

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,  
профессор

\_\_\_\_\_ Л.Б. Соколинский

«\_\_\_» \_\_\_\_\_ 2024 г.

**Разработка Telegram-бота для корпоративной системы  
ООО «ЭнергоИнжиниринг»**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 02.03.02.2024.308-313.ВКР

Научный руководитель,  
доцент кафедры СП, к.ф.-м.н.  
\_\_\_\_\_ А.Т. Латипова

Автор работы,  
студент группы КЭ-401  
\_\_\_\_\_ Г.В. Селиванов

Ученый секретарь  
(нормоконтролер)  
\_\_\_\_\_ И.Д. Володченко  
«\_\_\_» \_\_\_\_\_ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_ Л.Б. Соколинский

29.01.2024 г.

### **ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра**

студенту группы КЭ-401

Селиванову Григорию Викторовичу,

обучающемуся по направлению

02.03.02 «Фундаментальная информатика и информационные технологии»

**1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)

Разработка Telegram-бота для корпоративной системы

ООО «ЭнергоИнжиниринг».

**2. Срок сдачи студентом законченной работы:** 03.06.2024 г.

**3. Исходные данные к работе**

3.1. Приказ ООО «ЭнергоИнжиниринг» №18 от 10.02.2024 г. «О разработке корпоративной системы электронного документооборота» (дата обращения: 11.02.2024 г.).

3.2. Moodle Developer Documentation. [Электронный ресурс] URL:  
[https://docs.moodle.org/dev/Main\\_Page](https://docs.moodle.org/dev/Main_Page) (дата обращения: 11.02.2024 г.).

3.3. System.IO library. [Электронный ресурс] URL:  
<https://learn.microsoft.com/ru-ru/dotnet/api/system.io?view=net-8.0> (дата обращения: 11.02.2024 г.).

3.4. Microsoft Documentation. [Электронный ресурс] URL:  
<https://learn.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-8.0> (дата обращения: 11.02.2024 г.).

3.5. Metanit Documentation. [Электронный ресурс] URL:  
<https://metanit.com/sharp/mvc/12.1.php> (дата обращения: 12.02.2024 г.).

3.6. Шарапов Н.Р. Эффективность применения технологии Asp. Net Core для разработки веб-приложений. // Вопросы науки и образования, 2018. – №. 13 (25). – С. 1–2.

**4. Перечень подлежащих разработке вопросов**

- 4.1. Провести анализ предметной области.
- 4.2. Определить требования к разрабатываемому Telegram-боту.
- 4.3. Спроектировать и реализовать Telegram-бота.
- 4.4. Провести тестирование реализованного Telegram-бота.

**5. Дата выдачи задания:** 29.01.2024 г.

**Научный руководитель,**  
доцент кафедры СП, к.ф.-м.н.

А.Т. Латипова

**Задание принял к исполнению**

Г.В. Селиванов

## **ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	7
1.1. Описание предметной области .....	7
1.2. Анализ существующих решений.....	8
1.3. Обзор средств разработки .....	12
2. ПРОЕКТИРОВАНИЕ .....	16
2.1. Определение требований системы .....	16
2.2. Диаграмма вариантов использования .....	17
3. АРХИТЕКТУРА СИСТЕМЫ.....	20
3.1. Описание классов и моделей системы.....	20
3.2. Описание компонентов системы .....	22
3.3. Диаграмма последовательности .....	23
3.4. Диаграмма деятельности.....	24
4. РЕАЛИЗАЦИЯ СИСТЕМЫ .....	26
4.1. Подготовка к реализации .....	26
4.2. Реализация Telegram-бота.....	27
4.3. Реализация интерфейса пользователя.....	33
4.4. Развертывание Telegram-бота на удаленном сервере .....	36
5. ТЕСТИРОВАНИЕ СИСТЕМЫ.....	38
5.1. Функциональное тестирование .....	38
5.2. Юзабилити-тестирование.....	39
ЗАКЛЮЧЕНИЕ .....	41
ЛИТЕРАТУРА.....	42
ПРИЛОЖЕНИЕ. Варианты использования системы .....	44

## **ВВЕДЕНИЕ**

### **Актуальность**

В современном цифровом мире все люди пользуются различными мессенджерами, для которых можно разрабатывать боты. Поэтому с точки зрения управления предприятием боты являются удобным инструментом для оперативного получения уведомлений или просмотра необходимой информации [1].

Также актуальность бота в мессенджере Telegram обусловлена необходимостью ООО «ЭнергоИнжиниринг» в оперативном донесении информации до сотрудников, автоматизации процессов просмотра некоторой информации. Внедрение данного продукта дает возможность сотрудникам, не заходя в основную систему, иметь всегда актуальную информацию о ходе работы.

Еще одно преимущество использования Telegram-бота – это удобство и доступность: так как Telegram – широко распространенная программа обмена сообщениями, которую многие люди уже используют на своих мобильных устройствах.

### **Постановка задачи**

Целью выпускной квалификационной работы является разработка Telegram-бота для корпоративной системы ООО «ЭнергоИнжиниринг». Для достижения поставленной цели необходимо решить следующие задачи:

- 1) проанализировать предметную область;
- 2) определить требования к разрабатываемому Telegram-боту;
- 3) спроектировать архитектуру Telegram-бота;
- 4) реализовать Telegram-бота;
- 5) провести тестирование Telegram-бота.

## **Структура и содержание работы**

Работа состоит из введения, пяти глав, заключения и списка литературы. Объем работы составляет 48 страниц, объем списка литературы – 20 источников.

В первой главе производится анализ предметной области работы. Обзор существующих аналогов и выбор инструментов для выполнения поставленных задач.

Вторая глава посвящена определению функциональных и нефункциональных требований по разработке Telegram-бота. На основании данных требований представлена диаграмма вариантов использования системы и ее спецификации.

В третьей главе описана архитектура разрабатываемой системы. Представлена диаграмма классов с подробным описанием ее участников. Также спроектированы поведенческие диаграммы деятельности и последовательности.

В четвертой главе описаны программные средства необходимые для разработки, приведены детали реализации Telegram-бота, а также его развертывание на удаленном сервере.

В пятой главе произведено тестирование разработанного бота. В ней описаны виды тестирования системы, а также результаты тестирования.

В приложении содержится информация по каждому варианту использования системы.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Описание предметной области

Telegram-бот – это программа, которая запрограммирована на совершение определённых действий, автоматизации задач и взаимодействие с пользователями в мессенджере Telegram [2].

ООО «ЭнергоИнжиниринг» – строительная компания, предлагающая полный спектр услуг и работ в сфере строительства, осуществляющая строительство объектов любой сложности. Одни из направлений деятельности компании: строительство современных мини-ТЭС (теплоэлектростанций), электрических станций и подстанций, линий электропередач любой сложности, выполнение пусконаладочных работ [3].

Для строительной компании очень важно оптимизировать бизнес-процессы и взаимодействие своих работников. Ввиду специфики деятельности компании, на рынке нет готового продукта, который бы решил все поставленные задачи, поэтому компанией было принято решение разработать собственную корпоративную систему, объединяющую и автоматизирующую различные бизнес-процессы в единую экосистему. Цель, предметная область системы, а также постановка задач описаны в приказе о разработке корпоративной системы ООО «ЭнергоИнжиниринг». Исходя из текста приказа, основная задача Telegram-бота будет заключаться в оптимизации выполнения различных задач. В рамках текущей работы предлагается следующее решение поставленной задачи: разработка Telegram-бота для работы с корпоративной системой.

Пользователи разрабатываемого продукта смогут:

- 1) получать активные уведомления о состоянии задач, в которых они являются исполнителями или создателями;
- 2) просматривать список своих задач в основной системе;
- 3) просматривать список своих писем в основной системе;
- 4) получать выгрузку по поставщикам определенной заявки в виде excel-документа;

5) оперативно получать информацию по работе определенных элементов основной системы.

## 1.2. Анализ существующих решений

Перед проектированием Telegram-бота для корпоративной системы ООО «ЭнергоИнжиниринг» необходимо провести обзор существующих реализаций схожей тематики. Это поможет определиться с общей структурой, а также выделить положительные аспекты реализации и закономерности, которые могут быть учтены в работе.

### **Telegram-бот компании «ИнфоСофт»**

Данный Telegram-бот используется в корпоративной системе строительной компании для быстрого согласования заявок с контрагентами [4]. Его основным назначением является оперативное утверждение заявок, присланных контрагентами на согласование. При появлении различных заявок, которые необходимо согласовать, Telegram-бот оповещает об этом пользователя.

Главное меню состоит из нескольких пунктов в виде кнопок (рисунок 1).

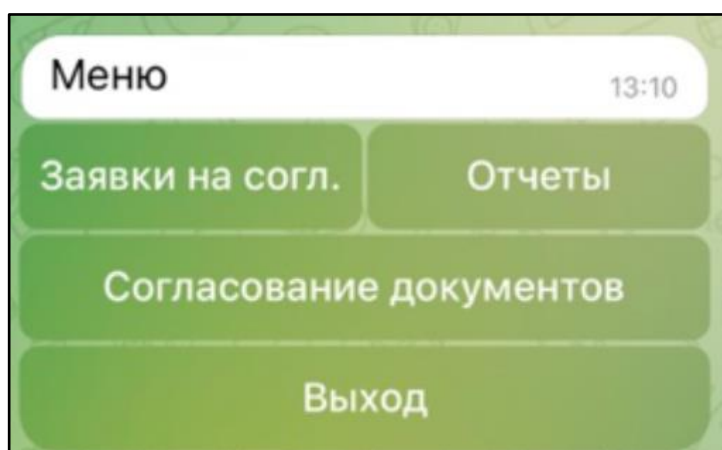


Рисунок 1 – Главное меню Telegram-бота «ИнфоСофт»



При переходе в пункт «Заявки на согласование» открывается список заявок с указанием основных данных. Далее выбирается определенная заявка путем нажатия на кнопку.

В информации о заявке содержатся данные о контрагенте, описание заявки, сумма, дата оплаты, а также отправленные контрагентом документы, которые можно сохранить к себе на телефон или посмотреть подлинный документ формата pdf. Пример вывода информации о заявке представлен на рисунке 2.

Внизу можно найти три кнопки, предназначенные для удобного выбора действий с текущей заявкой, а также кнопку «Назад» для быстрой навигации по системе.

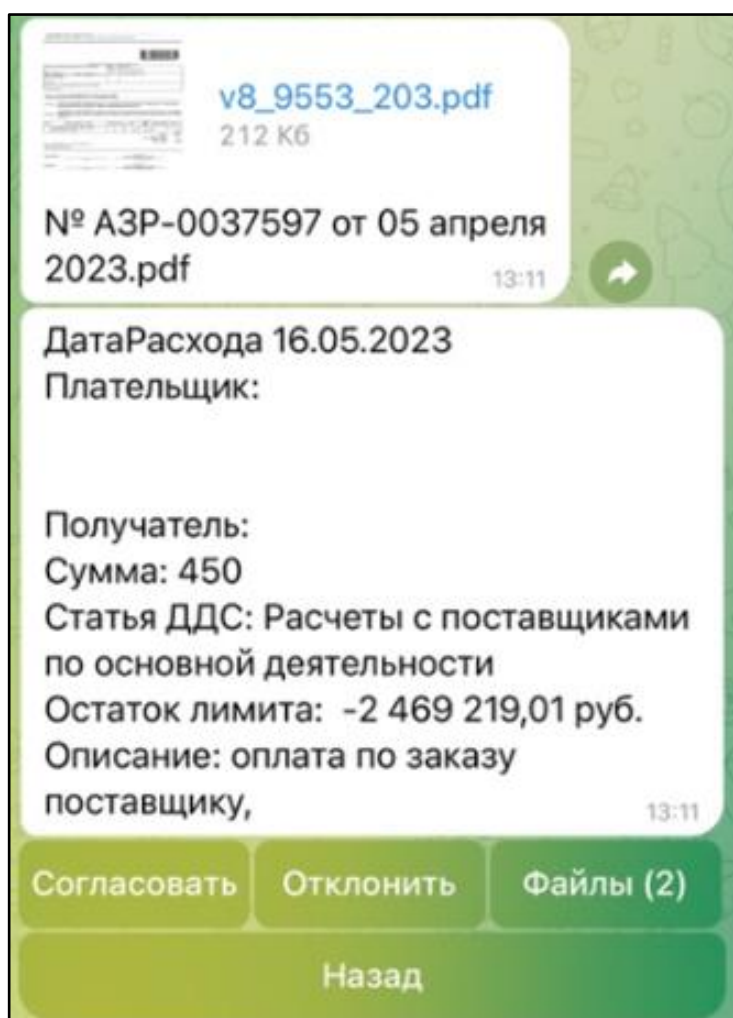


Рисунок 2 – Информация о заявке Telegram-бота «ИнфоСофт»

Достоинства:

- 1) удобное расположение функций в виде кнопок;
- 2) при получении информации о заявке дополнительно подгружается основной документ.

Недостатки:

- 1) большое количество свободного места, если не заполнены какие-то поля (например, платательщик на рисунке 2);
- 2) отсутствие начального описания бота.

### **Telegram-бот для 1С системы**

К основным функциям Telegram-бота относится заполнение фактического рабочего времени, просмотр текущих задач и просмотр данных о контрагенте [5].

Прежде чем взаимодействовать с ботом, пользователь должен авторизоваться в системе. Для этого он вводит ФИО пользователя, зарегистрированного в системе 1С. Далее, при нахождении соответствующего пользователя, бот отправляет на почту письмо с кодом, в котором находится специальный код. Данный код необходимо сообщить Telegram-боту (рисунок 3).

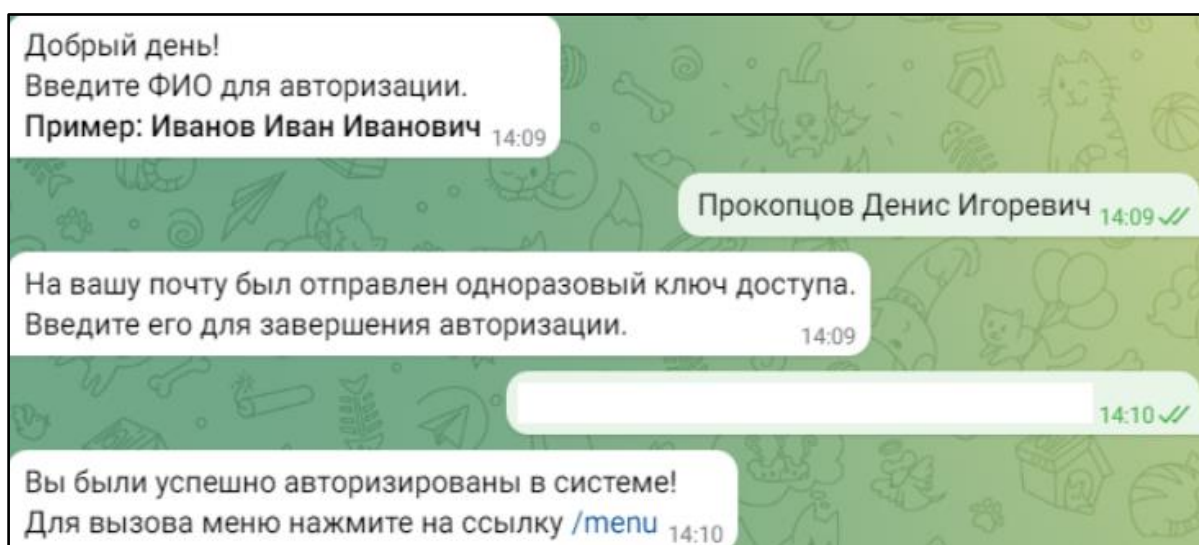


Рисунок 3 – Вход в систему Telegram-бота через аккаунт 1С

После успешной авторизации пользователю предлагается выбрать текущие незавершенные задачи либо задачи, которые находятся у пользователей на тестировании. Снизу доступны кнопки, нажав на которые пользователь попадает в самую задачу.

При нажатии на кнопку просмотра задачи, пользователь можно увидеть ее описание, название, контрагента, тип и номер. Также присутствуют основные контактные данные, необходимые для быстрого установления контакта и обсуждения различных вопросов. Пример такой задачи представлен на рисунке 4.

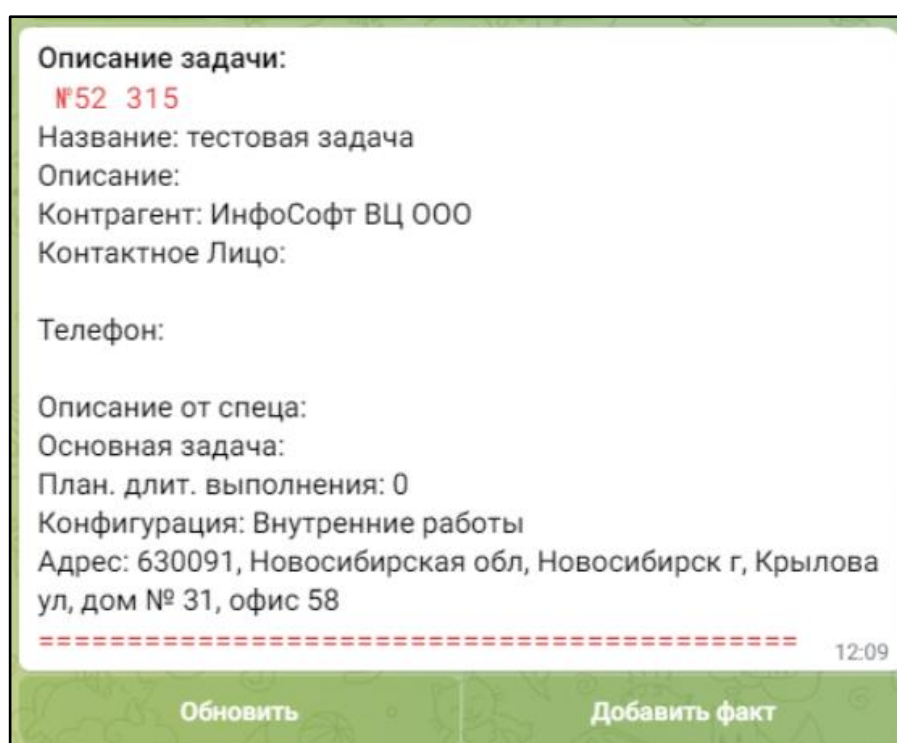


Рисунок 4 – Описание задачи

Если нажать на кнопку «Добавить факт», можно занести рабочее время. На рисунке 5 видно, как примерно может заполняться ввод фактического времени по задаче.

Достоинства:

1) лаконичное описание следующих действий, которые необходимо выполнить пользователю;

2) наличие дополнительной защиты в виде отправления одноразового ключа доступа на почту;

3) удобная навигация;

4) выделение статичного текста жирным шрифтом.

Недостатком является отсутствие полноразмерного календаря на месяц в виде кнопок.

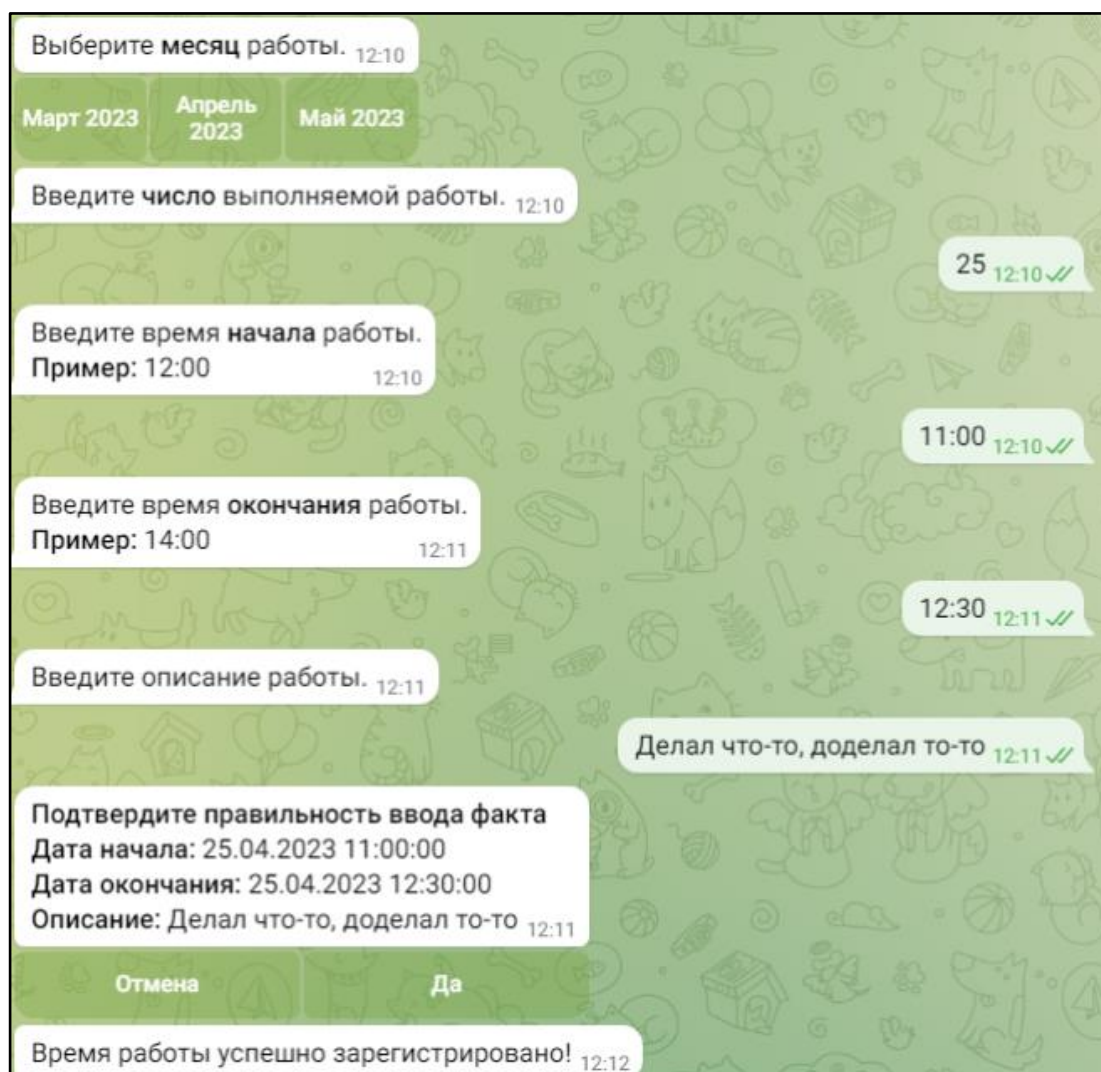


Рисунок 5 – Добавление табеля учета рабочего времени

### 1.3. Обзор средств разработки

Прежде всего требуется позаботиться о контроле за версиями разрабатываемой системы. Данную функцию отлично выполняет Git, который позволяет эффективно отслеживать изменения в коде, создавать резервные копии и легко откатываться к предыдущим версиям при необходимости,

что критически важно при внесении изменений в код. Еще одним преимуществом Git-репозитория является то, что код, находящийся под контролем Git, может быть легко перенесен на другие машины или в другие среды разработки. Это будет полезно при разворачивании Telegram-бота на удаленном сервере.

Для разработки серверной бизнес-логики был выбран язык программирования C#. Его преимущества состоят в том, что это кроссплатформенный язык общего назначения, который позволяет продуктивно писать высокопроизводительный код. С миллионами разработчиков C# является самым популярным языком .NET и имеет широкую поддержку в экосистеме. На основе объектно-ориентированных принципов он включает множество функций из других парадигм, а низкоуровневые функции поддерживают сценарии высокой эффективности без написания небезопасного кода [6].

Для написания кода также будет необходим фреймворк ASP.NET Core, который является кроссплатформенной, высокопроизводительной средой с открытым исходным кодом для создания современных облачных приложений, подключенных к Интернету [7].

Для хранения данных был выбран реляционный тип базы данных, потому как такой тип удобно использовать в случае, когда необходимо связать элементы данных между собой и безопасно управлять ими. При этом такая БД обеспечивает целостность данных в различных экземплярах базы в одно и то же время. Присутствует индексация, которая позволяет быстро находить данные. Также такой тип баз данных поддерживает модуль ORM, который в дальнейшем сильно упростит работу.

После детального изучения имеющихся реализаций таких типов баз данных, выбор пал на PostgreSQL. Такая база данных обладает всеми необходимыми требованиями и имеет огромное количество информации о решении всевозможных проблем и работе в целом [8].

Подключение к базе данных будет организовано через объектно-реляционный модуль сопоставления (ORM) Entity Framework (EF) Core.

Данная технология позволяет разработчикам .NET обеспечивать двустороннюю сохраняемость объектов во взаимодействии с источником данных. Это устраняет необходимость в большей части кода для доступа к данным, который разработчикам обычно приходится писать. Также нет необходимости создавать таблицы баз данных напрямую, потому что ORM будет самостоятельно создавать шаблон таблицы по созданной ранее модели, которую необходимо будет добавить в базу данных с помощью миграции [9].

Данная система позволит связать пользователя корпоративной системы и Telegram, не прилагая больших усилий. Для реализации логики будет необходима библиотека Telegram.Bot. Данная библиотека обладает обширными возможностями по созданию функционала Telegram-бота.

Webhook – это механизм, позволяющий веб-приложениям автоматически получать уведомления о событиях и обновлениях от других приложений и служб. Приложения, использующие веб-крючки, ждут, пока другие приложения отправят запросы на получение информации вместо того, чтобы постоянно опрашивать и проверять новые данные [10]. Это делает их более эффективными и экономными, поскольку они не тратят ресурсы на постоянные проверки. Вместо этого они получают нужные данные сразу, как только они становятся доступны. Веб-крючки широко используются для интеграции различных сервисов и приложений, где необходима мгновенная передача информации о событиях в режиме реального времени.

Ngrok – инструмент, который позволяет создавать временные туннели и обеспечивать публичный доступ к локально запущенным веб-серверам и приложениям. Он работает путем создания защищенного соединения между локальным сервером и облачным сервисом Ngrok, который проксирует входящие запросы к локальному серверу [11]. Данный сервис понадобится на этапе разработки Telegram-бота из-за отсутствия домена и SSL-сертификата на ЭВМ.

После реализации Telegram-бота необходимо позаботиться о его внедрении в компанию ООО «ЭнергоИнжиниринг». Для этого следует развернуть созданную систему в удаленной инфраструктуре операционной системы Linux. Подходящим сервисом для выполнения данных требований является AdminVPS [12]. Ключевыми преимуществами сервиса являются высокая производительность, обеспечиваемая мощными процессорами и быстрыми NVMe-накопителями, надежная и круглосуточная техническая поддержка с быстрым временем реакции, а также возможность выбора доменной зоны на территории Российской Федерации.

### **Вывод по первой главе**

После изучения средств разработки был сделан следующий вывод: необходимо создать Telegram-бота на языке программирования C# с использованием библиотеки Telegram.Bot. Приложение будет реализовано при помощи фреймворка ASP.NET Core, а работа механизма webhook на этапе разработки будет осуществлена благодаря инструменту Ngrok. Также стало понятно, что реализованного и идеально подходящего под запрос компании варианта на рынке нет. Заметим, что бот будет находиться в популярном приложении Telegram, благодаря чему пользователям не нужно будет ничего дополнительно устанавливать, а также изучать новый функционал для работы с системой.

## **2. ПРОЕКТИРОВАНИЕ**

### **2.1. Определение требований системы**

#### **Функциональные требования**

После обсуждения всех нюансов работы модуля были составлены следующие требования:

- 1) система должна авторизовать пользователя для работы с Telegram-ботом;
- 2) система должна выводить список всех возможных команд для удобства использования;
- 3) система должна позволять пользователям просматривать их активные уведомления и иметь возможность подписаться на автоматическую рассылку новых уведомлений;
- 4) система должна выводить список всех задач пользователя со сроками сдачи за сегодня или за любое время;
- 5) система должна выводить список входящих и исходящих писем пользователя;
- 6) система должна выгружать всех поставщиков из заявки в excel-документ и отправлять его пользователю;
- 7) система должна иметь справочник с инструкциями по работе в корпоративной системе.

#### **Нефункциональные требования**

Также в работе были выявлены нефункциональные требования:

- 1) обработка и синхронизация данных между проектируемой системой и ботом должны выполняться эффективно и в разумные сроки;
- 2) система должна быть доступной постоянно, чтобы пользователи могли иметь доступ к ней в любое время;
- 3) система должна быть интуитивно понятной и легкой в использовании, чтобы пользователи могли быстро освоить функционал и эффективно его использовать;
- 4) запросы к серверу должны осуществляться по протоколу HTTPS;



5) система должна обрабатывать ошибки и исключения, предоставляя информативные сообщения пользователю;

б) система должна быть способна обрабатывать одновременные запросы от нескольких пользователей.

## 2.2. Диаграмма вариантов использования

Для проектирования приложения был использован язык графического описания для объектного моделирования UML. На рисунке 6 представлена диаграмма вариантов использования Telegram-бота корпоративной системы ООО «ЭнергоИнжиниринг».

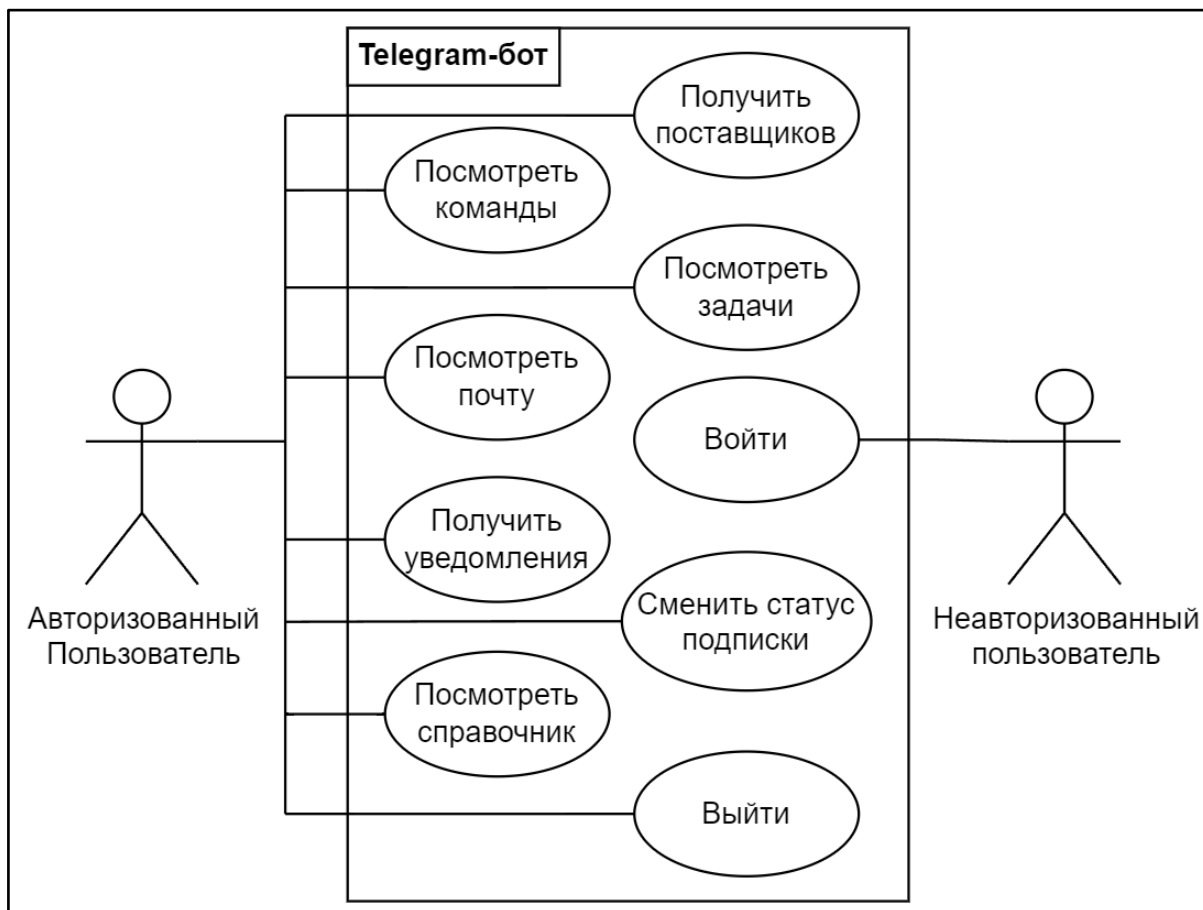


Рисунок 6 – Диаграмма вариантов использования Telegram-бота

Диаграмма имеет двух основных актеров: авторизованный пользователь системы и неавторизованный. Неавторизованный пользователь имеет всего один вариант использования.

Войти – зайти в систему и создать связь между своим Telegram-аккаунтом и аккаунтом корпоративной системы.

После авторизации, пользователю становятся доступны основные варианты использования.

1. Посмотреть команды – пользователь получает список всех основных команд.

2. Посмотреть задачи – пользователь получает список актуальных задач. Перед этим необходимо выбрать какие задачи хочет получить пользователь: свои, или те, в которых является ответственным; и за какой временной промежуток: за сегодня или за любое время.

3. Посмотреть почту – пользователь получает список писем, в которых он является ответственным. Перед этим необходимо выбрать какие письма хочет получить пользователь: входящие или исходящие.

4. Получить поставщиков – пользователь получает выборку поставщиков по конкретной заявке в формате excel-документа. Для этого необходимо выбрать объект, проект и саму заявку.

5. Получить уведомления – пользователь получает сообщение от Telegram-бота с информацией об изменении состояния заявок и задач, в которых он является ответственным за выполнение или создателем. А также уведомления о необходимости добавить таблицу за сегодняшний день при условии, что пользователь является ответственным за его создание.

6. Сменить статус подписки – пользователь меняет статус подписки на противоположный. Если подписка была неактивна, то она активируется и уведомления начинают приходить сразу при создании системой. Если подписка была активна, то она перестает быть активной, а уведомления больше не приходят в Telegram-бот.

7. Посмотреть справочник – пользователь получает информацию о работе с той или иной сущностью в формате видео или небольших сообщений.

8. Выйти из системы – пользователь выходит из системы и разрывает связь между своим Telegram-аккаунтом и аккаунтом проектируемой системы.

Данные варианты являются необходимыми и достаточными для выполнения требуемых задач.

### **Спецификация вариантов использования**

В приложении представлена спецификация основных вариантов использования системы.

### **Вывод по второй главе**

Из проведенного анализа предметной области были выведены как функциональные, так и нефункциональные требования к Telegram-боту. В ходе проектирования приложения была разработана основная концепция системы, включающая в себя цели, задачи и возможности, которые должен предоставлять Telegram-бот. Кроме того, на основе выявленных требований к системе была создана диаграмма вариантов использования, которая визуализировала взаимодействие пользователей с ботом в различных сценариях.

### 3. АРХИТЕКТУРА СИСТЕМЫ

#### 3.1. Описание классов и моделей системы

В результате анализа требований к системе была построена диаграмма классов, представленная на рисунке 7.

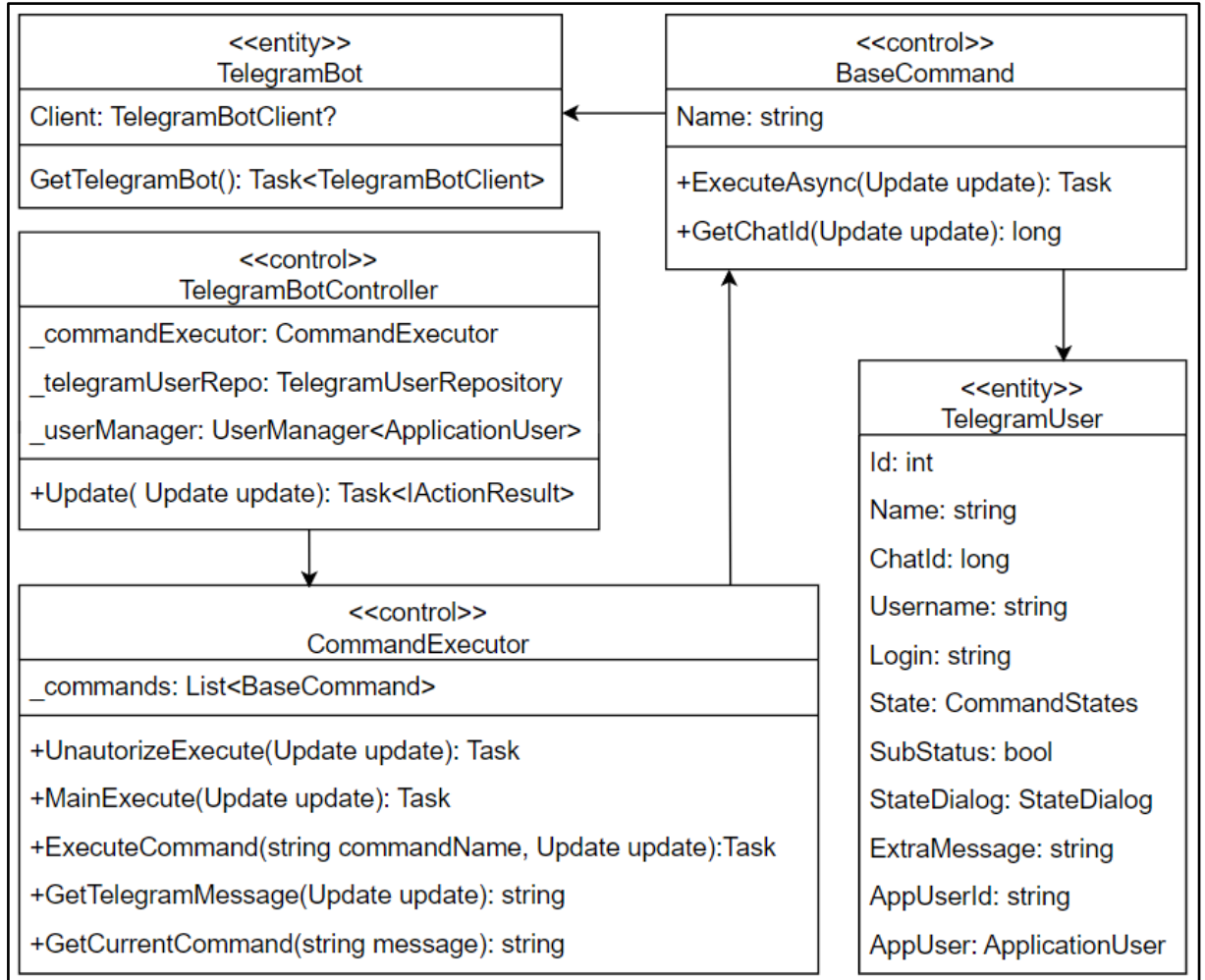


Рисунок 7 – Диаграмма классов

Для корректной работы были выделены классы-сущности пользователя и Telegram-бота.

1. `TelegramUser` – модель пользователя Telegram-ботом. Хранит часть персональных данных, состояние диалога и некоторую рабочую информацию о пользователе.

2. `TelegramBot` – готовая реализация клиента Telegram-бота, представленная сторонней библиотекой для использования Telegram Bot API.

Далее, для правильной обработки данных, были выделены классы поведения.

1. `TelegramBotController` – класс, отвечающий за получение контекста входящего сообщения, которое прислал бот.

2. `CommandExecutor` – класс, созданный для обработки входящего сообщения и определения команды для выполнения. Имеет два основных обработчика для авторизованных/неавторизованных пользователей.

3. `BaseCommand` – класс-родитель, является базовой реализацией для команд-наследников. Команды-наследники имеют название и два метода. В первом метода находится реализация команды Telegram-бота. Второй метод является вспомогательным для получения идентификатора чата.

### Модель базы данных

Для организации работы системы была разработана база данных, содержащая информацию о пользователях корпоративной системы и мессенджера Telegram. Схема разработанного хранилища данных представлена на рисунке 8.

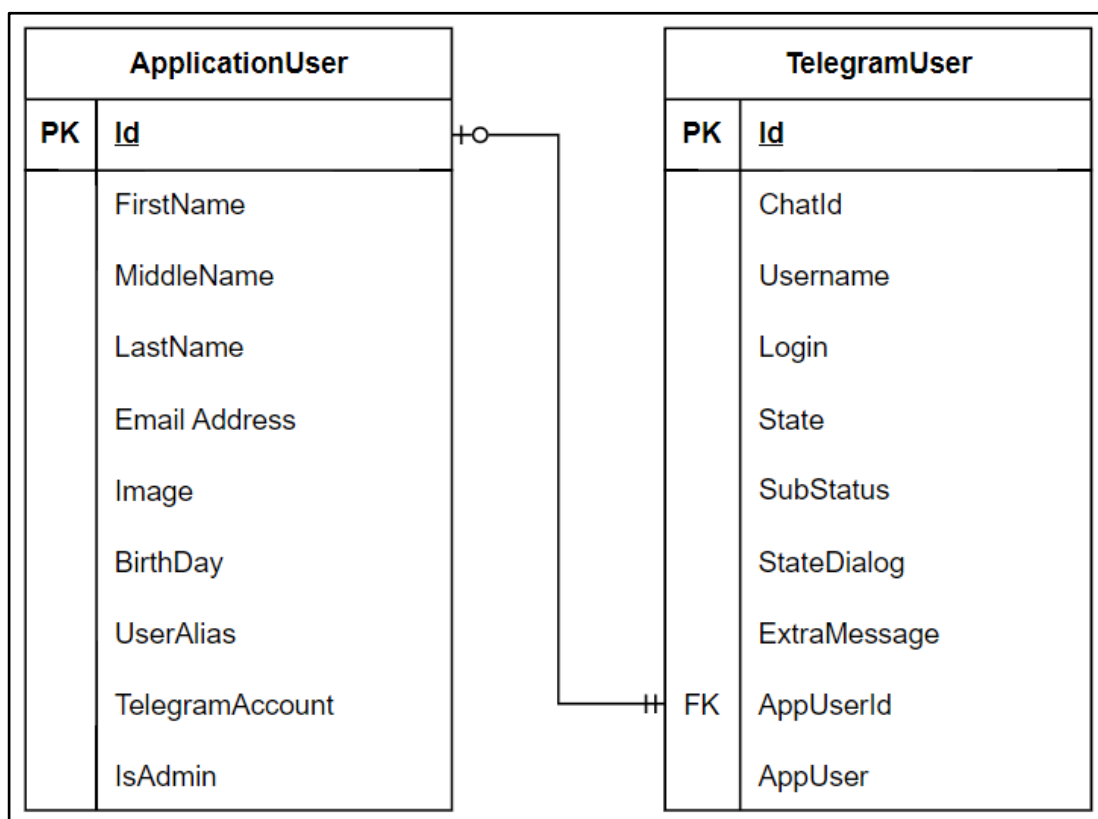


Рисунок 8 – Модель базы данных

Основная система использует базу данных с большим количеством моделей. В том числе в нее входят таблицы объектов, проектов, заявок, задач, уведомлений и писем, которые будут косвенно использоваться при реализации системы. Однако в контексте данной работы схема базы данных представляет собой сущность из двух основных таблиц.

1. `ApplicationUser` – основная модель пользователя корпоративной системы, которая содержит информацию о нем.

2. `TelegramUser` – модель пользователя мессенджером Telegram. Имеет внешний ключ, ссылающийся на идентификатор пользователя основной системы.

Стоит заметить, что Telegram-пользователь не может существовать отдельно от пользователя корпоративной системы.

### 3.2. Описание компонентов системы

На рисунке 9 описана диаграмма компонентов, которая представляет собой структурное разбиение программной системы, используемое для моделирования взаимодействия в системе.

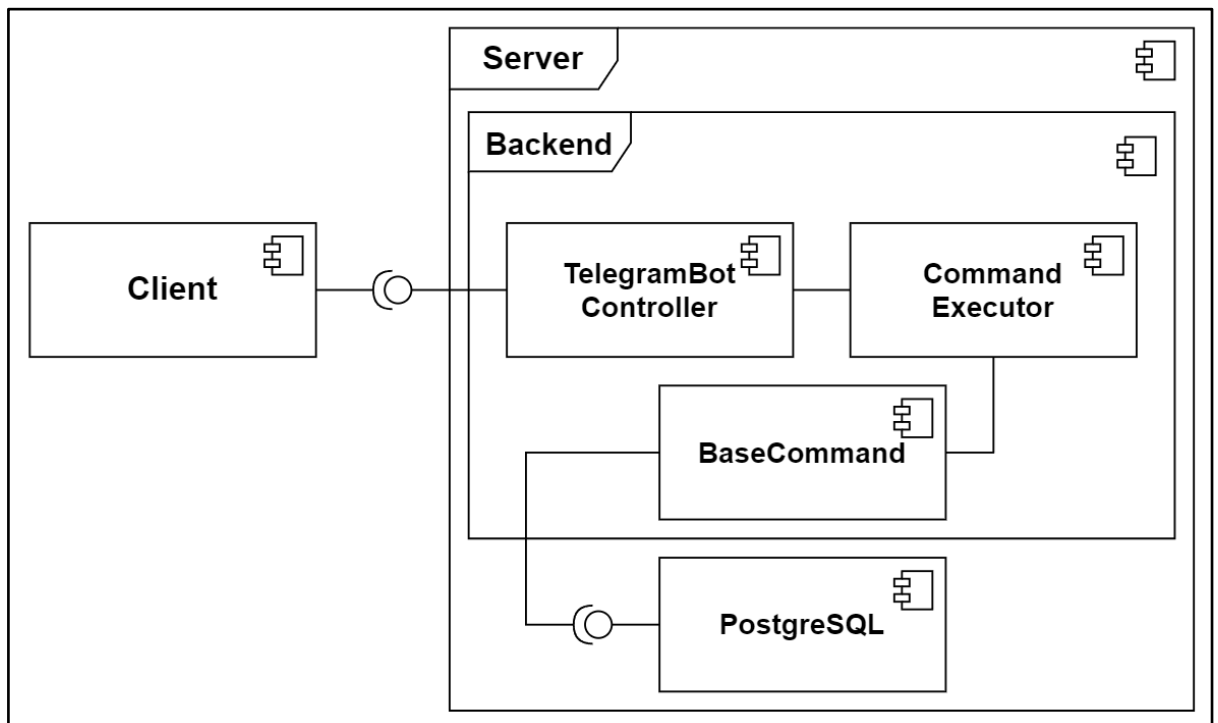


Рисунок 9 – Диаграмма компонентов

Все приложение делится на 4 основных компонента.

1. Client – это модуль, который содержит клиентскую часть системы. Интерфейс данной системы предоставляется Telegram.

2. Backend – текущий модуль реализует бизнес-логику системы. В данном компоненте происходит получение входящего сообщения от Telegram-бота, его обработка и переключение на необходимую команду с последующим выполнением.

3. PostgreSQL – данный компонент, отвечает за сохранение и поддержание целостности данных. В рамках разрабатываемой системы представлен в виде объектно-реляционной базы данных.

4. Server – компонент, в котором находится база данных и backend-часть приложения.

### **3.3. Диаграмма последовательности**

Пример взаимодействия пользователя с системой представлен на диаграмме последовательности для варианта использования «Получить уведомления», изображенной на рисунке 10.

Выполнение диаграммы начинается, когда пользователь нажимает кнопку: «Получить уведомления». Вызывается API-запрос для получения активных уведомлений пользователя. На первом этапе запрос попадает в TelegramBotController. В контроллере происходит получение идентификатора чата и проверка существования пользователя в системе. Далее, следующий код можно разделить на 2 части.

Первая часть наступает в том случае, если система не находит пользователя в системе. Происходит переход в обработчик для неавторизованных пользователей и вызов команды с сообщением о необходимости войти в аккаунт.

Вторая часть выполнится в том случае, если пользователь окажется авторизованным в системе. Произойдет вызов основного обработчика и переключение на команду по получению всех активных уведомлений

пользователя. Команда получит пользователя и выведет список его активных уведомлений.

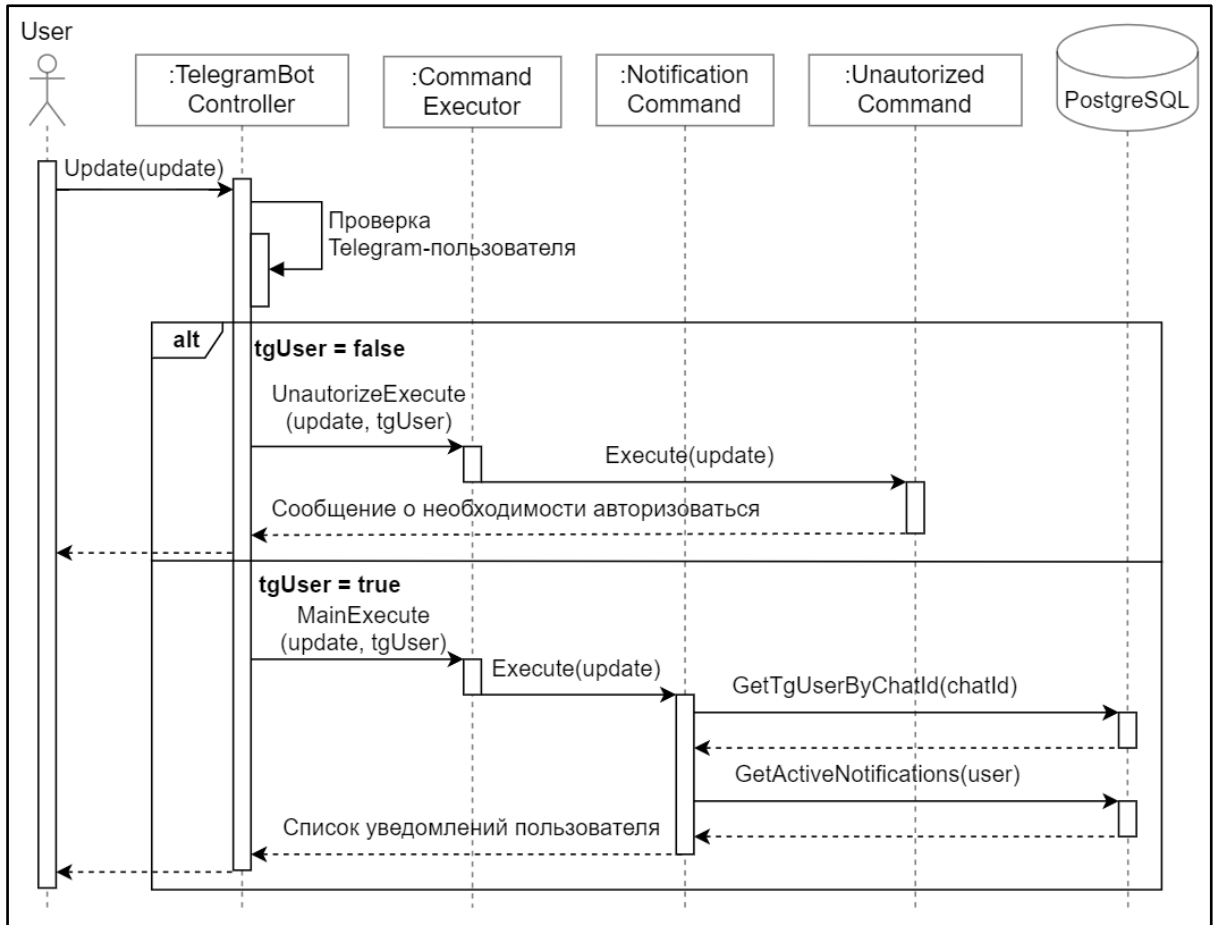


Рисунок 10 – Диаграмма последовательности

### 3.4. Диаграмма деятельности

Для детальной визуализации работы системы, ниже представлена диаграмма деятельности для варианта использования «Войти» (рисунок 11).



### **Вывод по третьей главе**

Было проведено описание архитектуры системы. Представлены диаграммы классов, компонентов, последовательности и деятельности, показывающие необходимые сущности и описывающие поведение.

## **4. РЕАЛИЗАЦИЯ СИСТЕМЫ**

### **4.1. Подготовка к реализации**

#### **Средства разработки**

В ходе обзора средств разработки в пункте 1.3, было выявлено, что наиболее удобной и подходящей средой разработки является Visual Studio 2022 – интегрированная среда разработки (IDE), которая предоставляет широкий набор функциональных возможностей для разработки программного обеспечения [13]. Ее использование позволит грамотно подойти как к написанию кода, так и к его последующей отладке.

В качестве основного языка разработки программной части был выбран язык C#. Для создания безопасного временного туннеля, обеспечивающего связь локального сервера с интернетом, был выбран сервис Ngrok версии 3.2.2.

## **4.2. Реализация Telegram-бота**













### **4.3. Реализация интерфейса пользователя**





#### **4.4. Развертывание Telegram-бота на удаленном сервере**

### **Вывод по четвертой главе**

Были определены основные средства для разработки и реализации работы логики бота. Кроме того, разобраны и описаны основные классы и модели для создания системы. Также были реализованы текстовый, графический и комбинированный интерфейсы для коммуникации пользователя с Telegram-ботом. После реализации бот развернут на сервере.

## 5. ТЕСТИРОВАНИЕ СИСТЕМЫ

### 5.1. Функциональное тестирование

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности ПО в определенных условиях решать задачи, необходимые пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает [19].

Итоги функционального тестирования разработанного Telegram-бота для корпоративной системы ООО «ЭнергоИнжиниринг» представлены в таблице 1.

Таблица 1 – Функциональное тестирование Telegram-бота

№	Наименование теста	Шаги	Ожидаемый результат	Результат теста
1	Начало работы.	1. Перейти к боту в первый раз. 2. Нажать кнопку «Начать».	Telegram-бот выведет приветственное сообщение.	Пройден.
2	Вход в систему с корректными данными для входа.	1. Нажать кнопку «Войти». 2. Ввести корректный логин пользователя. 3. Ввести корректный пароль пользователя.	Telegram-бот выведет сообщение о успешной авторизации и выведет кнопку «Посмотреть команды».	Пройден.
3	Отсутствие доступа у неавторизованного пользователя.	Отправить ключевое слово «/list» для получения списка команд.	Telegram-бот выведет сообщение о необходимости входа в систему и кнопку «Войти».	Пройден.
4	Просмотр команд.	Нажать на кнопку «Посмотреть команды».	Telegram-бот выведет основные команды в виде кнопок и сопроводительный текст.	Пройден.
5	Подписка на получение уведомлений.	1. Нажать на кнопку «Посмотреть команды». 2. Нажать на кнопку «Подписаться на уведомления».	Telegram-бот начнет отправлять сообщения пользователю автоматически.	Пройден.

№	Наименование теста	Шаги	Ожидаемый результат	Результат теста
6	Выгрузка excel-документа с поставщиками.	1. Нажать на кнопку «Посмотреть команды». 2. Нажать на кнопку «Объекты». 3. Выбрать объект. 4. Выбрать проект. 5. Выбрать заявку.	Telegram-бот создаст excel-документ с всеми утвержденными поставщиками из выбранной заявки и отправит этот документ пользователю.	Пройден.
7	Получение обучающего видео по созданию пользователя.	1. Нажать на кнопку «Посмотреть команды». 2. Нажать на кнопку «Справочник». 3. Нажать на кнопку «Создание пользователя».	Telegram-бот отправит обучающее видео, в котором показан алгоритм создания пользователя в корпоративной системе.	Пройден.
8	Просмотр своих задач на сегодня.	1. Нажать на кнопку «Посмотреть команды». 2. Нажать на кнопку «Задачи». 3. Нажать на кнопку «Мои задачи». 4. Нажать на кнопку «Получить за сегодня».	Telegram-бот выведет все задачи пользователя, где он является создателем задачи со сроком сдачи – сегодня.	Пройден.
9	Просмотр писем.	1. Нажать на кнопку «Посмотреть команды». 2. Нажать на кнопку «Письма». 3. Выбрать входящие или исходящие письма.	Telegram-бот выведет все входящие или исходящие письма пользователя, которые еще не завершены.	Пройден.
10	Выход из аккаунта.	1. Нажать на кнопку «Посмотреть команды». 2. Нажать кнопку «Выйти».	Telegram-бот удалит telegram-пользователя из базы и совершит выход пользователя.	Пройден.

## 5.2. Юзабилити-тестирование

Юзабилити-тестирование – это метод оценки удобства и эффективности интерфейса. В ходе оценки целевая аудитория продукта взаимодей-

ствуется с интерфейсом, выполняя специально подобранные задания. На основе поведения респондентов можно сделать выводы о наличии в интерфейсе проблем и их характере [20].

Текущая задача юзабилити-тестирования состояла в том, чтобы проверить дружелюбность, эффективность системы. Также необходимо проверить скорость ее работы под нагрузкой.

Для выполнения поставленной задачи было решено запустить Telegram-бота в проверочном режиме для использования сотрудниками ООО «ЭнергоИнжиниринг». Штат работников, тестируемых систему, составлял около 25 человек. Тестирование проводилось как на мобильных устройствах, так и с компьютерной версии Telegram.

Через некоторое время после начала тестирования были опрошены респонденты и получены отзывы и предложения по функционированию системы. Многие респонденты отметили интуитивность и простоту интерфейса. Также пользователи положительно высказались о скорости работы системы. Предложения и замечания также были учтены и внесены следующие правки: в некоторых местах был переписан текст ответных сообщений Telegram-бота, названия кнопок заменены на более короткие и лаконичные, изменена последовательность вывода кнопок в справочнике. После доработки системы Telegram-бот был отправлен на повторное юзабилити-тестирование.

После повторного тестирования все респонденты положительно высказались об использовании системы.

### **Вывод по пятой главе**

Для реализованного Telegram-бота было проведено функциональное тестирование, по итогам которого можно сказать, что все тесты пройдены успешно. Также было проведено юзабилити-тестирование, на основе его результатов были внесены небольшие правки и получена положительная реакция респондентов на систему в целом.



## **ЗАКЛЮЧЕНИЕ**

Целью данной работы являлась разработка Telegram-бота для корпоративной системы ООО «ЭнергоИнжиниринг». В ходе выполнения выпускной квалификационной работы были решены следующие задачи:

- 1) проанализирована предметная область;
- 2) определены требования к разрабатываемому Telegram-боту;
- 3) спроектирована архитектура Telegram-бота;
- 4) реализован Telegram-бот;
- 5) проведено тестирование Telegram-бота.

В настоящее время продолжается работа по совершенствованию и увеличению функционала Telegram-бота. В будущем планируется внедрить возможность утверждения задач, заявок и писем, а также добавить составление табеля учета рабочего времени.

На данный момент, система введена в промышленную эксплуатацию, что подтверждается актом о внедрении в строительную компанию ООО «ЭнергоИнжиниринг», и используется работниками для оперативного получения информации из корпоративной системы.

В результате выполнения выпускной квалификационной работы были достигнуты поставленные цели и решены все задачи.

## ЛИТЕРАТУРА

1. Бахтин И.В. Разработка чат-ботов для автоматизации бизнес-процессов. [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/razrabotka-chat-botov-dlya-avtomatizatsii-biznes-protsessov> (дата обращения 10.02.2024 г.).
2. Официальный сайт мессенджера Telegram. [Электронный ресурс] URL: <https://core.telegram.org/> (дата обращения 10.02.2024 г.).
3. Официальный сайт ООО «ЭнергоИнжиниринг». [Электронный ресурс] URL: <http://energoinz.ru/> (дата обращения: 10.02.2024 г.).
4. Telegram-бот компании «ИнфоСофт». [Электронный ресурс] URL: <https://is1c.ru/about/pc/article/telegram-boty/> (дата обращения: 10.02.2024 г.).
5. Telegram-боты для бизнеса. [Электронный ресурс] URL: <https://is1c.ru/services/avtomatizatsiya-torgovykh-predpriyatij/telegram-boty-dlya-biznesa/> (дата обращения: 10.02.2024 г.).
6. Документация по языку программирования C#. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 15.02.2024 г.).
7. Общие сведения о ASP.NET Core. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-8.0> (дата обращения: 15.02.2024 г.).
8. Документация базы данных PostgreSQL. [Электронный ресурс] URL: <https://postgrespro.ru/docs/postgresql/14/index> (дата обращения: 15.02.2024 г.).
9. Документация Entity Framework Core. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/ef/core/> (дата обращения: 15.02.2024 г.).
10. Marvin's Marvellous Guide to All Things Webhook. [Электронный ресурс] URL: <https://core.telegram.org/bots/webhooks> (дата обращения: 15.02.2024 г.).

11. Документация Ngrok. [Электронный ресурс] URL: <https://ngrok.com/docs/> (дата обращения: 15.02.2024 г.).
12. Облачное хранилище AdminVPS. [Электронный ресурс] URL: <https://adminvps.ru/> (дата обращения: 15.02.2024 г.).
13. Интегрированная среда разработки Visual Studio. [Электронный ресурс] URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 17.03.2024 г.).
14. Официальный сайт BotFather. [Электронный ресурс] URL: <https://telegram.me/BotFather> (дата обращения: 14.04.2024 г.).
15. Пользовательские интерфейсы. [Электронный ресурс] URL: <https://www.calltouch.ru/blog/glossary/interfejs/> (дата обращения: 14.04.2024 г.).
16. Использование SSH-ключей. [Электронный ресурс] URL: <https://selectel.ru/blog/ssh-keys/> (дата обращения: 14.04.2024 г.).
17. SSL-сертификат. [Электронный ресурс] URL: <https://www.kaspersky.ru/resource-center/definitions/what-is-a-ssl-certificate> (дата обращения: 14.04.2024 г.).
18. Центр сертификации Let's Encrypt. [Электронный ресурс] URL: <https://letsencrypt.org/ru/> (дата обращения: 14.04.2024 г.).
19. Медведев Ю.С., Терехов В.В. Автоматизация функционального тестирования web-приложений // Современные наукоемкие технологии. – 2016. – №. 5–1. – С. 58–61.
20. Речинский А.В., Сергеев С.Ф. Разработка пользовательских интерфейсов. Юзабилити-тестирование интерфейсов информационных систем: учебное пособие для студентов высших учебных заведений, обучающихся по направлению подготовки магистров «Информатика и вычислительная техника». – 2012. – №. 4–1. – С. 60–74.

## ПРИЛОЖЕНИЕ. Варианты использования системы

Спецификация вариантов использования (ВИ) представлена в таблицах 1–9.

Таблица 1 – Спецификация ВИ «Войти»

Вариант использования: Войти.
ID: 1
Краткое описание: Вход в систему с помощью корпоративного аккаунта пользователя.
Главные актеры: Неавторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает кнопку начать. 2. Telegram-бот выведет кнопку «Войти». 3. Пользователь нажмет на кнопку «Войти». 4. Telegram-бот попросит ввести логин пользователя. 5. Пользователь напишет свой логин и нажмет кнопку отправить. 6. Telegram-бот попросит ввести пароль пользователя. 7. Пользователь напишет свой пароль и нажмет кнопку отправить. 8. Если пользователь пройдет валидацию логина и пароля, то вызовется вариант использования номер 2. 9. Если пользователь не пройдет валидацию логина и пароля, то Telegram-бот выведет соответствующее сообщение.
Постусловия: Авторизованный пользователь.

Таблица 2 – Спецификация ВИ «Посмотреть команды»

Вариант использования: Посмотреть команды.
ID: 2
Краткое описание: Получение списка возможных команд пользователя.
Главные актеры: Авторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает кнопку «Посмотреть команды». 2. Telegram-бот выведет все основные команды.
Постусловия: Вывод основных команд Telegram-бота.

Таблица 3 – Спецификация ВИ «Посмотреть задачи»

Вариант использования: Посмотреть задачи.
ID: 3
Краткое описание: Просмотр задач пользователя.
Главные актеры: Авторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает кнопку «Задачи». 2. Telegram-бот выводит пояснительный текст и две кнопки: «Мои задачи» и «Назначенные мне». 3. Пользователь нажимает на любую из двух кнопок. 4. Telegram-бот выводит пояснительный текст и две кнопки: «Получить за сегодня» и «Получить все». 5. Если пользователь нажмет на кнопку «Получить за сегодня», Telegram-бот выведет список своих/назначенных задач пользователя со сроком сдачи до конца этого дня. 6. Если пользователь нажмет на кнопку «Получить все», Telegram-бот выведет список своих/назначенных задач пользователя за все время.
Постусловия: Получены требуемые задачи.

Таблица 4 – Спецификация ВИ «Посмотреть письма»

Вариант использования: Посмотреть письма.
ID: 4
Краткое описание: Просмотр писем пользователя.
Главные актеры: Авторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает на кнопку «Письма». 2. Telegram-бот выводит пояснительный текст и две кнопки: «Входящие» и «Исходящие». 3. Если пользователь нажимает на кнопку «Входящие», бот выводит список всех входящих писем пользователя. 4. Если пользователь нажимает на кнопку «Исходящие», бот выводит список всех исходящих писем пользователя.
Постусловия: Получены письма пользователя.

Таблица 5 – Спецификация ВИ «Получить поставщиков»

Вариант использования: Получить поставщиков.
ID: 5
Краткое описание: Просмотр поставщиков пользователя.
Главные актеры: Авторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает на кнопку «Объекты». 2. Telegram-бот выводит список объектов пользователя. 3. Пользователь нажимает на необходимый объект. 4. Telegram-бот выводит список проектов пользователя. 5. Пользователь нажимает на необходимый проект. 6. Telegram-бот выводит список заявок выбранного проекта. 7. Пользователь нажимает на необходимую заявку. 8. Telegram-бот отправляет excel-документ со всеми поставщиками из выбранной заявки.
Постусловия: Получены необходимые поставщики.

Таблица 6 – Спецификация ВИ «Получить уведомления»

Вариант использования: Получить уведомления.
ID: 6
Краткое описание: Вывод активных уведомления пользователя.
Главные актеры: Авторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает кнопку «Получить уведомления». 2. Telegram-бот выведет все активные уведомления пользователя.
Постусловия: Получение списка активных уведомлений.

Таблица 7 – Спецификация ВИ «Сменить статус подписки»

Вариант использования: Сменить статус подписки.
ID: 7
Краткое описание: Сменить статус подписки на уведомления.
Главные актеры: Авторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает кнопку «Подписаться на уведомления» или «Отменить подписку». 2. Если пользователь нажимает на кнопку «Подписаться на уведомления», бот подпишет пользователя на получение уведомлений и будет автоматически их присылать. 3. Если пользователь нажимает на кнопку «Отменить подписку», Telegram-бот отпишет пользователя от получения уведомлений и не будет присылать их в Telegram.
Постусловия: Смена статуса подписки на противоположный.

Таблица 8 – Спецификация ВИ «Посмотреть справочник»

Вариант использования: Посмотреть справочник.
ID: 8
Краткое описание: Просмотр информации о сущностях приложения.
Главные актеры: Авторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает на кнопку «Справочник». 2. Telegram-бот выводит список сущностей в виде кнопок и ссылку на текстовую документацию. 3. Если пользователь нажимает на текстовую документацию, то его переносит на word-документ с текстовой документацией всего приложения. 4. Если пользователь нажимает на кнопку какой-то сущности, бот выводит две кнопки: «Создание» и «Статусы». 5. Если пользователь нажимает на кнопку «Создание», бот отправляет обучающее видео по созданию сущности. 6. Если пользователь нажимает на кнопку «Статусы», бот выводит текстовую заметку с информацией о сущности.
Постусловия: Получена обучающая информация по работе с системой.

Таблица 9 – Спецификация ВИ «Выйти»

Вариант использования: Выйти.
ID: 9
Краткое описание: Выйти из аккаунта.
Главные актеры: Авторизованный пользователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток: 1. Вариант использования начинается, когда пользователь нажимает кнопку «Выйти». 2. Telegram-бот деактивирует аккаунт пользователя.
Постусловия: Неавторизованный пользователь.