboardMarkup()  
            a=area_list(database_user[str(call.message.chat.id)]["cashe"],database_user[str(call.message.chat.id)]["area_num"])  
            button_gen(database_user,bot, call, area_next_text, a, "area", "area")  
        else:  
            database_user[str(call.message.chat.id)]["area_num"] = int((len(database_user[str(call.message.chat.id)]["cashe"])/4))  
            markup=types.InlineKeyboardMarkup()  
            a=area_list(database_user[str(call.message.chat.id)]["cashe"],int((len(database_user[str(call.message.chat.id)]["cashe"])/4)))  
            button_gen(database_user,bot, call, area_next_text, a, "area", "area")
```

Рисунок 30 – Функция-переход между списками

Создадим функцию – генератор кнопок (рисунок 31).

Во время нажатия кнопки вызывается функция из «changenext». «Item3» показывает номер страницы, на которой находится пользователь. Количество страниц получаем из кэша и делим их на количество кнопок на странице. «Markup» – это контейнер, в котором хранятся кнопки. Цикл проходит по переменной a (список кнопок), в «item 1» создается кнопка, которая хранит в себе ключ для следующего шага, далее она добавляется в контейнер с кнопками.

```

def button_gen(database_user,bot , call , text_msg ,a ,main_bt ,second_bt):

    get_adres_text(call,database_user)

    markup=types.InlineKeyboardMarkup()

    for i in a:
        item1=types.InlineKeyboardButton(i[0],callback_data=(main_bt+"Code " + i[1]))
        markup.add(item1)

    item1=types.InlineKeyboardButton("назад",callback_data=main_bt+' -')
    item3=types.InlineKeyboardButton(["+str(database_user[str(call.message.chat.id)][second_bt+"_numb"])\
    +"/"+str(int((len(database_user[str(call.message.chat.id)][_cashe])/7)))+"]",callback_data='chet')
    item2=types.InlineKeyboardButton("далее",callback_data=main_bt+' +')
    markup.add(item1,item3,item2)
    markup.add(item_start)
    bot.delete_message(str(call.message.chat.id), call.message.id)
    bot.send_message(str(call.message.chat.id),text_msg,reply_markup=markup)

```

Рисунок 31 – Функция – генератор кнопок, лист 1

```

def button_gen_long(database_user,bot , call , text_msg ,a ,main_bt ,second_bt):

    get_adres_text(call,database_user)

    markup=types.InlineKeyboardMarkup()

    aa = []

    for i in a:
        if len(aa) == 3:
            markup.add(aa[0],aa[1],aa[2])
            aa=[]
        else:
            item1=types.InlineKeyboardButton(i[0],callback_data=(main_bt+"Code " + i[1]))
            #markup.add(item1)
            aa.append(item1)

    # if len(aa) <7:
    #     markup.add(aa)

    item1=types.InlineKeyboardButton("назад",callback_data=main_bt+' -')
    item3=types.InlineKeyboardButton(["+str(database_user[str(call.message.chat.id)][second_bt+"_numb"])\
    +"/"+str(int((len(database_user[str(call.message.chat.id)][_cashe])/7)))+"]",callback_data='chet')
    item2=types.InlineKeyboardButton("далее",callback_data=main_bt+' +')
    markup.add(item1,item3,item2)
    markup.add(item_start)
    bot.delete_message(str(call.message.chat.id), call.message.id)
    bot.send_message(str(call.message.chat.id),text_msg,reply_markup=markup)

```

Рисунок 31, лист 2

Функция, отвечающая за сохранение кэша, представлена на рисунке 32.

```
def areaCode_func(bot, call, database_user):
    database_user[str(call.message.chat.id)]["village_num"] = 0
    database_user[str(call.message.chat.id)]["area"] = call.data.split()[1]
    database_user[str(call.message.chat.id)]["cashe"] = gd.get_vilage(
        database_user[str(call.message.chat.id)]["regions"],
        database_user[str(call.message.chat.id)]["area"]
    )
    a=vilage_list(database_user[str(call.message.chat.id)]["cashe"],0)
    button_gen(database_user,bot, call, areaCode_func_text, a, "village", "area")
```

Рисунок 32 – Функция сохранения кэша

«Villagenumb» записывается в «database_user»,положение страницы (0 это первая страница), в «area» записывается ключ предыдущего действия пользователя. В «cashe»записывается результат от функции, в которую передали прошлые ключи и ключ, который получили только что. В переменную «a»записывается результат от функции, отвечающий за формирование списка кнопок, в которую передается кэш и номер страницы (0 это первая страница). Далее «a»передается в функцию «button_gen», которая создает меню и отправляет его пользователю.

При запуске бота будет выводиться список регионов (рисунок 33, рисунок 34).

```
@bot.callback_query_handler(func=lambda call: True)
def callback_worker(call):
    try:
        if call.data.split()[0] == "regions":
            cn.regions_next(bot, call, database_user, regions)
```

Рисунок 33 – Запуск бота



Рисунок 34 – Вывод списка регионов

Далее при выборе региона необходимо сделать кнопки выбора района и города (рисунок 35). «Item1» – кнопка «Город», «item 2» – кнопка «Район».

```
if call.data.split()[0] == "regionCode":
    database_user[str(call.message.chat.id)]["regions"] = call.data.split()[1]
    markup=types.InlineKeyboardMarkup()
    item1=types.InlineKeyboardButton("Город", callback_data='citi')
    item2=types.InlineKeyboardButton("Район", callback_data='area')
    markup.add(item1,item2)

    bot.delete_message(str(call.message.chat.id), call.message.id)
    bot.send_message(str(call.message.chat.id), 'Выберите город или район', reply_markup=markup)
```

Рисунок 35 – Программный код кнопок «Город» и «Район»

На рисунке 36 представлен интерфейс этих кнопок в Telegram-боте.



Рисунок 36 – Кнопки выбора района и города

При выборе района сделаем кнопки районов, поселков, улиц и домов (рисунок 37).

```
if call.data.split()[0]== "area":
    cn.area_next(bot,call,database_user)

if call.data.split()[0] == 'areaCode':
    cw.areaCode_func(bot, call ,database_user)

if call.data.split()[0]== "vilage":
    cn.vilage_next(bot,call,database_user)

if call.data.split()[0] == 'vilageCode':
    cw.vilageCode_func(bot, call ,database_user)

if call.data.split()[0] == 'settlementCode':
    cw.settlementCode_func(bot, call, database_user)
```

Рисунок 37 – Программный код кнопок районов, поселков, улиц и домов

На рисунках 38, 39, 40 и 41 представлены результаты вывода этих кнопок.

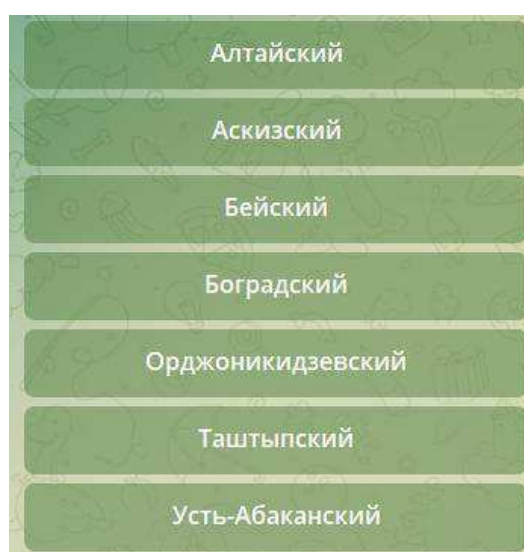


Рисунок 38 – Кнопки районов



Рисунок 39 – Кнопки поселков

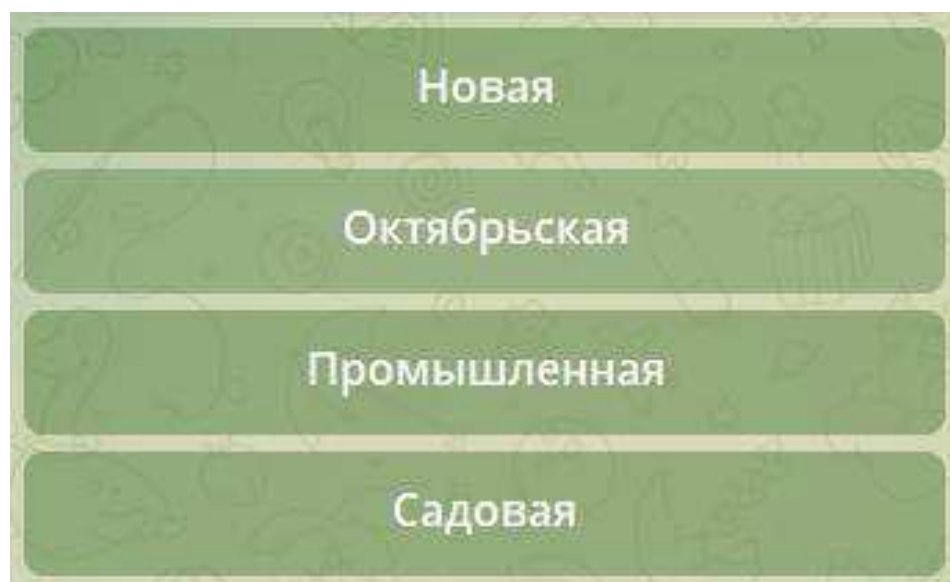


Рисунок 40 – Кнопки улиц поселков

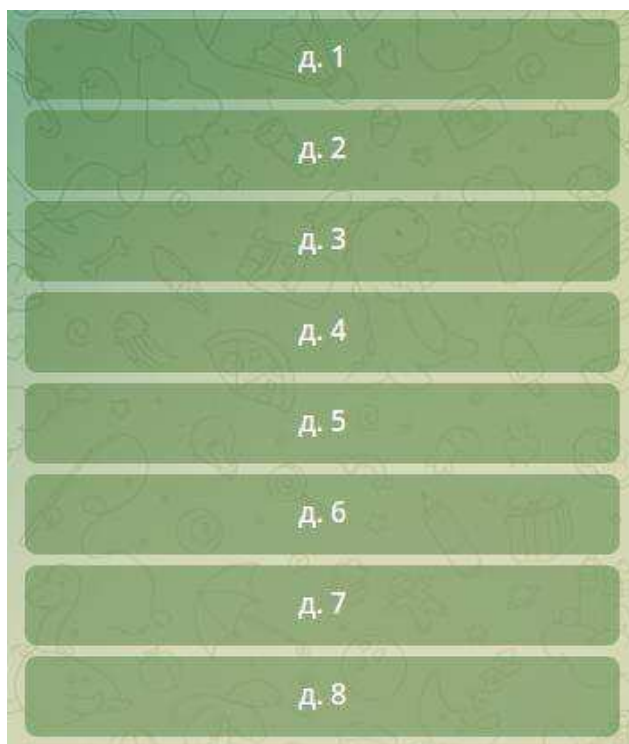


Рисунок 41 – Кнопки номеров домов

Если же человек выбирает город, необходимо выводить кнопки улиц, а далее домов города (рисунок 42).

```
#город-----  
  
if call.data.split()[0]== "citi":  
    cn.citi_next(bot,call,database_user)  
  
#улицы-----  
  
if call.data.split()[0] == 'citiCode':  
    cw.citiCode_func(bot, call,database_user)
```

Рисунок 42 – Программный код вывода кнопок улиц и домов города

На рисунках 43 и 44 представлены результаты вывода этих кнопок.

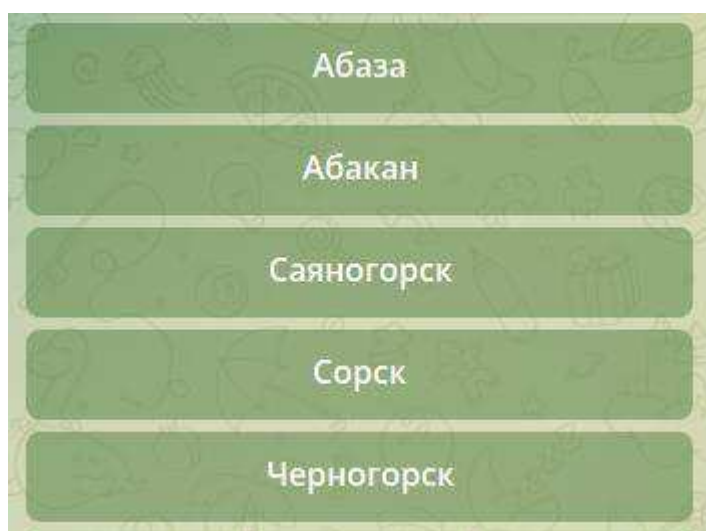


Рисунок 43 – Кнопки городов

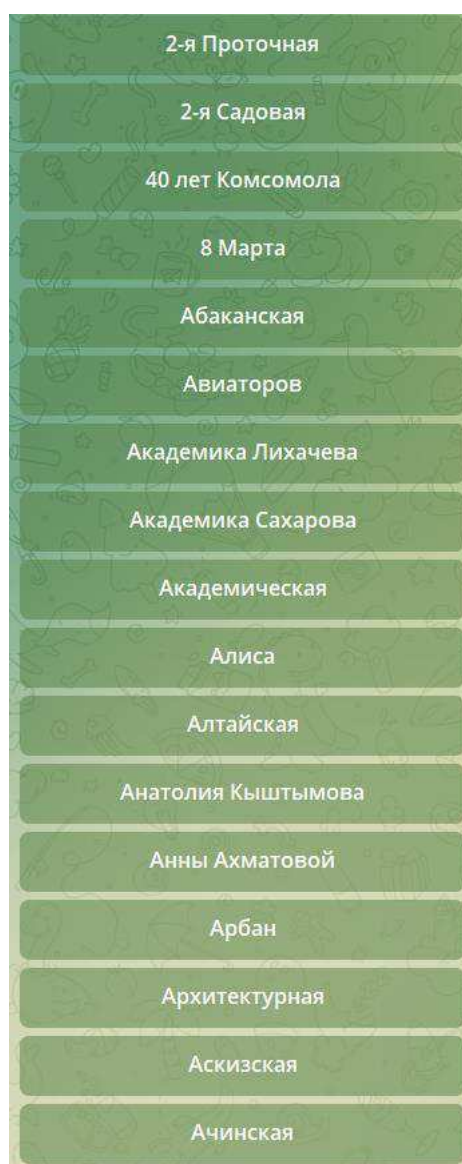


Рисунок 44 – Кнопки улиц города

После того как человек выберет номер дома, Telegram-бот должен переспрашивать пользователя о правильности введенного адреса (рисунок 45). Когда пользователь удостоверился в правильности выбранного адреса, нажимая на кнопку «верно», следующим шагом выводится список организаций.

```
if call.data.split()[0] == 'data':
    markup=types.InlineKeyboardMarkup()
    item1=types.InlineKeyboardButton("верно ?", callback_data="data_text")
    markup.add(item1)
    markup.add(item_start)
    cw.get_adres_text(call, database_user)
    database_user[str(call.message.chat.id)]["home"] = call.data.split()[1]
    text = database_user[str(call.message.chat.id)]["adres"]
    bot.send_message(str(call.message.chat.id), text, reply_markup=markup)

if call.data.split()[0] == 'data_vilage':
    markup=types.InlineKeyboardMarkup()
    item1=types.InlineKeyboardButton("верно ?", callback_data="data_vilage_text")
    markup.add(item1)
    markup.add(item_start)

    cw.get_adres_text(call, database_user)
    database_user[str(call.message.chat.id)]["home"] = call.data.split()[1]
    text = database_user[str(call.message.chat.id)]["adres"]
    bot.send_message(str(call.message.chat.id), text, reply_markup=markup)
```

Рисунок 45 – Вопрос правильности ввода

На рисунке 46 представлена кнопка «верно» в интерфейсе Telegram-бота.



Рисунок 46 – Кнопка «верно»

2.3. Тестирование программного продукта

Теперь запустим бота, на первой странице видно, что выводятся кнопки регионов, где Республика Хакасия занимает первую строку, т.к. основная часть пользователей будут являться жителями Хакасии. Так же отображаются кнопки «назад» и «далее», счетчик и кнопка «в начало» (рисунок 47).



Рисунок 47 – Список регионов

При выборе региона следующим шагом является выбор муниципального или городского округа (рисунок 48).

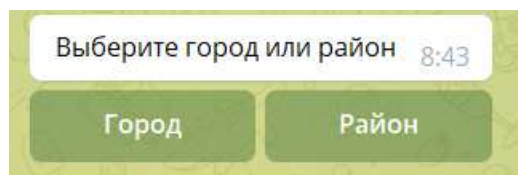


Рисунок 48 – Выбор города или района

Если открыть кнопку города, то, как и должно, выводятся города (рисунок 49). Срайонами аналогично (рисунок 50).



Рисунок 49 – Список городов



Рисунок 50 – Список районов

Следующим шагом после выбора района является выбор поселка, улицы и дома (рисунки 51, 52 и 53).

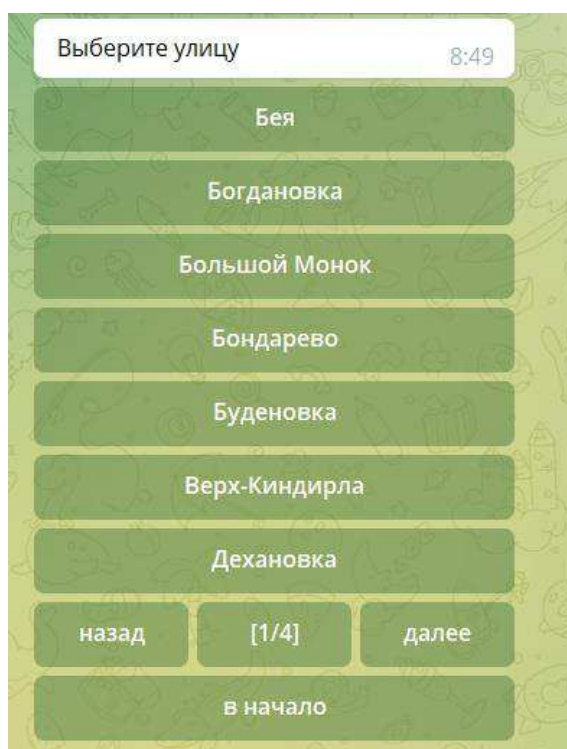


Рисунок 51– Список населенных пунктов

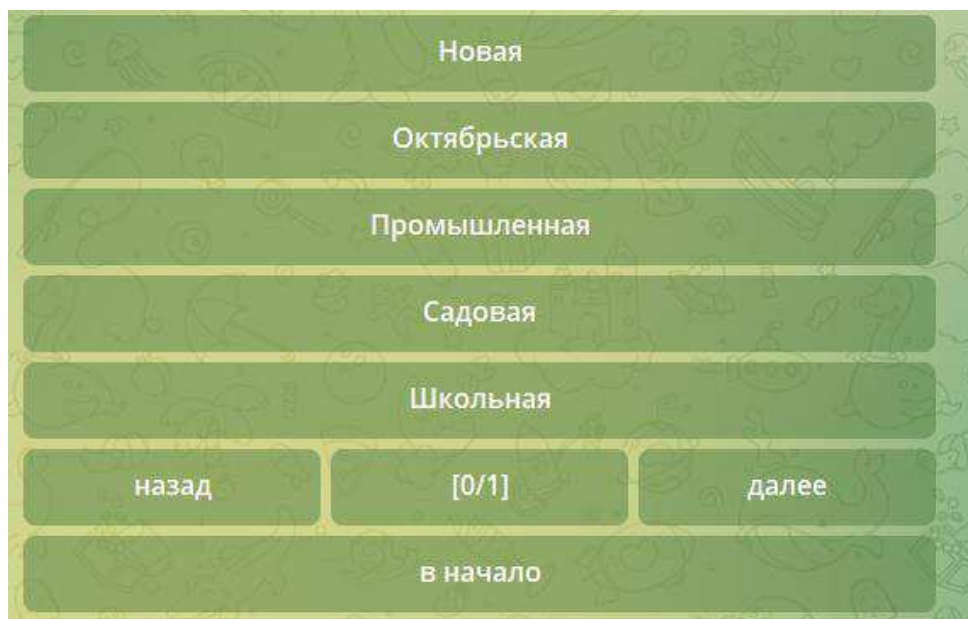


Рисунок 52 – Список улиц

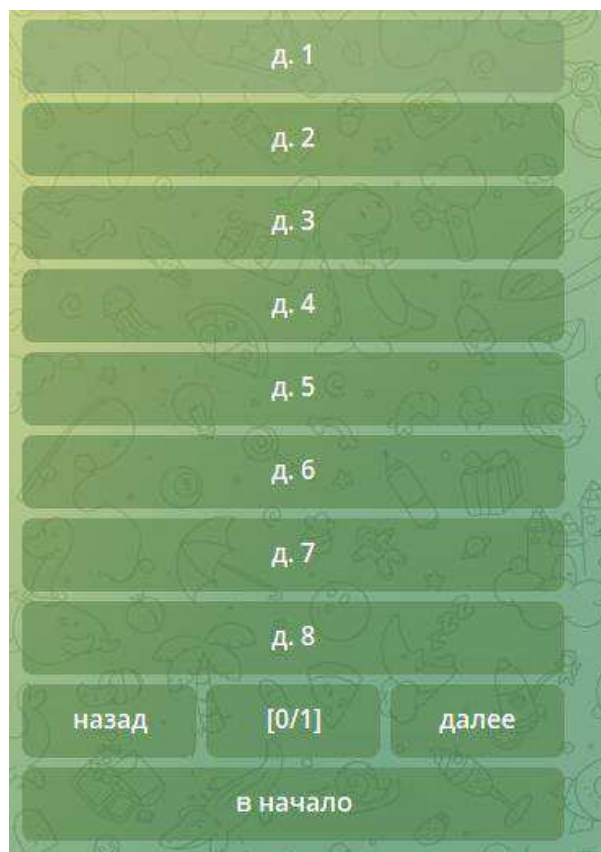


Рисунок 53 – Список номеров домов

При выборе города открываются улицы городов, а далее ситуация аналогичная.

После выбора номера дома Telegram-бот переспрашивает у пользователя верность введенных данных (рисунок 54), а далее выводит список организаций(рисунок 55).

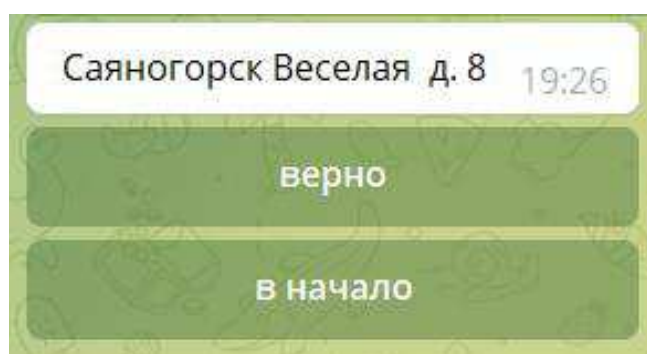


Рисунок 54 – Вопрос правильности введенных данных

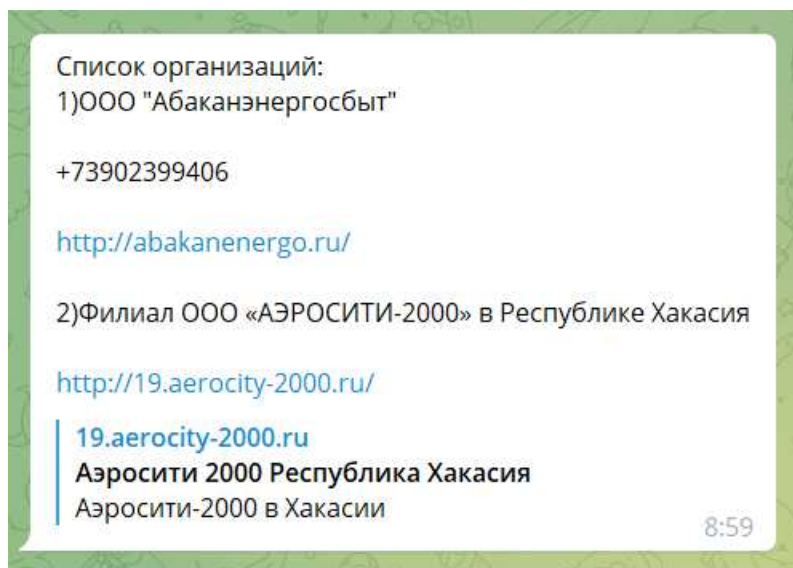


Рисунок 55 – Вывод списка организаций

На данный момент в боте можно выбрать субъект, район, город, улицу и дом с помощью кнопок. Бот же в свою очередь при выборе адреса уже выводит управляющие и ресурсоснабжающие организации.

2.4. Выводы по разделу «Разработка Telegram-бота поиска управляющих и ресурсоснабжающих организаций»

В данном разделе был описан процесс установки языка программирования Python и среды разработки PyCharm, с помощью которого создаются различные программные продукты. Были подключены и описаны библиотекитеlebot, json, requests, math, difflib, numpy.

Разработан Telegram-бот поиска управляющих и ресурсоснабжающих организаций. Интерфейс бота включает в себя кнопки регионов, городов, улиц, населенных пунктов, районов, домов. Произведена интеграция с сайтом ГИС ЖКХ, итогом которой стал вывод списка организаций по выбранному адресу.

3Расчёт затрат и оценка экономической эффективности реализации проекта «Разработка Telegram-бота поиска управляющих и ресурсоснабжающих организаций»

В экономическом разделе рассчитываются затраты на проектирование системы, совокупная стоимость владения системой, оценка рисков реализации проекта.

3.1 Анализ состава и стоимости ресурсов, необходимых для реализации проекта

Для создания программного продукта необходима вычислительная техника, на которой будет производиться разработка программного продукта. Перечень ресурсов и их стоимость представлены в таблице 4.

Таблица 4 – Перечень ресурсов, необходимых на этапах создания

Название оборудования	Количество, шт.	Цена, руб.	Нормативный срок
Монитор HUAWEI AD80HW	1	16299	7 лет
Процессор Intel Core i3-10105F	1	10999	4 года
Видеокарта NVIDIA GeForce GT 730	1	8999	5 лет
Оперативная память AMD Radeon R5 Entertainment Series 8 ГБ	1	3099	7 лет
Блокпитания Corsair CV 650W 80 Plus Bronze	1	4999	8 лет
Клавиатура проводная Logitech K120	1	1399	4 года
Мышь проводная Logitech M90	1	899	1 год

Используя список необходимого оборудования с таблицы, вычислим годовую норму амортизации по формуле

$$H_a = \frac{100\%}{\text{СПИ}}, \quad (1)$$

где СПИ – срок полезного использования.

Срок полезного использования (СПИ) – 5 мес.

Годовая норма амортизации согласно формуле (1): $100\%/5 = 20\%$.

Ежегодная сумма амортизации вычисляется по формуле

$$A_{\text{год}} = C_б * H_a, \quad (2)$$

где $C_б$ – общая стоимость;

H_a – годовая норма амортизации.

Ежегодная амортизация: $46693 * 20\% = 9338,6$ рублей.

Ежемесячная амортизация вычисляется по формуле

$$A_{\text{мес}} = \frac{A_{\text{год}}}{12} \quad (3)$$

Ежемесячная амортизация: $9338,6/12=778$ рублей.

Список персонала, которые будут разрабатывать проект, их срок работы и рабочее время представлены в таблице 5.

Таблица 5 – Персонал

Должность	Срок работы	Дни работы
Программист	1 месяц	18 рабочих дней

Для разработки Telegram-бота необходимо программное обеспечение. Перечень ПО, которое понадобится на этапе создания проекта и его стоимость представлен в таблице 6.

Таблица 6 – ПО на этапе создания

Наименование ПО	Цена, руб.
PyCharmCommunityEdition	бесплатно
SQLiteStudio	бесплатно
MicrosoftWindows 10 Home	11 618

Срок полезного использования – 4 мес.

Годовая норма амортизации, согласно формуле (1): $100\%/4=25\%$.

Ежегодная амортизация, согласно формуле (2): $11\ 618*25\%=2904,5$ рублей.

Ежемесячная амортизация, согласно формуле (3): $2904,5/12=242$ рублей.

Список персонала, которые будут обслуживать проект после внедрения, представлен в таблице 7.

Таблица 7 – Обслуживающий персонал

Квалифицированный персонал	Зарботная плата в месяц, руб.
Инженер-программист	24000

Перечень ПО, которое понадобится на этапе эксплуатации проекта. Список ПО представлен в таблице 8.

Таблица 8 – ПО, необходимое для эксплуатации

Наименование ПО	Цена, руб.
LinuxOS	Бесплатно
PyCharm	Бесплатно

Разработанный чат-бот на этапе внедрения будет размещен на компьютере имеющемся на предприятии в распоряжении инженера программиста, на нем должно быть установлено базовое ПО и интегрированная среда разработки для языка программирования Python – PyCharm.

3.2 Расчет проектных затрат

Затраты на проектирование программного продукта вычисляются по следующей формуле

$$K_{\text{пр}} = K_{\text{зп}} + K_{\text{инс}} + K_{\text{свт}} + K_{\text{проч}}, \quad (4)$$

где $K_{\text{зп}}$ – затраты на заработную плату программиста;

$K_{\text{инс}}$ – затраты на инструментальные программные средства проектирования;

$K_{\text{свт}}$ – затраты на средства вычислительной техники для проектирования;

$K_{\text{проч}}$ – прочие затраты на проектирование.

Минимальный оклад труда в Абакане на момент 2022 года составляет 13890 рублей[7]. Для того чтобы вычислить, сколько программист зарабатывает в день, разделим минимальный оклад на количество рабочих дней в месяце. Затраты на заработную плату включают в себя северную надбавку (для Хакасии 30% от фонда заработной платы), обязательные отчисления во внебюджетные фонды (30,2% от фонда заработной платы), а районный коэффициент не учитывается, т.к. нет опыта работы.

$$K_{\text{зп}} = (13890 * 1,3 * 1,302) / 24 * 18 = 17633 \text{ руб.}$$

Из средств вычислительной техники язык программирования Python распространяется на бесплатной основе. Интегрированная среда разработки для языка программирования Python – PyCharm имеет 2 версии: Professional

(платная версия для научной и веб-разработки на Python.) и Community (бесплатная версия для разработки только на Python). В нашем случае использована бесплатная версия PyCharm Community. Операционная система Microsoft Windows 10 Home стоит 11 618. Значение переменной $K_{ипс}$ будет равнозначно ежемесячной амортизации ПО на этапе создания.

$$K_{ипс} = 242 \text{ руб.}$$

Значение переменной $K_{свт}$ определяется по амортизации компьютера разработчика на период работы, который составляет один месяц.

$$K_{свт} = 778 \text{ рублей.}$$

Для поддержания в рабочем состоянии компьютера и его комплектующих, а также программного обеспечения необходимы дополнительные затраты. Затраты на $K_{проч}$ определяются суммой $K_{зп}$, $K_{ипс}$, $K_{свт}$ от 1% до 5%. Так же в затраты $K_{проч}$ входят: затраты на электричество и интернет, коммунальные услуги за рабочее время программиста. В нашем случае затраты будут составлять 3%.

$$K_{проч} = (K_{зп} + K_{ипс} + K_{свт}) * 3\%$$

$$K_{проч} = (17633 + 242 + 778) * 3\% = 560 \text{ рублей.}$$

$$K_{пр} = K_{зп} + K_{ипс} + K_{свт} + K_{проч}$$

$$K_{пр} = 17633 + 242 + 778 + 560 = 19213 \text{ рублей.}$$

Проектные затраты представлены в таблице 9.

Таблица 9 – Проектные затраты

Наименование затрат	Затраты, руб.
К _{зп}	17633
К _{ипс}	242
К _{свт}	778
К _{проч}	560
К _{пр}	19213

На диаграмме отобразим данные таблицы 9. Диаграмма представлена на рисунке 56.

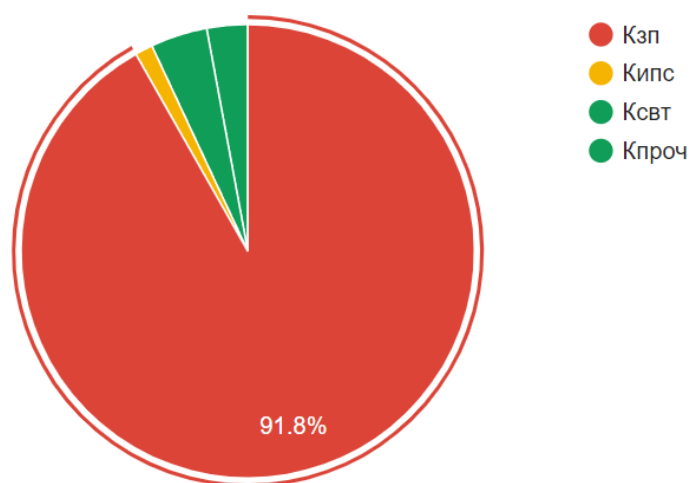


Рисунок 56 – Структура проектных затрат

Согласно данным диаграммы наибольшие проектные затраты составляют

К_{зп} – 91,8%, К_{свт}– 4%, К_{проч} – 2,9%, К_{ипс} – 1,3%.

3.3 Расчет капитальных затрат

Для того чтобы вычислить капитальные затраты необходимо использовать формулу

$$K = K_{\text{пр}} + K_{\text{тс}} + K_{\text{лс}} + K_{\text{по}} + K_{\text{ио}} + K_{\text{об}} + K_{\text{оэ}}, \quad (5)$$

где $K_{\text{пр}}$ – проектные затраты;

$K_{\text{тс}}$ – затраты на технические средства для эксплуатации;

$K_{\text{лс}}$ – затраты на линии связи;

$K_{\text{по}}$ – затраты на программное обеспечение;

$K_{\text{ио}}$ – затраты на формирование информационной базы;

$K_{\text{об}}$ – затраты на обучение персонала;

$K_{\text{оэ}}$ – затраты на опытную эксплуатацию.

Как отмечалось ранее, разработанный Telegram-бот на этапе внедрения будет размещен на компьютере, имеющемся на предприятии в распоряжении инженера-программиста. На компьютере должно быть установлено базовое ПО и интегрированная среда разработки для языка программирования Python – PyCharm. В этой связи $K_{\text{тс}}$ и $K_{\text{по}}$ будут равны нулю.

$$K_{\text{тс}} = 0 \text{ рублей.}$$

$$K_{\text{по}} = 0 \text{ рублей.}$$

Затраты на использование линий связи также будут равны нулю, так как при разработке Telegram-бота будет использоваться локальный компьютер.

$$K_{\text{лс}} = 0 \text{ рублей.}$$

Затраты на программные средства будут равны нулю, так как для внедрения Telegram-бота понадобится только бесплатное ПО.

$$K_{\text{по}} = 0.$$

Затраты на формирование информационной базы отсутствуют в связи с тем, что Telegram-бот не содержит базы данных.

$$K_{\text{но}} = 0 \text{ рублей.}$$

Затраты на обучение персонала равны нулю, потому что для Telegram-бота, сервер которого будет размещен на компьютере предприятия, в этом нет необходимости.

$$K_{\text{об}} = 0 \text{ рублей.}$$

$K_{\text{оз}}$ представляют из себя затраты на заработную плату инженера программиста ЦУР в течении одного рабочего дня. Необходимо вычислить, сколько программист зарабатывает в день, для этого разделим заработную плату на количество рабочих дней в месяце. Указанные затраты в таблице 7 на заработную плату включает в себя северную надбавку и районный коэффициент (для Хакасии по 30% от фонда заработной платы). Для расчета необходимо добавить обязательные отчисления во внебюджетные фонды (30,2% от фонда заработной платы).

$$K_{\text{оз}} = (24000/24) * 1,302 = 1302 \text{ рублей.}$$

В таблице 10 представлены вычисленные затраты.

Таблица 10 – Капитальные затраты

Виды затрат	Затраты, руб.
$K_{\text{пр}}$	19213
$K_{\text{тс}}$	0
$K_{\text{лс}}$	0
$K_{\text{по}}$	0
$K_{\text{но}}$	0
$K_{\text{об}}$	0
$K_{\text{оз}}$	1302

Вычислим капитальные затраты согласно формуле (4).

$$K = 19213 + 0 + 0 + 0 + 0 + 0 + 1302 = 20515 \text{ рублей.}$$

На диаграмме отразим данные из таблицы 10. Диаграмма представлена на рисунке 57.

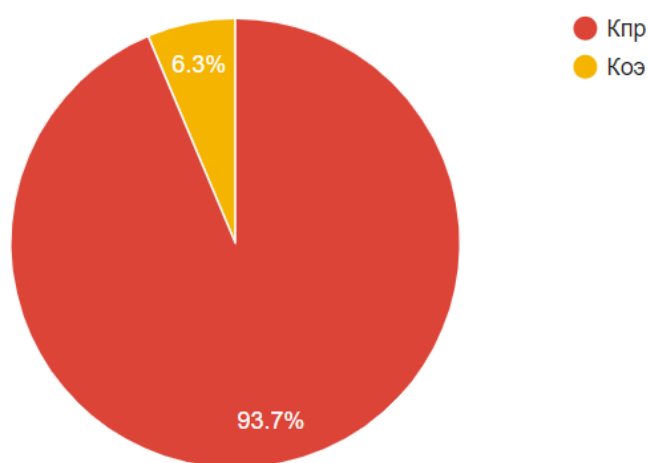


Рисунок 57 –Капитальные затраты

На диаграмме наглядно показано что большую часть затрат составляют $K_{пр}$ – 93,7%. Показатель $K_{оз}$ составляет 6,3%. $K_{тс}$, $K_{лс}$, $K_{по}$, $K_{ио}$, $K_{об}$ не представлены на диаграмме, так как не имеют затрат.

1.4 Расчет эксплуатационных затрат

Telegram-бот после размещения на сервере предприятия не требует дальнейшего взаимодействия. Все данные, которые предоставляет разработанный Telegram-бот, являются продуктом интеграции с сайтом ГИС

ЖКХ. По мере обновления этих данных на сайте, они автоматически будут обновляться и в Telegram-боте, при запросах.

3.5 Расчет совокупной стоимости владения системой

Для расчета затрат на реализацию проекта применяется следующая формула TCO

$$TCO = DE + IC_1 + IC_2, \quad (6)$$

где DE (direct expenses) – прямые расходы;

IC₁ (indirect costs) – косвенные расходы первой группы;

IC₂ (indirect costs) – косвенный расходы второй группы.

Прямые расходы рассчитываются по формуле

$$DE = DE_1 + DE_2 + DE_3 + DE_4 + DE_5 + DE_6 + DE_7 + DE_8, \quad (7)$$

где DE₁ – капитальные затраты;

DE₂ – расходы на управление информационными технологиями;

DE₃ – расходы на техническую поддержку автоматизированного обеспечения и программного обеспечения;

DE₄ – расходы на разработку прикладного программного обеспечения внутренними силами;

DE₅ – расходы на аутсорсинг;

DE₆ – командировочные расходы;

DE₇ – расходы на услуги связи;

DE₈ – другие группы расходов.

Произведем вычисления:

$DE1 = K = 20515$ рублей.

$DE2 = C_{зп} = 0$ рублей.

$DE3 = C_{то}$ и $C_{ао} = 0$ рублей.

$DE4 = 0$ рублей, т.к. разработка дополнительного ПО внутренними силами не потребуется.

$DE5 = 300$ рублей. Данные расходы связанные с необходимостью консультирования со специалистом Министерства строительства и ЖКХ Республики Хакасия по проверке правильности работы Telegram-бота. Данные затраты включают величину заработной платы одного часа специалиста в среднем равную 300 р.

$DE6 = 0$ рублей, т.к. разработка Telegram-бота осуществляется на месте и для этого не требуются командировочные расходы.

$DE7 = 0$ рублей, так как дополнительных расходов на услуги связи не потребуется.

$DE8 = 0$ рублей, т.к. в другую группу расходов входят оставшиеся эксплуатационные расходы, которые в нашем случае не предусмотрены.

$DE = 20515 + 300 = 20815$ рублей.

Теперь вычислим косвенные расходы. В косвенные расходы первой группы может входить риск доработки проекта. В нашем случае разработчик будет исправлять существующие ошибки бесплатно. В косвенные расходы второй группы может входить риск того, что заказчику не понравится готовый программный продукт, в этом случае разработчику будет необходимо доработать его. В случае обнаружения необходимости доработки Telegram-бота заработная плата разработчика за один день будет составлять двойной оклад, т.е. 2604 рубля.

Итого, косвенные расходы первой группы IC_1 равны 0 рублей, косвенные расходы второй группы IC_2 равны 2604 рублей.

$ТСО = 20815 + 0 + 2604 = 22879$ рублей.

3.6 Оценка рисков реализации проекта

Для оценки вероятности успешности программного продукта, следует провести анализ рисков реализации проекта. Для этого оценивается возможность возникновения, какого-либо риска и определение уровня вероятности влияния на проект.

Риски проекта представлены в таблице 11.

Таблица 11 – Риски проекта

№	Группы рисков	Перечень рисков проекта	Уровень влияния риска на проект	Вероятность риска	Возможность предотвращения или снижения риска
1.	Реализационные риски	Проблемы с сайтом ГИС ЖКХ	Высокий уровень	Низкая	Постоянное тестирование на правильность работы и корректность вывода данных.
		Блокировка мессенджера Telegram	Высокий уровень	Низкая	Применение разработки алгоритма для других мессенджеров.

Данный проект низкорискованный, т.к. архитектура разрабатываемого Telegram-бота проста, и состоит из сервера и сайта ГИС ЖКХ. Это значит, что этот программный продукт будет работать непрерывно только при постоянной работоспособности сайта.

Существует вероятность того, что возникнут проблемы с сайтом. Вероятность риска – низкая, т.к. сайт является государственным порталом. Любое изменение сайта или его обновление, связанное с выводом управляющих и ресурсоснабжающих организаций будет влиять и на

работоспособность Telegram-бота. Необходимо отметить, что обычное обновление списка регионов, городов или улиц, а также организаций никак не влияют на работу Telegram-бота, т.к. данные с помощью интеграции с сайтом обновляются автоматически. Для снижения данного риска необходимо проводить постоянное тестирование на правильность работы и корректность вывода данных.

Второй и последний риск – блокировка мессенджера Telegram на территории Российской Федерации. Данный риск имеет низкую вероятность т.к. в истории этого мессенджера уже были подобные случаи, когда он был на грани блокировки, но на сегодняшний день у разработчиков и модераторов Telegram есть определенные договоренности с контролирующими органами в IT-сфере. К тому же на данный момент мессенджер Telegram полностью соблюдает российское законодательство, и постоянно совершенствует его работу. Если данный риск все же окажется возможным, то он будет иметь высокий уровень влияния на проект. В данном случае будет только один выход – применение разработки алгоритма для других мессенджеров. Все же в короткие сроки сделать аналогичный бот для другого мессенджера не удастся, т.к. каждый мессенджер уникален и имеет разные возможности и функции для создания ботов.

3.7 Экономическая эффективность реализации проекта

Проект «Telegram-бот поиска управляющих и ресурсоснабжающих организаций» создан для информирования граждан республики Хакасия, а именно получения списка таких организаций и контактных данных о них.

Данная информация предоставляется бесплатно, соответственно прямого экономического эффекта от внедрения проекта не предусмотрено. Косвенная эффективность может быть обусловлена снижением количества обращений граждан в органы власти за информацией об управляющих и ресурсоснабжающих организациях. Тем самым, гражданин экономит свое

время и время сотрудников государственных учреждений. Автоматизировать полностью вопрос информирования граждан об обслуживающих организациях данный проект не предусматривает. Соответственно рассчитать эффективность методом оценки автоматизации (сравнения базового и разработанного процессов) не представляется возможным.

Все же, хотелось отметить, что затраты на создание проекта оправданы растущим спросом на цифровизацию регионов России. Создавая и внедряя подобные проекты, заказчик – Центр управления регионом республики Хакасия, может планировать более глобальные разработки и претендовать на получения финансирования в рамках реализации государственных проектов области цифровой экономики.

3.8 Выводы по разделу «Расчёт затрат и оценка экономической эффективности реализации проекта «Разработка Telegram-бота для информирования о управляющих и ресурсоснабжающих организаций»»

Капитальные затраты на создание Telegram-бота поиска управляющих и ресурсоснабжающих организаций равны 20515 рублей, что является средней стоимостью создания Telegram-ботов такого уровня. Эксплуатационные затраты для данного проекта не предусмотрены.

Совокупная стоимость владения равна 22879 рублей. Этот показатель является полной стоимостью Telegram-бота для заказчика с самого начала разработки данного программного продукта.

Необходимо отметить что информация, предоставляемая ботом, является бесплатной, соответственно прямого экономического эффекта от внедрения проекта нет. Косвенная эффективность может быть обусловлена снижением количества обращений граждан в органы власти за информацией об управляющих и ресурсоснабжающих организациях.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной поставленная цель была достигнута.

По результатам аналитического раздела была обоснована актуальность, проанализированы аналоги созданного Telegram-бота.

Выполнен выбор и обоснование средств разработки.

По результатам проектного раздела был разработан Telegram-бот и описаны этапы его создания, а именно зарегистрирован Telegram-бот, получен токен, созданы и описаны его основные функции, произведена интеграция с сайтом с помощью geti postзапросов.

В экономическом разделе были рассчитаны капитальные, и проектные затраты, а также был проведен расчет экономической эффективности проекта. Также был проведен расчет рисков реализации проекта, описаны мероприятия по снижению вероятности рисков.

В конечном итоге получили программный продукт, который будет полезен жителям Республики Хакасия в плане предоставления информации об управляющих и ресурсоснабжающих организациях. Telegram-бот будет размещен на сервере ЦУР Хакасии для непрерывной работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ПОСТАНОВЛЕНИЕ №120-п. О создании и функционировании Центра управления регионом Республики Хакасия: национальный стандарт Российской Федерации: издание официальное: утверждено постановлением Президиума Правительства Республики Хакасия: введен впервые: дата введения 04.09.2020 / разработан Государственным комитетом СССР по стандартам, Министерством приборостроения, средств автоматизации и систем управления СССР. – Москва: Стандартиформ, 2009. – V, 43, [1] с.

2. SharkDevelop: разработка мобильных приложений: [сайт]. – Новосибирск, 2013. – URL: <https://sharkdevelop.com/boty-v-telegram> (дата обращения: 20.05.2022).

3. Wibe: Агентство стратегического маркетинга: [сайт]. – NewYork, 2020. – URL: <https://wibe.ua/sozдание-bota-v-telegram/> (дата обращения: 20.05.2022).

4. Python: Python Software Foundation [сайт]. – Delaware, 2022. – URL: <https://www.python.org/downloads/> (дата обращения: 20.05.2022).

5. JetBrains: Производитель программного обеспечения [сайт]. – Praha, 2022. – URL: <https://www.jetbrains.com/ruru/pycharm/download/#section=windows> (дата обращения: 20.05.2022).

6. ГИС ЖКХ: Государственная информационная система жилищно-коммунального хозяйства [сайт]. – Москва, 2022. – URL: <https://dom.gosuslugi.ru> (дата обращения: 20.05.2022).

7. Роструд : Государственная инспекция труда в Республике Хакасия : официальный сайт. – 2022. – URL: <https://git19.rostrud.gov.ru/news/928937.html#:~:text=Новый%20МРОТ%20с%201%20января,он%20составляет%2020%20467%20рублей> (дата обращения: 20.05.2022).

8. Консультант Плюс : НАЛОГОВЫЙ КОДЕКС РОССИЙСКОЙ ФЕДЕРАЦИИ : официальный сайт. – 2022. – URL:http://www.consultant.ru/document/cons_doc_LAW_19671/ (дата обращения: 29.03.2022).

9. СТУ 7.5–07–2021 СТАНДАРТ УНИВЕРСИТЕТА : Система менеджмента качества : Общие требования к построению, изложению и оформлению документов учебной деятельности : сайт / Сибирский Федеральный Университет. – Красноярск : СФУ, 2021 – . – URL: <https://about.sfu-kras.ru/docs/8127/pdf/808588> (дата обращения: 05.04.2022).

10. Выполнение и защита выпускной квалификационной работы по направлению 09.03.03 «Прикладная информатика» [Электронный ресурс] : метод. указания / сост. Е. Н. Скурятенко, В. И. Кокова, И. В. Янченко ; Сиб. федер. ун-т, ХТИ – филиал СФУ. – Абакан : ХТИ – филиал СФУ, 2017. 78 с.

11. ГОСТ Р ИСО 9001-2001 Системы менеджмента качества. Требования. Введ. 31.08.2001


Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ

Заведующий кафедрой


Е. Н. Скуратенко

подпись

« 17 » июня 2022 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

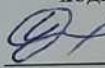
Разработка Telegram-бота для информирования о управляющих и
ресурсоснабжающих организациях

Руководитель


17.06.22
подпись, дата

доцент, канд. техн. наук Е. Н. Скуратенко

Выпускник


17.06.22
подпись, дата

А. А. Чаптыков


Консультанты
по разделам:

Экономический


17.06.22
подпись, дата

Е. Н. Скуратенко

Нормоконтролер


17.06.22
подпись, дата

В. И. Кокова

Абакан 2022