

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

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки/ специальности)

Корпоративные информационные системы

(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему Разработка веб-приложения для работы с данными ФИАС

Обучающийся

А.М. Мамбеталиев

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Рогова

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

Консультант

к.п.н., доцент Т.С. Якушева

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

Тольятти 2022

## Аннотация

Тема бакалаврской работы является «Разработка веб-приложения для работы с данными ФИАС».

Ключевые слова: веб-приложение, моделирование информационной системы, база данных, логическое моделирование, диаграмма последовательности, диаграмма деятельности, ит-специалист и тестирование.

Данная выпускная квалификационная работа посвящена разработке веб-приложения для работы с данными справочника ФИАС.

Работа состоит из трех глав. В первой главе проведен анализ работ ООО «Управляющая компания» с предоставлением организационной структуры, проведен разбор существующих веб-приложений для работы с данными ФИАС, декомпозиции бизнес-процессов, разработана модель «как есть» и «как должно быть».

Во второй главе описан этап логического проектирования разрабатываемого веб-приложения с использованием объектно-ориентированного подхода.

В третьей главе рассмотрена разработка веб-приложения, а именно ее компонентов – сервера, панели ИТ-специалиста, базы данных. Также в третьей главе проводится описание процесса тестирования и представление результатов тестирования. В заключении показаны выводы и результаты о проделанной работе.

Итогом выпускной квалификационной работы является веб-приложение для организации быстрого поиска данных в федеральной информационной адресной системе.

Выпускная квалификационная работа состоит из пояснительной записки 48 страниц, ведения, включая 18 рисунков 10 таблиц и 20 источников.

## **Abstract**

This graduation work is about "Development of a web application for working with FIAS data.

Keywords: web application, information system modeling, database, logical modeling, sequence diagram, activity diagram, IT specialist and testing.

This graduation project is devoted to the development of a web application for working with the data of the FIAS directory.

The graduation work consists of three chapters and management.

In the first chapter, an analysis of the work of Management Company LLC with the provision of an organizational structure is carried out, an analysis of existing web applications for working with FIAS data, decomposition of business processes is carried out, a model "as is it" and "as it should be" was developed.

The second chapter describes the stage of logical design of the web application being developed using an object-oriented approach.

The third chapter discusses the development of a web application, namely its components - servers, IT panel, database. The third chapter also describes the testing process and provides the test results. The conclusion shows the conclusions and results of the work done.

The result of the final qualifying work is a web application for organizing a quick data search in the federal information address system.

The graduation project consists of an explanatory note on 48 pages, introduction, including 18 figures, 10 tables, the list of 20 references

## Оглавление

Введение .....	6
Глава 1 Функциональное моделирование предметной области .....	8
1.1 Техничко-экономическая характеристика ООО «Управляющая компания» .....	8
1.2 Концептуальное моделирование предметной области .....	10
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям.....	15
1.4 Постановка задачи на разработку веб-приложения .....	19
1.5 Разработка модели бизнес-процесса «как должно быть».....	20
Глава 2 Логическое моделирование веб-приложения для работы с данными ФИАС.....	23
2.1 Выбор технологии логического моделирования веб-приложения	23
2.2 Разработка логической модели веб-приложения.....	24
2.3 Реализация диаграммы последовательности .....	28
2.4 Реализация диаграммы деятельности .....	29
2.5 Требования к аппаратно-программному обеспечению веб-приложения.....	30
Глава 3 Физическое проектирование веб-приложения .....	32
3.1 Выбор архитектуры автоматизированной веб-приложения.....	32
3.2 Выбор технологии разработки программного обеспечения автоматизированной веб-приложения .....	33
3.3 Выбор системы управления базы данных веб-приложения .....	35
3.4 Описание функциональности, автоматизированной веб-приложение.....	37

3.5 Тестирование веб-приложения .....	39
Заключение .....	43
Список используемых источников.....	44
Приложение А Листинг программы.....	46

## Введение

В настоящее время интернет занимает большую часть в жизни человека. С каждым годом компьютерные сети развиваются и широкое распространение получили веб-приложения, позволяющие пользователю взаимодействовать с веб-сервером с помощью браузера для решения различных задач.

Актуальность темы заключается в поставленной задаче создание веб-приложение для работы с данными справочника федеральная информационная адресная система (ФИАС). Данное решение облегчит и ускорит пользование справочника ФИАС. Это приведет к точному определению адреса нужных мест вплоть до домов.

Объектом исследования данной выпускной квалификационной работы является организация для работы с данными справочника ФИАС.

Предмет исследования выпускной квалификационной работы является автоматизация процесса использования справочника ФИАС.

Целью выпускной квалификационной работы является разработка веб-приложения для работы с данными справочника ФИАС.

Для достижение поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области;
- провести анализы аналогов веб-приложения;
- разработать концептуальную модель;
- разработать логическую модель;
- выбрать среду разработки;
- разработать веб-приложение.

Данная бакалаврская работа заключается в выполнении поставленной задачи от компании ООО «Управляющая компания». Данная задача позволит компании упростить производить поиск информации об адресе справочника ФИАС.

Бакалаврская работа состоит из трех глав.

В первом главе анализируется технология концептуального моделирования, которая проектирует бизнес-процесс «как есть» и «как должно быть». Проведен анализ существующих веб-приложение для работы с данными ФИАС и выполнен сравнительный анализ существующих веб-приложения, а также требования к проектируемой веб-приложение.

Во втором главе описывается логическое проектирование веб-приложение. Во вторую главу входит технологи логического моделирование, построение логической модели веб-приложения для работы с данными ФИАС и описаны требования к аппаратно-программному обеспечению веб-приложения.

В третьем главе описывается физическое проектирование веб-приложение. Также выбор технологий разработки программного обеспечения, архитектура веб-приложения, СУБД и разработанной веб-приложения, а также осуществлена тестирование веб-приложения.

В заключение описывается вывод об эффективности разработанного веб-приложения.

# Глава 1 Функциональное моделирование предметной области

## 1.1 Технико-экономическая характеристика ООО «Управляющая компания»

Объектом для автоматизации является компания ООО «Управляющая компания». ООО «Управляющая компания» основным видом является оказание услуг по управлению многоквартирными жилыми домами.

Рассмотрим организационную структуру предприятия. В компании ООО «Управляющая компания» работает семь высших административных должностей: зам директор по производству, главный лаборант, зам директор по финансам, коммерческий директор, исполнительный директор, директор склада и логистики, директор по маркетингу.

Организационная структура представлена на рисунке 1

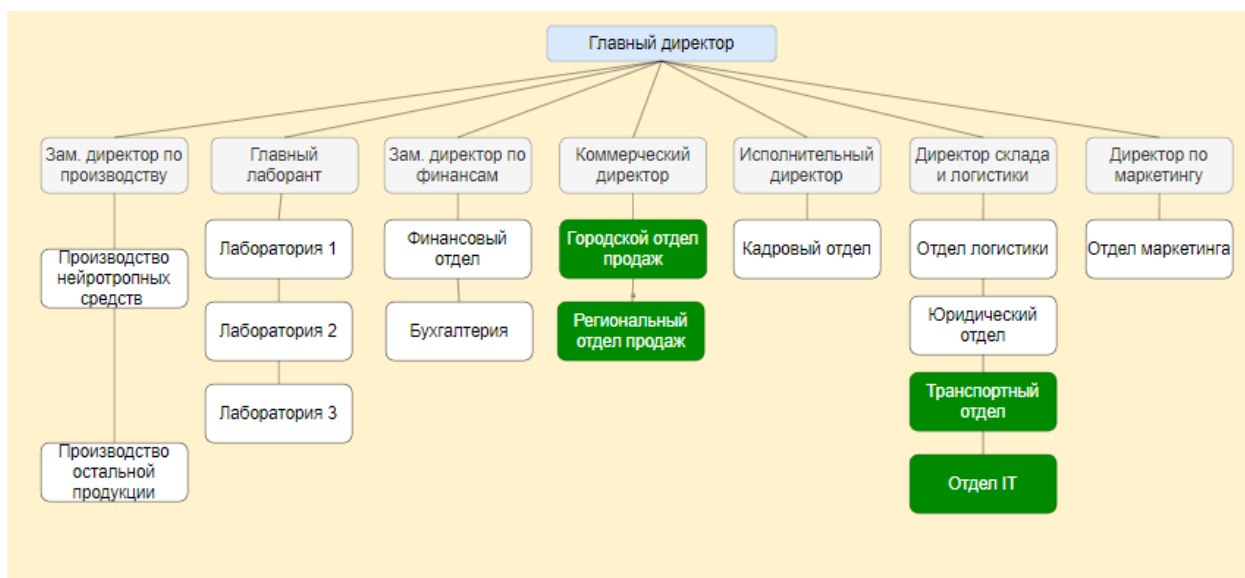


Рисунок 1 – Организационная структура ООО «Управляющая компания»

Все сотрудники организации напрямую подчиняются генеральному директору.



- директор реализовывает осмотр всех процессов предприятия, регулирует службу сотрудников, принимает сознательные заключения в критических обстановках и воспаляется раскруткой компании;
- исполнительный директор - это один из руководителей компании. Обязанностями исполнительного директора является разработка и внедрение долгосрочных планов для компании;
- коммерческий директор – это главный менеджер, который руководит продажами и покупками. Данный отдел будет взаимодействовать с нашим веб-приложением так как им надо получать информацию об адресах, с которыми они будут взаимодействовать.
- ИТ отдел обеспечение информационной безопасности, защиты информации, находящейся на электронных носителях от несанкционированного доступа, искажения и уничтожения при её передаче, обработке и хранении с применением средств вычислительной техники, телекоммуникаций и защиты информации. Данный отдел будет заниматься исправлением ошибок и редактированием информации;
- заместитель директора по производству отвечает за своевременный выпуск и качество продуктов.
- заместитель директора по финансовым вопросам относится к категории руководителей, принимается на работу и увольняется с работы приказом руководителя организации;
- главный лаборант относится к категории специалистов. Он выполняет лабораторные исследования и другие виды работы под руководством врача.

Процесс работа с данными ФИАС планируется осуществлять через веб-приложение. «Веб-приложения – приложения, обеспечивающие работу веб - сайтов. В большинстве своем, работа веб-приложений сводится к приему http-

запросов, обработке (иногда достаточно сложной) и формированию http-ответов».

Пользователь вводит поисковик нужный им адрес после чего происходит запрос на сервер, где происходит поиск данных по нужному адресу и найденные данные по адресу будут отправлены в веб-приложение, где пользователь может посмотреть всю нужную информацию. Данное веб-приложение будет работать через сервер компании, где будет осуществляться поиск нужных адресов для компании. Позволит компании облегчит и упростит поиск нужных им информации об адресе.

Далее рассмотрим процесс концептуального моделирования, то есть выберем технологии концептуального моделирования предметной области.

## **1.2 Концептуальное моделирование предметной области**

### **1.2.1 Выбор технологии концептуального моделирования предметной области**

Моделирование предметной области является одним из самых важных этапов. На данном этапе устанавливается все фрагменты предметной области. Для выбора методологии следует проанализировать ключевые аспекты методологии концептуального моделирования: BPMN, ARIS и IDEF0.

«Главное преимущество BPMN – она лучше всего подходит, если нужно описать именно бизнес-процесс, сделав его понятным даже для рядовых сотрудников. Сегодня BPMN пользуется популярностью: большинство вендоров, предлагающих системы BPM, предусматривают работу с BPMN: схему, созданную с её помощью, можно сделать исполняемой, подключив возможности веб-приложение» [11].

«Достоинство ARIS – простота для восприятия. Разноцветные элементы делают модель более «живой», приятной для глаз, а это немаловажно, если требуется нарисовать схему для сотрудников или провести презентацию. В отличие от предыдущей нотации, эта позволяет выстроить сложные развилки

и длинные параллельные ряды событий. Каждый элемент можно разложить на более мелкие элементы, построив для него отдельную схему» [11].

«Модель IDEF0 разворачивается одновременно слева направо и сверху вниз, по диагонали. Объекты, расположенные левее выше, доминируют над теми, которые находятся правее ниже. Доминирующие объекты могут включать в себя зависимые: например, доставка заказа – это элемент, входящий в состав более масштабного процесса управления заказами» [11].

Результат сравнительного анализа методологий представлен в таблице 1

Таблица 1 – Сравнительный анализ методологий

Критерий сравнений	Нотация BPMN	Нотация ARIS	Нотация IDEF0
Легкость познания	сложная	среднее	легко
Удобство по созданию моделей	сложно	сложно	легко
Подход к проектированию	Функциональный	Объектно-ориентированный	Функциональный
Система хранения данных модели	Модели хранятся в файлах	Модели хранятся в файлах	Модели хранятся в файлах
Возможность декомпозиции	неограниченная	неограниченная	Неограниченная

Из анализа, представленного в таблице 1 можно сделать вывод, что нотация IDEF0 является более подходящей для моделирования бизнес-процессов предприятия.

После разбора и выбора технологии концептуального моделирования, перейдем к проектированию модели бизнес-процесса «как есть».

### **1.2.2 Разработка и анализ модели бизнес-процесса «как есть»**

«Модель AS-IS («как есть») позволит отразить текущую ситуацию, систематизировать протекающие в организации процессы и информационные потоки в рамках этих процессов. На основе данной модели выявляются проблемные (узкие) места при выполнении и взаимодействии процессов, определяется необходимость внесения тех или иных изменений.

В результате анализа заказчик получает актуальную картину текущего состояния процессов организации и предложения по оптимизации ее деятельности. Данный подход может применяться как с целью определения процессов, подлежащих дальнейшей автоматизации, так и вести только к организационным изменениям.

Отдельно следует заметить, что в рамках проводимых преобразований одно из важнейших мест должно отводиться работе с людьми. Без воли и поддержки высшего руководства и без желания что-то изменить, разработанные модели бизнес-процессов могут так и остаться лишь описаниями с красивыми картинками, по которым никто не работает» [1].

Контекстная модель «Как есть» процесса «Запрос получение данных ФИАС» в нотации IDEF0, представлена на рисунке 2.



Рисунок 2 – Контекстная модель «Как есть»

Рассмотрим более подробно процесс «Запрос получение данных ФИАС», на вход поступает запрос данных, на выходе пользователь получает уже готовую информацию об определенном адресе.

Для подробного представление необыкновенности ключевого процесса осуществим его декомпозицию, то есть выделим подпроцессы. Модель выполненной декомпозиции «как есть» представлена на рисунке 3.

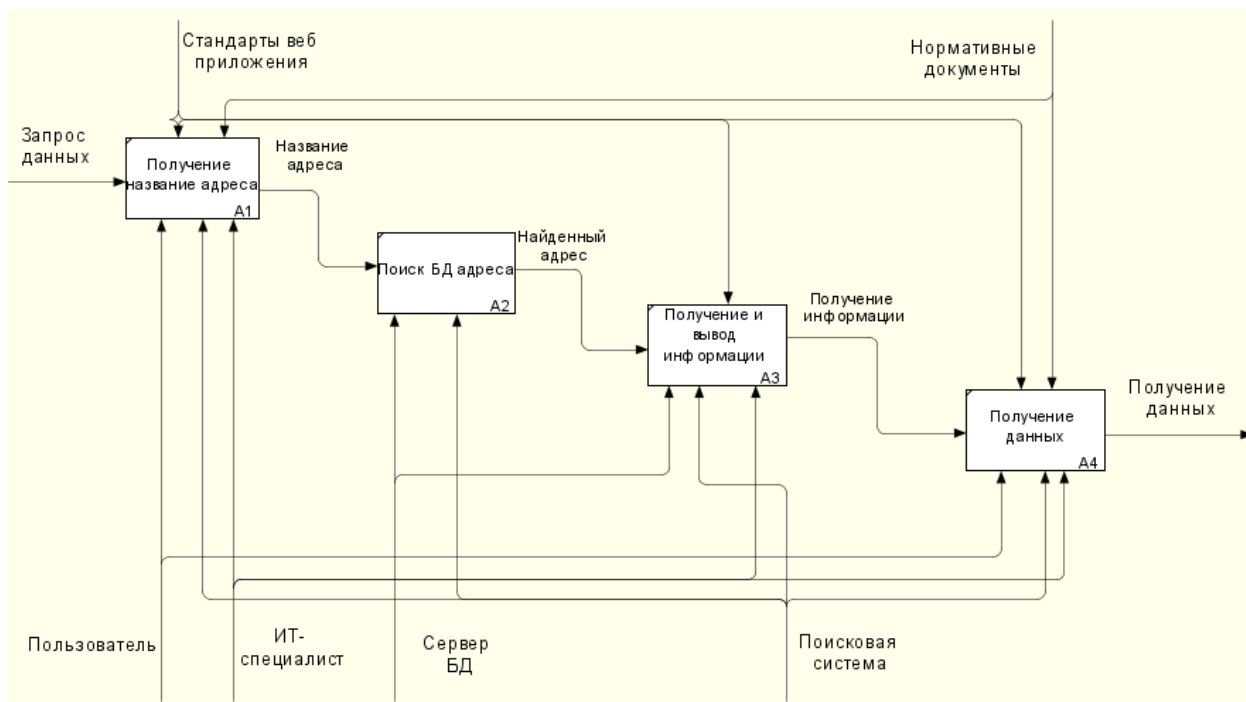


Рисунок 3 – Декомпозиция контекстной модели «как есть»

После постройки контекстной модели «как есть» рассмотрим поподробнее каждый процесс.

Процесс А1 «Получение наименование адреса» подразумевает, что пользователь делает запрос о получение подробной информации об определенном адресе, также пользователь сам может выбрать нужный им адрес из списка.

Процесс А2 «поиск БД адреса» подразумевает, что после того как пользователь выбрал нужный им адрес веб-приложение отправляет запрос в сервер для начала поиска в базе данных где хранятся информация об адресах.

Процесс А3 «получение и вывод информация об адресе» подразумевается, что веб-приложение получает и выводит подробную

информацию об определенном адресе, который ввел в поиск или выбрал пользователь.

Процесс А4 «получение данных» является заключительным и подразумевает, что веб-приложение получает данные информации об адресе в виде отчета для просмотра.

### **1.2.3 Разработка требования к веб-приложению для работы с данными ФИАС**

Разбор модели «как есть» показал, что существует бизнес-процесс имеют следующие недостатки:

- отсутствие быстрого поиска;
- отсутствие изменять или дополнять данные в базу пользователям;
- отсутствие выбора нескольких адресов для быстрого поиска.

После всего этого как были обнаружены все недостатки имеющего бизнес-процесса, нужно описать условия проектирования веб-приложения для работы с данными. Перейдем к отображению требований, которым соответственно будет согласоваться будущее веб-приложение.

«FURPS — это одна из распространенных классификаций требований к информационным системам. Название произошло от первых букв слов Functionality, Usability, Reliability, Performance, Supportability (Функциональность, Удобство использования, Надежность, Производительность, Поддерживаемость)» [9].

FURPS расшифровывается последующим образом:

- functionality – требования к функциональности: просмотр информации, просмотр данных, редактирование информации и просмотр информации через модальные окна;
- usability – требования к удобству: эффективность использования и минималистический дизайн;
- reliability – требования к надежности: проверка вероятных сбоев и проверка измененных информации;

- performance – требования к производительности: время отклика и эффективность;
- supportability – требования к поддержке: уведомление к службе поддержки, обновление и наращивания дополнительного функционала.

«Стоит отметить, что классификация FURPS+ содержит не только требования к ИС, но и ограничения, которые выделены в названии знаком «плюса». В рамках рассматриваемой классификации выделяют четыре вида ограничений:» [9]

- implementation (ограничения разработки) – ограничение на технологии, процесс средства разработки;
- interface (ограничения на интерфейсы) – стандартные разработки, стандарты качества;
- design (ограничения проектирования) – форматы данных, протоколы взаимодействия, внешние системы;
- physical (физические ограничения) – накладывается на технические средства.

Определив и описав условия проектируемого веб-приложения, перейдем к анализу имеющихся заключений в данной предметной области.

### **1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям**

Осуществим сравнительный разбор наиболее известных веб-приложения для работы с методанными ФИАС. Это необходимо для того, чтобы избежать ошибки других разработчиков при дальнейшей работе будущего веб-приложения.

Рассмотрим уже существующие аналоги веб-приложений для работы с методанными.

1. Федеральная информационная адресная система представлена на рисунке 4

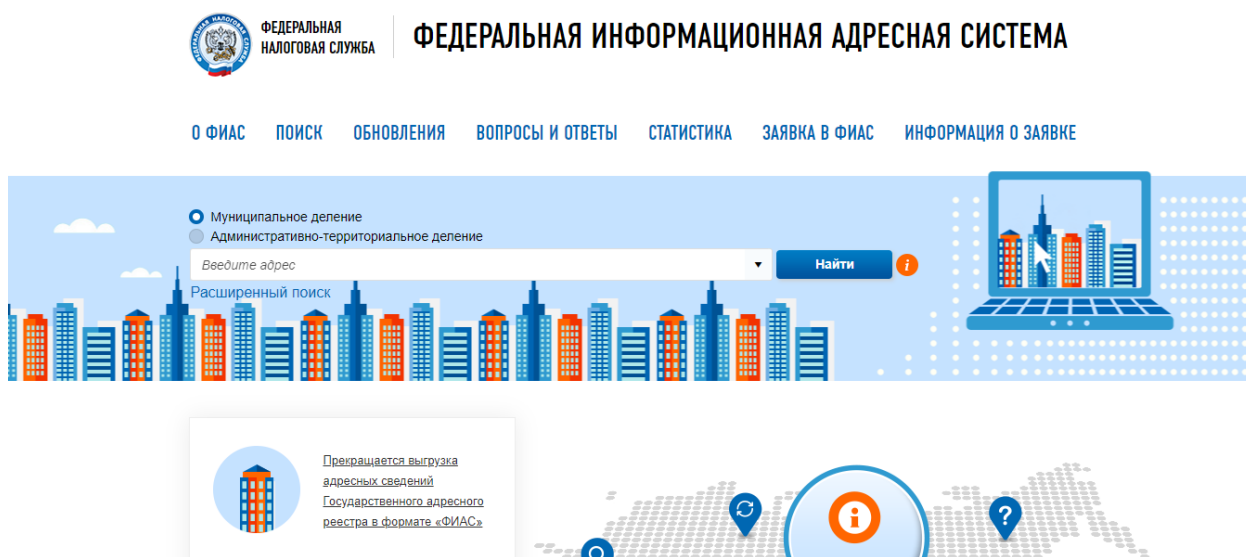


Рисунок 4 - федеральная информационная адресная система

Располагается по адресу <https://fias.nalog.ru/>, представляет собой веб-приложение который работает с данными ФИАС. Данная веб-приложение обладает рядом преимуществ.

- открытый доступ к базе ФИАС;
- детально показывает нужный нам адрес;
- хороший дизайн приложения;
- возможность посмотреть на карте адрес.

Данная веб-приложения рядом недостатков:

- неудобный интерфейс;
- низкая поспешность работы;
- необходимо время, чтобы сориентироваться как пользоваться

представленным сервисом.

2. ФИАС онлайн представлена на рисунке 5



## ФИАС онлайн

Федеральная информационная адресная система (ФИАС) - обеспечивает формирование, ведение и использование государственного адресного реестра. Целью создания ФИАС является формирование единого федерального ресурса, содержащего достоверную, единообразную, общедоступную, структурированную адресную информацию. Ведением ФИАС занимается [Федеральная Налоговая Служба Российской Федерации](#).

Проект ФИАС онлайн создан для бесплатного представления общедоступных данных из базы ФИАС в наглядной удобной форме.

### Субъекты Российской Федерации

Алтай Республика  
Архангельская область Область  
Белгородская область Область  
Владимирская область Область  
Воронежская область Область  
Город Санкт-Петербург Город

Ивановская Область  
Кабардино-Балкарская Республика Республика  
Камчатский край Край  
Кировская область Область  
Красноярский край Край

Алтайский край Край  
Астраханская область Область  
Брянская область Область  
Волгоградская область Область  
Город Байконур Город  
Еврейская автономная область Автономная область  
Ингушетия Республика  
Калининградская область Область  
Карачаево-Черкесская Республика Республика  
Костромская область Область  
Крым Республика

Амурская область Область  
Башкортостан Республика  
Бурятия Республика  
Вологодская Область  
Город Москва Город  
Забайкальский край Край  
Иркутская область Область  
Калужская область Область  
Кемеровская область - Кузбасс Область  
Краснодарский край Край  
Курганская область Область

Рисунок 5 - ФИАС онлайн

Располагается по адресу <https://xn--80ap2aj.xn--80asehdb/>, представляет собой сервер работающий с данными ФИАС. Из преимуществ можно выделить несколько:

- легко разобраться как работает данный сервер;
- быстрая скорость работы.

Данный сервер также обладает рядом недостатков:

- нет возможности посмотреть на карте адрес;
- плохой дизайн приложение;
- нет поиска для упрощения поиска.

### 3. Альта-софт представлен на рисунке 6

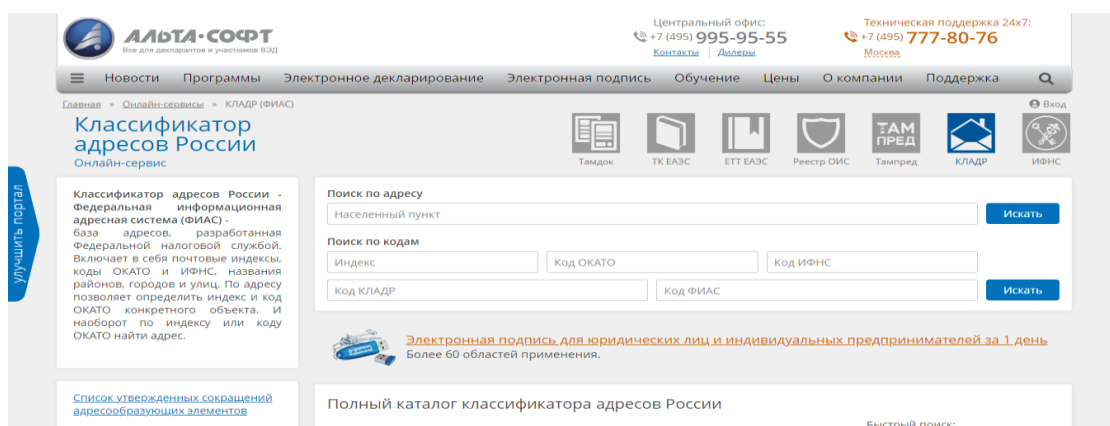


Рисунок 6 – Альта-софт

Располагается по адресу <https://www.alt.ru/fias/>, представляет собой веб-приложение который так же работает с данными ФИАС. Из преимуществ моно отметить несколько:

- удобный интерфейс;
- есть подробный поиск;
- быстрый поиск;
- легко разобраться как работает данный сервер;
- возможность посмотреть на карте адрес.

Также у этого сайта есть недостатки:

- довольно скучный дизайн;
- нет возможности посмотреть на карте адрес;
- не полная информация о нужном нам адресе.

Далее рассмотрим результат анализа веб-приложений для работы с данными ФИАС представлена в таблице 2.

Таблица 2 – Сравнение существующих веб-приложений

Требования	Fias.nalog	ФИАС онлайн	Альто-софт
Открытый доступ к базе	+	+	+
Возможность редактировать	-	-	-
Поиск	+	-	+
Удобный интерфейс	-	-	+
Итого	2/4	1/4	3/4

Исходя из анализа таблицы 2, можно сделать вывод, что каждая из выше перечисленных веб-приложений для работы с данными ФИАС имеет некоторые недостатки.

После того, как был проведен сравнительный анализ существующих веб-приложений для работы с данными ФИАС, следует сделать итог о необходимости проектирование и разработки нового веб-приложения для

работы с данными ФИАС, а также описать цели ее создания и основные требования к предстоящему веб-приложению.

#### **1.4 Постановка задачи на разработку веб-приложения**

По результату относительного анализа, выполненного в таблице 2 осуществляем вывод, что ни одна из ранее обозначенных веб-приложений в представленной предметной области не отвечает всем необходимым условиям компании. Следовательно, вновь удостоверяемся в том, что есть надобность к разработке нового веб-приложение для работы с данными ФИАС.

Сформулируем цель и основные условия к грядущему веб-приложению.

Целью проектирование веб-приложения является предоставление удобного веб-приложение для просмотра данных ФИАС.

Для достижения установленной цели были сформированы ключевые запросы к веб-приложению.

- удобный интерфейс;
- поиск адресов по определенным параметрам;
- редактирование и добавление адресов;
- всплывающее окно с подробной информацией.

Компания ООО «Управляющая компания» тратит много времени для получения информации об адресе. Именно поэтому необходимо создать веб-приложение, которое позволит сократить время получения информации об адресах РФ, также обеспечит более качественное и надежное выполнение работы.

После того, как задачи разработки определены, переходим к разработке модели бизнес-процесса «как должно быть».

## 1.5 Разработка модели бизнес-процесса «как должно быть»

Построенная контекстная диаграмма «как должно быть» демонстрирует будущее положение предметной области. Представленная диаграмма проектируется на основе контекстной модели «как есть» с исключением изъянов, а также, с их усовершенствованием и оптимизации в существующей веб-приложения. Это достигается посредством отстранения обнаруженных в течение разбора контекстной модели «как есть» узких мест. Контекстная модель «как должно быть» представлена на рисунке 7. Сравнительно с моделью «как есть» добавляется разрабатываемый механизм – «Веб-приложение», которое позволит взаимодействовать с базой данных ФИАС через веб-приложение

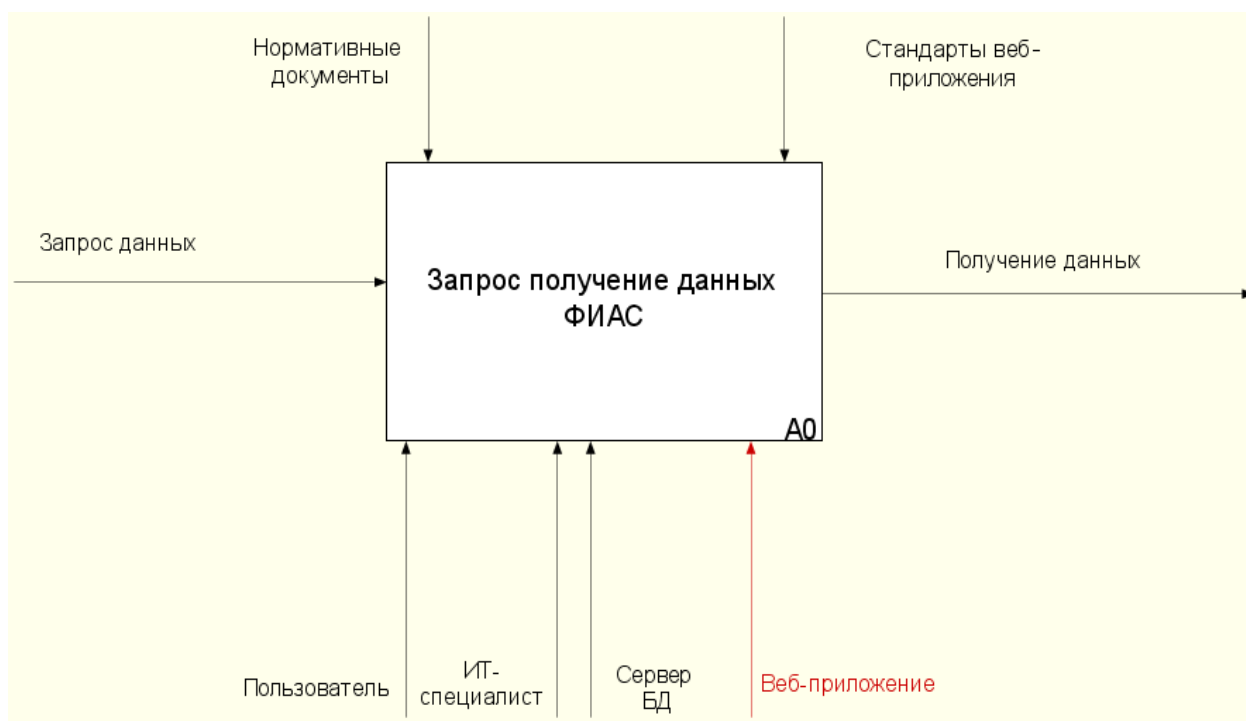


Рисунок 7 – Контекстная модель «как должно быть»

С помощью веб-приложения по работе с данными ФИАС пользователи могут получать информации об адресе в режиме реального времени. Это нужно для того что бы в компании всегда была достоверная информация. Так

же пользователю будет доступен список адресов, благодаря которому он может выбрать нужный ему адрес.

С помощью веб-приложение организация может присваивать определенным местам индивидуальный номер. Данная особенность позволит упростить поиск тех мест, которые уже были просмотрены.

Также проведем декомпозицию контекстной модели «как должно быть» результат представлен на рисунке 8.

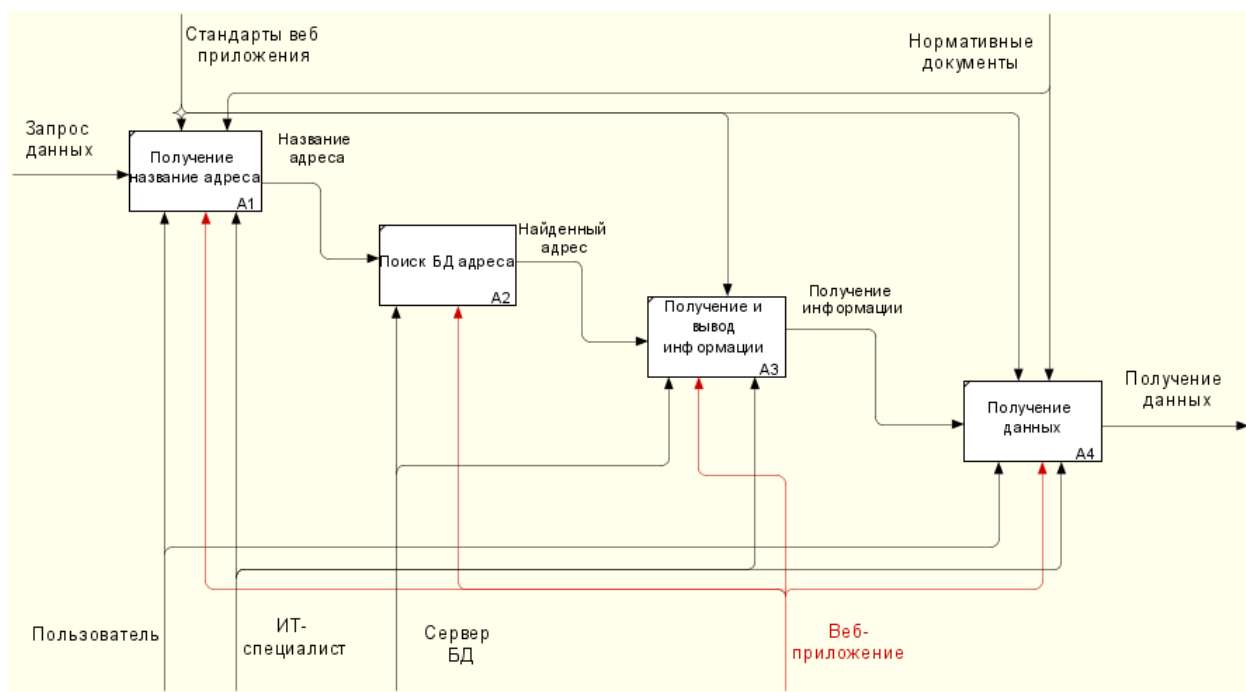


Рисунок 8 – Декомпозиция контекстной модели «как должно быть»

Рассмотрим декомпозицию контекстной модели «как должно быть» на данной диаграмме показано как именно будет взаимодействовать созданная веб-приложение. Данная веб-приложение упрощает пользователю просматривать нужную ему информацию об адресе.

После добавление веб-приложение поиск по адресу будет осуществляться значительно быстрее так как раньше все это проходило через поисковую систему где поиск информации по адресу проводилось дольше чем проектируемой веб-приложение.

После проектирования контекстной модели бизнес-процесса «как должно быть».

#### Выводы по первой главе

Первая глава записывает собой представление функционального моделирования предметной области. Проведен анализ технологии концептуального моделирования. При этом была разработана и проанализирована модель бизнес-процесса «как есть».

Затем был подготовлен результат, что часть процессов необходимо автоматизировать. Так же приведен сравнительный разбор имеющихся веб-приложений ради извлечения информации об адресах ФИАС и их сравнения.

Согласно проведенному разбору ни одна из выше перечисленных веб-приложений для работы с данными ФИАС не удовлетворяет всем достаточным условиям заказчика, как следует из данного осуществляем вывод что разработка новой веб-приложении положительно необходимо. Вдобавок были сформированы и описаны условия к проектируемого веб-приложению, затем была осуществлена установка задачи для разработки нового веб-приложению и проектирование модели бизнес-процессов «как должно быть», была представлена в первой главе.

## **Глава 2 Логическое моделирование веб-приложения для работы с данными ФИАС**

### **2.1 Выбор технологии логического моделирования веб-приложения**

«Логическое моделирование основано на анализе возможных исходов вследствие изменения отдельных элементов в состоянии управляемой системы, при функционировании которой сложилась конкретная ситуация. Такое моделирование представляет собой неотъемлемую составную часть традиционной системы управления в области принятия решений только на основе мысленного обобщения полученной технической или технико-экономической информации. Общение с памятью ЭВМ при данном методе проводится редко, использование этого метода не позволяет, как правило, решать задачи по текущему отысканию и согласованию технического и экономического оптимумов, поскольку традиционными методами сделать это невозможно» [10].

Логическое моделирование предполагает построение логической модели веб-приложения и модели его информационной базы на основе концептуальной модели «как должно быть», разработанной и рассмотренной в первой главе. Логическое моделирование веб-приложение будет спроектированы при поддержке диаграммы вариантов использования и диаграммы классов. Формирование диаграммы вариантов использования и диаграммы классов, а также проектирование диаграммы деятельности и диаграммы последовательности с применением языка моделирование UML. С помощью CASE-инструментов можно значительно повысить эффективность работы, сократиться время разработки и уменьшиться количество программных ошибок.

«StarUML — это программный инструмент визуального моделирования с открытым исходным кодом, который поддерживает стандартизованный язык

графического описания UML (Unified Modeling Language) для моделирования систем и программного обеспечения» [23].

После выбора технологии логического моделирования приступим к разработке логической модели веб-приложения для работы с данными ФИАС.

## **2.2 Разработка логической модели веб-приложения**

Механизм проектирование логической модели веб-приложение предполагает переход от контекстной модели «как должно быть» к диаграмме вариантов использования.

«Диаграмма вариантов использования в UML — диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему» [7].

Представленная диаграмма вариантов использования позволяет представить основные наблюдаемые процессы в порядке их взаимосвязи, а также реализовать выделенную функциональную структуру веб-приложений. Также необходимо создать диаграмму классов, описывающую структуру веб-приложения в классах профессиональной разработки объектно-ориентированного программирования и их взаимосвязь между ними, атрибутами и методами [8].

В таблицах 3-7 представлена краткая оценка прецедентов (вариантов использования) диаграммы.

Созданная диаграмма вариантов использования презентована на рисунке 9.



Таблица 3 – Краткая характеристика прецедентов

Прецедент	Характеристика
Поиск адреса	Поиск адреса позволит пользователю искать адреса через поисковую строку
Просмотр адреса	Просмотр адреса позволит пользователю просматривать адреса которые он искал.
Редактирование	Редактирование будет доступно только ИТ – админу он может добавлять или редактировать информации об адресах.

Таблица 4 – Прецедент «Поиск адреса»

Раздел	Описание
Прецедент «Поиск адреса»	
Актер	1. пользователь; 2. ит-специалист.
Предусловие	Пользователь заходит на начальную страницу веб-приложения
Основной поток	1. зайти в веб-приложение; 2. написать в поисковик нужный адрес для поиска.
Альтернативный поток	Нет
Постусловие	Произведен поиск адреса.

Таблица 5 – Прецедент «Просмотр адреса»

Раздел	Описание
Прецедент «Просмотр адреса»	
Актер	1. пользователь; 2. ит-специалист.
Предусловие	Пользователь заходит на начальную страницу веб-приложения
Основной поток	1. зайти веб-приложение; 2. Просмотр нужного адреса.
Альтернативный поток	Нет
постусловие	Произведен просмотр адреса

Таблица 6 – Прецедент «Получение данных по адресу»

Раздел	Описание
1	2
Прецедент «Получение данных по адресу»	
Актер	1. Пользователь; 2. ит-специалист.

Продолжение таблицы 6

1	2
Предусловие	Пользователь заходит на начальную страницу веб-приложения
Основной поток	<ol style="list-style-type: none"> <li>1. Зайти в веб-приложение;</li> <li>2. Выбрать нужный адрес;</li> <li>3. Просмотр полученные данные по адресу.</li> </ol>
Альтернативный поток	<ol style="list-style-type: none"> <li>1. Зайти в веб-приложение;</li> <li>2. Сделать поиск нужного адреса;</li> <li>3. Просмотр полученные данные по адресу.</li> </ol>
постусловие	Произведен получение данных по адресу

Таблица 7 – Прецедент «Редактирование»

Раздел	Описание
Прецедент «Редактирование»	
Актер	1. ит-специалист.
Предусловие	ИТ-специалист выбирает нужный адрес после чего нажимает кнопку редактировать
Основной поток	<ol style="list-style-type: none"> <li>1. Зайти в веб-приложение;</li> <li>2. Выбрать нужный адрес;</li> <li>3. Нажать кнопку редактировать;</li> <li>4. Выполнить редактирование.</li> </ol>
Альтернативный поток	Нет
постусловие	Произведено редактирование.



Рисунок 9 – Диаграмма вариантов использования

На диаграмме представлены следующие действующие лица (актер):

- пользователь – пользователь веб приложения, который может только делать поиск адресов, которые ему нужны и просматривать результат поиска адреса, который он искал.
- ИТ-специалист - пользователь веб-приложения, который может редактировать и добавление новых адресов. Заполняет необходимую информацию которые были обновлены адресов.

Впоследствии необходимо проанализировать исследованную диаграмму классов, данная диаграмма представлена на рисунке 10.

«Диаграмма классов — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования» [6].

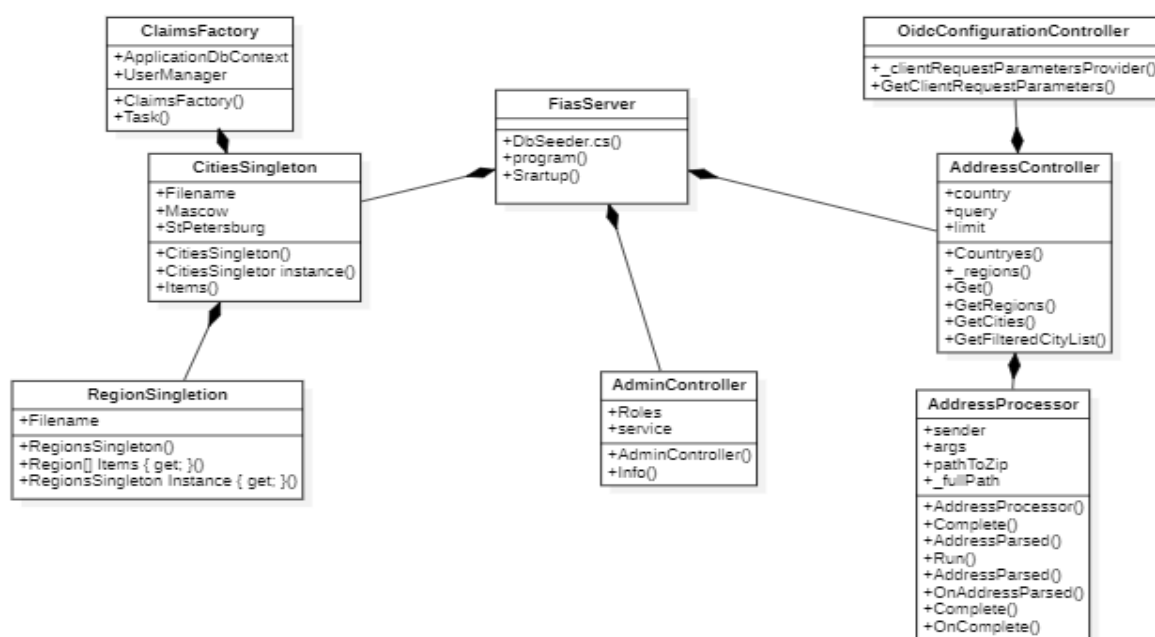


Рисунок 10 – Диаграмма классов

На рисунке 10 представлена диаграмма классов. Класс «ClaimsFactory» и класс «RegionSingleton» наследуются от класса «CitesSingleton» данный класс наследуется от более общего класса «FiasServer». Другой класс «AdminController» который наследуется от класса «FiasServer».

Так же есть такие классы как «AddressProcessor» и класс «OidcConfigurationController» эти два класса наследуются от класса «AddressController».

Класс «AddressController» наследуется от более общего класса «FiasServer». Данная диаграмма не требует пояснения и ясно показывает различные классы и как они взаимодействуют.

После построения логической модели веб-приложение приступим к разработке диаграммы последовательности и диаграмма деятельности.

### **2.3 Реализация диаграммы последовательности**

Для понимания жизненного цикла веб-приложения и взаимодействия с пользователем в течение времени требуется визуализация данного процесса. В основном лучшим решением является представление представленного процесса в виде диаграммы последовательности (рисунок 11).

«Диаграмма последовательности (англ. sequence diagram) — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента» [7].

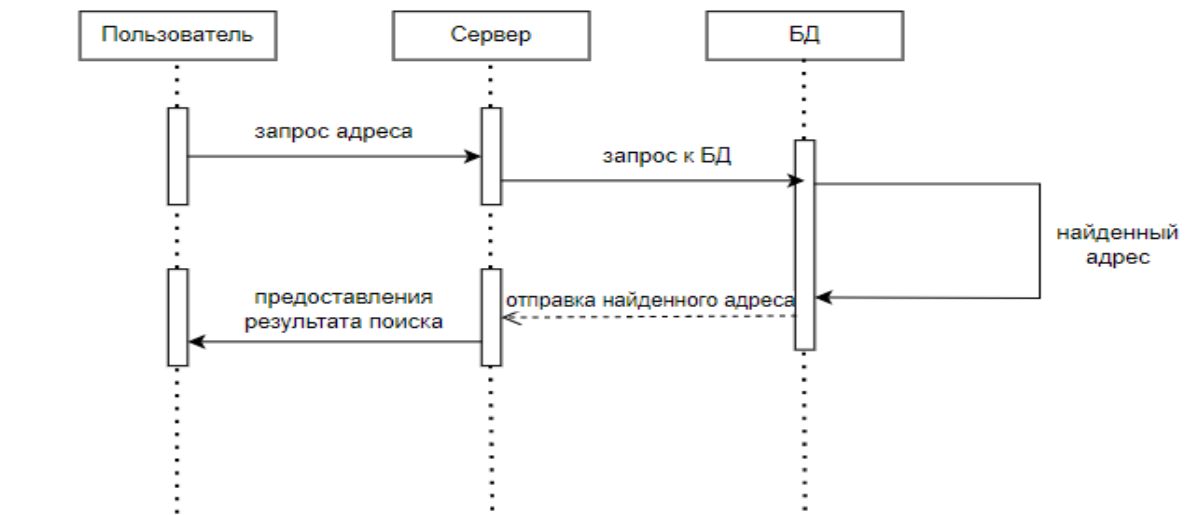


Рисунок 11 – Диаграмма последовательности

Пользователь делает запрос на поиск нужного нам адреса и отправляет его на сервер после чего сервер обращается базу данных где находится информация об адресах и начинается поиск нужного нам адреса. После успешного нахождения информация об адресе информация проходит через сервер и отправляется пользователю для просмотра.

После проектирование диаграммы последовательности перейдем к проектированию диаграммы деятельности.

## 2.4 Реализация диаграммы деятельности

Для того, чтобы понять последовательность действий пользователя в веб-приложении, сервера и базы данных, необходимо визуализировать этот процесс в виде схемы. Наиболее лучшим вариантом построения диаграмм для этого процесса является диаграмма деятельности (рисунок 12).

«Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения

подчинённых элементов — вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого» [5].

На рисунке 12 показана реализация диаграммы деятельности

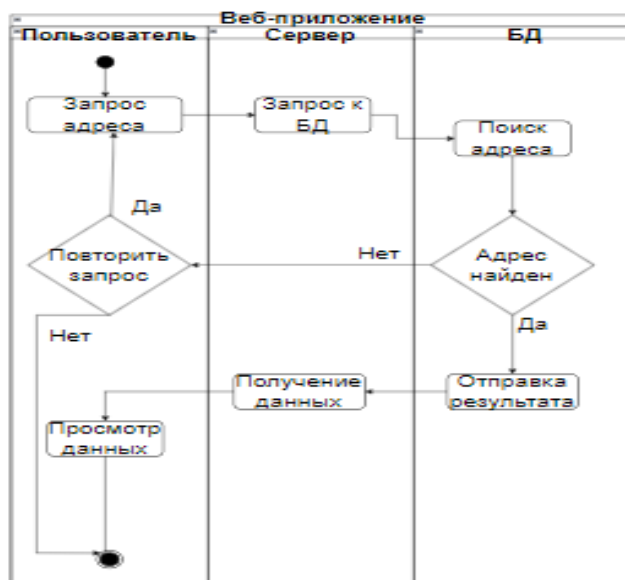


Рисунок 12 – Диаграмма деятельности

Пользователь делает запрос на необходимый ему адрес и отправляет его в сервер после чего сервер производит запрос к базе данных где свершается поиск адреса если адрес не находится, то спрашиваем у пользователя повторить запрос если нет прекращение работы. Если пользователь решит повторить запрос, то действия начнется с начало. Если во время поиска адрес будет найден, то отправляется результат в сервер после чего отправляется пользователю для просмотра информации об адресе.

## 2.5 Требования к аппаратно-программному обеспечению веб-приложения

Техническое обеспечение системы должно быть максимально и наиболее эффективно использоваться.

В составе технического комплекса входит:

- сервер СУБД;
- ПК пользователей;
- мобильное устройство пользователя.

Минимальные требования к СУБД:

- процессор – частота не менее 2.4 ГГц;
- оперативная память (ОЗУ) – не менее 4 Гб;
- свободное место на жестком диске – не менее 25 Гб;
- подключение к интернету;
- операционная система – Windows.

Минимальные требования к ПК:

- процессор частота – от 2 ГГц;
- оперативная память (ОЗУ) – не менее 2 Гб;
- подключения к сети интернета;
- ПК под любой популярной ОС (Windows, Linux, MacOS и т.д.);
- любой из популярных браузеров (Google, Mozilla, Яндекс и др.)

Вслед за тем, как были предоставлены и описаны условия к аппаратному и программному обеспечению для веб-приложения, перейдем к подведению результатов второй главы.

Вывод по второй главе

Вторая глава содержит логическую структуру веб-приложения. Проведен анализ технологии логического моделирования. Затем была построена логическая модель веб-приложения, содержащая диаграмму вариантов использования и диаграмму классов. Вдобавок была спроектирована диаграмма последовательности и диаграмма деятельности. Затем были определены и описаны все требования к аппаратной и программной части веб-приложения..

## Глава 3 Физическое проектирование веб-приложения

### 3.1 Выбор архитектуры автоматизированной веб-приложения

Успех проекта зависит от подбора технологии разработки. На сегодняшний период имеются огромное количество вариантов реализации веб-приложения. В момент выбора средств реализации предусматривались условия заказчика.

Архитектура веб-приложений – это расположение функций между его подсистемами и элементами определяет, как границы подсистем, а также их согласование друг с другом. Начало архитектуры веб-приложения презентует собой трехуровневую клиент-серверную архитектура представлена на рисунке 13.

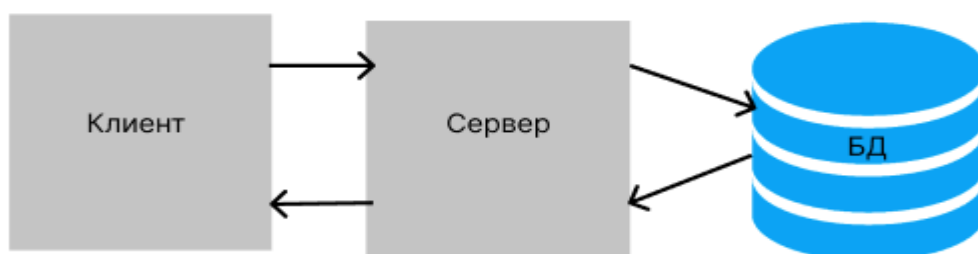


Рисунок 13 – Клиент-серверная архитектура

Структура веб-приложение, расписанная на рисунке 13 содержит три звена: клиент — сервер — сервер базы данных:

- первое звено – клиент представляет собой всякий браузер, которым может пользоваться пользователь такие как (Chrome, Mozilla, Explorer и др.)
- второе звено – это сервер приложений (слой логики) веб-сервер на платформе asp.net. core.
- третье звено – это сервер базы данных под управлением СУБД MySQL.



Клиент взаимодействует с сервером приложений после того, как сервер приложений взаимодействует с сервером базы данных с помощью SQL-запросов.

Эта архитектура владеет превосходной масштабируемостью, гибкостью. Кроме того, эта архитектура известна своей высокой надежностью и хорошей безопасностью. Все функции назначены между сервером приложений и сервером базы данных, что гарантирует хорошую скорость. Именно благодаря данной архитектуре имеет невысокие требования к скорости сети между клиентом и сервером приложений, невысокие требования к техническим потребительским характеристикам и производительности. По этой причине для веб-приложения лучше всего подходит трехуровневая структура клиент-сервер.

После подбора архитектуры веб-приложение перейдем к подбору технологии разработки ПО веб-приложения

### **3.2 Выбор технологии разработки программного обеспечения автоматизированной веб-приложения**

Чтобы подобрать технологию разработки программного обеспечения веб-приложение. Проведем относительный разбор таких средств как: WebStorm и Visual Studio, и текстовые редакторы, Atom и Brackets. По конечным результатам сравнительного разбора подберем преимущественно правильную среду для разработки программного обеспечения веб-приложение.

«WebStorm — это интегрированная среда для разработки на JavaScript и связанных с ним технологиях. Как и другие IDE JetBrains, WebStorm позволяет автоматизировать рутинную работу и легко справляться со сложными задачами, делая разработку более увлекательной» [3].

«Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том

числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight» [19].

«Atom — бесплатный текстовый редактор с открытым исходным кодом для Linux, macOS, Windows с поддержкой плагинов, написанных на JavaScript, и встраиваемых под управлением Git. Большинство плагинов имеют статус свободного программного обеспечения, разрабатываются и поддерживаются сообществом» [16].

«Brackets - это редактор с открытым исходным кодом для веб-дизайна и разработки, основанный на веб-технологиях, таких как HTML, CSS и JavaScript. Проект был создан и поддерживается компанией Adobe и выпущен под лицензией MIT (Массачусетского технологического института)» [2].

Итоги сравнительного разбора безопасности технологий разработки программного обеспечения приведен в таблице 8.

Таблица 8 – Анализ технологий разработки

Критерии	Brackets	Atom	Visual Studio	WebStorm
Бесплатное пользование	+	+	+	-
Кроссплатформенность	+	+	+	-
Удобная навигация по коду	-	+	+	-
Быстродействие	-	-	+	+
Итог	2/4	3/4	4/4	2/4

Вследствие разбора таблицы 8 осуществляем вывод, что текстовый редактор Visual Studio — наилучший инструмент для разработки веб-приложения. Данный инструмент заработал высочайший балл и соответствует всем аспектам оценки. Данная веб-приложение будет разработана с помощью asp.net [13].

«ASP.NET Core – это кросс-платформенный фреймворк с открытым исходным кодом для разработки веб-приложений, который можно использовать для быстрого создания динамических приложений с отрисовкой на стороне сервера» [3].

После подбора технологии разработки программного обеспечения веб-приложение, перейдем к выбору СУБД для веб-приложения.

### **3.3 Выбор системы управления базы данных веб-приложения**

Для выбора СУБД нужно проанализировать самые используемые СУБД и провести их сравнительный анализ. Для анализа были представлены следующие. СУБД: MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database и Firebird. По конечным результатам сравнительного разбора подберем преимущественно подходящую СУБД.

«Система баз данных — это, по сути, не что иное, как компьютеризированная система хранения однотипных записей» [4].

«MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации» [20].

«PostgreSQL не просто реляционная, а объектно-реляционная СУБД. Это даёт ему некоторые преимущества над другими SQL базами данных с открытым исходным кодом, такими как MySQL, MariaDB и Firebird.

PostgreSQL может похвастаться поддержкой uuid, денежного, перечисляемого, геометрического, бинарного типов, сетевых адресов,

битовых строк, текстового поиска, xml, json, массивов, композитных типов и диапазонов, а также некоторых внутренних типов для идентификации объектов и местоположения логов» [15].

«Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка» [18].

«Oracle Database — это объектно-реляционная система управления базами данных (СУБД) от компании Oracle. Она используется для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации».

«Firebird (FirebirdSQL) — свободная кроссплатформенная реляционная система управления базами данных, работающая на macOS, Linux, Microsoft Windows и некоторых Unix-платформах» [17].

Результат сравнительного анализа СУБД показан на таблице 9.

Таблица 9 – Анализ СУБД

Критерии	MySQL	Firebird	Oracle Database	Microsoft SQL Server	PostgreSQL
Бесплатное использование	+	+	+	+	-
Быстродействие	+	-	+	+	+
Наличие информации документации по работе СУБД	+	+	-	+	-
Распространённость	+	+	-	+	-
Опыт работы	+	-	-	-	-
Итог	5/5	3/5	2/5	4/5	1/5

Из таблицы 9 делаем вывод, что СУБД MySQL представляется лучшим выбором для нашей работы, поэтому выбираем его в качестве СУБД. Представленная СУБД заработал высокую оценку и соответствует всем аспектам оценки.

Представленный СУБД MySQL сервер был выбран так как с данной базой функционирует компания, которая поставила задачу на разработку веб-приложение для работы с данными ФИАС

После подбора СУБД для веб-приложения, перейдем к разработке веб-приложение для работы с данными.

### 3.4 Описание функциональности, автоматизированной веб-приложение

В текущем параграфе будут описаны ключевые функции разработанного веб-приложение для работы с данными ФИАС. Организаторы будут работать с этой системой события и участники. Эти роли были рассмотрены и описаны в вторая глава, пункт 2.2.

На начальной странице веб-приложения пользователь встретит главное окно где будет показан список адресов, которые он захочет выбрать для детального просмотра. Данная страница изображена на рисунке 14.

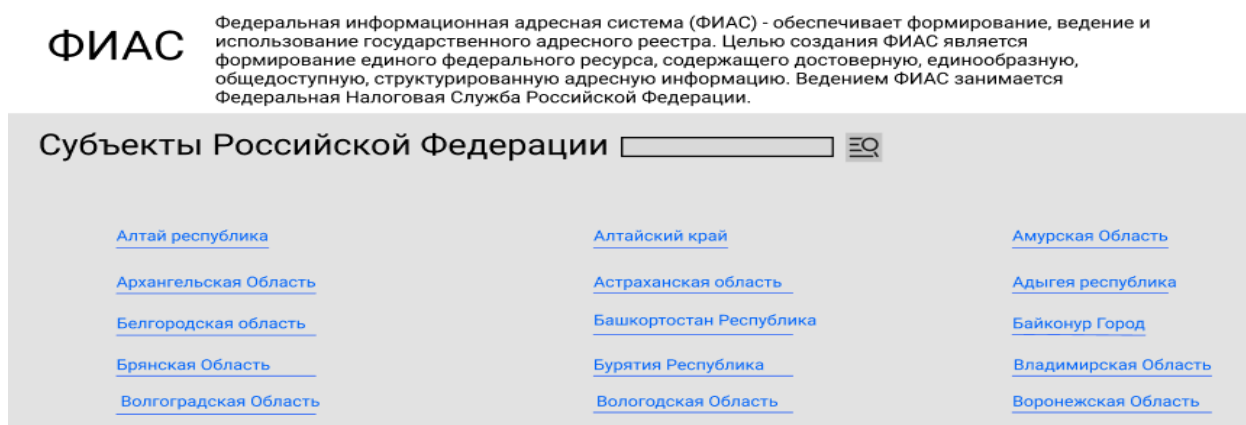


Рисунок 14 - Главная страница веб-приложения

Пользователь может вбить в поисковую строку название адреса после чего начнется поиск информации об адресе. Данная страница показано на рисунке 15.

## ФИАС

Федеральная информационная адресная система (ФИАС) - обеспечивает формирование, ведение и использование государственного адресного реестра. Целью создания ФИАС является формирование единого федерального ресурса, содержащего достоверную, единообразную, общедоступную, структурированную адресную информацию. Ведением ФИАС занимается Федеральная Налоговая Служба Российской Федерации.

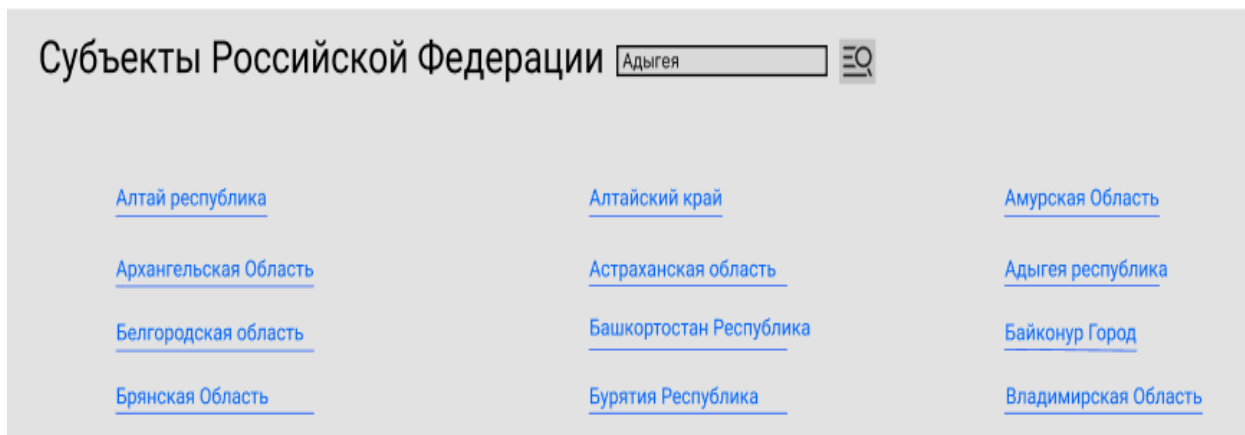


Рисунок 15 – Поиск адреса

После выбора нужного адреса пользователю будет показано информация о выбранном адресе. Данная страница показано на рисунке 16.

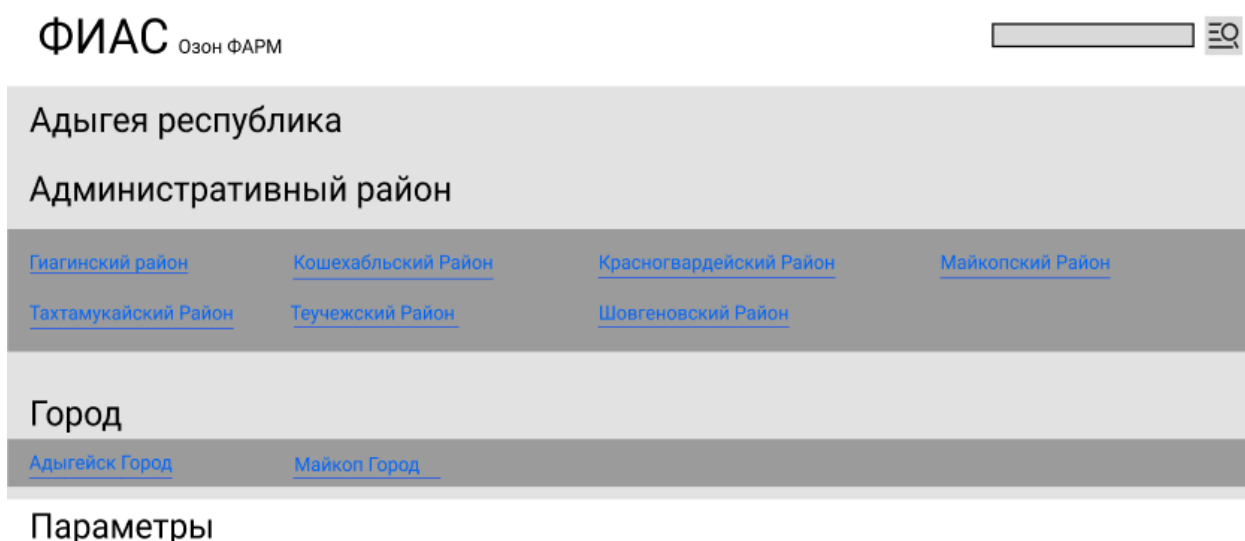


Рисунок 16 – Итоги поиска

На рисунке 17 представлены параметры адреса.

Наименование:	Адыгея
Тип:	Республика
Уникальный идентификатор ФИАС:	d8327a56-80de-4df2-815c-4f6ab1224c50
ИФНС ФЛ:	0100
ИФНС ЮЛ:	0100
ОКАТО:	79000000000
ОКТМО:	79000000
Код адресного объекта одной строкой с признаком актуальности:	0100000000000
Код адресного объекта одной строкой без признака актуальности:	010000000000
Реестровый номер адресного объекта:	7900000000000000000001
Признак в каком делении присвоен адрес, муниципальном/ административном:	0

Рисунок 17 – Параметр адреса

После основного функционала, разработанного веб-приложения для работы с данными ФИАС. Перейдем к тестированию веб-приложение для работы с данными ФИАС.

### 3.5 Тестирование веб-приложения

Тестирование веб-приложения для работы с данными ФИАС включает в себя проверку соответствие между действительным и ожидаемым поведением веб-приложения.

Для разрабатываемой веб-приложения работы с данными ФИАС был выбран метод тестирование «черного ящика».

«Тестирование чёрного ящика или поведенческое тестирование — стратегия (метод) тестирования функционального поведения объекта (программы, системы) с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве (коде) тестируемого объекта.

Иначе говоря, тестированием чёрного ящика занимаются тестировщики, не имеющие доступ к исходному коду приложения. Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из технических требований и их спецификаций» [14].

«Достоинства метода «черный ящик»:

- тестирование методом «черного ящика» позволяет найти ошибки, которые невозможно обнаружить методом «белого ящика»;
- «Черный ящик» позволяет быстро выявить ошибки в функциональных спецификациях (в них описаны не только входные значения, но и то, что мы должны в итоге получить);
- тестировщику не нужна дополнительная квалификация;
- тестирование проходит «с позиции пользователя»;
- составлять тест-кейсы можно сразу после подготовки спецификации.

Недостатки метода:

- основным недостатком метода «черного ящика» является возможность пропуска границ и переходов, которые не очевидны из спецификации, но есть в реализации кода;
- можно протестировать только небольшое количество возможных входных (входящих) значений» [12].

Таблица 10 – Реализация тестирование веб-приложения

№ этапа	Тип информации	Назначение	Результат
1	2	3	4
1	Информирующие	Информация о передачи сообщение.	Передача информации прошла успешно
2	Сообщение об успешной операции	Информирование об успешной обработке запроса клиента	Обработка запроса клиента прошла успешно



Продолжение таблицы 10

1	2	3	4
3	Проверка адаптивности системы	Проверка на устройства пользователя	Веб-приложение полностью подстраивается под устройство пользователя
4	Ошибка на стороне клиента	Указание на ошибки со стороны клиента	Во время работоспособности не найдены
5	Ошибка на стороне сервера	Указание на ошибки со стороны сервера	Во время работоспособности ошибки не выявлены

В результате тестирования разработанной веб-приложения для работы с данными ФИАС методом «черного ящика» ошибок в работе не выявлены.

Результат тестирования представлен в виде графика. Тестирование было проведено вручную и оценивалось по пятибалльной шкале оценки. График проведенного тестирования представлен на рисунке 18.

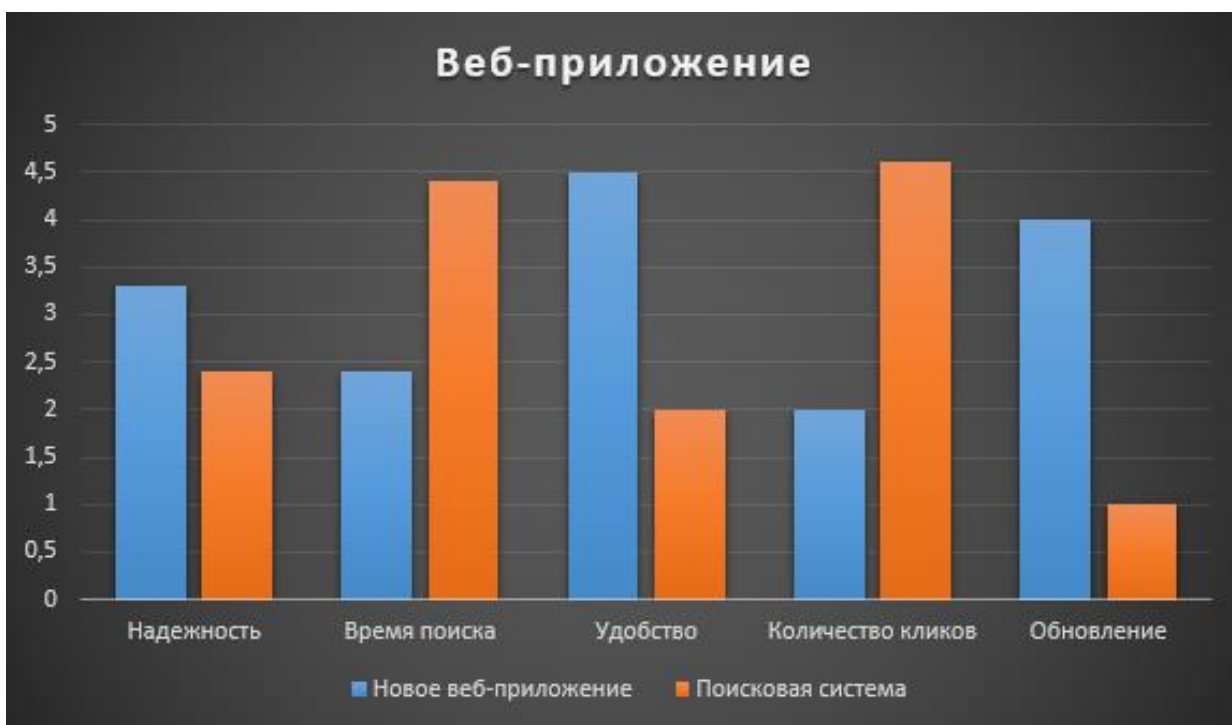


Рисунок 18 – График тестирования

После тестирования разработанного веб-приложения для работы с данными ФИАС, перейдем к подведению итогов третьей главы.

#### Вывод по третьей главе

Третья глава содержит выбор архитектуры. Был проведен анализ и выбор технологии для разработки веб-приложения, также был проведен анализ и выбор СУБД. После чего были описаны функциональные возможности автоматизированной веб-приложение. Проведен тестирование веб-приложение, в котором был выбран метод тестирование «черный ящик» в последствии был приведен график проведенного тестирование.

## Заключение

В процессе выполнения бакалаврской работы реализовано веб-приложение для работы с данными ФИАС. Данное веб-приложение было разработано на платформе разработке asp.net.

Также был произведен анализ предметной области, который показал, что уже ранее существующие веб-приложение имеет некоторые изъяны. Для понятия работы со справочником ФИАС была построена контекстная модель «как есть». Благодаря этому было принято решение создать новое веб-приложение для работы с данными ФИАС. Для того чтобы понять, что нужно улучшить было создано контекстная модель «как должно быть».

В ходе выполнения бакалаврской работы были проанализированы требования к разрабатываемой веб-приложение для работы с данными ФИАС. Построены логические модели, что позволило проверить логику работы веб-приложения. Также были спроектированы диаграмма деятельности и диаграмма последовательности.

После чего был выбран СУБД для хранения информации справочника ФИАС. И самым оптимальным решением стал MySQL так как данная СУБД легче к освоению и пониманию. Так же была выбрана технология Visual Studio для реализации веб-приложения. Далее было реализовано веб-приложение для работы со справочниками ФИАС. Так же был произведена тестирование веб-приложение, где было проанализирована работоспособность. В дальнейшем предполагается доработать веб-приложение добавит функцию просмотр адреса по карте, возможность ит-специалисту редактировать, добавление модального окна для просмотра информации и доработать дизайн веб-приложения.

Все поставленные задачи были выполнены и проверены.

## Список используемых источников

1. Анализ и оптимизация бизнес процессов [Электронный ресурс] URL <https://www.bazt.ru/services/gov/business-process-gos> (Дата обращения 12.05.2022)
2. Аналогии Brackets [Электронный ресурс] URL <https://ruprogi.ru/software/brackets> (дата обращения 12.05.2022)
3. Беликова Д. А. ASP.NET CORE в действии - М.: ДМК Пресс, 2021. - 906 с.
4. Дейт К. Дж. Введение в системы баз данных – Вильяме, 2005. – 1316 с.
5. Диаграмма деятельности [Электронный ресурс] URL [https://ru.wikipedia.org/wiki/Диаграмма\\_деятельности](https://ru.wikipedia.org/wiki/Диаграмма_деятельности) (дата обращения 12.05.2022)
6. Диаграмма классов [Электронный ресурс] URL [https://ru.wikipedia.org/wiki/Диаграмма\\_классов](https://ru.wikipedia.org/wiki/Диаграмма_классов) (Дата обращение 12.05.2022)
7. Диаграмма последовательности [Электронный ресурс] URL [https://ru.wikipedia.org/wiki/Диаграмма\\_последовательности](https://ru.wikipedia.org/wiki/Диаграмма_последовательности) (дата обращения 12.05.2022)
8. Диаграмма прецедентов [Электронный ресурс] URL [https://ru.wikipedia.org/wiki/Диаграмма\\_прецедентов](https://ru.wikipedia.org/wiki/Диаграмма_прецедентов) дата обращения 12.05.2022)
9. Классификация требований по FURPS+ [Электронный ресурс] URL [https://studme.org/226097/informatika/klassifikatsiya\\_trebovaniy\\_furps](https://studme.org/226097/informatika/klassifikatsiya_trebovaniy_furps) (дата обращение 29.05.2022)
10. Логическое моделирование [Электронный ресурс] URL <https://www.ngpedia.ru/id159031p1.html> (дата обращения 07.04.2022)
11. Нотации бизнес-процессов IDEF0. EPC. BPMN [Электронный ресурс] URL <https://www.comindware.com/ru/blog-нотации-бизнес-процессов-idef0-epc-bpmn/> (дата обращения 29.04.2022).

12. Особенности тестирования «черного ящика» [Электронный ресурс] URL <https://quality-lab.ru/blog/key-principles-of-black-box-testing/> (дата обращения 12.05.2022)
13. Смелов В. В. Программирование интернет-изданий – Полиграфический комплекс, 2014 – 59 с.
14. Тестирование по стратегии чёрного ящика [Электронный ресурс] URL [https://ru.wikipedia.org/wiki/Тестирование\\_по\\_стратегии\\_чёрного\\_ящика](https://ru.wikipedia.org/wiki/Тестирование_по_стратегии_чёрного_ящика) (дата обращения 12.05.2022)
15. Хабр [Электронный ресурс] URL <https://habr.com/ru/post/282764/> (дата обращения 12.05.2022)
16. Atom [Электронный ресурс] URL [https://ru.wikipedia.org/wiki/Atom\\_\(текстовый\\_редактор\)](https://ru.wikipedia.org/wiki/Atom_(текстовый_редактор)) (дата обращения 12.05.2022)
17. Firebird [Электронный ресурс] URL <https://ru.wikipedia.org/wiki/Firebird> (дата обращения 12.05.2022)
18. Microsoft SQL Server [Электронный ресурс] URL [https://ru.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server) (дата обращения 12.05.2022)
19. Microsoft Visual Studio [Электронный ресурс] URL [https://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio) (дата обращения 12.05.2022)
20. MySQL [Электронный ресурс] URL <https://ru.wikipedia.org/wiki/MySQL> (дата обращения 12.05.2022)

## Приложение А

### Листинг программы

```
using System.Globalization;
using System.Linq;
using FiasServer.Code;
using FiasServer.Models;
using Microsoft.AspNetCore.Mvc;

namespace FiasServer.Controllers
{
    /// <summary>
    /// Адресный контроллер. Занимается всеми вопросами связанными с адресами, регионами и прочим.
    /// </summary>
    [Route("api/address")]
    [ApiController]
    Ссылка: 0
    public class AddressController : Controller
    {
        private static readonly Country[] Countries = new Country[]
        {
            new Country{Id = 643,
                Alpha2 = "RU",
                Alpha3 = "RUS",
                Name = "Russia"},
        };
        private readonly Region[] _regions = RegionsSingleton.Instance.Items;
        private readonly City[] _cities = CitiesSingleton.Instance.Items;
        /// <summary>
        /// Список стран в системе.
        /// </summary>
        /// <returns></returns>
        [HttpGet]
        [ResponseCache(Location = ResponseCacheLocation.Any, Duration = 86000)]
        Ссылка: 0
        public IActionResult Get()
        {
            return Ok(Countries);
        }

        /// <summary>
        ///
        /// </summary>
        /// <param name="country"></param>
        /// <returns></returns>
        [HttpGet("{country}")]
        [ResponseCache(Location = ResponseCacheLocation.Any, Duration = 86000)]
        Ссылка: 0
        public ActionResult<Country> Get(string country)
        {
            var res = Countries.FirstOrDefault(x => x.Alpha2 == country.ToUpper(CultureInfo.InvariantCulture));
            if (res == null)
                return NotFound(new { error = "Not found country" });
            return res;
        }
        /// <summary>
        /// Список регионов для страны.
        /// </summary>
        /// <param name="country"></param>
        /// <returns></returns>
    }
}
```

## Продолжение приложения А

```
/// <returns></returns>
[HttpGet("{country}/regions")]
[ResponseCache(Location = ResponseCacheLocation.Any, Duration = 86000)]
Ссылка: 0
public ActionResult GetRegions(string country)
{
    var res = Countries.FirstOrDefault(x => x.Alpha2 == country.ToUpper(CultureInfo.InvariantCulture));
    if (res == null)
        return NotFound(new { error = "Not found country" });
    if (res.Alpha2 == "RU")
        return Ok(_regions);
    return NoContent();
}
/// <summary>
/// Список городов для страны. Только Россия.
/// </summary>
/// <param name="country">Код страны, ru</param>
/// <param name="query">Поисковая фраза, по умолчанию все</param>
/// <param name="limit">Лимит для запроса</param>
/// <returns></returns>
[HttpGet("{country}/cities")]
[ResponseCache(Location = ResponseCacheLocation.Any, Duration = 86000)]
Ссылка: 0
public ActionResult GetCities(string country, [FromQuery] string query = "", [FromQuery] int limit = 10)
{
    var res = Countries.FirstOrDefault(x => x.Alpha2 == country.ToUpper(CultureInfo.InvariantCulture));
    if (res == null)
        return NotFound(new { error = "Not found country" });
    if (res.Alpha2 == "RU")
        return Ok(GetFilteredCityList(query, limit));
    return NoContent();
}
```

```
var res = Countries.FirstOrDefault(x => x.Alpha2 == country.ToUpper(CultureInfo.InvariantCulture));
if (res == null)
    return NotFound(new { error = "Not found country" });
if (res.Alpha2 == "RU")
{
    return Ok(GetFilteredCityList(query, limit));
}
return NoContent();
/// <summary>
/// Отдает лимитированное количество городов.
/// </summary>
/// <param name="query">Если строка не пустая то применяется фильтр</param>
/// <param name="limit"></param>
/// <returns></returns>
Ссылка: 1
private City[] GetFilteredCityList(string query, int limit)
{
    if (string.IsNullOrEmpty(query))
        return _cities.Take(limit).ToArray();

    return _cities.Where(x => x.Name.Contains(query, System.StringComparison.InvariantCultureIgnoreCase) ||
        x.Slug.Contains(query, System.StringComparison.InvariantCultureIgnoreCase))
        .Take(limit)
        .ToArray();
}
}
```

## Продолжение приложения А

```
using FiasServer.Data;
using FiasServer.Models;
using IdentityModel;
using Microsoft.AspNetCore.Identity;
using Microsoft.Extensions.Options;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;

namespace FiasServer.Code
{
    Ссылка: 2
    public class ClaimsFactory : UserClaimsPrincipalFactory<ApplicationUser>
    {
        private readonly ApplicationDbContext _context;
        private readonly UserManager<ApplicationUser> _userManager;

        Ссылка: 0
        public ClaimsFactory(
            UserManager<ApplicationUser> userManager,
            IOptions<IdentityOptions> optionsAccessor,
            ApplicationDbContext context) : base(userManager, optionsAccessor)
        {
            _context = context;
            _userManager = userManager;
        }
    }
    Ссылка: 0

```

```
    Ссылка: 0
    protected override async Task<ClaimsIdentity> GenerateClaimsAsync(ApplicationUser user)
    {
        var identity = await base.GenerateClaimsAsync(user);
        var roles = await _userManager.GetRolesAsync(user);

        identity.AddClaims(roles.Select(role => new Claim(JwtClaimTypes.Role, role)));

        return identity;
    }
}

```