

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Цифровая трансформация бизнеса

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка чат-бота Telegram для взаимодействия и информационной поддержки клиентов расчетно-кассового центра ресурсоснабжающей организации»

Обучающийся

Р.И. Алексеенко

(Инициалы Фамилия)

(личная подпись)

Руководитель

д-р социол. наук, доцент Е.В Желнина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

В выпускной квалификационной работе рассмотрен процесс разработки и внедрения Telegram-бота для расчетно-кассового центра «РКЦ-Эксперт» с целью автоматизации приема показаний приборов учета и улучшения взаимодействия с клиентами. В условиях высокой нагрузки на операторов и низкой эффективности классических способов коммуникации, было обосновано применение современных цифровых решений.

В работе проведен анализ предметной области, выявлены основные проблемы текущей модели бизнес-процессов и рассмотрены аналоги существующих решений. На основании полученных данных предложено автоматизированное решение на базе Telegram-бота, реализованное с использованием языка программирования Python и Google Таблиц для хранения данных. Описана архитектура системы, логика взаимодействия с пользователем, алгоритмы обработки и валидации данных.

Проект прошёл этапы разработки, тестирования и внедрения. Также проведена оценка экономической эффективности.

Работа состоит из 46 страниц печатного текста, 29 рисунков, 6 таблиц и 22 источника информации.

Оглавление

Введение	4
Глава 1 Анализ деятельности Расчетно-кассового центра	6
1.1 Описание предприятия и его организационной структуры	6
1.2 Организационная структура предприятия.....	7
1.3 Моделирование бизнес-процесса обработка показаний приборов учета.....	8
1.4 Сравнительный анализ аналогов решений по сбору показаний	10
1.5 Преимущества автоматизации.....	14
Глава 2: Проектирование программы.....	17
2.1 Обзор платформ для создания чат-ботов	17
2.2 Хранение данных	19
2.3 Сравнительный анализ языков программирования.....	20
2.4 Техническое задание и реализация интерфейса программного средства	23
Глава 3: Физическое проектирование Telegram-бота по приему показаний	27
3.1 Получение токена Телеграм и создание Google таблицы.....	27
3.2 Написание и запуск кода.....	29
3.3 Перспективы развития.....	39
3.4 Методика расчета экономической эффективности	40
3.5 Расчет экономической эффективности проекта	41
Заключение	43
Список используемой литературы и источников	45

Введение

В современном мире цифровые бизнес-процессы ресурсоснабжающих организаций сталкиваются с необходимостью автоматизации взаимодействия с клиентами [3]. Одной из основных задач расчетно-кассового центра (РКЦ) является выставление счетов на оплату, прием показаний приборов учета, консультирование потребителей и обработка их обращений. Классические методы взаимодействия, такие как телефонные звонки и личные посещения, часто приводят к перегрузке сотрудников, увеличивают время обработки запросов и создают неудобства для клиентов. В часы пик операторы не успевают отвечать на телефонные звонки, что приводит к образованию очередей и негативным отзывам клиентов. Среднее время ожидания ответа на телефонный звонок составляет 10 минут, а количество необработанных обращений в день достигает 345. Использование чат-ботов в мессенджерах, таких как Telegram, Вконтакте, позволяет решить эти проблемы за счет автоматизации взаимодействия, снижения нагрузки на сотрудников и повышения лояльности потребителей [7], [8], [9], [22]. Чат-бот способен работать круглосуточно, принимая обращения и показания счетчиков. Кроме того, он может предоставлять информацию о задолженности, принимать заявки на техническое обслуживание и отвечать на часто задаваемые вопросы.

Из исследований были выявлены ключевые показатели, которые демонстрируют эффективность внедрения ботов в сфере ЖКХ. – Исследование компании HeadWork Analytics показало, что автоматизация обработки типовых обращений позволяет снизить нагрузку на операторов более чем на 40%. [20]. Согласно отчету РБК, до 50% управляющих компаний рассматривают внедрение чат-ботов как приоритетную задачу на ближайшие два года. [13]. Внедрение чат-ботов позволило сократить время обработки заявок на 50%, согласно отчету компании «Стек» [11].goog

Таким образом, разработка и внедрение Telegram-бота для расчетно-кассового центра «РКЦ-Эксперт» является актуальной задачей, способствующей повышению эффективности работы организации и улучшению клиентского сервиса.

Цель работы – разработка и внедрение чат-бота Telegram для автоматизированного приема показаний приборов учета и предоставления информации клиентам расчетно-кассового центра «РКЦ-Эксперт».

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области и бизнес-процессов РКЦ, включая процессы приема показаний;
- определить функциональные требования к чат-боту;
- разработать архитектуру и логику работы чат-бота, обеспечивающую масштабируемость и надежность;
- разработать удобный и интуитивно понятный пользовательский интерфейс чат-бота;
- реализовать интеграцию чат-бота с Google Таблицами хранения данных;
- провести тестирование и оценку работоспособности системы, интеграционное тестирование и пользовательское тестирование;
- рассчитать экономическую эффективность внедрения чат-бота.

В работе используются следующие методы:

- анализ предметной области и изучение текущих бизнес-процессов РКЦ и существующих проблем;
- моделирование бизнес-процессов, построение диаграмм «как есть» и «как должно быть»;
- программная разработка;
- проектирование архитектуры чат-бота и его интеграция с Google Таблицами с использованием Python [1].
- экономический анализ и расчет показателей эффективности внедрения.

Глава 1 Анализ деятельности Расчетно-кассового центра

1.1 Описание предприятия и его организационной структуры

ООО «РКЦ-Эксперт» – это компания, специализирующаяся на предоставлении услуги по расчету и начислению платежей за жилищно-коммунальные услуги для населения. Партнерами являются крупные организации, такие как ООО «РКС» (обеспечивающее водоснабжение и водоотведение) и ряд управляющих компаний («МУЖКХ», «Время Перемен» и другие).

Основные услуги включают:

Расчет платы за ЖКУ: Начисление платежей за воду, водоотведение, отопление и другие коммунальные услуги. Расчеты производятся на основе показаний приборов учета и нормативов потребления.

- Прием и обработка показаний: Показания ИПУ принимаются лично в офисах и по телефону. Ограниченное количество телефонных линий создает очереди и нагрузку на персонал. Ввод данных осуществляется вручную в информационную систему;
- формирование и печать квитанций: Квитанции формируются, распечатываются и доставляются абонентам;
- сверка данных с поставщиками ресурсов: Компания получает данные о поставленных ресурсах от РКС и сверяет их с данными о начисленных платежах, составляют платежи и исключают расхождения;
- работа с должниками: проводится оповещение абонентов о задолженности (по телефону, почте), а также формирование списков и взаимодействие с юридическими службами;
- консультации абонентов: Предоставление консультаций абонентам по вопросам начислений, задолженности и другим вопросам, связанным с коммунальными платежами.

Основное ПО – 1С Предприятие, которая автоматизирует ключевые процессы, связанные с расчетами, документооборотом и работой с клиентской базой. Система включает в себя модули:

- модуль расчета начислений: обеспечивает автоматизированный расчет начислений на основании различных условий, таких как тарифы, объемы потребления, индивидуальные параметры абонента.
- модуль расчета начислений: автоматически формирует квитанции для оплаты. Квитанции генерируются в печатные формы, электронные документы и отправлены абонентам различными способами: курьерами в почтовые ящики, электронной почте или личный кабинет на сайте.
- модуль составления отчетности: широкие возможности для формирования различных отчетов, необходимых для анализа деятельности компании и контроля финансовых показателей.

Компания имеет 4 офиса по г. Тольятти с рабочими местами для сотрудников, общая численность организации 11 человек. Распределение персонала: на обслуживании у РКЦ более 100 тысяч абонентов, сотрудники заняты в основном приемом показаний и расчетами. Несколько человек занимаются консультированием и работой с должниками. Компания планирует расширить клиентскую базу (заключить договоры с новыми управляющими организациями, ресурсоснабжающими организациями) и автоматизировать бизнес-процессы.

1.2 Организационная структура предприятия

Организационная структура РКЦ включает следующие подразделения:

Отдел по работе с населением – консультирует клиентов по начислениям, сбору и внесение показаний, проверок счетчиков и смену

лицевых счетов в Билингвой системе компании. Состоит из операторов call-центра, специалисты по документообороту и приему населения.

Отдел информационных технологий – отвечает за автоматизацию процессов, поддержку базы данных и работу онлайн-сервисов.

Финансовый отдел – включает в себя главного бухгалтера, кассира и специалистов по расчетам и контролю платежей, ведут финансовый учет.

На рисунке 1 представлена организационная структура РКЦ, включающая основные подразделения и распределение функций между ними.

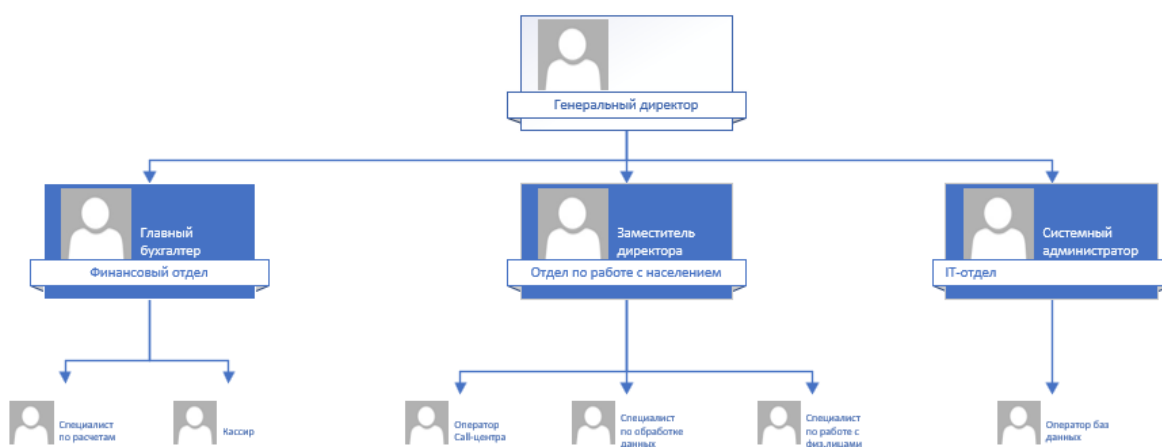


Рисунок 1 - Схема с подразделениями ООО «РКЦ-Эксперт»

1.3 Моделирование бизнес-процесса обработка показаний приборов учета

Функциональная модель «как есть» (AS-IS) предусматривает [10], [19] взаимодействие клиентов с РКЦ происходит следующими способами:

- лично в офисе: клиент приходит в РКЦ, передает показания, получает консультацию, по телефону – операторы вручную фиксируют показания и отвечают на вопросы.
- через электронную почту – клиенты отправляют показания и запросы, которые обрабатываются вручную

Контекстная диаграмма процесса приема показаний на рисунке 2.

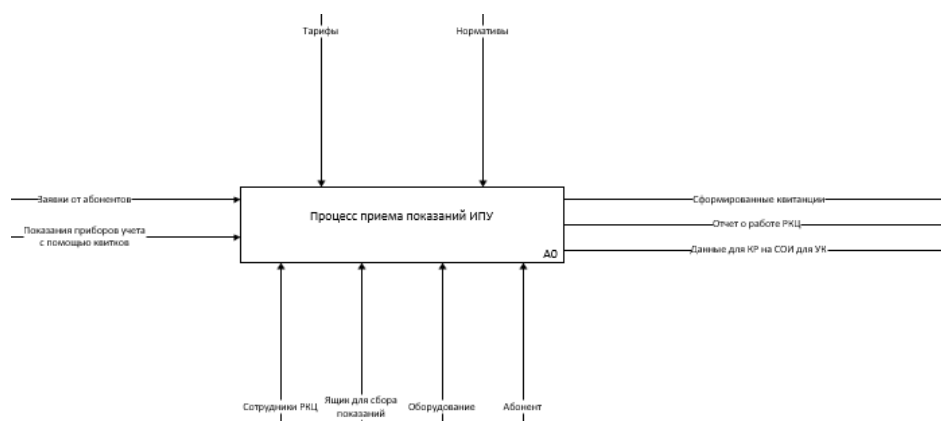


Рисунок 2 – Контекстная диаграмма процесса.

Декомпозиция модели процесса приема показаний на текущем этапе автоматизации на рисунке 3.

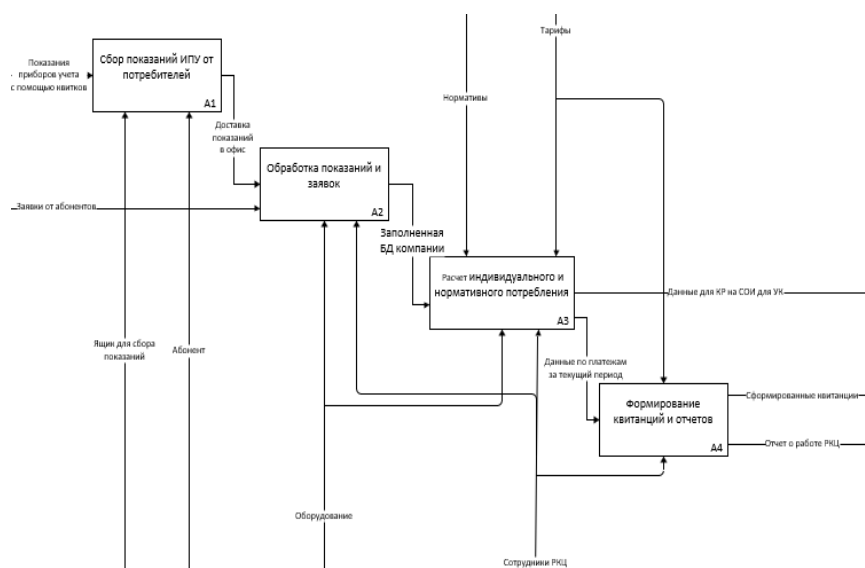


Рисунок 3 - Декомпозиция процесса приема показаний «как есть»

- Процесс A1 «Сбор показаний ИПУ от потребителей» подразумевает, что потребитель заполняет свои данные в отрывном купоне на квитанции и относит в специализированные ящики для приема

показаний развешенные по городу. В конце приема показаний, курьеры собирают показания и развозят их в офисы компании.

- Процесс А2 «Обработка показаний и заявок» сотрудники РКЦ заносят показания в билингвную систему компании для дальнейшего расчета.
- Процесс А3 «Расчет индивидуального и нормативного потребления» происходит расчет всех потребителей по индивидуальным приборам учета, в случае отсутствия показаний расчет идет по среднему потреблению за последние 6 месяцев, а затем по нормативу. Если у потребителя не установлены счетчики, расчет идет относительно норматива, который устанавливается индивидуально для каждого дома.
- Процесс А4 «Формирование квитанций и отчетов» происходит массовый расчет всех клиентов компании, формируются квитанции и направляются потребителям.

Основными недостатками текущей модели являются:

- высокая нагрузка на операторов;
- длинное время ожидания в очереди или на телефоне;
- ошибки при ручном вводе данных;
- неудобство для клиентов, которым приходится тратить время на походы в офис или звонки.

Таким образом процесс приема показаний является самым проблемным и неавтоматизированным бизнес-процессом компании

1.4 Сравнительный анализ аналогов решений по сбору показаний

Рассмотрим преимущества и недостатки аналогов [2], доступные потребителей для передачи показаний счетчиков, определим наиболее эффективные и удобные решения.

«СамараЭнерго» – Телеграм бот с позволяет передавать показания по счетчикам, оплачивать квитанции, отслеживать задолженность и следить за историей показаний.

Бота можно найти по нику @SamaraEnergoBot, сервис по приему показаний электроэнергии. Весь функционал доступен после подтверждения ФИО Нанимателя. Пример интерфейса Telegram-бота «СамараЭнерго», используемого для передачи показаний на рисунке 4.

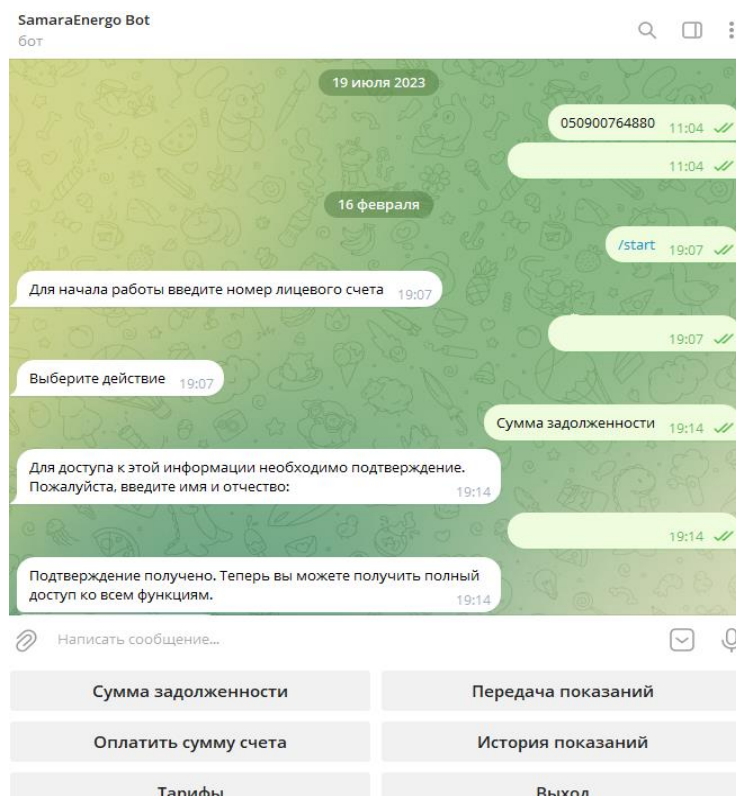


Рисунок 4 – Бот «Самараэнерго»

Его преимущества:

- просмотр истории переданных показаний;
- возможность узнать задолженность и оплатить;
- простой и интуитивно понятный интерфейс.

Недостатки:

- некорректная работа с разными типами счетчиков: возникают проблемы, когда у потребителя установлен сложный многотарифный счетчик;
- нестабильная работа: во время приема показаний периодически бот «зависает», недоступен из-за проблем на сервере.

«ЕРЦ г. Тольятти» - @erc163ru_bot предлагает передать показания, узнать данные по счетчикам, передать документы и записаться на очный прием.

Интерфейс Telegram-бота для пользователей Единого расчетного центра рисунок 5.

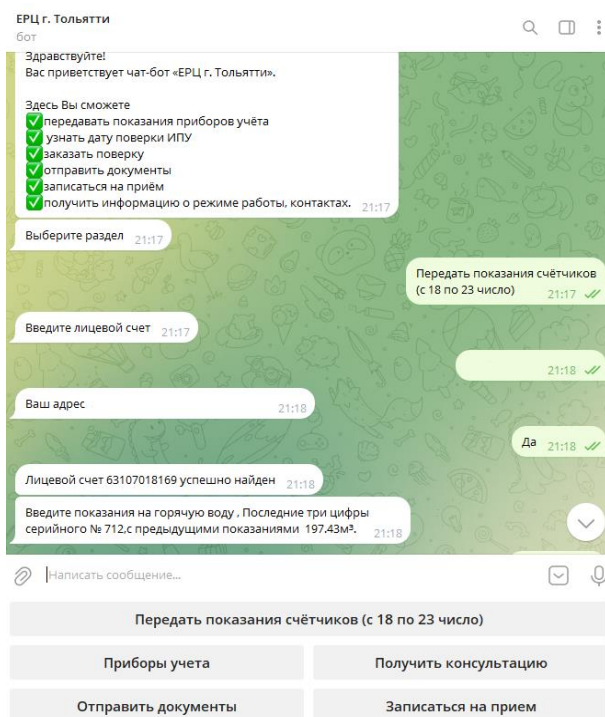


Рисунок 5 – Бот «ЕРЦ Тольятти»

Преимущества сервиса:

- легкая авторизация, достаточно ввести свой лицевой счет, весь функционал доступен без дополнительных подтверждений;

- простой и понятный интерфейс, без лишней информации;
- получение уведомлений о необходимости передать показания и изменения тарифов.

Недостаток: нет возможности сохранять данные для авторизации нескольких квартир. На рисунке 6, показан бот РКС-Тольятти



Рисунок 6 – «РКС» бот

Огромное количество возможностей бота от приема показаний до получения консультации и передачи документов в режиме онлайн, доступен по нику @Roscomsysru_bot.

Авторизация и возможности чат-бота компании «РКС» на рисунке 1.6

Преимущества бота:

- большой спектр возможностей, оплата, передача показаний, документов, настройка уведомлений, квитанции;

- возможность активировать до 5 лицевого счета и управлять ими;
- удобное управление и меню;
- интеграция с Биллингом, все изменения в реальном времени.

Недостатки:

- сложная авторизация, простому пользователю сложно разобраться и раскрыть весь функционал;
- большое количество кнопок в подменю бота.

Приведен анализ ключевых характеристик, существующих Telegram-ботов, предназначенных для приема показаний в таблице 1

Таблицы 1 – сравнительный анализ характеристик

	СамараЭнерго	ЕРЦ Тольятти	РКС
Удобный и понятный интерфейс	Средний	Легкий	Сложный
Управление лицевыми счетами	Нет возможности управлять несколькими ЛС	Нет возможности управлять несколькими ЛС	До 5 лицевых счетов
Уведомления	Нет	Да	Да
Авторизация	Средний	Легкий	Сложный

1.5 Преимущества автоматизации

Разработанный Telegram-бот решит ряд проблем: прием показаний без участия операторов, станет доступным и удобным способом взаимодействия с потребителями, значительно снизит вероятность ошибок [6], [23]. Модель бизнес-процесса [16] в будущем состоянии после внедрения чат-бота на рисунке 7.

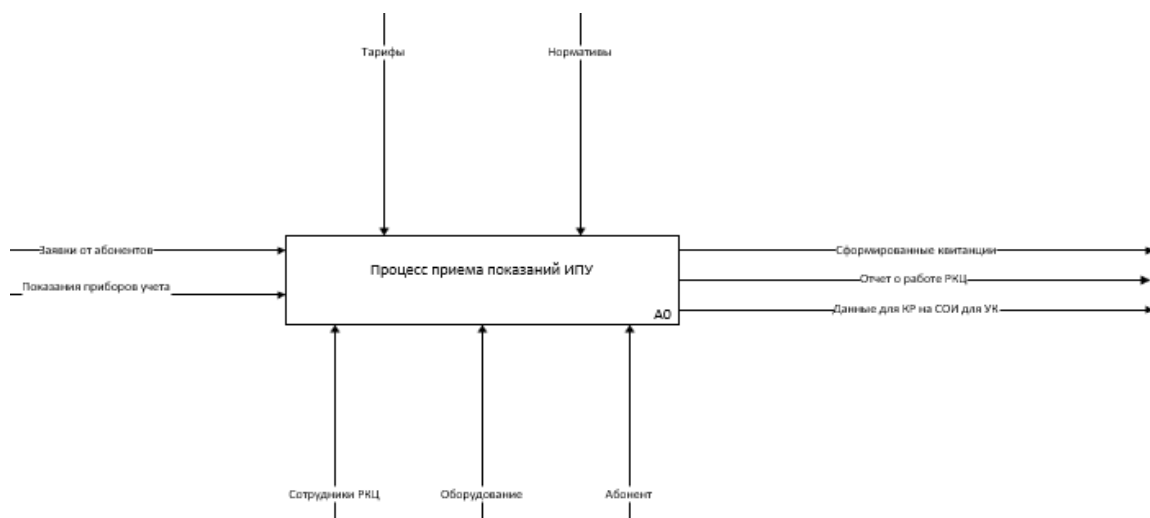


Рисунок 7 – IDEF0 «to be»

Декомпозиция бизнес-процесса по приему показаний с использованием чат-бота, рисунок 8.

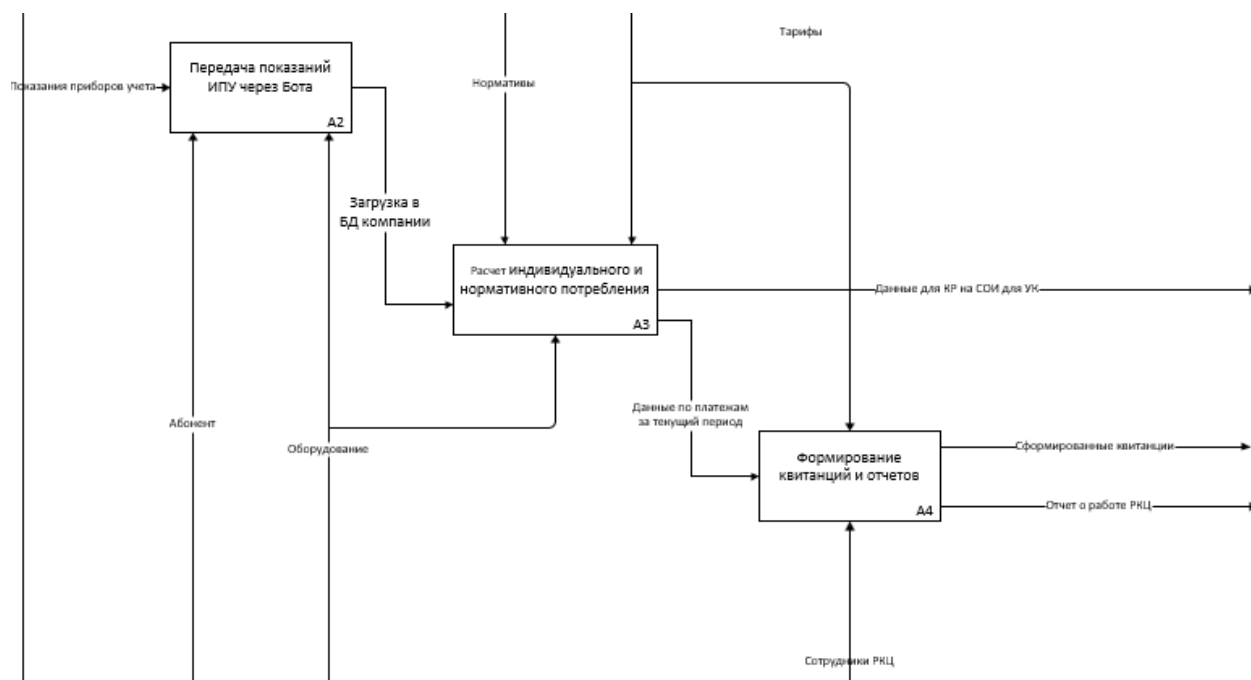


Рисунок 8 – IDEF0 декомпозиция «to be»

Внедрение чат-бота Telegram позволит:

- снизить нагрузку на операторов – бот берет на себя до 70% показаний;
- моментальная обработка данных и доступность сервиса 24/7 сократит очереди в центрах обслуживания и дает возможность сотрудникам уделять время на более важные проблемы;
- повышает удобство для клиентов – нет необходимости звонить или посещать офис;
- сократить затраты на обработку заявок, уменьшение количества операторов.

Выводы по первой главе

В первой главе была проведена оценка предметной области, включая обзор существующих решений, выявление проблем и постановку задач исследования. Был проведен анализ требований к системе, рассмотрены ключевые факторы, влияющие на эффективность работы.

На основе анализа стало очевидно, что существует необходимость в разработке более эффективного программного продукта, который позволит повысить производительность, автоматизировать процессы и снизить затраты.

Глава 2 Проектирование программы

2.1 Обзор платформ для создания чат-ботов

Для разработки чат-бота существуют различные платформы, каждая из которых имеет свои преимущества и ограничения. Рассмотрим наиболее популярные решения и выберем оптимальный вариант для проекта. Сравнительный анализ платформы (Telegram, WhatsApp, VK) по функциональности, доступности и возможностям для разработки ботов представлен в таблице 1

Таблицы 2 – Сравнительный анализ платформ для бота

	Telegram	WhatsApp	VK
API для ботов	Бесплатный, прост в настройке	Доступен через Business API, платный, сложный	Бесплатный, но менее популярный для ботов
Доступность	Глобально, без значительных ограничений	Глобально, но есть ограничения в России	Глобально, незначительные ограничения
Популярность ботов	Высокая, развитая экосистема.	Значительно меньше, чем у Telegram	Практически не развита
Функциональность	Широкие возможности, поддержка различного контента, инструменты для платежей	Ограниченная функциональность	Ориентирован на Общение и рассылки. Широкие возможности, интеграция с сервисами VK
Простота разработки	Прост, много библиотек и документации	Сложно, требуется интеграция с WhatsApp Business	Средняя, зависит от сложности задачи
Интеграция с внешними сервисами	Хорошая, множество библиотек и готовых решений	Ограниченная, требуется настройка через API	Хорошая, особенно с сервисами VK

Преимущества Telegram:

- бесплатный API для ботов;

- широкая функциональность;
- развитая экосистема ботов и сообщество разработчиков;
- простота разработки.

Недостаток: не такой широкий охват, как у WhatsApp (но быстро растет).

Преимущества WhatsApp API:

- огромный охват аудитории;
- привычный интерфейс для многих пользователей.

Недостатки:

- платный API;
- более сложная разработка и интеграция, ограниченная функциональность по сравнению с Telegram;
- строгие правила использования (необходимо согласие пользователя на получение сообщений).

Преимущества VK (ВКонтакте):

- бесплатный API для ботов;
- широкие возможности платформы;
- интеграция с другими сервисами VK.

Недостаток: меньшая популярность ботов.

Произведя сравнительный анализ платформ, Telegram является наиболее оптимальным вариантом проекта.

Бесплатный API: Telegram предоставляет все необходимые инструменты для разработки функционального бота без каких-либо затрат.

Простота разработки: Легкость разработки и наличие большого количества библиотек и документации.

Функциональность: Telegram позволяет реализовать широкий спектр функций, включая сбор и обработку данных, отправку уведомлений и интеграцию с другими сервисами.

Популярность ботов: Пользователи Telegram привыкли к использованию ботов, что повышает вероятность того, что потребители будут активно использовать бота для отправки показаний.

2.2 Хранение данных

При создании чат-бота для Telegram важно выбрать подходящую базу данных для хранения и обработки данных. Характеристики популярных платформ для хранения данных: Google Таблицы, SQLite, MySQL в таблице 3.

Таблица 3 – сравнительный анализ платформ для хранения данных

	Google таблица	SQLite	MySQL
Простота настройки	Очень простая	Простая	Средняя
Скорость работы	Низкая	Высокая	Высокая
Управление данными	Удобный интерфейс	CLI и GUI	Требуется администрирование
Стоимость	Бесплатно	Бесплатно	Базовый уровень, бесплатно
API	Есть	Нет	Есть
Поддержка в реальном времени	Через API	Нет	Ограниченно
Масштабируемость	Низкая	Низкая	Высокая

Преимущества выбора Google таблицы [18].

- простота настройки;
- не требует установки на сервер, достаточно создать таблицу на диске;
- подключение через API с помощью Python-библиотеки gspread.

Комфортный интерфейс.

- легко просматривать, редактировать данные;
- подходит для любого пользователя, не обязательно обладать навыками программирования или администрирования;
- бесплатно;
- достаточно личного аккаунта, использование таблиц бесплатно.

Google таблицы – это простой и популярный инструмент, несложная настройка и не требует специальных навыков администрирования. Для моей задачи по чтению и записи данных ботом объема бесплатного хранилища Google будет достаточно.

Данные можно легко просматривать и редактировать напрямую в таблице, интеграция с Google Apps Script позволяет автоматизировать взаимодействие с таблицей, получая и записывая данные, так же данные в таблице можно легко анализировать, строить графики, использовать формулы.

При необходимости, к таблице можно предоставить доступ другим пользователям для просмотра или редактирования.

Для решения оптимизации будет создано автоматизированное решение – Telegram-бот. Он позволит уменьшить нагрузку на сотрудников за счет автоматического сбора данных. Для этого сотрудники будут выгружать данные счетчиков и показаний из базы в Google таблицу, которые бот будет читать и записывать данные потребителей.

2.3 Сравнительный анализ языков программирования

Анализ языков программирования Python, Java и C# по критериям удобства, производительности и интеграции с API, таблица 4.

Таблица 4 – Сравнительный анализ языков программирования

Критерий	Python	Java	C#
Простота настройки	Очень высокая	Средняя	Средняя
Производительность	Средняя	Высокая	Высокая
Поддержка API	Полная поддержка	Ограниченная	Ограниченная
Масштабируемость	Средняя	Высокая	Высокая
Доступность библиотек	Очень высокая	Средняя	Средняя
Затраты на разработку	Низкие	Высокая	Высокая

Для разработки был выбран python [21], [24], [25]. Python известен своим простым синтаксисом, который упрощает написание, чтение и поддержку кода. Telegram-бот на Python [14] может быть создан буквально за несколько строк кода благодаря доступным библиотекам, таким как python-telegram-bot или aiogram. Например, простой бот на Python [15]:

```
from telegram import Update
from telegram.ext import ApplicationBuilder, CommandHandler,
ContextTypes

async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text('ДоБрОе УТРо.')
```

```
app = ApplicationBuilder().token("ВАШ_ТОКЕН").build()
app.add_handler(CommandHandler("start", start))
app.run_polling()
```

В других языках подобный код будет громоздким и сложным.

Python имеет огромное количество библиотек и фреймворков, упрощающих разработку бота:

- работа с данными: pandas, openpyxl, csv, json
- базы данных: gspread, SQLite, PostgreSQL
- веб-фреймворки: FastAPI, Flask, Django
- машинное обучение и ИИ: TensorFlow, scikit-learn

Благодаря простому синтаксису, большому количеству готовых проектов и активному сообществу, разработка на Python требует меньше времени и ресурсов. Это снижает стоимость проекта по сравнению с использованием более сложных языков

Как итог он является оптимальным выбором для разработки Telegram-бота. Простота: лаконичный синтаксис и хорошие библиотеки позволяют быстро создать функционального бота. Поддержка API Telegram: множество библиотек упрощают интеграцию с Telegram. Асинхронность: возможность обработки множества запросов одновременно. Экосистема и

кроссплатформенность: python работает на любой платформе и имеет обширную библиотеку модулей. Экономичность: быстрое развертывание и низкая стоимость разработки.

На рисунке 9 представлена архитектура взаимодействия Telegram-бота с сервисом Google Таблицы для автоматизации процесса учёта данных.

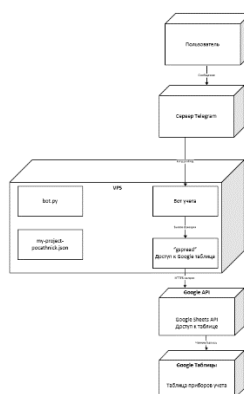


Рисунок 9 – Диаграмма развертывания

Пользователь отправляет сообщение с показаниями приборов в Telegram-бота. Сообщение обрабатывается сервером Telegram и перенаправляется на сервер, где размещён бот учёта. Основной компонент бота – скрипт bot.py реализующий логику обработки входящих сообщений и выполнения операций.

Для взаимодействия с Google Таблицами используется библиотека gspread, обеспечивающая доступ к таблице через Google Sheets API. Авторизация осуществляется с помощью JSON-файла (my-project-rosatchnik.json), содержащего ключи доступа к Google API.

После обработки данных бот формирует запрос к Google Sheets API, он взаимодействует с конкретной таблицей в сервисе Google Таблицы. В результате показания пользователя записываются в соответствующую строку таблицы.

2.4 Техническое задание и реализация интерфейса программного средства

Процесс: Приём показаний от пользователя

Цель: обеспечить корректный прием, валидацию и сохранение показаний в базе данных с подтверждением пользователю

Описание бизнес-процесса

Процесс начинается с запроса пользователя на отправку показаний. Система проходит несколько этапов, включая проверку лицевого счета, валидацию формата показаний и сохранение информации в базе данных. Диаграмма построена с использованием логических операторов XOR для ветвления сценариев.

Этапы бизнес-процесса:

- 1) запрос на отправку показаний Система получает запрос от пользователя. Переходит к следующему этапу – запрос номера лицевого счета;
- 2) запрос номера лицевого счета. Система запрашивает у пользователя ввод лицевого счета. Используется логическое ветвление (XOR). Если номер не введен или в неверном формате - ошибка Лицевой счет не найден. Если номер введен корректно → переходит к запросу показаний;
- 3) запрос показаний. Система запрашивает у пользователя ввод показаний. После ввода: если формат неверный → сообщение об ошибке «Ваши показания меньше предыдущих»; если формат верный → переход к сохранению в БД;
- 4) сохранение показаний в БД. При корректном вводе показаний осуществляется сохранение данных в базе. При успешном сохранении отображается сообщение «Показания сохранены»;

5) подтверждение пользователю. После сохранения система отправляет подтверждение пользователю. Завершение процесса: «Показания приняты». Логика обработки ошибок. На каждом этапе предусмотрена проверка формата вводимых данных. В случае некорректных данных пользователь получает сообщение об ошибке. Только при корректном выполнении всех шагов данные фиксируются в системе. Форматы и ограничения: номер лицевого счета: только числовой формат, длина 11 символов; показания: только положительные числа, показания не могут быть меньше предыдущих.

Для реализации проекта был выбран язык программирования Python – самый популярный выбор для работы с Telegram API. В Python достаточно простой синтаксис и большое количество библиотек

Для того чтобы бот был доступен 24/7, он должен быть размещен на сервере. Платформа «my.ishosting», предоставляющая виртуальный частный сервер, простой способ аренды и дает больше гибкости и контроля

Модель, рисунок 10, описывает последовательность операций взаимодействия пользователя с чат-ботом.

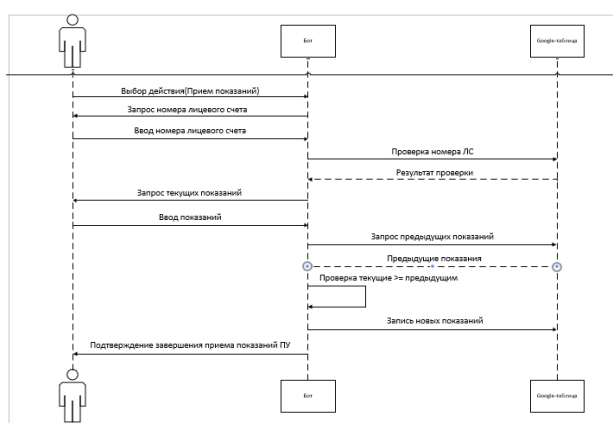


Рисунок 10 – Диаграмма активности

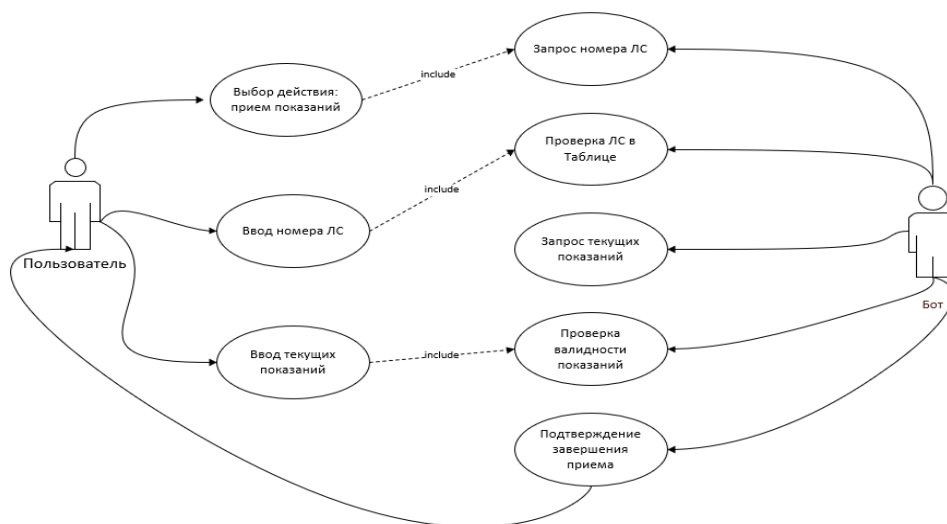


Рисунок 11 – Диаграмма вариантов использования.

Для понимания всех действий, в таблице 5 представлены прецеденты и описания, которые изображены на рисунке 11.

Таблица 5 – описание прецедентов

Прецеденты	Описания
Выбор действия: прием показаний	Пользователь инициирует процесс передачи показаний.
Запрос номера ЛС	Бот запрашивает у пользователя ввести лицевой счёт.
Ввод номера ЛС	Пользователь вводит номер лицевого счёта.
Проверка ЛС в таблице	Бот отправляет запрос в Google-таблицу для проверки существования ЛС.
Запрос текущих показаний	После успешной проверки, бот запрашивает текущие показания.
Ввод текущих показаний	Пользователь вводит текущие показания.
Проверка валидности показаний	Бот сравнивает новые показания с предыдущими (они не должны быть меньше).
Подтверждение приема показаний	После успешной проверки бот записывает данные и подтверждает приём.

Вывод по главе 2

Во второй главе дипломной работы был проведён анализ и обоснование выбора технологических решений, необходимых для реализации

программного продукта. Основное внимание было уделено выбору платформы для создания чат-бота, системы хранения данных, языка программирования, а также логики взаимодействия пользователя с системой. Важной частью главы стал выбор языка программирования. В рамках анализа языков Python, Java и C# было установлено, что Python лучше всего отвечает задачам проекта. Его простой синтаксис, обширная библиотечная поддержка, наличие готовых решений для работы с Telegram API, а также экономичность разработки делают его оптимальным инструментом для создания чат-бота.

Завершающим этапом главы стало описание логики реализации программного решения, была проработана структура бизнес-процесса взаимодействия пользователя с ботом на каждом этапе — от ввода лицевого счета до подтверждения успешной отправки данных.

Таким образом, в ходе данной главы были сформированы технологические и логические основы разработки Telegram-бота, что позволило перейти к этапу его практической реализации.

Глава 3 Физическое проектирование Telegram-бота по приему показаний

3.1 Получение токена Телеграм и создание Google таблицы

Процесс разработки Telegram-бота состоит из нескольких этапов. Получение токена с помощью Телеграм бота «BotFather». Это специальный бот, который генерирует токен, необходимый для управления ботом через API. Для работы с Телеграм API и другими функциями, необходимо установить библиотеки python-telegram-bot. Процесс регистрации и получения токена для Telegram-бота через BotFather на рисунке 12.

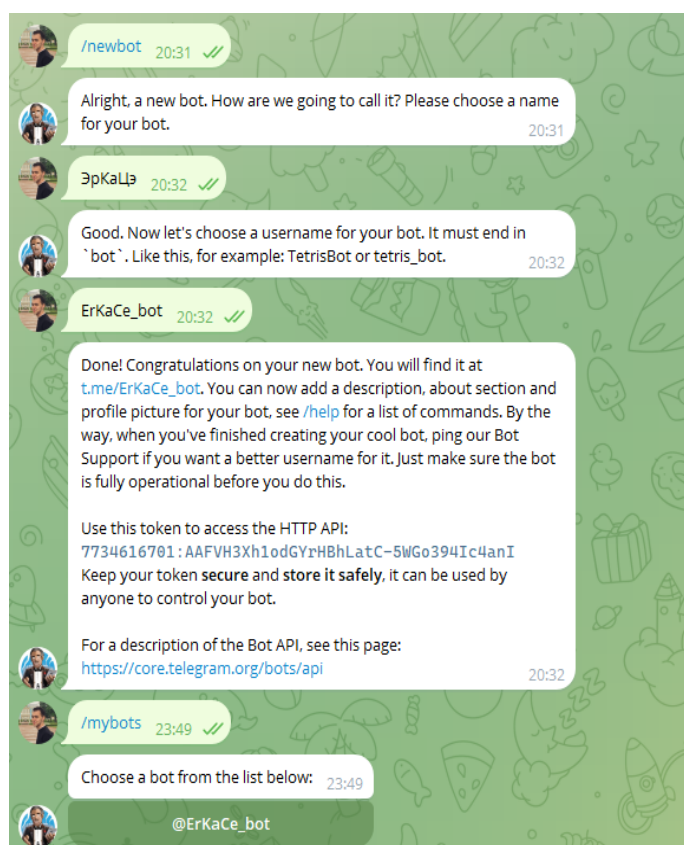


Рисунок 12 – получение токена Телеграм бота

Для хранения, чтения и записи будет использоваться Google таблица.

Пример структуры таблицы, используемой для хранения данных абонентов и их показаний рисунок 13.

№ п/п	№ базы	Count id	Лицевой счет	Адрес	Прибор учета	Срок поверки прибора	Вид показаний	Показания прошлого месяца	Показания текущего месяца
1		3	6045664	63207018680	Матросова ул. 15А квартира 1	25277067 29.05.2030	Холодное водоснабжение	476.000	
2		3	6045665	63207018680	Матросова ул. 15А квартира 1	25277062 29.05.2028	Горячее водоснабжение	212.000	
3		3	6045666	63207018680	Матросова ул. 15А квартира 1	25277079 29.05.2030	Холодное водоснабжение	1432.000	
4		3	6045667	63207018680	Матросова ул. 15А квартира 1	25297125 29.05.2028	Горячее водоснабжение	715.000	
5		3	7199884	63207018689	Матросова ул. 15А квартира 10	30196656 30.01.2029	Холодное водоснабжение	332.000	
6		3	7199885	63207018689	Матросова ул. 15А квартира 10	29933949 30.01.2029	Горячее водоснабжение	240.170	
7		3	3355166	63207018778	Матросова ул. 15А квартира 100	17007138 01.03.2030	Холодное водоснабжение	135.000	
8		3	3355169	63207018778	Матросова ул. 15А квартира 100	16994924 01.03.2028	Горячее водоснабжение	107.000	
9		3	3355171	63207018778	Матросова ул. 15А квартира 100	17007177 01.03.2030	Холодное водоснабжение	509.000	
10		3	3355172	63207018778	Матросова ул. 15А квартира 100	17007253 01.03.2028	Горячее водоснабжение	290.000	
11		3	3296958	63207018779	Матросова ул. 15А квартира 101	15764243 07.07.2029	Холодное водоснабжение	140.100	
12		3	3296959	63207018779	Матросова ул. 15А квартира 101	15762859 07.07.2029	Холодное водоснабжение	697.800	
13		3	7143605	63207018779	Матросова ул. 15А квартира 101	15763595 07.07.2027	Горячее водоснабжение	220.300	
14		3	7143606	63207018779	Матросова ул. 15А квартира 101	15764278 07.07.2027	Горячее водоснабжение	581.300	
15		3	3351639	63207018780	Матросова ул. 15А квартира 102	16885965 22.08.2028	Горячее водоснабжение	281.300	
16		3	7225703	63207018780	Матросова ул. 15А квартира 102	16885960 27.09.2030	Холодное водоснабжение	440.000	
17		3	5573844	63207018781	Матросова ул. 15А квартира 103	13501284 23.11.2026	Горячее водоснабжение	672.229	
18		3	7206319	63207018781	Матросова ул. 15А квартира 103	13553387 23.11.2028	Холодное водоснабжение	2053.380	
19		3	7103695	63207018782	Матросова ул. 15А квартира 104	32037920 23.08.2025	Горячее водоснабжение	171.300	
20		3	7141259	63207018782	Матросова ул. 15А квартира 104	12191405 15.07.2027	Холодное водоснабжение	790.000	
21		3	3150543	63207018783	Матросова ул. 15А квартира 105	13574763 16.06.2028	Холодное водоснабжение	247.400	
22		3	3150545	63207018783	Матросова ул. 15А квартира 105	3574749 16.06.2026	Горячее водоснабжение	379.400	
23		3	3150548	63207018783	Матросова ул. 15А квартира 105	13538259 16.06.2028	Холодное водоснабжение	1032.100	
24		3	3150550	63207018783	Матросова ул. 15А квартира 105	3539412 16.06.2026	Горячее водоснабжение	525.300	
25		3	7182257	63207018784	Матросова ул. 15А квартира 106	14013716 23.08.2028	Холодное водоснабжение	1287.000	
26		3	9042655	63207018784	Матросова ул. 15А квартира 106	14009486 23.08.2026	Горячее водоснабжение	972.000	
27		3	3089642	63207018785	Матросова ул. 15А квартира 107	12191838 13.07.2027	Холодное водоснабжение	681.000	
28		3	7103633	63207018785	Матросова ул. 15А квартира 107	31576126 23.07.2025	Горячее водоснабжение	306.000	
29		3	7087175	63207018786	Матросова ул. 15А квартира 108	363327 17.09.2030	Холодное водоснабжение	1555.000	
30		3	5579168	63207018787	Матросова ул. 15А квартира 109	346963 10.03.2029	Холодное водоснабжение	110.000	
31		3	5579169	63207018787	Матросова ул. 15А квартира 109	336590 10.03.2029	Холодное водоснабжение	251.000	
32		3	7120857	63207018787	Матросова ул. 15А квартира 109	17097221 04.02.2027	Горячее водоснабжение	201.000	
33		3	7120858	63207018787	Матросова ул. 15А квартира 109	17093090 04.02.2027	Горячее водоснабжение	303.000	
34		3	3129025	63207018690	Матросова ул. 15А квартира 11	12798851 14.01.2028	Холодное водоснабжение	1788.000	
35		3	7123575	63207018690	Матросова ул. 15А квартира 11	10639735 23.01.2026	Горячее водоснабжение	711.000	
36		3	3340829	63207018788	Матросова ул. 15А квартира 110	686439 05.12.2027	Горячее водоснабжение	267.000	
37		3	7214599	63207018788	Матросова ул. 15А квартира 110	686434 05.12.2029	Холодное водоснабжение	409.000	
38		3	3359573	63207018789	Матросова ул. 15А квартира 111	17058403 27.03.2030	Холодное водоснабжение	301.730	
39		3	5579354	63207018789	Матросова ул. 15А квартира 111	17058229 06.04.2027	Горячее водоснабжение	173.790	
40		3	5576816	63207018790	Матросова ул. 15А квартира 112	14010083 10.11.2028	Холодное водоснабжение	869.790	
41		3	5576817	63207018790	Матросова ул. 15А квартира 112	14012706 10.11.2026	Горячее водоснабжение	130.200	
42		3	5576818	63207018790	Матросова ул. 15А квартира 112	14010140 10.11.2026	Горячее водоснабжение	476.510	
43		3	7185316	63207018790	Матросова ул. 15А квартира 112	14012601 10.11.2028	Холодное водоснабжение	504.070	
44		3	3179875	63207018791	Матросова ул. 15А квартира 113	14108714 29.09.2028	Холодное водоснабжение	188.000	
45		3	3179878	63207018791	Матросова ул. 15А квартира 113	14114469 29.09.2026	Горячее водоснабжение	282.000	
46		3	3179881	63207018791	Матросова ул. 15А квартира 113	14297071 29.09.2028	Холодное водоснабжение	978.000	
47		3	3179884	63207018791	Матросова ул. 15А квартира 113	14296582 29.09.2026	Горячее водоснабжение	373.000	
48		3	3290338	63207018792	Матросова ул. 15А квартира 114	15043014 30.03.2029	Холодное водоснабжение	638.890	
49		3	7130741	63207018792	Матросова ул. 15А квартира 114	15030675 30.03.2027	Горячее водоснабжение	309.000	
50		3							

Рисунок 13 – Google таблица с данными пользователей

Создаем таблицу, выгрузив данные из биллинга. Бот будет запрашивать индивидуальный лицевой счёт указанный на платежных документах находить

его в таблице и поочерёдно запрашивать новые показания индивидуальных приборов учета, в формате - наименование услуги (столбец Н) - предыдущие показания (столбец І). Пользователь введет свои показания, а бот запишет их в таблицу (столбец Ј).

После приема показаний администратор РКЦ загрузит ведомость показаний в систему компании.

3.2 Написание и запуск кода.

Устанавливаем и импортируем библиотеки:

«telegram» – основная библиотека для работы с TelegramBot API. Импортирует классы для создания кнопок в сообщении, объекты, содержащие информацию о входящих сообщениях и действий.

«telegram.ext» – расширение для библиотеки «telegram» предоставляет класс для запуска бота и инструменты для управления ботом, обработку команд и сообщений.

«gspread» – библиотека для интеграции бота и таблицы, она позволяет читать и записывать данные из Google таблицы.

«asyncio» – для асинхронной обработки сообщений и команд

Список основных библиотек, используемых для реализации Telegram-бота рисунок 14.

```
from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup, KeyboardButton, ReplyKeyboardMarkup, InputMediaPhoto
from telegram.ext import Application, CommandHandler, ConversationHandler, MessageHandler, CallbackQueryHandler, filters, ContextTypes
from functools import partial
import gspread
import nest_asyncio
import asyncio
```

Рисунок 14 – Библиотеки Python

Функции для работы с Google таблицей

Импортируем классы BOT_MESSENGE и BOT_BUTTONS из модулей components.messenge и components.buttons соответственно. Классы содержат текстовые сообщения и кнопки для бота.

Создаются экземпляры классов BOT_MESSENGE и BOT_BUTTONS, которые будут использоваться для получения сообщений и кнопок.

Функция «read_columns_with_row_number», читает данные из Google таблицы и возвращает словарь с данными.

Объект gc, который авторизуется в Google API с использованием JSON-файла с ключами. Далее открываем таблицу по id, открываем конкретный лист в таблице по его имени.

write_to_column_j записывает значение в ячейку столбца J

Фрагмент кода, реализующий чтение и запись данных в таблицу, рисунок 15.

```
8  from components.messenge import BOT_MESSENGE
9  from components.buttons import BOT_BUTTONS
10
11  MESSENGE_LIST = BOT_MESSENGE()
12  BUTTONS_LIST = BOT_BUTTONS()
13
14  WRITE_RAHHUN, WRITE_LEN = range(2)
15
16  def read_columns_with_row_number(json_keyfile: str, sheet_id: str, worksheet_name: str):
17
18      gc = gspread.service_account(filename=json_keyfile)
19      sheet = gc.open_by_key(sheet_id)
20
21      worksheet = sheet.worksheet(worksheet_name)
22      all_rows = worksheet.get_all_values()
23
24      result_dict = {}
25      for row_number, row in enumerate(all_rows, start=1):
26          if len(row) >= 9 and row[3] and row[8]:
27              key = row[3]
28              if key not in result_dict:
29                  value = [[row_number, row[8], row[7]]]
30
31                  result_dict[key] = value
32              else:
33                  result_dict[key].append([row_number, row[8], row[7]])
34
35      return result_dict
36
37  def write_to_column_j(json_keyfile: str, sheet_id: str, worksheet_name: str, row_number: int, value: str):
38
39      gc = gspread.service_account(filename=json_keyfile)
40
41      sheet = gc.open_by_key(sheet_id)
42      worksheet = sheet.worksheet(worksheet_name)
43
44      worksheet.update_cell(row_number, 10, value)
45
```

Рисунок 15 - Функции для работы с Google таблицы

Параметры проекта, доступа к API и таблице на рисунке 16

```
46 json_keyfile = 'my-project-pocathnick-78b606d811e9.json'
47 sheet_id = '1-0ppzQ0iiQGSYGBMxVvS4lsDSwCZBk7tTCeWbA7Mn7I'
48 worksheet_name = 'list1'
49
50 TOKEN = '7734616701:AAFVH3Xh1odGYrHBhLatC-5WGo394Ic4anI'
```

Рисунок 16 – Конфигурация бота

Параметры для работы с Google таблицей: путь к JSON-файлу с ключами, ID таблицы, имя листа и токен для доступа к Telegram Bot API.

На рисунке 17 продемонстрирована обработка команд пользователя и отправка сообщений через Telegram.

```
51
52 v async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
53     chat_id = update.effective_chat.id
54     user = update.effective_user
55     full_message = update.message.text
56     reply_markup = InlineKeyboardMarkup(BUTTONS_LIST.button_start)
57     await update.message.reply_text(MESSENGE_LIST.messenge_start, reply_markup=reply_markup)
58
59 v async def cancel(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
60     await update.message.reply_text(f"Непонятный запрос")
61     return ConversationHandler.END
62
63 v async def button_click(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
64     try:
65         user = update.effective_user
66         query = update.callback_query
67         response = query.data
68         bot = context.bot
69         await query.answer()
70         if response == 'consultation':
71             reply_markup = InlineKeyboardMarkup(BUTTONS_LIST.button_consultation)
72             await bot.send_message(chat_id=user.id, text=MESSENGE_LIST.messenge_consultaton, reply_markup=reply_markup)
73         elif response == 'main_menu':
74             reply_markup = InlineKeyboardMarkup(BUTTONS_LIST.button_start)
75             await bot.send_message(chat_id=user.id, text=MESSENGE_LIST.messenge_start, reply_markup=reply_markup)
76         elif response == 'add_pocathatel':
77             await bot.send_message(chat_id=user.id, text=MESSENGE_LIST.messenge_add_pocethatel)
78             return WRITE_RAHUN
79     except Exception as e:
80         print(f'err all: {e}')
81     return ConversationHandler.END
82
```

Рисунок 17 - команды и сообщения

`async def start...` определяется асинхронная функция, которая обрабатывает `/start`

Получаем ID чата, пользователя, текст сообщения и создание клавиатуры с кнопками.

`update.message.reply_text` - Отправляем сообщение с текстом и клавиатурой.

`def cancel` - Определяется асинхронная функция `cancel`, которая завершает текущий диалог.

`def button_click` - Определяется асинхронная функция `button_click`, которая обрабатывает нажатия на кнопки.

Получаем данные о пользователе, запросе и боте.

Если нажата кнопка «consultation», отправляем соответствующее сообщение и клавиатуру. Если нажата кнопка «main_menu», возвращаемся в главное меню. Если нажата кнопка «add_pocathatel», начинаем процесс добавления данных и переходим в состояние `WRITE_RAHUN`

Функции, обрабатывающие ввод пользователем данных показаний рисунок 18.

```
81 async def func_add_pocathatel(update: Update, context: ContextTypes.DEFAULT_TYPE, field_name: str) -> int:
82     user = update.effective_user
83     info_text = update.message.text
84     bot = context.bot
85     print(field_name)
86
87     if field_name == 'write_rahun':
88         print(info_text)
89         column_data = read_columns_with_row_number(json_keyfile, sheet_id, worksheet_name)
90         if info_text in column_data:
91             await bot.send_message(chat_id=user.id, text=MESSAGE_LIST.message_good_rah(info_text))
92             print(column_data[info_text])
93             context.user_data['n_rahun'] = info_text
94             context.user_data['n_rahun_use'] = 0
95             await bot.send_message(chat_id=user.id, text=MESSAGE_LIST.message_wr_next_poc(column_data[info_text][0][1], column_data[info_text][0][2]))
96             return WRITE_LEN
97         else:
98             await bot.send_message(chat_id=user.id, text=MESSAGE_LIST.message_add_pocathatel_not_rahun)
99             return WRITE_RAHUN
100     elif field_name == 'write_len':
101         print(info_text)
102         try:
103             n_rahun = context.user_data.get('n_rahun')
104             n_rahun_use = int(context.user_data.get('n_rahun_use'))
105             column_data = read_columns_with_row_number(json_keyfile, sheet_id, worksheet_name)
106             if float(info_text) >= float(column_data[n_rahun][n_rahun_use][1]):
107                 await bot.send_message(chat_id=user.id, text=MESSAGE_LIST.message_good_len(info_text))
108                 write_to_column_j(json_keyfile, sheet_id, worksheet_name, column_data[n_rahun][n_rahun_use][0], info_text)
109                 n_rahun_use += 1
110                 if n_rahun_use < len(column_data[n_rahun]):
111                     context.user_data['n_rahun_use'] = n_rahun_use
112                     await bot.send_message(chat_id=user.id, text=MESSAGE_LIST.message_wr_next_poc(column_data[n_rahun][n_rahun_use][1], column_data[n_rahun][n_rahun_use][2]))
113                     return WRITE_LEN
114                 else:
115                     reply_markup = InlineKeyboardMarkup(BUTTONS_LIST.button_start)
116                     await bot.send_message(chat_id=user.id, text=MESSAGE_LIST.message_fin_wr, reply_markup=reply_markup)
117             else:
118                 await bot.send_message(chat_id=user.id, text=MESSAGE_LIST.message_add_len_not_rahun)
119             return WRITE_LEN
120         except Exception as e:
121             print(e)
122             await bot.send_message(chat_id=user.id, text=MESSAGE_LIST.message_add_len_not_rahun)
123             return WRITE_LEN
124     return ConversationHandler.END
```

Рисунок 18 – Функции для обработки ввода данных

Определяется асинхронная функция «funk_add_pocath», которая обрабатывает ввод данных пользователем.

Получаем данные о пользователе

Если поле «field_name» равно «writ_rahun», проверяем, есть ли введенный текст в данных из таблицы. Если есть, переходим в состояние WRITE_LEN, иначе просим ввести данные заново.

Если поле «field_name» равно «writ_len», проверяют введенное значение и записывает его в таблицу. Если все данные введены, завершается процесс, иначе продолжают запрашивать данные

Завершаем диалог.

Настройка асинхронной работы Telegram-бота рисунок 19.

```
127
128 v async def main():
129     application = Application.builder().token(TOKEN).build()
130     application.add_handler(CommandHandler("start", start))
131 v   handler = ConversationHandler(
132       entry_points=[CallbackQueryHandler(button_click)],
133 v     states={
134         WRITE_RAHAUN: [MessageHandler(filters.TEXT & ~filters.COMMAND, partial(funk_add_pocath, field_name='writ_rahun'))],
135         WRITE_LEN: [MessageHandler(filters.TEXT & ~filters.COMMAND, partial(funk_add_pocath, field_name='writ_len'))],
136     },
137     fallbacks=[CommandHandler('cancel', cancel)],
138 )
139 application.add_handler(handler)
140 await application.run_polling()
141
```

Рисунок 19 - Основная функция бота

Определяем асинхронную функцию main, которая запускает бота, создается функция application для управления ботом.

Создаем обработчик команды start

ConversationHandler для управления диалогами с пользователем.

application.add_handler(handler) Добавляется ConversationHandler (управление диалогами)

await application.run_polling () - Запускается бот в режиме polling.

Инициализация бота рисунок 20.

```

141
142     if __name__ == '__main__':
143         nest_asyncio.apply()
144         asyncio.run(main())

```

Рисунок 20 - Запуск бота

JSON файл содержит все необходимые данные для аутентификации и авторизации сервисного аккаунта в Google.

Используется для доступа к API Google таблицы, фрагмент файла с ключами доступа к Google API показан на рисунке 21

```

1 my-project-pocathnick-78b606d811e8.json > ...
2 {
3   "type": "service_account",
4   "project_id": "my-project-pocathnick",
5   "private_key_id": "78b606d811e866ab59d9758398ca1643211ad85",
6   "private_key": "-----BEGIN PRIVATE KEY-----vMTIEvIBADMBgkqk1G6w4BAQEFASCBKwgs1AgEAAQIBAQDAmuscbsBjM1vWNgp058/b7nawWuyH64/fjK81j1puVQ16CrZewfCG6f68eHtX5qqVb703LAdVn2qZ6WYX86Rzqlhex2TqDv74",
7   "client_email": "test45345my-project-pocathnick.iam.serviceaccount.com",
8   "client_id": "118381147897948858875",
9   "auth_uri": "https://accounts.google.com/o/oauth2/auth",
10  "token_uri": "https://oauth2.googleapis.com/token",
11  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
12  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/test4534540my-project-pocathnick.iam.serviceaccount.com",
13  "universe_domain": "googleapis.com"
14 }

```

Рисунок 21 - данные для аутентификации в Google

VOT_BUTTONS, Рисунок 3.11 - Класс содержащий набор кнопок для бота.

Атрибут `button_start` - главное меню бота, которое отображается при запуске или возврате в главное меню. Строка содержит две кнопки «Сбор показаний» и «Консультация»

Атрибут `button_consultation` — меню, отображающее при выборе «Консультация» так же содержит 2 кнопки, «Оставить обращение» и «Часто

задаваемые вопросы», при нажатии на них направит на форму с обращениями или на сайт компании соответственно.

```
1 from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup
2
3 class BOT_BUTTONS:
4     button_start = [
5         [InlineKeyboardButton("Сброс показаний", callback_data='add_pocethatel'), InlineKeyboardButton("Консультация", callback_data='consultation')],
6     ]
7     button_consultation = [
8         [InlineKeyboardButton("Оставить обращение", url='https://forms.gle/UpKlovZy18S9Vrzik6'), InlineKeyboardButton("Часто задаваемые вопросы", url='http://rkhelg.tilda.ws')],
9         [InlineKeyboardButton("Главное меню", callback_data='main_menu')],
10    ]
```

Рисунок 22 – класс BOT_BUTTONS

Класс BOT_MESSENGE на рисунке 23 – содержит текстовые сообщения, отправляемые ботом пользователю.

```
1 class BOT_MESSENGE:
2     message_start = """Здравствуйте!
3     Вас приветствует чат бот РХЦ
4     Здесь Вы сможете передать показания
5     приборов учёта
6     """
7     message_consultation = """Получить консультацию, вопросы начислений, платежей и другим. Вы можете по телефону Контакт-Центра +7(8482)008-00-00; ежедневно с 08:00 до 17:00.
8     В случае необходимости очной консультации, специалист запишет Вас на приём.
9     Здесь Вы можете оставить обращение и посмотреть подборку ответов на частые вопросы.
10    """
11    message_add_pocethatel = """Вводите лицевой счет"""
12    message_add_pocethatel_not_rahun = """Лицевой счет не найден попробуйте снова или обратитесь в центр обслуживания"""
13    message_add_len_not_rahun = """Ваши показания меньше
14    предыдущих, попробуйте снова
15    или обратитесь в центр
16    обслуживания
17    """
18    message_fin_w = """Показания счетчиков сохранены"""
19    def message_good_rah(self, rahun):
20        return f"Лицевой счет {rahun} успешно найден"
21    def message_w_text_pos(self, len_bul, vod_tupe):
22        if vod_tupe == "Холодное водоснабжение":
23            tup = "холодную"
24        else:
25            tup = "горячую"
26        return f"Вводите показания на {tup} воду с предыдущими показаниями {len_bul}!"
27    def message_good_len(self, len_bul):
28        return f"Вы ввели {len_bul}"
```

Рисунок 23 – Сообщения бота

После написания кода запускаю проект

Команда /start запускает бота.

Выводится приветственное сообщение с выбором действия, нажимая на «Консультация», бот предоставляет данные по разделу и предоставляет возможность нажать на нужную пользователю кнопку и перейти по ссылке для дальнейшей работы или вернуться в главное меню.

Визуализация взаимодействия с ботом при его запуске на рисунке 24

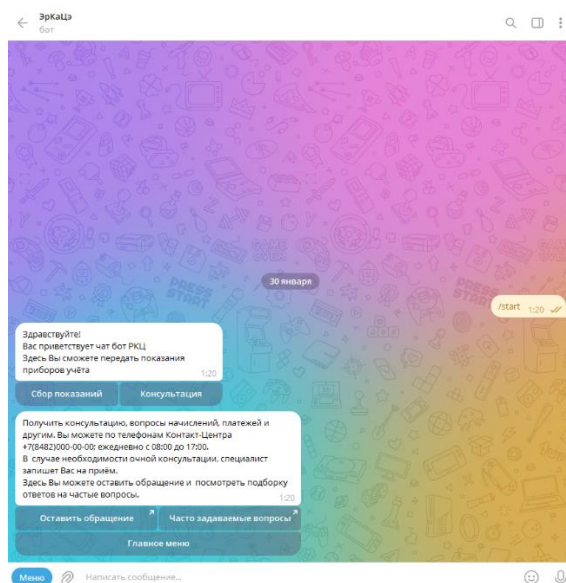


Рисунок 24 – Запуск бота

Навигационные кнопки, предоставляющие пользователю доступ к справочной информации, рисунок 25

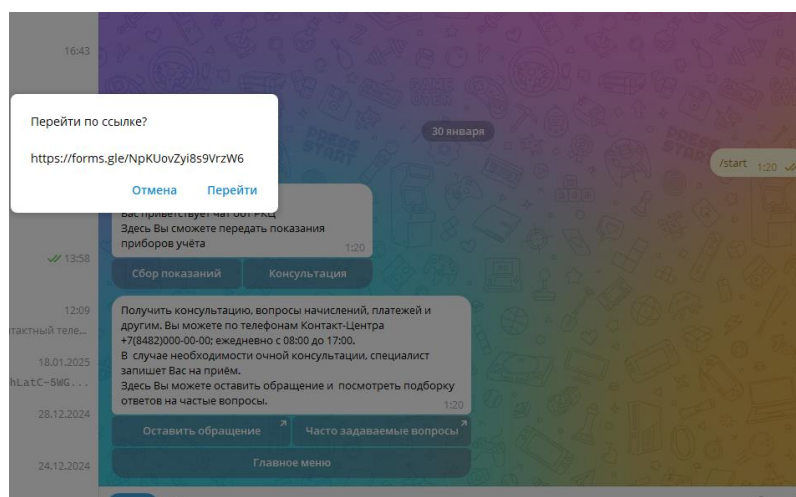


Рисунок 25 – Ссылки на FAQ и обращения

При нажатии команды «Сбор показаний», бот запрашивает индивидуальный лицевой счет пользователя, в случае ввода верного лс, он находит его в нашей Google таблице и запускает процесс сбора показаний.

Формат сообщения в случае успешного ввода «Лицевой счет «gaһun» успешно найден» и далее запрашивает показания по всем счетчикам на данном лицевом с указанием: типа прибора учета, холодная или горячая вода и предыдущими показаниями. Пример на рисунке 26 - взаимодействия пользователя с ботом при вводе показаний.

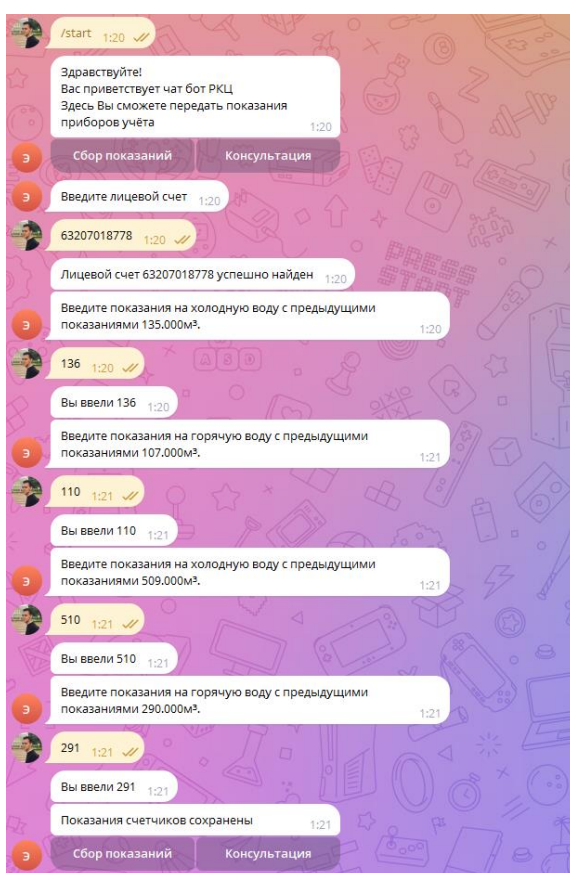


Рисунок 26 – Процесс передачи показаний

Пользователь ввел данные, отображение процесса записи данных из бота в Google таблицу на рисунке 27

3	7199885	83207018889	Матросова ул. 15А квартира 10	29933849	30.01.2029	Горячее водоснабжение	240.170	
3	3355186	83207018778	Матросова ул. 15А квартира 100	17007139	01.03.2030	Холодное водоснабжение	135.000	136
3	3355189	83207018778	Матросова ул. 15А квартира 100	18994924	01.03.2028	Горячее водоснабжение	107.000	110
3	3355171	83207018778	Матросова ул. 15А квартира 100	17007177	01.03.2030	Холодное водоснабжение	509.000	510
3	3355172	83207018778	Матросова ул. 15А квартира 100	17007253	01.03.2028	Горячее водоснабжение	290.000	291

Рисунок 27 – сохранение данных в Google таблицы

Отсылает сообщение об успешном приеме показаний, если у пользователя несколько квартир, бот предлагает запустить команду сбора показаний снова.

В случае если лицевой счет введен не верно или его нет в нашей таблице, бот выведет ошибку и предложит ввести его снова

Сообщение, отображающее ошибку при вводе некорректных показаний или лицевого счета показан на рисунке 28 и 29

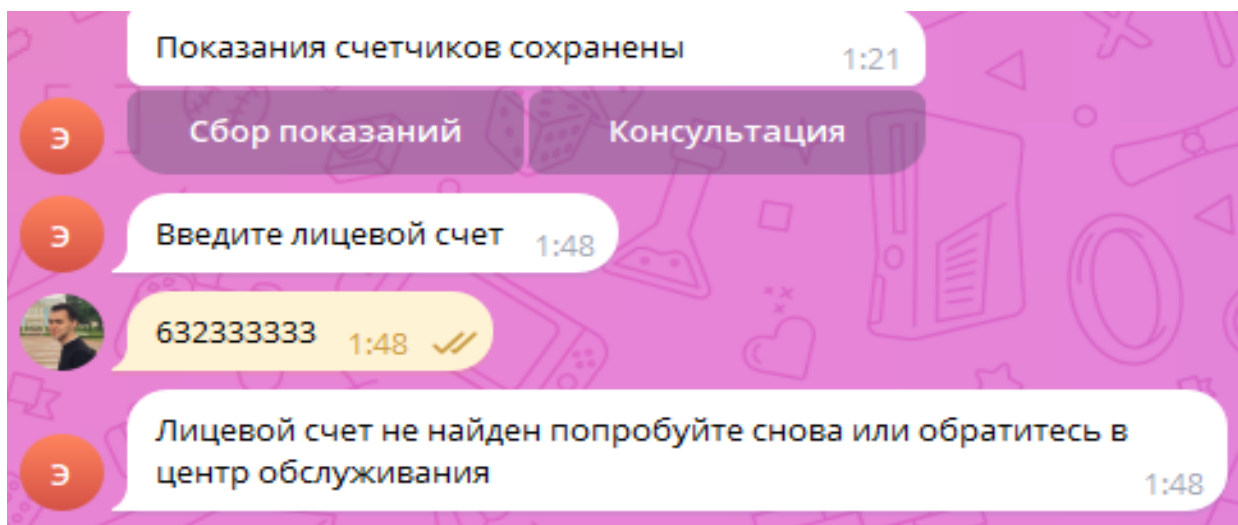


Рисунок 28 – Ошибка Лицевого счета

Вводить данные по ИПУ меньше предыдущих запрещено поэтому бот так же укажет ошибку и запросит ввести новые показатели.

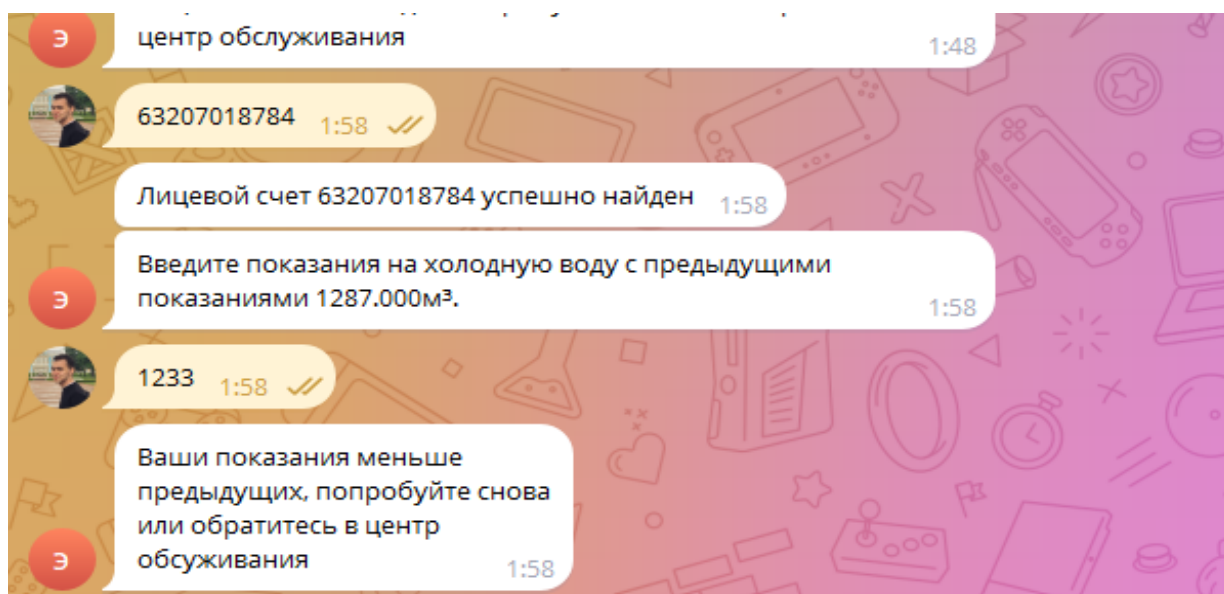


Рисунок 29 – Ошибка ввода показаний

3.3 Перспективы развития

1. Возможность прикреплять фото счетчиков для подтверждения и проверки счетчика.

- бот предлагает пользователю выбрать тип счётчика;
- после выбора бот запрашивает показания и предлагает загрузить фото для подтверждения. Хранение фото может быть организовано двумя способами: локально (на сервере бота) и в облаке (Google Drive, Yandex Disk).

2. Напоминания о необходимости передачи показаний

- при регистрации бот спрашивает предпочитаемую дату и время для напоминаний;
- напоминание можно настроить. Напоминание приходит в виде сообщения с кнопками;
- ввести показания;
- загрузить фото;

- пропустить;
- если пользователь игнорирует сообщение, бот может отправить повторное уведомление через несколько дней;
- пользователь может изменить настройки напоминаний: Время, частоту;
- отключить напоминания.

3. Подключение к системам ЖКХ и банковским сервисам для автоматических расчётов [5], [12], [17].

- управляющие компании и водоканалы часто предоставляют API для передачи данных;
- интеграция с платёжными системами. Для автоматического расчёта и оплаты можно подключить сервисы: Ю-Kassa, Сбербанк, Тинькофф;
- после приёма показаний бот рассчитывает сумму и предлагает оплату.
- Пользователь оплачивает через встроенную форму Telegram

После успешной оплаты бот отправляет подтверждение в систему ЖКХ. Вся история платежей и чеки сохраняются в профиле пользователя.

3.4 Методика расчета экономической эффективности

ROI (Return on Investment, возврат инвестиций) – показывает, насколько быстро окупятся затраты на разработку [4]. Таблица 6 описывает возможные риски при реализации проекта и меры по их минимизации.

Таблица 6 - Оценка рисков реализации проекта

Тип риска	Описание	Способы минимизации
Технический	Возможные сбои в работе бота или API Telegram	Тестирование, мониторинг сервера
Финансовый	Возможные дополнительные расходы на сервер	Оптимизация затрат, выбор доступного хостинга
Юридический	Обработка персональных данных клиентов	Соответствие ФЗ-152 (Закон о персональных данных)

3.5 Расчет экономической эффективности проекта

Исходные данные:

Число клиентов, передающих показания с помощью бумажных квитков – 13000 чел./мес.

Среднее время обработки одной заявки оператором – 1,5 мин.

Число операторов в РКЦ, занятых обработкой показаний – 5 чел.

Средняя зарплата оператора – 30 000 руб./мес.

Время обработки одной заявки чат-ботом – <10 сек.

Расчет экономии времени и затрат

До внедрения чат-бота: 13000 заявок * 1,5 мин = 325 часов работы операторов в месяц. При 5 операторах каждый тратит на обработку 65 часов ежемесячно.

После внедрения чат-бота:

Автоматизация 70% заявок → остается только 3900 заявок для операторов: 3900 * 1,5 мин = 97 часов работы операторов в месяц

Экономия рабочего времени: 238 часов в месяц (3,5 оператора)

Расчет экономии зарплатных затрат:

Средняя зарплата оператора = 30000 руб./мес, Экономия: 3,5 оператора × 30 000 руб. = 105 000 руб./мес

Расчет ROI

Инвестиции в проект:

Разработка бота: 80 000 руб.

Настройка базы данных и серверов: 10 000 руб.

Хостинг: 500 руб./мес

Общие затраты на проект = 90 500 руб.

Ожидаемая экономия в первый год:

$105\,000 \text{ руб./мес} \times 12 \text{ мес} = 1\,260\,000 \text{ руб.}$

$ROI = (\text{экономия} - \text{затраты}) / \text{затраты} \times 100\%$

$ROI = (1\,260\,000 - 90\,500) / 90\,500 \times 100\% = 1292\%$

Выводы по главе 3

Во третьей главе была выполнена разработка программного продукта, базы данных и пользовательского интерфейса, произведена оценка экономической целесообразности разработки. Были проведены расчеты затрат на реализацию проекта,

Таким образом, была создана эффективная и производительная система, полностью соответствующая задачам проекта.

Анализ показал, что разработанная система позволяет значительно снизить затраты на выполнение определенных операций, повысить производительность труда и сократить временные затраты. Таким образом, внедрение предложенного решения является экономически обоснованным и выгодным, что подтверждает его практическую ценность.

Заключение

В ходе выполнения работы был проведен анализ и оценка организационной структуры компании, особое внимание было уделено ключевым бизнес-процессам. Анализ показал, что самое требующие оптимизации направление является процесс приема показаний и обращений от клиентов. Существующая ручная обработка данных имеет ряд недостатков: высокая нагрузка на персонал, риск ошибок при вводе, замедление обработки запросов.

Разработка и внедрение чат-ботов – перспективное направление для автоматизации бизнес-процессов и повышения качества обслуживания клиентов. Был разработан программный продукт Telegram-bot для использования потребителями для приема показаний.

Чат-бот успешно решает следующие задачи:

- 1) автоматизация приема показаний: Клиенты получают возможность самостоятельно и в любое удобное время передавать показания, что значительно повысит их удовлетворенность обслуживанием;
- 2) снижение нагрузки на персонал: благодаря автоматизированной обработке показаний, сотрудники больше уделяют время сложным и приоритетным задачам;
- 3) минимизация ошибок: Исключение ошибок, при вводе данных позволило повысить точность учета показаний;
- 4) круглосуточная подача показаний: бот интегрирован с Google таблицами и загружен на сервер что позволяет его использовать круглосуточно.

Внедрение чат-бота стало первым шагом на пути к автоматизации и повышению эффективности взаимодействия с клиентами. Благодаря интеграции с Google Таблицами и размещению на сервере, бот функционирует в режиме 24/7.

Реализация бота представляет собой важный шаг к автоматизации и повышению эффективности взаимодействия с клиентами. В перспективе планируется дальнейшее развитие проекта, включая интеграцию с другими системами компании и расширение функциональности для охвата дополнительных задач клиентского сервиса. Внедрение чат-бота стало важным этапом развития компании в области цифровизации процессов обслуживания клиентов. Интеграция с Google Таблицами позволила автоматизировать обработку запросов пользователей, обеспечивая бесперебойное функционирование системы в круглосуточном режиме.

Перспективы дальнейшего развития проекта включают следующие направления:

1. Интеграция с дополнительными сервисами. Подключение к другим внутренним информационным системам позволит расширить возможности обработки запросов клиентов и повысить точность предоставления информации.
2. Расширение функционала. Планируется внедрение новых функций, направленных на улучшение качества обслуживания — от упрощенной регистрации заявок до персонализированных рекомендаций и поддержки оплаты онлайн-сервисов.
3. Анализ данных и машинное обучение. Анализ больших объемов данных, полученных в ходе общения с пользователями, даст возможность оптимизировать процессы коммуникации и предлагать клиентам наиболее релевантные решения.

Таким образом, создание и совершенствование чат-бота станет залогом повышения лояльности наших клиентов и роста конкурентоспособности бизнеса в условиях быстро меняющегося рынка цифровых технологий.

Список используемой литературы и используемых источников

1. Воробьев, А. А. Базы данных и Google Таблицы для бизнеса. – М.: Альпина Паблишер, 2022. – 224 с.
2. Государственная информационная система ЖКХ [Электронный ресурс]. – URL: <https://dom.gosuslugi.ru>
3. Горшкова, И. Ю., Сидоров, П. Н. Автоматизация бизнес-процессов с использованием чат-ботов. – М.: Инфра-М, 2022. – 256 с.
4. Дмитриев, В. А. Методы оценки экономической эффективности IT-проектов. – М.: Финансы и статистика, 2021. – 192 с.
5. ЖКХ Контроль. Технологические решения для ЖКХ [Электронный ресурс]. – 2024. – URL: <https://gkhkontrol.ru>
6. Козлова, Н. В. Автоматизация клиентских сервисов на базе мессенджеров // Цифровая экономика. – 2022. – №2. – С. 43–51.
7. Кузнецов, В. А. Эффективность внедрения информационных систем в ЖКХ. – СПб.: Питер, 2021. – 198 с
8. Министерство ЖКХ РФ. Цифровизация жилищно-коммунального хозяйства: нормативно-правовая база и перспективы развития [Электронный ресурс]. – 2023. – URL: <https://www.minstroy.gov.ru>
9. Николаев, С. В. Telegram API: Разработка ботов на Python. – М.: ДМК Пресс, 2023. – 312 с.
10. Олейников, А. Н. Моделирование бизнес-процессов с использованием BPMN 2.0. – М.: Юрайт, 2022. – 288 с.
11. Официальный сайт компании "Стек" [Электронный ресурс]. – 2024. – URL: <https://stek.company>
12. Петров, И. И., Иванова, А. В. Цифровизация процессов в сфере ЖКХ: современные подходы и решения // Вестник ЖКХ. – 2022. – №4. – С. 54–61.
13. РБК. Будущее ЖКХ: цифровые технологии и автоматизация [Электронный ресурс]. – 2024. – URL: <https://www.rbc.ru>

14. Савельев, П. Р. Python для начинающих: от основ до создания ботов. – СПб.: БХВ-Петербург, 2023. – 348 с.
15. Федоров, С. В. Разработка телеграм-ботов с интеграцией Google Sheets. – СПб.: Питер, 2023. – 240 с.
16. Aiogram Documentation [Электронный ресурс]. – URL: <https://docs.aiogram.dev>
17. Digital Housing: How Technology is Changing the Housing Sector [Электронный ресурс]. – McKinsey & Company, 2023. – URL: <https://www.mckinsey.com> .
18. Google Workspace API Documentation [Электронный ресурс]. – URL: <https://developers.google.com/workspace>
19. gspread Documentation [Электронный ресурс]. – URL: <https://docs.gspread.org/en/v6.1.3/> (дата обращения: 16.05.2024).
20. HeadWork Analytics. Оптимизация ЖКХ с помощью чат-ботов [Электронный ресурс]. – 2023. – URL: <https://headwork.io>
21. Python. Python 3.14.0 documentation. Python Software Foundation, 2001-2025. URL: <https://docs.python.org/3.14/>
22. Telegram Bot API [Электронный ресурс]. – URL: <https://core.telegram.org/bots/api>.
23. Unreal Engine. Vulkan Support Documentation [Электронный ресурс]. – Режим доступа: [<https://docs.unrealengine.com>] (<https://docs.unrealengine.com>), свободный. - Дата обращения: 11.05.2025.
24. Unity Technologies. Graphics API Support [Электронный ресурс]. – Режим доступа: [<https://docs.unity3d.com>] (<https://docs.unity3d.com>), свободный. - Дата обращения: 11.05.2025.
25. Valve. Source Engine Documentation [Электронный ресурс]. – Режим доступа: [https://developer.valvesoftware.com/wiki/Main_Page] (https://developer.valvesoftware.com/wiki/Main_Page), свободный. - Дата обращения: 11.05.2025.