

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО
«БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра «Информатика и программное обеспечение»

«У Т В Е Р Ж Д А Ю»

Зав. кафедрой «И и ПО», к.т.н., доцент

_____ Копелиович Д.И.

«____» _____ 2023 г.

ВЕБ-СЕРВИС ДЛЯ ПУБЛИКАЦИИ И ПРОСМОТРА
ОБЪЯВЛЕНИЙ О НЕДВИЖИМОСТИ

ДИПЛОМНАЯ РАБОТА

Документы текстовые

Всего листов _____

Руководитель:

_____ к.т.н., доц. Михалева О.А.

«____» _____ 2023 г.

Консультанты:

по экономической части

_____ к.т.н., доц. Титарёв Д.В.

«____» _____ 2023 г.

по экономической части

_____ к.б.н., доц. Быкова И.В.

«____» _____ 2023 г.

Нормконтролер:

_____ к.т.н., доц. Булатицкий Д.И.

«____» _____ 2023 г.

Студент:

_____ Исаков К.Э.

«____» _____ 2023 г.

БРЯНСК 2023

БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**Направление 09.03.01 – Информатика и вычислительная техника
профиль «Программное обеспечение вычислительной техники
и автоматизированных систем»**

З А Д А Н И Е

на дипломную работу

студенту группы О-19-ИВТ-1-по-Б

Исакову Константину Эдуардовичу

1. Тема работы «Веб-сервис для публикации и просмотра объявлений о недвижимости»

Утверждена приказом по БГТУ № 602-3 от 17. 05. 2023 г.

2. Срок предоставления законченной работы июнь 2023г.

3. Исходные данные

Цель работы автоматизация процессов публикации и поиска объявлений о недвижимости, путем разработки веб-сервиса, обеспечивающего данный функционал

Инструментальные средства React, Typescript, CSS, HTML, Node.js, Express, PostgreSQL, pgAdmin4, Multer, Postman.

Основные требования аутентификация пользователей, просмотр и публикация объявлений о недвижимости, страница для управления ролями пользователей, страница модерации.

Дата выдачи задания _____ 2023г.

Заведующий кафедрой _____ Копелиович Д.И.

Руководитель работы _____ Михалева О.А.

Задание принял к исполнению _____ 2023г.

Студент _____ Исаков К.Э.

АННОТАЦИЯ

В рамках данной дипломной работы был спроектирован веб-сервис для публикации и просмотра объявлений о недвижимости.

В разделе «Анализ требований» описаны основные понятия, связанные с недвижимостью, рассмотрены программы-аналоги. Также спроектирована функциональная модель разрабатываемого веб-приложения, определено основание и назначение разработки. Также описаны функциональные требования и этапы разработки данного продукта.

В разделе «Экономический анализ» представлено экономическое обоснование создания приложения, а также приведены вычисления себестоимости программного продукта.

В разделе «Разработка программного продукта» содержится описание архитектуры разрабатываемого программного продукта, приведены основные используемые технологии, описана модель данных. Также выполнено низкоуровневое проектирование и проектирование интерфейса.

В разделе «Экспериментальная часть» описаны этапы и виды тестирования, а также проверка выполнения функциональных требований.

В разделе «Организационная часть» содержится основные вредные и опасные факторы при работе с ПЭВМ и расчет освещенности на рабочем месте.

1. СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1. Анализ требований	8
1.1. Обзор предметной области.....	8
1.2. Обзор программ-аналогов	10
1.2.1. Сервис недвижимости Домклик	10
1.2.2. Агентство недвижимости Этажи	11
1.2.3. Сайт недвижимости Циан.....	12
1.2.4. Цифровая платформа Restate	13
1.2.5. Интернет-сервис Avito.....	14
1.2.6. Сравнение аналогов	15
1.3. Функциональная модель разрабатываемой системы	18
1.4. Функциональные требования.....	20
1.5. Технические требования.....	21
1.6. Выводы.....	22
2. Экономический анализ.....	23
2.1. Организационная структура проекта	23
2.2. Календарный план проекта	23
2.3. Расчёт затрат на разработку и себестоимости продукта	26
2.3.1. Расчёт заработной платы исполнителей работ по созданию программного продукта	27
2.3.2. Расчёт отчислений на социальные нужды (страховые взносы)	28
2.3.3. Арендные платежи за производственные (офисные) помещения	29

2.3.4. Амортизация используемых основных средств и нематериальных активов.....	29
2.3.5. Расходы на модернизацию и приобретение основных средств.....	30
2.3.6. Расходы на приобретение необходимого ПО.....	30
2.3.7. Расходы на интернет и связь.....	31
2.3.8. Расходы на канцелярские товары и расходные материалы	31
2.3.9. Прочие расходы.....	31
2.3.10. Расчёт себестоимости программного продукта.....	31
3. Разработка программного продукта	33
3.1. Архитектура программного обеспечения	33
3.2. Модель данных.....	35
3.3. Низкоуровневое проектирование	37
3.3.1. Посмотреть список объявлений о недвижимости.....	37
3.3.2. Фильтр объектов.....	38
3.3.3. Опубликовать объявление.....	39
3.4. Проектирование интерфейса.....	40
3.5. API Документация.....	44
3.5.1. Обзор	44
3.5.2. Базовый URL.....	44
3.5.3. Эндпоинты для работы с пользователями	45
3.5.4. Эндпоинты для работы с объектами	47
3.5.5. Эндпоинты для работы с изображениями объектов	49
3.5.6. Эндпоинты для работы с утилитами	49
4. Экспериментальная часть	50
4.1. Испытания.....	50
4.2. План испытаний	50

4.3. Проверка функциональных требований.....	51
4.3.1. Проверка фильтра и поиска.....	51
4.3.2. Проверка отображения на карте	53
4.3.3. Проверка страницы мои объявления.....	54
4.3.4. Проверка страницы модерации.....	56
5. Организационная часть.....	57
5.1. Опасные и вредные факторы.....	58
5.2. Микроклимат	59
5.3. Профилактическая гимнастика	60
5.4. Освещенность рабочего места	62
5.5. Расчет системы искусственного освещения	63
5.6. Пожарная безопасность	67
ЗАКЛЮЧЕНИЕ.....	68
СПИСОК ЛИТЕРАТУРЫ	69
ПРИЛОЖЕНИЕ	71

ВВЕДЕНИЕ

С учетом роста интернет-технологий и увеличения их доступности, виртуальные площадки для обмена информацией получают все большее распространение.

Целью данного дипломного проекта является автоматизация процессов публикации и поиска объявлений о недвижимости, путем разработки веб-сервиса, обеспечивающего данный функционал.

Этот веб-сервис предложит пользователям интуитивно понятный и удобный интерфейс, позволяющий с легкостью размещать объявления и находить интересующую недвижимость.

Поставленная цель достигается путем решения следующих **задач**:

1. Анализ предметной области;
2. Сравнительный анализ аналогов, определение преимуществ, которые обеспечит разработка системы;
3. Проектирование системы;
4. Разработка методики тестирования системы.

Объектом исследования является процесс публикации и поиска объявлений о недвижимости.

Предметом исследования является автоматизация данного процесса с использованием современных технологий, направленная на повышение эффективности и удобства для конечных пользователей.

1. АНАЛИЗ ТРЕБОВАНИЙ

1.1. Обзор предметной области

Недвижимость – вид собственности, которое на законодательном уровне признается как недвижимое имущество. В соответствии с происхождением, к недвижимости причисляются аграрные участки, участки недр, а также все то, что основательно относится к земельным участкам, то есть объекты, которые нельзя переместить без причинения несоразмерного ущерба их назначению, к этому также относятся сооружения, здания и объекты незавершённого строительства.

Область недвижимости представляет собой широкий сектор, включающий в себя поиск, покупку, продажу, аренду и управление объектами недвижимости. В этой области находятся различные участники:

Покупатели – лица или организации, которые планируют приобрести недвижимость. Они сталкиваются со сложностями в поиске подходящих объектов из-за огромного количества предложений на рынке. Возможность удобной фильтрации и сортировки объявлений в веб-приложении может значительно облегчить процесс отбора подходящих вариантов.

Продавцы – владельцы объектов недвижимости, стремящиеся их продать. Они сталкиваются со сложностью сбора информации, так как для получения полной информации о конкретном объекте недвижимости, часто требуется обращаться к различным источникам и вести множество переговоров. Веб-приложение может централизовать всю необходимую информацию, представив её в удобном и доступном формате.

Арендаторы и арендодатели – лица или организации, которые сдают или берут в аренду недвижимость. Они также сталкиваются с проблемой сравнения предложений, которая может быть затруднительной из-за разрозненности информации и необходимости помнить множество деталей. Веб-приложение может предоставить инструменты для удобного сравнения и анализа вариантов.

Агенты по недвижимости – лица или организации, которые выступают в роли посредников между продавцами и покупателями или арендаторами и арендодателями. Они помогают обеим сторонам вести переговоры и заключать сделки.

Также все вышеперечисленные участники могут использовать веб-сервисы для упрощения различных процессов.

Веб-сервис для недвижимости – это цифровая платформа, которая позволяет пользователям публиковать, просматривать и управлять объявлениями о недвижимости.

Веб-сервисы для недвижимости предоставляют возможности для решения целого ряда проблем, с которыми сталкиваются участники рынка недвижимости:

- *Сложности в поиске подходящих вариантов.*

Из-за огромного количества предложений на рынке, поиск подходящего объекта недвижимости может быть сложным и трудоемким процессом. Поэтому возможность удобной фильтрации и сортировки объявлений могут намного облегчить процесс отбора подходящих вариантов.

- *Сложность сбора информации.*

Для получения полной информации о объекте недвижимости, часто приходится обращаться к разным источникам и много разговаривать с владельцем или агентами, что занимает много времени и усилий. Веб-приложение может централизовать всю необходимую информацию и представить её в доступной и понятной форме, тем самым экономя время пользователя.

- *Сложность сравнения предложений.*

Сравнение различных предложений может быть затруднительно из-за разрозненности информации и необходимости держать в голове множество деталей. Веб-приложение может предложить инструменты для удобного сравнения и анализа вариантов, что поможет клиенту принять обоснованное решение.

В этом контексте, разработка веб-сервиса для публикации и просмотра объявлений по недвижимости представляется крайне актуальной [1]. С помощью передовых технологий и глубокого понимания предметной области можно создать приложение, которое предоставит клиентам доступ к ассортименту недвижимости и к информации о рынке. Этой значительно упростит процесс поиска и выбора недвижимости, делая его более удобным, быстрым и безопасным для всех участников.

1.2. Обзор программ-аналогов

Необходимо провести анализ аналогичных программных продуктов, имеющие функции или часть функций, которые совпадают с темой дипломного проекта. Нужно выявить достоинства и недостатки систем, и выполнить их сравнение.

1.2.1. Сервис недвижимости Домклик

Домклик – это онлайн-сервис для поиска недвижимости и размещения объявлений о продаже и аренде в России. На сайте представлен широкий выбор объектов недвижимости, удобный интерфейс, подробная информация о каждом объекте, широкие возможности поиска. Интерфейс сайта представлен на рис. 1.1

Достоинства:

- Удобный интерфейс;
- Удобный поиск;
- Наличие карты;
- Возможность сравнения объектов.

Недостатки:

- Ограничения работы с фотографиями;
- Отсутствие перевода основных элементов сайта на другой язык

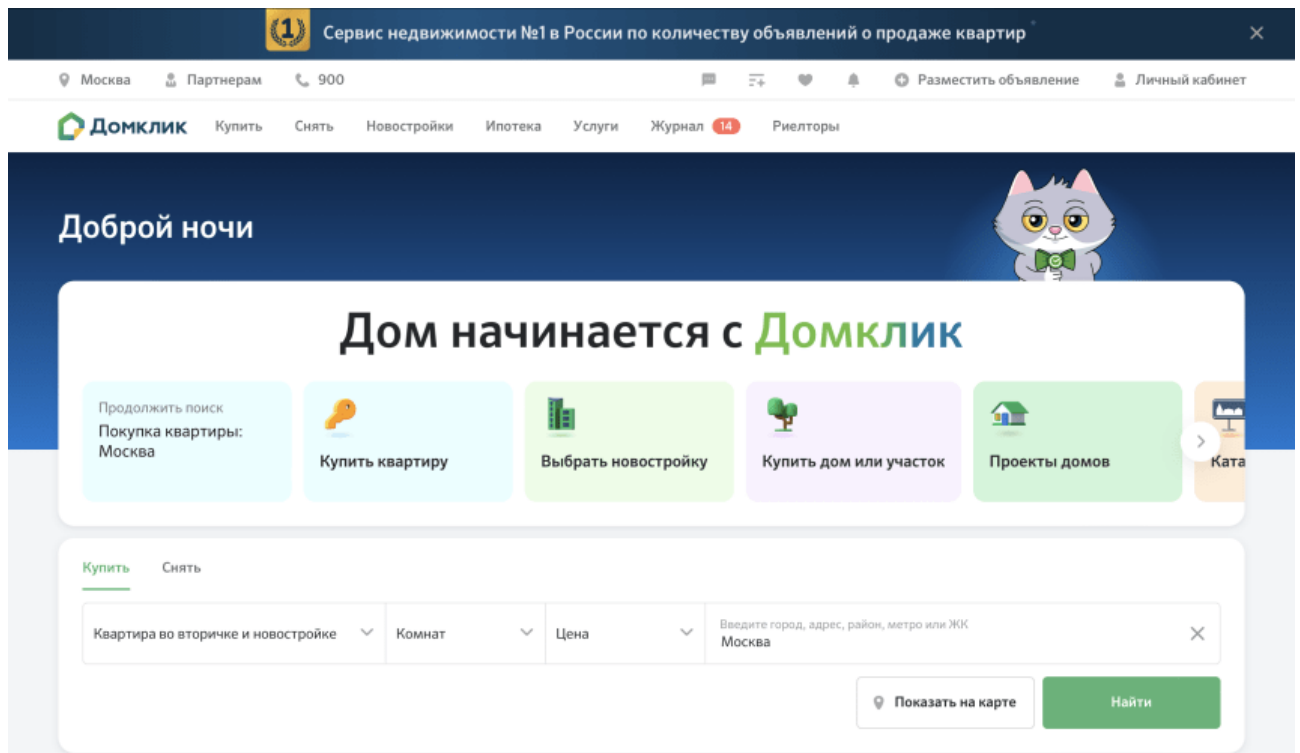


Рис. 1.1. Интерфейс сайта Домклик

1.2.2. Агентство недвижимости Этажи

Этажи – агентство, которое предоставляет онлайн-платформу для поиска недвижимости, публикации квартир, комнат, домов и других объектов в разных городах России. Интерфейс сайта представлен на рис. 1.2. Далее представлены некоторые из достоинств и недостатков этого сайта.

Достоинства:

- Удобный поиск;
- Наличие карты;

Недостатки:

- Не все объявления имеют фотографии;
- Ограниченные фильтры поиска
- Отсутствие перевода основных элементов сайта на другой язык

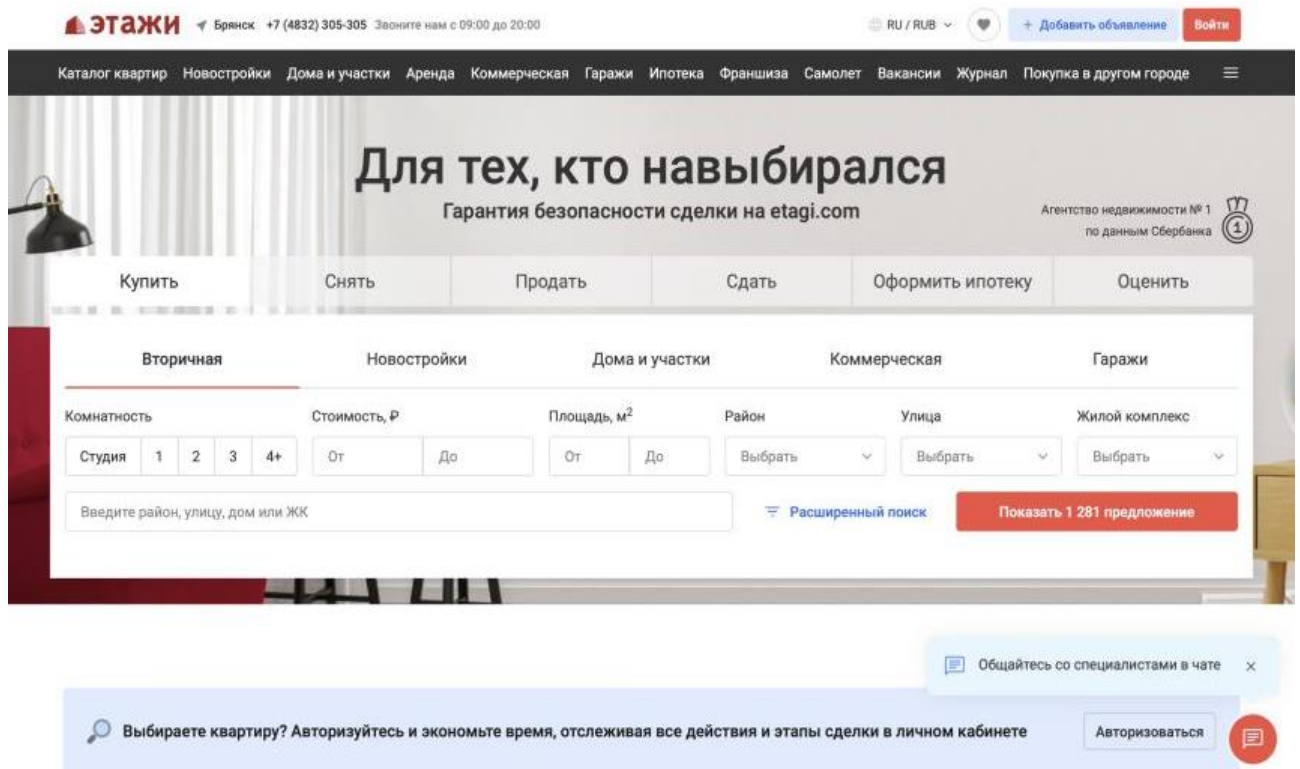


Рис. 1.2. Интерфейс сайта этажи

1.2.3. Сайт недвижимости Циан

Циан – один из самых крупных и известных в России онлайн-сервисов по поиску недвижимости и публикации объявлений для поиска покупателей. Интерфейс сайта на рис. 1.3. Далее представлены некоторые достоинства и недостатки данного сайта.

Достоинства:

- Широкий функционал сайта;
- Удобный поиск;
- Наличие сортировки;
- Наличие карты.

Недостатки:

- Сложность в навигации из-за большого объема информации;
- Не все объявления имеют фотографии;
- Перегруженные фильтры поиска;
- Отсутствие перевода основных элементов сайта на другой язык.

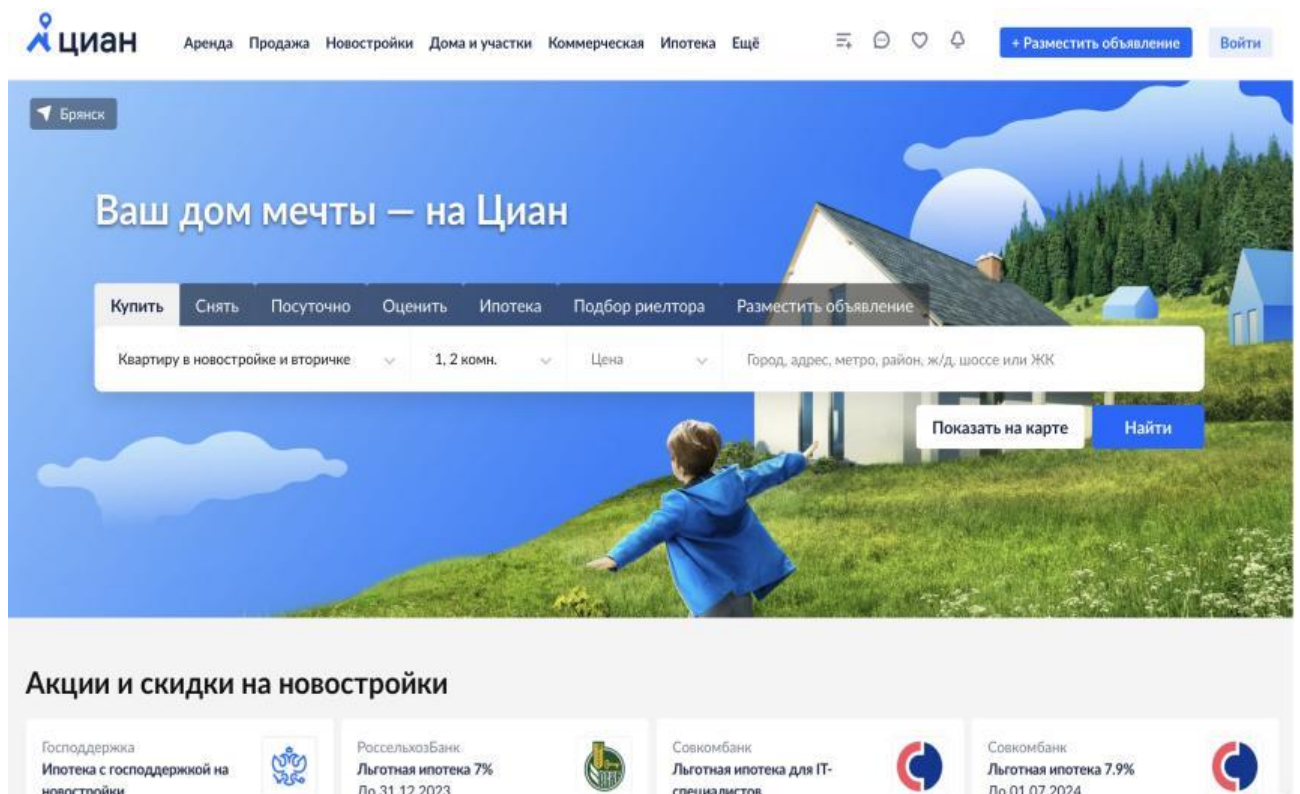


Рис. 1.3. Интерфейс сайта Циан

1.2.4. Цифровая платформа Restate

Restate — это цифровая платформа для поиска недвижимости и людей, которые заинтересованы в поиске объектов недвижимости в России. На рис. 1.4 изображен интерфейс главной страницы сайта. Ниже описаны некоторые достоинства и недостатки данного сайта.

Достоинства:

- Детальное описание объектов недвижимости;
- Наличие карты.

Недостатки:

- Небольшое количество объявлений, по сравнению с конкурентами;
- Сложность в навигации из-за большого объема информации;
- Ограниченные фильтры поиска;
- Не все объявления имеют фотографии;
- Недостаточная структурированность страниц, что может затруднить поиск.

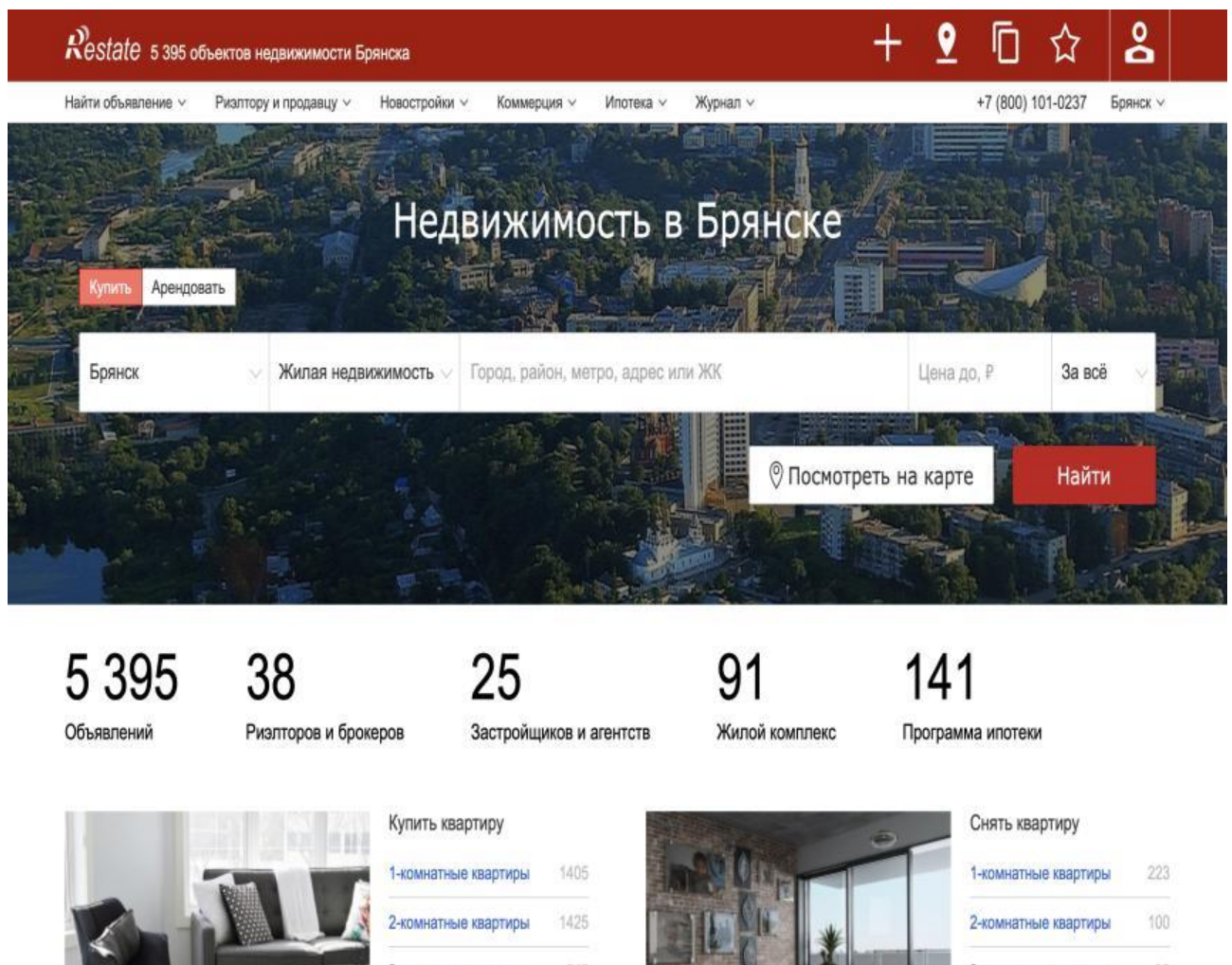


Рис. 1.4. Интерфейс сайта Restate

1.2.5. Интернет-сервис Avito

Avito Недвижимость – это раздел популярного интернет-сервиса Avito, предназначенный для публикации объявлений объектов недвижимости и также для поиска подходящих объектов. Интерфейс сервиса представлен на рис. 1.5. Далее указаны достоинства и недостатки данного раздела сервиса.

Достоинства:

- Возможность сравнения объектов;
- Детальное описание объектов недвижимости;
- Наличие карты;
- Поддержка фильтров поиска;
- Широкий выбор фильтров;

Недостатки:

- Множество неактуальных объявлений;
- Неудобный фильтр;
- Некоторые объявления могут быть недостоверными;
- Много рекламы, что может отвлекать от поиска.

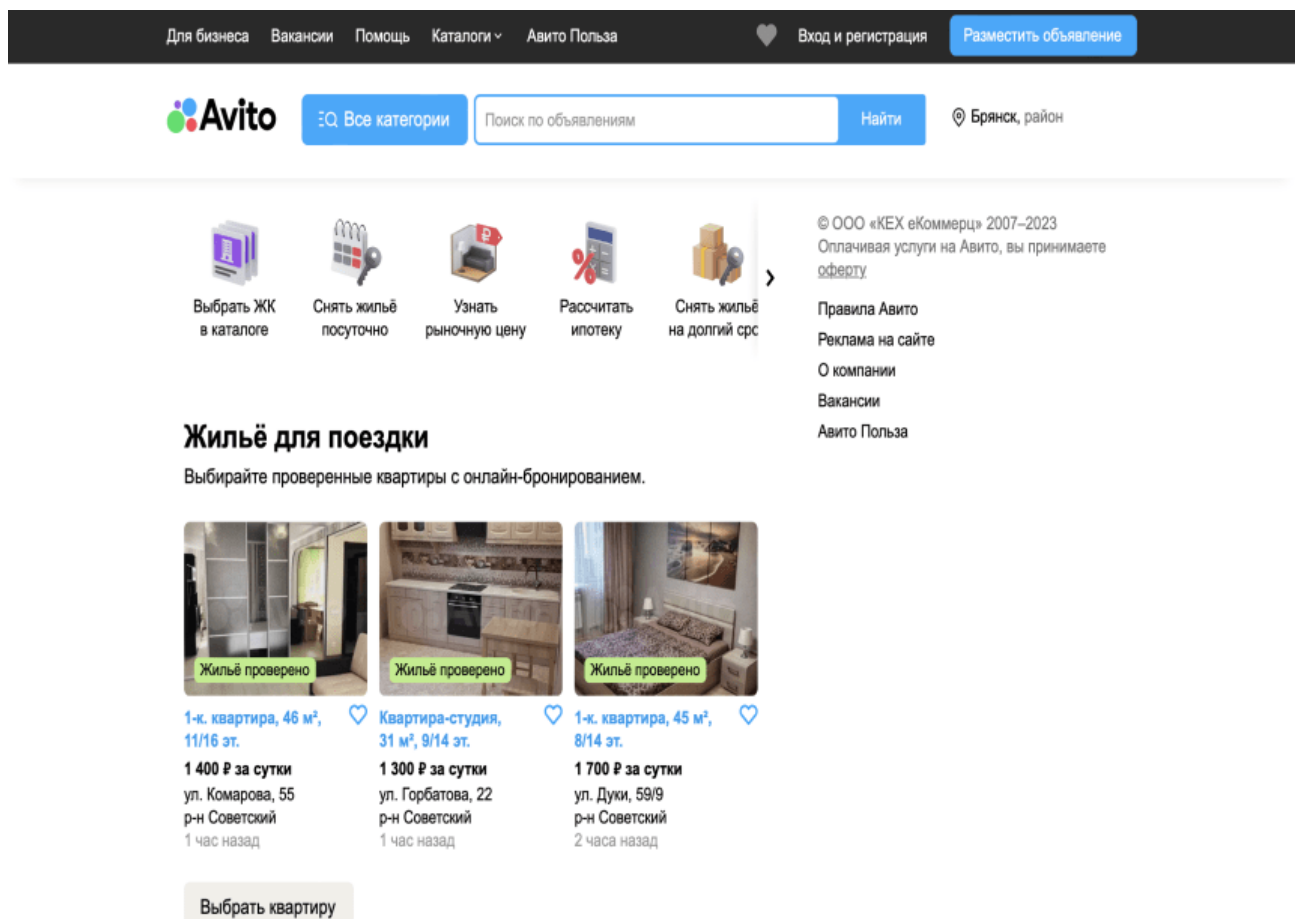


Рис. 1.5. Интерфейс сайта Avito раздела недвижимость

1.2.6. Сравнение аналогов

После детального анализа аналогов, мы можем выделить ключевые характеристики, которые определяют успешность платформы для публикации и просмотра объявлений о недвижимости. Оценивая каждую платформу по этим параметрам, можно определить преимущества и недостатки каждого решения, а также выявить возможности для улучшения в рамках разработки нового веб-приложения (таблица 1.1).

Таблица сравнения аналогов

Аналоги Характеристики	Домклик	Этажи	Циан	Restate	Avito
Удобство интерфейса /не загруженность интерфейса	+	–	+	–	+
Удобство поиска	+	–	+	–	+
Детализированные описания объектов	+	+	+	–	+
Страницы структурированы и логически понятны	+	+	+	–	+
Поддержка мультиязычности	–	–	–	–	–
Удобный и функциональный фильтр поиска	+	–	+	–	+
Адаптация под разные устройства	+	+	+	+	+

В результате сравнения программ-аналогов, можно сделать вывод, что каждая из рассмотренных платформ имеет свои сильные и слабые стороны.

Из преимуществ, большинство платформ предоставляют удобный и интуитивно понятный интерфейс, что позволяет пользователям легко ориентироваться и быстро найти нужную информацию. Почти все платформы предоставляют детализированные описания недвижимости, что позволяет пользователям получить полное представление о объекте прямо на сайте. Платформы, такие как "Домклик", "Этажи", "Циан" и "Avito", предоставляют адаптивный дизайн, который оптимизирован для различных устройств.

Все платформы стараются сделать свой интерфейс максимально удобным и дружелюбным для пользователя, однако стилизация страниц, карточек объектов и особенно фильтра, в особенности на платформе Restate, оставляет желать лучшего, что может оттолкнуть пользователя от пользования сайтом. К тому же, не все сайты предоставляют полноценные и удобные фильтры для поиска недвижимости, что может затруднить процесс выбора подходящего

объекта. Также на всех платформах отсутствует какая-либо локализация, что затрудняет взаимодействие иностранных пользователей с платформой.

При анализе текущего состояния рынка веб-приложений в сфере недвижимости, можно заметить ряд общих тенденций и ключевых особенностей, которые будут полезны при разработке нового решения.

Следует отметить значительную важность удобства интерфейса для обеспечения положительного пользовательского опыта. Интерфейс должен быть максимально простым и понятным, что позволяет пользователям без проблем перемещаться по веб-приложению, даже не имея опыта работы с подобными платформами.

При разработке нового веб-приложения стоит стремиться к созданию поискового фильтра, который бы сочетал в себе лучшие практики конкурентов и вместе с тем устранял выявленные слабости. Это может означать создание более интуитивно понятного и удобного фильтра с хорошим дизайном, предоставляющего пользователям возможность гибко настраивать параметры поиска.

Также следует уделить внимание вопросу локализации. Многие существующие решения ограничиваются одним языком, что может затруднить использование платформы для иностранных пользователей. Разработка локализации в новом веб-приложении позволит привлечь более широкую аудиторию пользователей, увеличивая охват и потенциальную базу клиентов.

Наконец, важно заметить, что структура страниц веб-приложений в этой области часто требует оптимизации. Более логичная и понятная структура страниц сможет облегчить навигацию и улучшить общее впечатление от использования веб-приложения. Это позволит пользователю легче ориентироваться в приложении и сделает его использование более приятным и комфортным.

Таким образом, анализ конкурентов позволяет выявить не только их сильные стороны, которые стоит перенять, но и слабые места, которые могут быть улучшены в процессе разработки нового веб-приложения.

1.3. Функциональная модель разрабатываемой системы

Для функционального моделирования программного комплекса используется Use-Case [3] диаграмма (рис. 1.6).

Сценарии вариантов использования:

1. Регистрация и аутентификация пользователей.

Исходные условия: пользователь открывает веб-сервис.

Последовательность действий: пользователь кликает на "Регистрация", заполняет форму и подтверждает регистрацию. После этого, пользователь входит в систему с использованием своих учетных данных.

Результат: Пользователь зарегистрирован и аутентифицирован в системе.

2. Поиск и фильтрация объявлений.

Исходные условия: аутентифицированный пользователь хочет найти подходящий объект недвижимости.

Последовательность действий: пользователь задает критерии поиска и фильтрации (тип недвижимости, цена, площадь и т.д.), затем запускает поиск.

Результат: пользователь получает список объявлений, соответствующих заданным критериям.

3. Размещение объявлений о недвижимости.

Исходные условия: аутентифицированный пользователь хочет разместить объявление о недвижимости.

Последовательность действий: пользователь заполняет форму для создания нового объявления, добавляет необходимую информацию и подтверждает создание объявления.

Результат: новое объявление отправлено на модерацию. После модерации объявление либо публикуется на веб-сервисе на вкладке недвижимость, либо отклоняется, и пользователь сможет увидеть состояние объявления во вкладке «Мои объявления».

4. Управление ролями пользователей (для администратора).

Исходные условия: аутентифицированный администратор хочет назначить модератором пользователя.

Последовательность действий: администратор просматривает список пользователей, либо с помощью поиска находит по почте, имени или номеру телефона необходимого пользователя и изменяет роль пользователя на модераторе.

Результат: роль выбранного пользователя изменена на модератор.

5. Проверка и модерация новых объявлений перед публикацией (для модератора).

Исходные условия: аутентифицированный модератор получает уведомление о новом объявлении, ожидающем проверки.

Последовательность действий: модератор просматривает объявление, проверяет его на соответствие стандартам и либо утверждает его, либо отклоняет.

Результат: новое объявление проверено и либо опубликовано, либо отклонено.

Диаграмма вариантов использования [2] для веб-сервиса для публикации и просмотра объявлений о недвижимости представлена далее (Рис. 1.6).

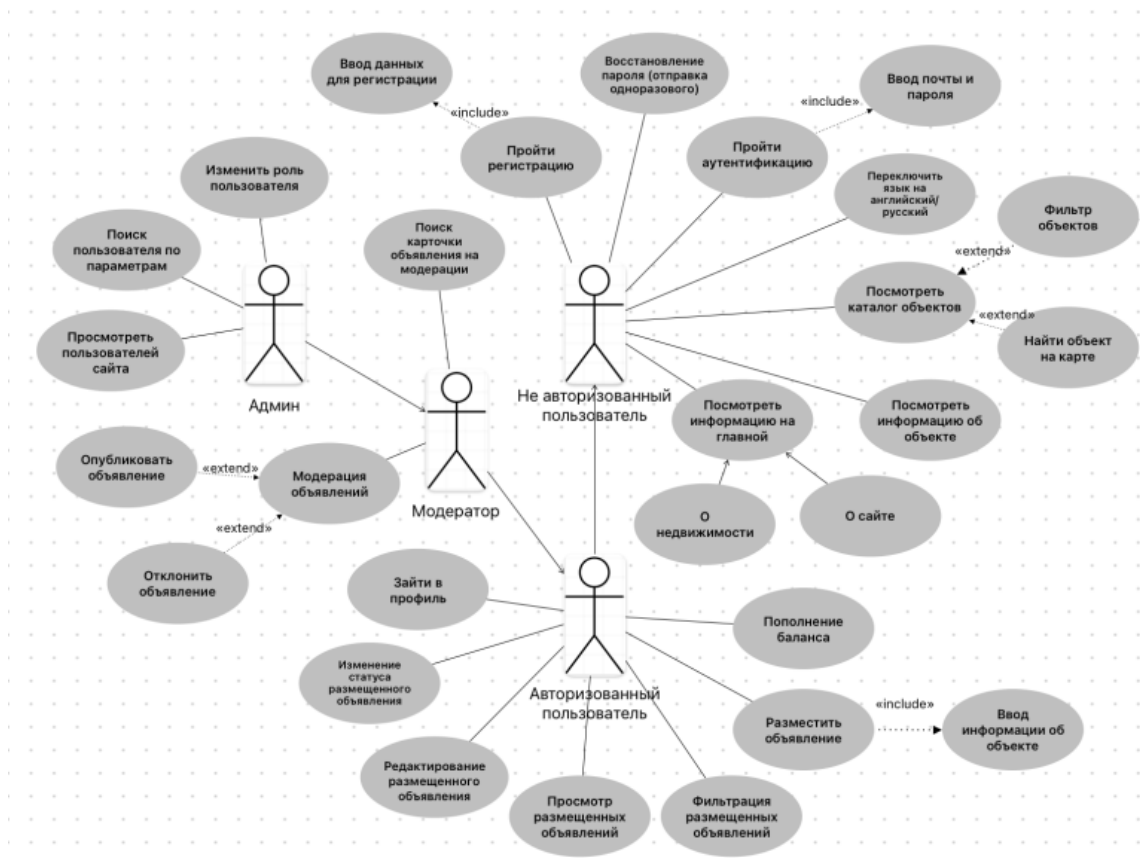


Рис. 1.6. Диаграмма вариантов использования

1.4.Функциональные требования

Общие требования:

- Регистрация и авторизация пользователей.
- Возможность войти с использованием одноразового пароля, отправленного на почту, если пользователь уже был зарегистрирован;
- Просмотр описания веб-сервиса;
- Возможность переключить язык основных компонентов веб-сайта (функция мультиязычности)
- Возможность просмотреть карточки объектов с краткой информацией и стоимостью;
- Поиск и фильтрация объявлений по различным критериям, таким как тип недвижимости, цена, площадь и т.д;
- Отображение объектов недвижимости на интерактивной карте;
- Отображение карточки объекта, при выборе на карте;
- Отображение подробной информации о каждом объекте недвижимости, включая фотографии и данные о владельце;

Для пользователя:

- Размещение объявлений о недвижимости;
- Возможность добавление фотографий о недвижимости;
- Возможность просмотра размещенных объявлений;
- Функция редактирования размещенных объявлений;
- Опция изменения статуса размещенных объявлений (Черновик, Публикация, Архив, Удалить);
- Возможность просмотра профиля;
- Функция редактирования информации пользователя в своем профиле;
- Опция пополнения баланса в профиле;

Для администратора:

- Возможность модерации объявлений (опубликовать, отклонить);
- Страница администрации для просмотра пользователей,

зарегистрированных в системе;

- Возможность поиска нужного пользователя, по почте, имени или номеру телефона;
- Опция изменения роли у нужного пользователя;

Для модератора:

- Проверка и модерация новых объявлений перед публикацией, чтобы обеспечить соответствие стандартам веб-приложения;
- Возможность фильтрации и поиска карточки объекта среди карточек, которые находятся на модерации.

1.5. Технические требования

- Использование современных веб-технологий [7], таких как HTML, CSS [14], JavaScript [7], React [8], TypeScript [9];
- Использование сервера на Node.js [10] и Express [11] с API [13] для доступа к данным БД;
- Использование PostgreSQL [12] для базы данных, с возможностью редактирования в pgAdmin4 [22];
- Тестирование API-запросов через Postman [21];
- Загрузка файлов на сервер, используя Multer [17];
- Интеграция с сервисами карт Яндекс карты, для отображения местоположения недвижимости;
- Интеграция с сервисом оплаты ЮКасса [23], для пополнения личного счета;
- Реализация адаптивной верстки для отображения на различных устройствах, включая десктопы, планшеты и смартфоны;
- Защита персональных данных пользователей;
- Реализация возможности перевода сайта на английский язык [18] [19];

1.6. Выводы

На сегодняшний день на рынке недвижимости в России существует множество онлайн-платформ, каждая из которых имеет свои положительные и отрицательные стороны. Положительные стороны включают в себя наличие удобного и интуитивно понятного интерфейса, который облегчает ориентацию пользователей и поиск нужной информации. Большинство платформ, предоставляют детальные описания объектов недвижимости, а также адаптивный дизайн. Однако, есть области для улучшения, такие как стилизация страниц и фильтров поиска и работа над их эффективностью в области поиска информации. Также важно отметить, что отсутствие локализации на всех рассмотренных платформах затрудняет взаимодействие иностранных пользователей с платформой, что необходимо учитывать при разработке собственного веб-сервиса.

Целью данного дипломного проекта является автоматизация процессов публикации и поиска объявлений о недвижимости, путем разработки веб-сервиса, обеспечивающего данный функционал.

В рамках работы ставятся следующие задачи:

- проведение анализа существующих платформ для публикации и просмотра объявлений о недвижимости;
- определение требований к новому веб-приложению;
- проектирование архитектуры программного продукта;
- проектирование пользовательского интерфейса;
- разработка серверной части веб-приложения;
- разработка клиентской части веб-приложения;
- создание плана испытаний программного продукта.

Объектом исследования является процесс публикации и поиска объявлений о недвижимости.

Предметом исследования является автоматизация данного процесса с использованием современных технологий, направленная на повышение эффективности и удобства для конечных пользователей.

2. ЭКОНОМИЧЕСКИЙ АНАЛИЗ

Перед началом разработки любого программного продукта необходимо выяснить, насколько целесообразна его разработка, как с точки зрения полезности, так и с точки зрения экономической эффективности. В технико-экономической части работы рассматриваются вопросы организации работ по созданию программного продукта, а также приводится расчёт ее себестоимости.

2.1. Организационная структура проекта

Организационная структура проекта (OBS) приведена на рис. 2.1.

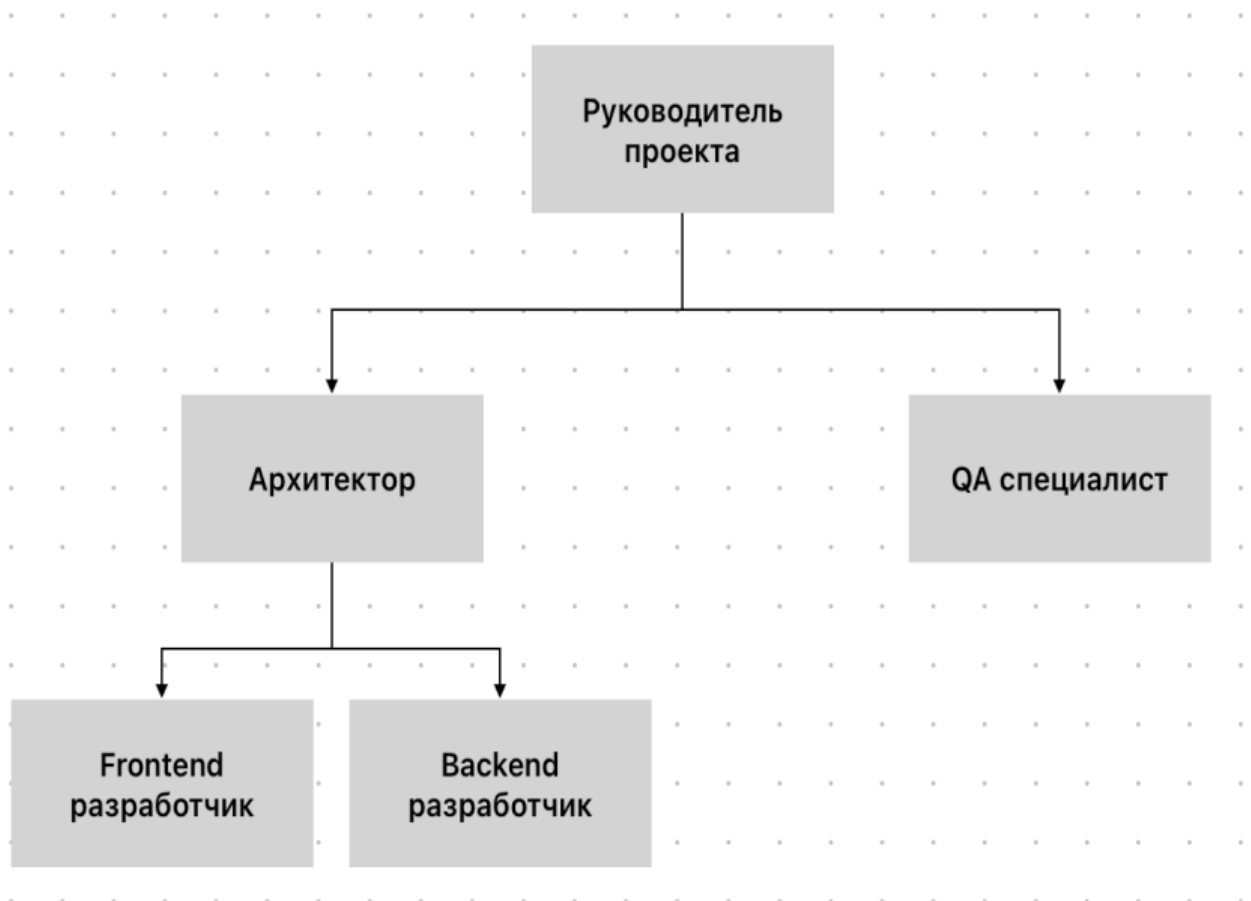


Рис. 2.1. Организационная структура проекта

2.2. Календарный план проекта

Для оценки расходов на реализацию проекта в числе прочих необходимо определить временные затраты на его реализацию. Для определения временных

затрат проекта необходимо разработать календарный план проекта. Реализуемый проект является типовым для Компании, исходя из этого, был сформирован состав работ проекта, определена их длительность, а также распределение ресурсов по ним. При разработке календарного плана были учтены ограничения и допущения, накладываемые на проект Заказчиком.

Для реализации проекта необходимо выполнить следующие работы:

1. Сбор требований Заказчика к разрабатываемому ПО.
2. Разработка и согласование технического задания.
3. Разработка технического проекта.
4. Разработка серверной части.
5. Разработка API для доступа к БД.
6. Разработка клиентской части.
7. Разработка пользовательской документации.
8. Тестирование ПО.

Распределение человеческих ресурсов по работам проекта и степень их загрузки приведены в таблица 2.1.

Таблица 2.1

Структура общего времени на создание программного продукта

№ этапа	Этап работ	Ответственные исполнители (занятость на этапе)	Длительность, дней
1	Сбор требований Заказчика к разрабатываемому ПО	<ul style="list-style-type: none"> Руководитель проекта [100%] 	3
2	Разработка и согласование технического задания	<ul style="list-style-type: none"> Руководитель проекта [100%] Архитектор [40%] 	3
3	Разработка серверной части	<ul style="list-style-type: none"> Backend разработчик [100%] Архитектор [40%] QA специалист [30%] Руководитель проекта [30%] 	15
4	Разработка API для доступа к БД	<ul style="list-style-type: none"> Backend разработчик [100%] Руководитель проекта [40%] Архитектор [50%] 	7
5	Разработка клиентской части	<ul style="list-style-type: none"> Frontend разработчик [100%] Руководитель проекта [30%] Архитектор [40%] QA специалист [30%] 	20
6	Разработка пользовательской документации	<ul style="list-style-type: none"> Руководитель проекта [50%] Frontend разработчик [30%] Backend разработчик [30%] 	2
7	Тестирование ПО, подготовка к запуску и поддержка	<ul style="list-style-type: none"> QA специалист [100%] Руководитель проекта [40%] Frontend разработчик [30%] Backend разработчик [30%] 	10

При реализации данного проекта работы выполняются последовательно. Диаграмма Ганта приведена на рис. 2.2.

	Режим задачи	Название задачи	Длитель	Начало	Окончание	П	Названия ресурсов
1		Сбор требований Заказчика к разрабатываемому ПО	3 дней	Пн 13.02.23	Ср 15.02.23		Руководитель проекта
2		Разработка и согласование технического задания	3 дней	Чт 16.02.23	Пн 20.02.23	1	Руководитель проекта;Архитектор[40%]
3		Разработка серверной части	15 дней	Вт 21.02.23	Пн 13.03.23	2	Backend разработчик;QA специалист[30%];Архитектор[40%];Руководитель проекта[30%]
4		Разработка API для доступа к БД	7 дней	Вт 14.03.23	Ср 22.03.23	3	Backend разработчик;Руководитель проекта[40%];Архитектор[50%]
5		Разработка клиентской части	20 дней	Чт 23.03.23	Ср 19.04.23	4	Frontend разработчик;QA специалист[30%];Архитектор[40%];Руководитель проекта[30%]
6		Разработка пользовательской документации	2 дней	Чт 20.04.23	Пт 21.04.23	5	Backend разработчик[30%];Frontend разработчик[30%];Руководитель проекта[50%];Архитектор[50%]
7		Тестирование ПО	10 дней	Пн 24.04.23	Пт 05.05.23	6	Backend разработчик[30%];Frontend разработчик[30%];QA специалист[30%];Руководитель проекта[50%];Архитектор[50%]

Рис. 2.2. Табличное представление Диаграммы Ганта

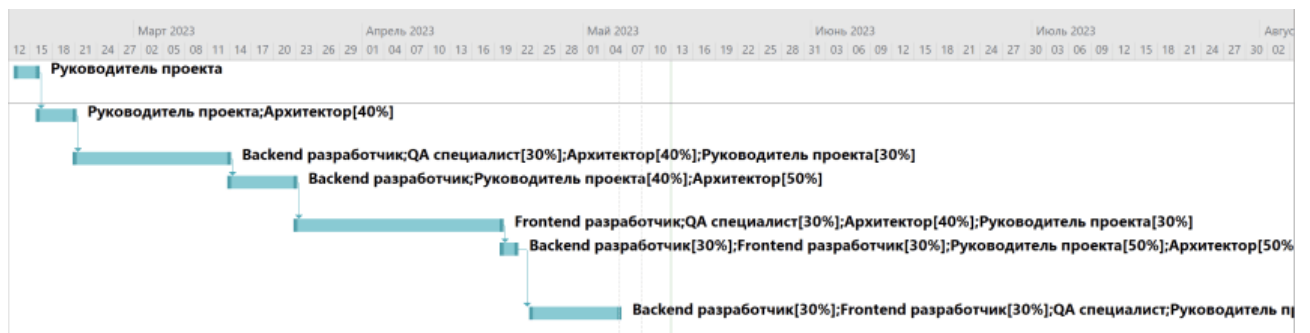


Рис. 2.3. Графическое представление Диаграммы Ганта

Исходя из длительности работ и коэффициента загрузки членов проектной команды, определим их трудозатраты при реализации проекта (таблица 2.2).

Таблица 2.2

Трудозатраты членов проектной команды

№	Исполнитель	Трудозатраты, человеко-часов
1	Руководитель проекта	178,4
2	Frontend разработчик	188,8
3	Backend разработчик	204,8
4	QA специалист	164
5	Архитектор	173,6

2.3.Расчёт затрат на разработку и себестоимости продукта

Расчет затрат на создание веб-сервиса для публикации и просмотра объявлений о недвижимости включает следующие составляющие с последующим их графическим представлением в виде круговой диаграммы:

- заработная плата исполнителей работ по проекту – $ЗП_{осн}$;
- дополнительная заработная плата $ЗП_{дон}$;

- заработная плата обслуживающего и административного персонала;
- отчисления на социальные нужды (страховые взносы) – $H_{зн}$;
- арендные платежи за производственные (офисные) помещения – $A_{пм}$;
- амортизация используемых основных средств и нематериальных активов – A ;
- расходы на модернизацию и приобретение основных средств – $P_{мод}$;
- расходы на приобретение необходимого ПО – $P_{ПО}$;
- расходы на интернет, связь – $P_{тел}$;
- расходы на канцелярские товары и расходные материалы – $P_{р.м.}$;
- прочие расходы – $П_{р.р.}$.

2.3.1. Расчёт заработной платы исполнителей работ по созданию программного продукта

Основная ЗП определяется по формуле:

$$ЗП_{осн} = \frac{M \cdot T}{Ч_p \cdot t_{p.д.}} \left(1 + \frac{П}{100} \right), \text{ руб.}, \quad (2.1)$$

где M – месячная зарплата (руб.), T – общие трудозатраты (чел.-ч), $Ч_p$ – число рабочих дней в месяц, $t_{p.д.}$ – продолжительность рабочего дня в часах, $П$ – процент премии. В данной работе $Ч_p = 20$ дней, $t_{p.д.} = 8$ ч, $П = 0$.

Значение месячной заработной платы (М), суммарные трудозатраты членов, а также рассчитанная по формуле (2.1) основная заработная плата проектной команды приведены в таблица 2.3.

Таблица 2.3

Основная заработная плата членов проектной команды

№	Исполнитель	Месячная заработная плата (М), руб.	Трудозатраты, человеко- часов	ЗП _{осн} , руб.
1	Руководитель проекта	80 000	178,4	89 200
2	Frontend разработчик	60 000	188,8	70 800
3	Backend разработчик	60 000	204,8	76 800
4	QA специалист	50 000	164	51 250
5	Архитектор	80 000	173,6	86 800

Суммарное значение основной заработной платы проектной команды на период реализации проекта составит 374 850 (руб.).

Дополнительная заработная плата берется в размере 15% от основной.

$$\text{ЗП}_{\text{доп}} = 56\,228 \text{ (руб.)}$$

Общая заработная плата составит 431 078 (руб.).

Проект реализуется в небольшой ИТ-компании, где доля вспомогательного и административного персонала по отношению к основному персоналу не велика. Большая часть административного персонала задействована в проектной деятельности в качестве руководителей проекта. Кадровый учет, бухгалтерский и налоговый учет в Компании отдан на аутсорсинг. Затраты на аутсорсинг войдут в прочие расходы. В связи с этим прием заработную плату обслуживающего персонала равной 0 руб.

2.3.2. Расчёт отчислений на социальные нужды (страховые взносы)

Теперь можно рассчитать величину отчислений на социальные нужды (страховые взносы), которые начисляются на заработную плату. С 1-го января 2023 года по страховым взносам был введен единый страховой тариф и единая предельная база для исчисления взносов. При этом налоги и взносы уплачиваются единым платежом с уведомлением об этом налоговой инспекции на единый налоговый счет. В дальнейшем средства поступают в Социальный фонд России.

Базовый тариф для расчета единого платежа составляет 30%, при этом он включает отчисления на ОПС, ОМС и ВНиМ. Отчисления на травматизм по-прежнему производятся по индивидуальному тарифу страхователя. Базовый тариф используется в расчетах до достижения предельной базы, после ставки снижается до 15,1%.

Таким образом, $H_{zn} = 129\,323$ (руб.).

2.3.3. Арендные платежи за производственные (офисные) помещения

Компания, реализующая веб-сервис для публикации и просмотра объявлений о недвижимости, арендует офисные помещения в г. Брянск.

Стоимость аренды составляет 1000 руб/м² в месяц.

Арендная плата включает в себя оплату как площади, занимаемых компанией помещений, так и электроэнергии, отопления, водоснабжения, кондиционирования и уборки помещений, вывоза и утилизации технико-бытовых отходов, парковочных мест на автостоянке.

На каждого члена проектной команды приходится 7,2 м² арендуемого офисного помещения. На период данного проекта члены проектной команды в других проектах не задействованы.

Исходя из изложенного выше, затраты на аренду помещений, отнесенные на проект составят $A_{nm} = 144\,000$ (руб.).

2.3.4. Амортизация используемых основных средств и нематериальных активов

При реализации веб-сервиса для публикации и просмотра объявлений о недвижимости задействованы следующие основные средства:

- 4 персональных компьютеров в сборе первоначальной стоимостью 30 000 (руб.) каждый.
- 1 ноутбук первоначальной стоимостью 45 000 (руб.).

Срок полезного использования для задействованных в проекте основных средств определен в 3 года. Метод начисления амортизации – линейный.

Амортизационные отчисления для персонального компьютера на 1 месяц составят

$$30\,000 / 36 = 833,33 \text{ (руб.)}.$$

Амортизационные отчисления для ноутбука на 1 месяц составят

$$45\,000 / 36 = 1\,250 \text{ (руб.)}.$$

Амортизационные отчисления по ОС, относящиеся на проект, составят:

$$A_{\text{ос}} = 4 \cdot 4 \cdot 833,33 + 1250 \cdot 4 = 18\,333,28 \text{ (руб.)}.$$

Ранее (до начала проекта) Компанией были приобретены:

- Корпоративная лицензия на WebStorm на 3 рабочих мест стоимостью 39 900 (руб.).
- Корпоративная лицензия на Windows 10 на 4 рабочих мест стоимостью 60 000 (руб.).

ПО Webstorm принимается Компанией к учету как расходы будущих периодов со сроком списания 1 год, ПО Windows берется на срок использования компьютеров, то есть на 3 года. Метод списания – линейный.

В качестве ОС используется Windows 10. В качестве сервера БД используется свободно распространяемое ПО – PostgreSQL.

Амортизационные отчисления по РБП, относящиеся на проект, составят:

$$A_{\text{РБП}} = 39\,900 \cdot 4 / 12 + 60\,000 \cdot 4 / 36 = 19\,996,67 \text{ (руб.)}.$$

Суммарные амортизационные отчисления составят: $A = 38\,299,95$ руб.

2.3.5. Расходы на модернизацию и приобретение основных средств

При реализации веб-сервиса для публикации и просмотра объявлений о недвижимости не планируется приобретение новых и модернизация существующих основных средств.

2.3.6. Расходы на приобретение необходимого ПО

При реализации проекта не планируется приобретение ПО.

2.3.7. Расходы на интернет и связь

Так как в Компании, реализующей проект не производится биллинг и тарификация телекоммуникационных услуг в разрезе сотрудников, затраты на интернет и связь войдут в прочие затраты, рассчитываемые как процент от прямых затрат.

2.3.8. Расходы на канцелярские товары и расходные материалы

Затраты на расходные материалы берутся по факту и составляют $P_{p.m.} = 3\,500$ (руб.). К данным затратам относятся затраты на канцтовары, тонер и бумагу для принтера и т.д.

2.3.9. Прочие расходы

Прочие расходы составляют 30% от суммы следующих элементов структуры затрат: $ЗП_{осн}$, $ЗП_{доп}$, $H_{зн}$, $A_{пм}$, A , $P_{мод}$, $P_{ПО}$, $P_{тел}$ и $P_{p.m.}$.

$$P_{p.p.} = 0.3(ЗП_{осн} + ЗП_{доп} + H_{зн} + A_{пм} + A + P_{мод} + P_{ПО} + P_{тел} + P_{p.m.}) \quad (2.2)$$

Таким образом, $P_{p.p.} = 223\,860,21$ (руб.).

2.3.10. Расчёт себестоимости программного продукта

В себестоимость программного продукта входят следующие элементы: $ЗП_{осн}$, $ЗП_{доп}$, $H_{зн}$, $A_{пм}$, A , $P_{мод}$, $P_{ПО}$, $P_{тел}$, $P_{p.m.}$ и $P_{p.p.}$.

Сложив все элементы, можно определить себестоимость программного продукта и услуг по его внедрению: $C_{п.п.} = 970\,060,91$ (руб.).

Структура себестоимости программного продукта отражена в таблица 2.4 и представлена на рис. 2.4.

Таблица 2.4

Структура себестоимости программного продукта

№	Элементы себестоимости	Сумма (руб.)	% в общ. сумме себестоимости
1	Основная заработная плата исполнителя	374 850	38,64
2	Дополнительная заработная плата исполнителя	56 228	5,80
3	Отчисления на социальные нужды (страховые взносы)	129 323	13,33
4	Арендные платежи за производственные (офисные) помещения	144 000	14,84
5	Амортизация используемых основных средств и нематериальных активов	38 299,95	3,95
6	Расходы на модернизацию и приобретение основных средств	-	-
7	Расходы на приобретение необходимого ПО	-	-
8	Расходы на интернет, связь	-	-
9	Расходы на канцелярские товары и расходные материалы	3 500	0,36
10	Прочие расходы	223 860,21	23,08
Итого:		970 060,91	100



Рис. 2.4. Структура себестоимости программного продукта

3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

3.1. Архитектура программного обеспечения

Разрабатываемое программное обеспечение состоит из веб-приложения, а также сервера с базой данных. Архитектуры этих приложений представлены на Рис. 3.1.

Для разработки была выбрана клиент-серверная архитектура, так как клиент (веб-приложение) будет выполнять api-запросы к серверу, в роли серверной части будет выступать сервер на Node.js. Выбор данной архитектуры основан на возможности централизованного хранения данных, которую предоставляет клиент-серверная архитектура, а также возможности поддержки различных типов клиентов.

Архитектура проектируемой системы [2] [6] должна быть гибкой и масштабируемой. Она основана на компонентной архитектуре, где каждый компонент отвечает за конкретную функциональность и может быть многократно использован в других частях приложения [4].

Серверная часть системы разрабатывается на платформе Node.js [10] с использованием фреймворка Express [11]. Это позволяет создать мощный и быстродействующий сервер, способный обрабатывать множество параллельных запросов. Основной функцией сервера в данной архитектуре является обработка API-запросов от клиентской части и взаимодействие с базой данных PostgreSQL [12].

На серверной стороне также используется хранилище для фотографий. Загрузка и выгрузка изображений из данного хранилища осуществляется с помощью библиотеки для загрузки файлов на сервер Multier [17].

Nodemailer [29] – модуль для Node.js, который позволяет отправлять электронные письма с сервера. Он поддерживает Unicode и работает как с SMTP, так и с Sendmail, что делает его универсальным инструментом для отправки электронной почты.

Dadata [30] – сервис, который предоставляет API для структурирования, очистки и подсказок данных. Он может быть использован для автоматического заполнения форм, проверки и коррекции адресов, имён, телефонов и других данных.

ЮKassa [23] – сервис для приема электронных платежей. Он предоставляет API, которое позволяет интегрировать функциональность приема платежей в веб-приложение. ЮKassa поддерживает различные способы оплаты, включая банковские карты, электронные деньги, мобильные платежи и другие.

API [13] сервера играет ключевую роль в взаимодействии между клиентской частью и сервером [7]. Он позволяет выполнять все необходимые CRUD-операции (создание, чтение, обновление, удаление), обрабатывая запросы от клиента и возвращая соответствующие ответы после взаимодействия с базой данных.

База данных PostgreSQL [12] используется для хранения и обработки всех данных приложения. Это мощная, открытая и надежная система управления базами данных, которая обеспечивает все необходимые для работы системы функции, включая транзакционность, поддержку различных типов данных и сложные запросы.

После обработки запросов на сервере, данные отправляются обратно в клиентскую часть системы. Для разработки клиентской части используется фреймворк React [8] [15] в сочетании с языком программирования TypeScript [9]. Использование TypeScript [16] обеспечивает строгую типизацию, что упрощает процесс разработки и последующего сопровождения приложения.

Такой подход к проектированию архитектуры позволяет создать высокопроизводительное, масштабируемое и гибкое веб-приложение, способное адаптироваться к изменяющимся требованиям и условиям использования.

Тип приложения – Web-приложение. Ниже приведена архитектура проектируемой системы (рис. 3.1).

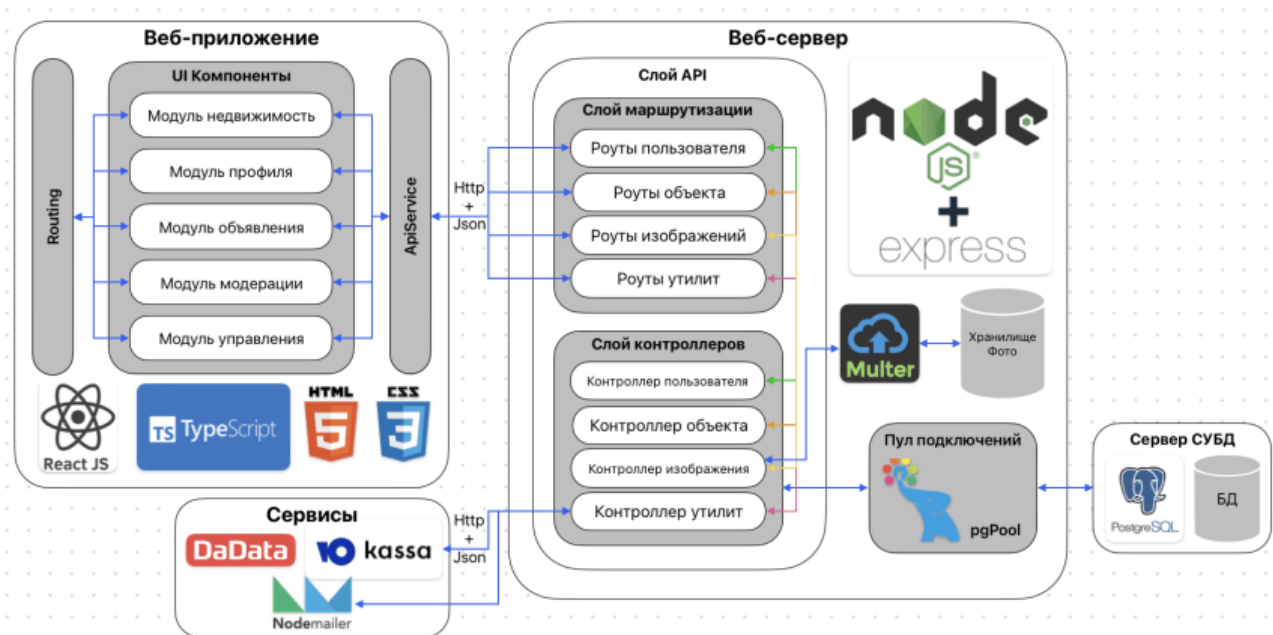


Рис. 3.1. Архитектура проектируемой системы

3.2. Модель данных

Благодаря диаграмме классов имеется возможность строить логическое представление системы, на основании которого формируется исходный код описанных классов.

Диаграмма классов позволяет создавать логическое представление системы, на основе которого создается исходный код описанных классов.

Для разрабатываемого программного обеспечения была разработана нормализованная модель представления данных (рис. 3.2). Модель базы данных соответствует четвертой нормальной форме. На ней представлены следующие основные таблицы системы:

users – содержит информацию о пользователях системы. Каждая запись в таблице представляет собой уникального пользователя с полями.

objects – содержит информацию об объектах недвижимости. Каждая запись представляет собой уникальный объект с полями.

ownership – связывает объекты недвижимости с их владельцами и агентами.

object_images – содержит изображения объектов недвижимости. Каждая запись представляет собой уникальное изображение.

features – содержит информацию о характеристиках объектов недвижимости. Каждая запись представляет собой уникальную характеристику.

object_features – связывает объекты недвижимости с их характеристиками. Каждая запись представляет собой уникальное соответствие между объектом и его характеристикой.

schema_version – содержит информацию о текущей версии схемы базы данных.

payments – таблица содержит информацию о платежах пользователей.

purchases – содержит информацию о пополнениях, совершенных пользователями.

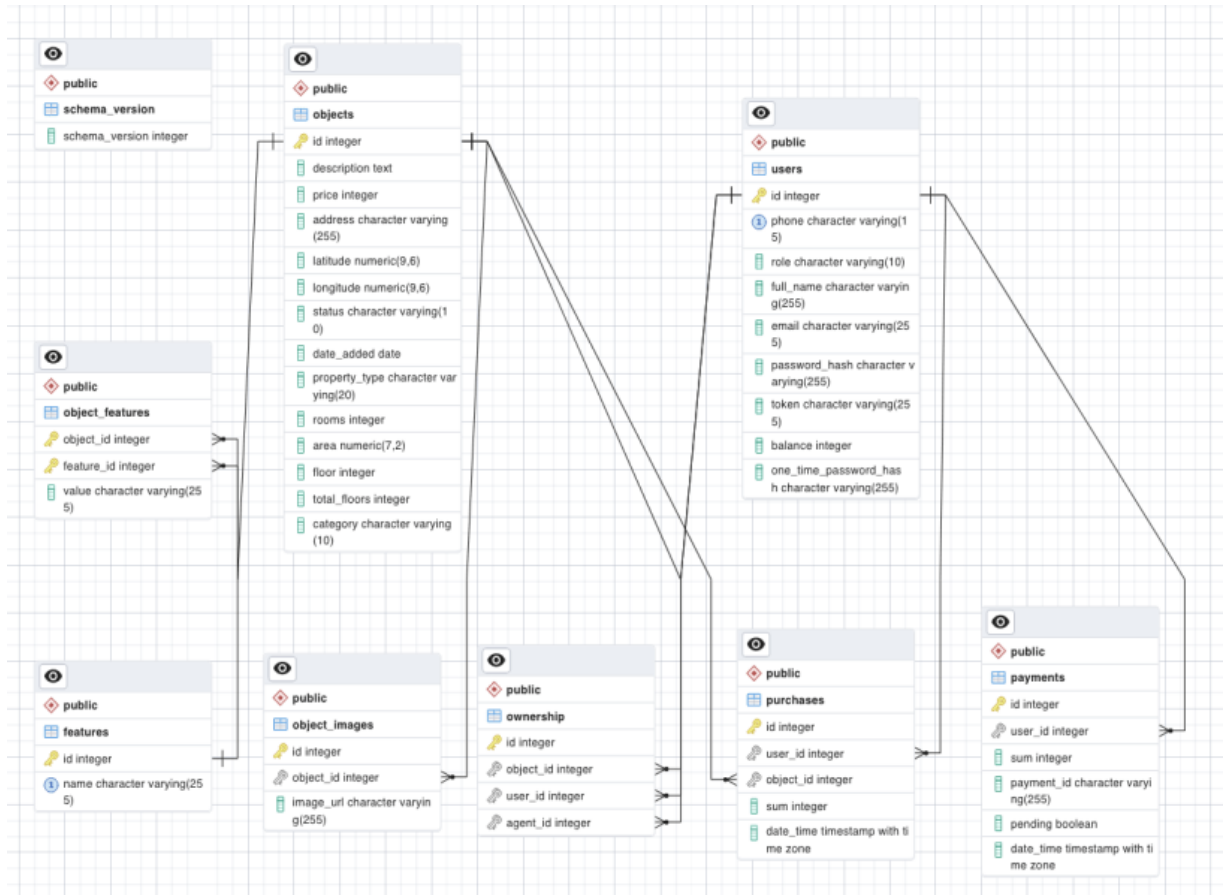


Рис. 3.2. Диаграмма классов для проектируемой системы

3.3. Низкоуровневое проектирование

3.3.1. Посмотреть список объявлений о недвижимости

Вариант использования «Посмотреть каталог объектов» позволяет любому пользователю просмотреть карточки объектов для краткого ознакомления и выбора подходящего объявления.

1. Пользователь находится на главной странице;
2. Пользователь нажимает на кнопку каталог;
3. Система отображает список всех объявлений;

Для этого сценария была разработана диаграмма коммуникации (рис. 3.3).

Также с реализацией компоненты карточки объявления можно ознакомиться в приложении (Приложение А).

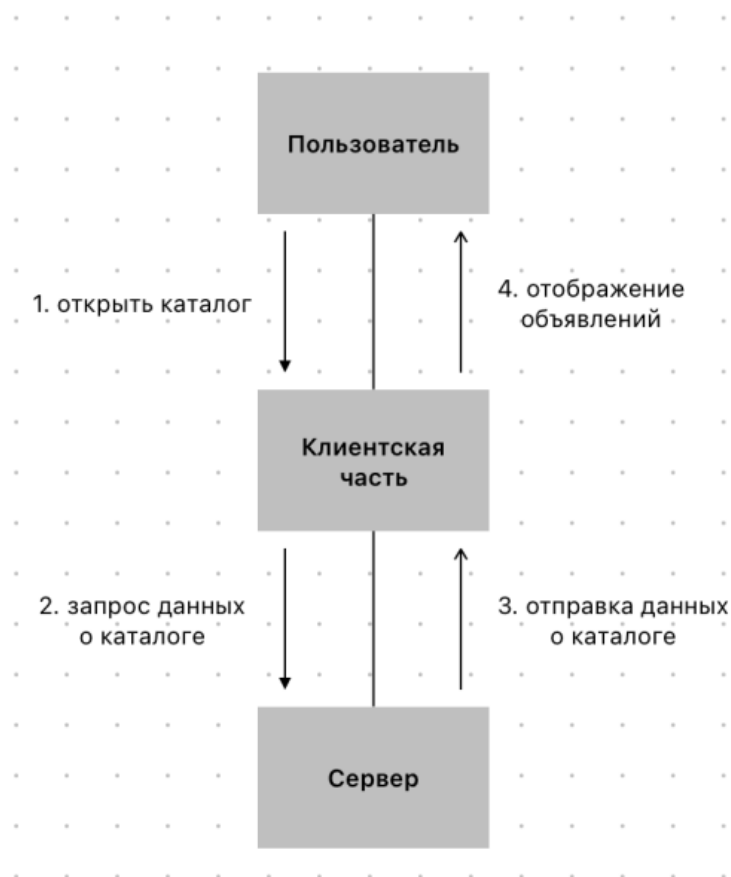


Рис. 3.3. Диаграмма коммуникаций для варианта использования «Посмотреть каталог объектов»

3.3.2. Фильтр объектов

Вариант использования «Фильтр объектов» позволяет пользователю отфильтровать объекты по определенным выбранным параметрам.

1. Пользователь находится в каталоге объектов;
2. Пользователь нажимает на кнопку фильтр;
3. Система отображает допустимые критерии объектов;
4. Выбирая нужные чек-боксы фильтра и заполняя необходимую информацию, пользователь меняет критерии искомого объекта;
5. Пользователь нажимает кнопку применить;
6. Система отображает объекты, соответствующие введенным критериям.

Для данного сценария была разработана диаграмма деятельности (рис. 3.4).

Также в приложении можно ознакомиться с реализацией компоненты searchBar, которая содержит в себе часть фильтра и поиска (Приложение В).

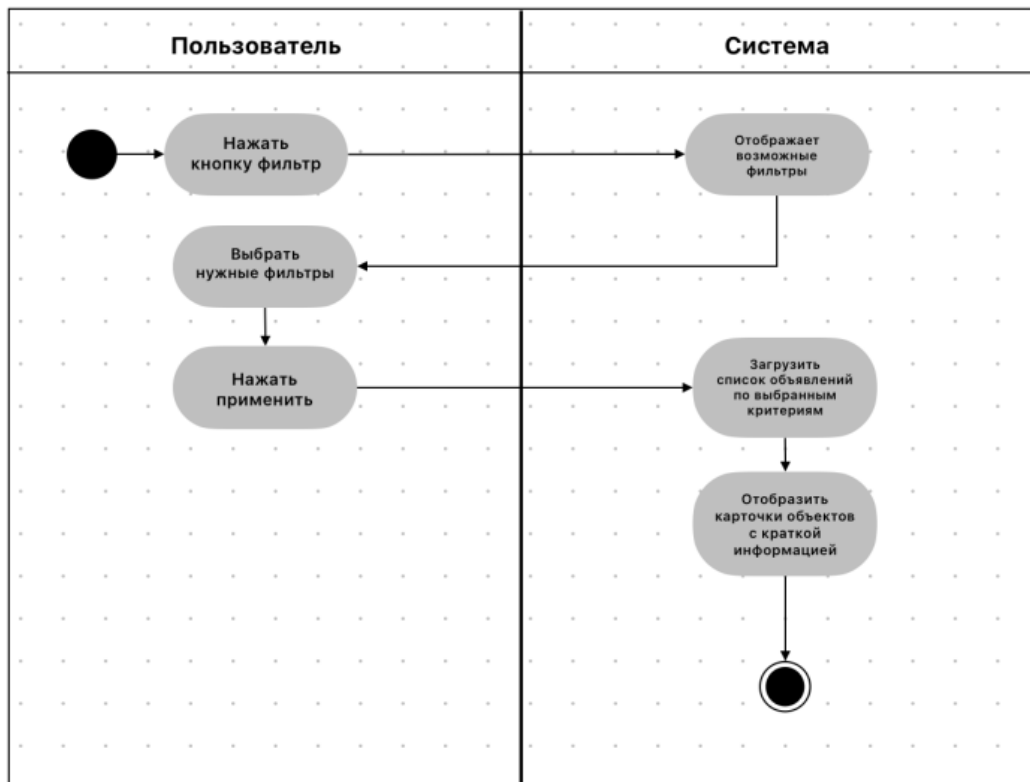


Рис. 3.4. Диаграмма деятельности для варианта использования «Фильтр объектов»

3.3.3. Опубликовать объявление

Пользователь для данного варианта использования должен пройти аутентификацию.

1. Авторизованный пользователь находится на главной странице;
2. Пользователь нажимает кнопку навигации «Опубликовать»;
3. Система открывает форму заполнения информации об объекте;
4. Пользователь вводит необходимые данные и нажимает кнопку «сохранить»;
5. Система открывает пользователю страницу размещенных объявлений пользователя, где появилось новое объявление.
6. Далее после смены пользователем статуса на публикацию, объявление отправляется на модерацию.

Для данного сценария была разработана диаграмма деятельности (рис. 3.5) и диаграмма состояний (рис. 3.6).

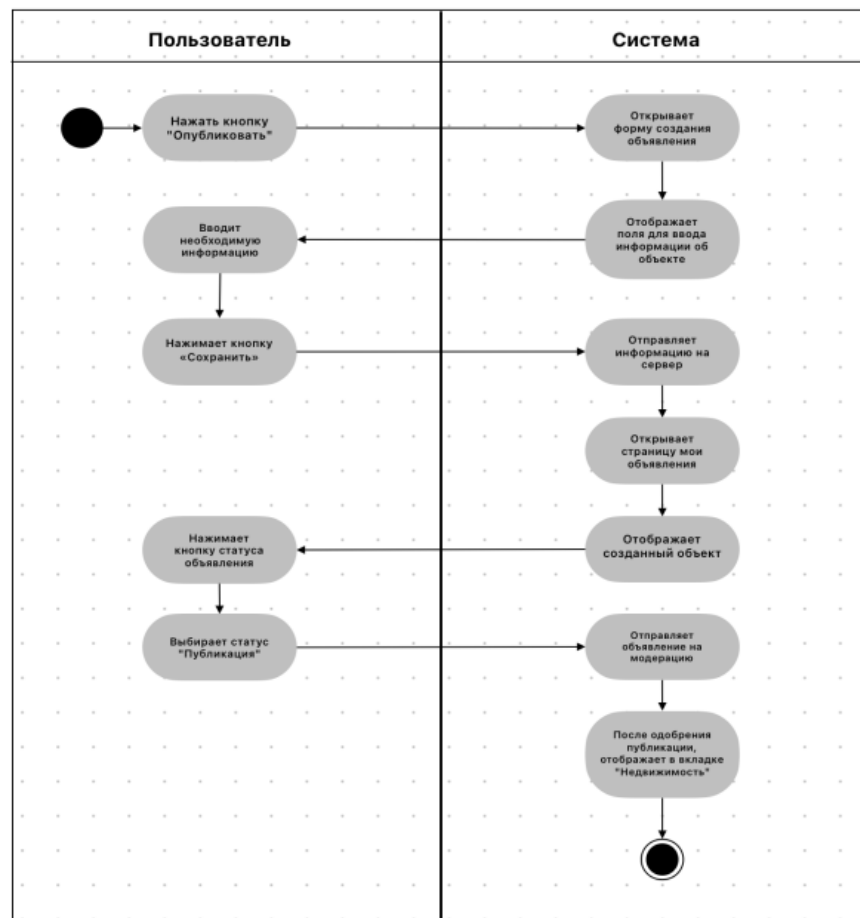


Рис. 3.5. Диаграмма деятельности для варианта использования «Опубликовать объявление»

Ниже представлена диаграмма состояний для варианта использования «Опубликовать объявление» (рис. 3.6).

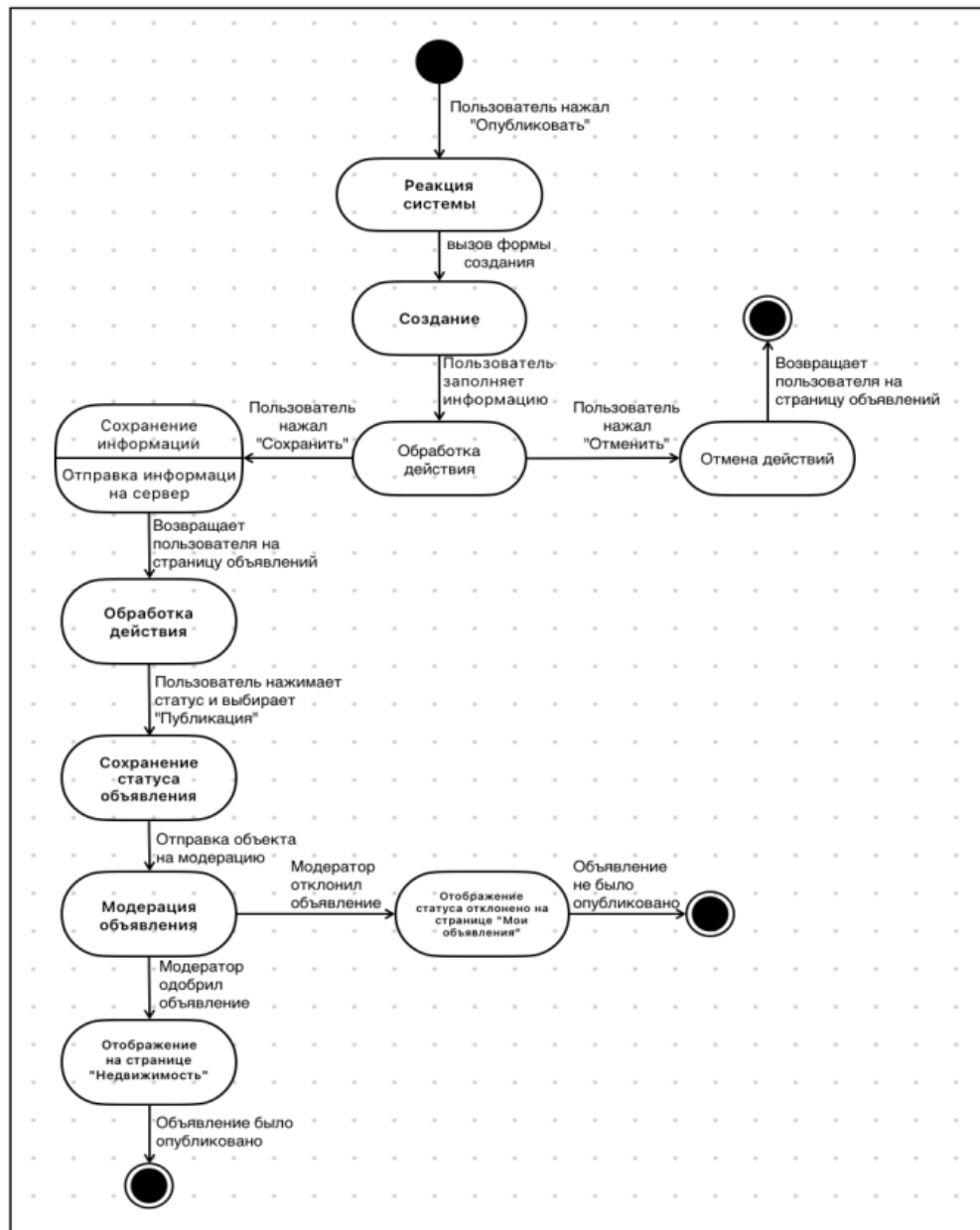


Рис. 3.6. Диаграмма состояний для варианта использования «Опубликовать объявление»

3.4. Проектирование интерфейса

При создании веб-сервиса для публикации и просмотра объявлений о недвижимости одним из ключевых моментов становится разработка простого, удобного в использовании и интуитивно понятного пользовательского интерфейса. Процесс проектирования интерфейса данного веб-сервиса осуществлялся с помощью Balsamiq Wireframes [20], инструмента для создания макетов и эскизов. В процессе проектирования учитывались выводы,

полученные в ходе анализа схожих веб-сайтов, а также требования технического задания.

Разработанный интерфейс адаптирован под различные устройства и использует шрифт, удобный для чтения. Весь текст на сайте сделан в едином стиле на каждой странице. Фон сайта выполнен в нейтральных светлых тонах. Интерфейс не перегружен лишними элементами, при наведении курсора на элементы срабатывает небольшая анимация сдвига и выделения контуров. На сайте отсутствуют элементы, способные отвлечь внимание пользователя.

Структура основных блоков и элементов страницы основывается на модульной сетке. Для минимизации возможных ошибок со стороны пользователя и повышения визуальной привлекательности интерфейса между элементами сохранены приемлемые отступы. В рамках модульной сетки элементы выровнены по левому краю.

Скорость работы пользователя повышается благодаря уменьшению количества шагов при переходе между страницами веб-приложения. Для упрощения процесса обучения пользователя все поля для ввода информации имеют подписи. Иконки, использованные в интерфейсе, являются понятными и выполнены в едином стиле.

На рис. 3.7 представлен макет страницы для администратора, из-за чего открывается расширенный функционал области навигации. На данной странице находятся все объявления о недвижимости и большую ее часть занимают карточки с краткой информацией об объекте и ее ценой. Чуть выше находится область с фильтром и поиском.

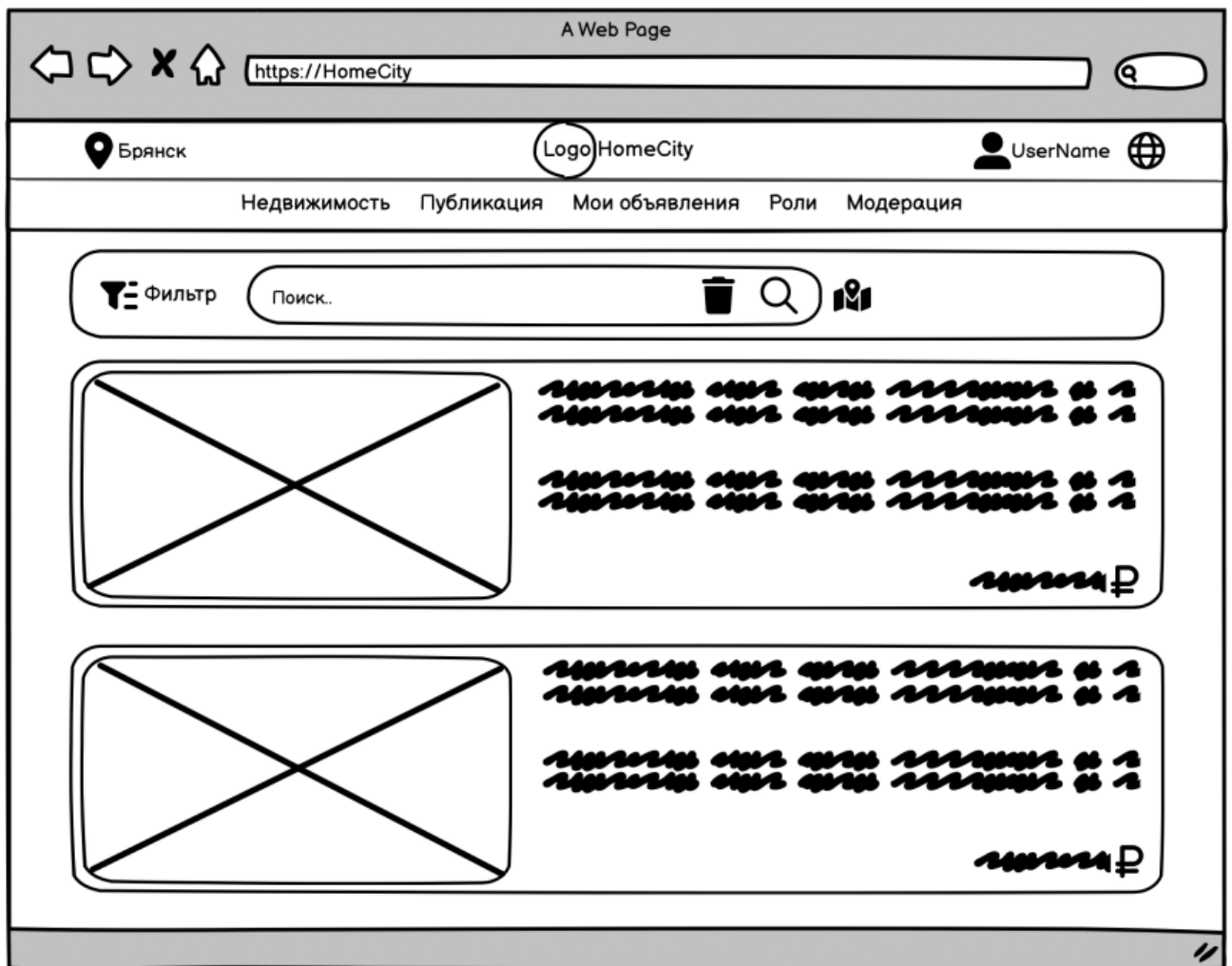


Рис. 3.7. Макет страницы объявлений недвижимости

Для создания собственного объявления необходимо нажать в навигации на «Публикация». Открывается страница создания объявления (рис. 3.8). На данной странице можно подгрузить необходимые фотографии, и заполнить информацию объекте недвижимости, и нажав на кнопку сохранить, объявление будет направлено на модерацию.

A Web Page

https://HomeCity

Брянск Logo HomeCity UserName

Недвижимость Публикация Мои объявления Роли Модерация

Создать объявление

Адрес

Описание

Цена

Тип Квартира Дом

Комнат

Площадь

Этаж

Всего этажей

Рис. 3.8. Макет страницы создания объявления

Для просмотра пользователей системы и редактирования их ролей необходима страница для администрации (рис. 3.9). На данной странице администратор сможет увидеть всех пользователей, информацию о пользователях и их роли. Также с помощью строки поиска он может найти нужного пользователя по почте, имени или номеру телефона. Также администратор может, нажав на кнопку роли, выбрать из списка необходимую роль для пользователя и изменить ее.

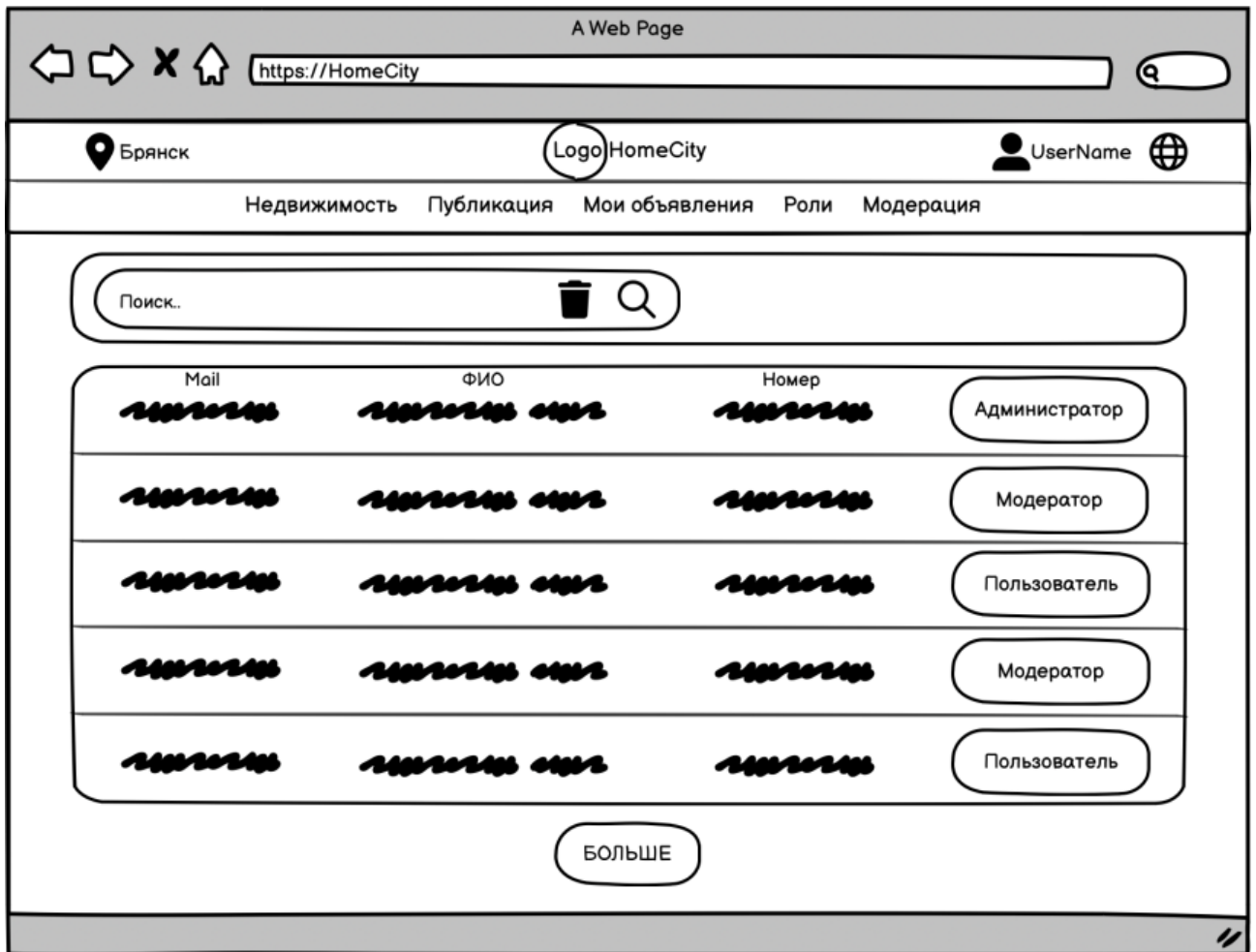


Рис. 3.9. Макет страницы администратора

3.5.API Документация

3.5.1. Обзор

Этот API позволяет управлять объектами, пользователями и изображениями объектов в базе данных.

Благодаря организации API можно создавать, читать, обновлять и удалять пользователей, объекты, а также загружать, читать и удалять изображения, связанные с определенным объектом.

3.5.2. Базовый URL

Базовый URL для всех запросов к API – <https://localhost:8080/api>.

3.5.3. Эндпоинты для работы с пользователями

1. Создание пользователя

- URL: /users
- Метод: POST
- Тело запроса (application/json):
 - phone: Номер телефона пользователя (строка)
 - full_name: Полное имя пользователя (строка)
 - email: Электронная почта пользователя (строка)

2. Получение списка пользователей

- URL: /users
- Метод: GET

3. Получение информации о конкретном пользователе

- URL: /users/:id
- Метод: GET
- Параметры URL: id - идентификатор пользователя

4. Обновление пользователя

- URL: /users
- Метод: PUT
- Тело запроса (application/json):
 - id: Идентификатор пользователя (строка)
 - phone: Номер телефона пользователя (строка, необязательный)
 - role: Роль пользователя (строка, необязательный, может быть 'user', 'admin', 'agent')
 - full_name: Полное имя пользователя (строка, необязательный)
 - email: Электронная почта пользователя (строка, необязательный)

5. Обновление роли пользователя

- URL: /users/:id
- Метод: PUT
- Параметры URL: id - идентификатор пользователя
- Тело запроса (application/json):

- role: Новая роль пользователя (строка, может быть 'user', 'admin', 'agent')

6. Удаление пользователя

- URL: /users/:id
- Метод: DELETE
- Параметры URL: id - идентификатор пользователя

7. Получение профиля пользователя

- URL: /profile
- Метод: GET

8. Обновление профиля пользователя

- URL: /profile
- Метод: PUT
- Тело запроса (application/json):
 - phone: Номер телефона пользователя (строка, необязательный)
 - password: Пароль пользователя (строка, необязательный)
 - full_name: Полное имя пользователя (строка, необязательный)
 - email: Электронная почта пользователя (строка, необязательный)

9. Отправка одноразового пароля

- URL: /sendOneTimePassword
- Метод: POST
- Тело запроса (application/json):
 - email: Электронная почта пользователя (строка)

1. Получение платежей пользователя

- URL: /user/:id/payments
- Метод: GET
- Параметры URL: id - идентификатор пользователя

2. Получение покупок пользователя

- URL: /user/:id/purchases
- Метод: GET
- Параметры URL: id - идентификатор пользователя

3.5.4. Эндпоинты для работы с объектами

1. Создание объекта

- URL: /objects
- Метод: POST
- Тело запроса (application/json):
 - description: Описание объекта (строка)
 - price: Цена объекта (число)
 - address: Адрес объекта (строка)
 - status: Статус объекта (строка)
 - user_id: Идентификатор пользователя, являющегося владельцем (строка)
 - property_type: Тип недвижимости (строка)
 - rooms: Количество комнат (число)
 - area: Площадь объекта (число)

2. Добавление агента к объекту

- URL: /objects/:object_id/agent
- Метод: PUT
- Параметры URL: object_id - идентификатор объекта
- Тело запроса (application/json):
 - agent_id: Идентификатор агента (строка)

3. Получение всех объектов

- URL: /objects
- Метод: GET

4. Получение информации об определенном объекте

- URL: /objects/:id
- Метод: GET
- Параметры URL: id - идентификатор объекта

5. Получение информации об владельце/агенте объекта

- URL: /objects/:object_id/ownership
- Метод: GET

- Параметры URL: object_id - идентификатор объекта

6. Обновление информации об объекте

- URL: /objects
- Метод: PUT
- Тело запроса (application/json):
 - id: Идентификатор объекта (строка)
 - description: Описание объекта (строка, необязательный)
 - price: Цена объекта (число, необязательный)
 - address: Адрес объекта (строка, необязательный)
 - status: Статус объекта (строка, необязательный)
 - agent_id: Идентификатор агента (строка, необязательный)

7. Обновление категории объекта

- URL: /object/:object_id/category
- Метод: PUT
- Параметры URL: object_id - идентификатор объекта
- Тело запроса (application/json):
 - category: Категория объекта (строка)

8. Обновление информации об агенте объекта

- URL: /objects/agent
- Метод: PUT
- Тело запроса (application/json):
 - id: Идентификатор объекта (строка)
 - agent_id: Идентификатор агента (строка)

9. Удаление объекта

- URL: /objects/:id
- Метод: DELETE
- Параметры URL: id - идентификатор объекта

3.5.5. Эндпоинты для работы с изображениями объектов

1. Загрузка изображения
 - URL: /object_images/:id
 - Метод: POST
 - Параметры URL: id - идентификатор объекта
 - Тело запроса: Файл изображения в формате multipart/form-data
2. Получение изображений для конкретного объекта
 - URL: /object_images/:id
 - Метод: GET
 - Параметры URL: id - идентификатор объекта
3. Удаление изображения
 - URL: /object_images/:object_id/:id
 - Метод: DELETE
 - Параметры URL:
 - object_id - идентификатор объекта
 - id - идентификатор изображения

3.5.6. Эндпоинты для работы с утилитами

1. Получение местоположений
 - URL: /utils/getLocations
 - Метод: POST
 - Тело запроса (application/json):
 - name: Название местоположения (строка)
2. Пополнение баланса
 - URL: /utils/refill
 - Метод: POST
 - Тело запроса (application/json):
 - sum: Сумма пополнения (число)
 - Требуется авторизация (cookies)

4. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

4.1. Испытания

Тестирование играет ключевую роль и является неотъемлемой частью процесса разработки программного обеспечения.

Тестирование – это процесс, в ходе которого оценивается, насколько реальная работа программы соответствует её предполагаемому поведению. Для проверки качества разрабатываемого продукта были выбраны следующие виды тестирования:

- функциональное тестирование;
- кросс-браузерное тестирование;
- тестирование пользовательского интерфейса;

4.2. План испытаний

Был разработан план тестирования, для выполнения тестов согласно выбранным методам.

В ходе функционального тестирования следует проверить:

- аутентификацию;
- изменение данных в профиле;
- отображение карточек объектов и объявлений пользователей;
- фильтр объектов;
- поиск объектов;
- отображение объектов на карте;
- создание/редактирования объявления об объекте;
- модерация объектов;
- администрирование ролей пользователей;
- перевод основных элементов сайта на английский.

Кросс-браузерное тестирование включает в себя проверку корректной работы на таких браузерах как:

- opera [24];
- microsoft edge [25];
- firefox [26];
- google chrome [27];
- яндекс [28];

Тестирование пользовательского интерфейса включает в себя:

- адаптивность элементов при изменении размера окна браузера;
- правильность перемещения фокуса в окне;
- изменение вида элемента при наведении курсора;
- тестирование ввода текстовых данных;
- правильность перехода по внутренним ссылкам;
- доступность главного меню с любой страницы сайта.

4.3.Проверка функциональных требований

Чтобы убедиться в полноте функционала разрабатываемой системы, проводится проверка на соответствие функциональным требованиям.

4.3.1. Проверка фильтра и поиска

При выборе фильтра цены до 8 000 000 и количества комнат до 2х (рис. 4.1) и при нажатии кнопки принять, на странице отобразились карточки объектов недвижимости, соответствующие этим 2м критериям (рис. 4.2). Также на данной странице видно по точке возле фильтра, что фильтр был применен и он не является пустым. При нажатии на карточку, можно перейти к детальной информации недвижимости. Также на данной странице, при нажатии на пиктограмму карты, которая находится правее строки поиска, можно перейти к отображению объектов на карте.

Фильтр

Цена	От	До 8000000
Тип	<input type="checkbox"/> КВАРТИРА <input type="checkbox"/> КОМНАТА <input type="checkbox"/> ДОМ <input type="checkbox"/> ЗЕМЛЯ	
Количество комнат	От	До 2
Площадь	От	До
Этаж	От	До
Всего этажей	От	До


[ОТМЕНА](#)
[ОЧИСТИТЬ](#)
[ПРИМЕНИТЬ](#)

Рис. 4.1. Диалоговое окно фильтра

Брянск
HomeCity
Konstantin Isakov

[Недвижимость](#)
[Опубликовать](#)
[Мои объявления](#)
[Пользователи](#)
[Модерация](#)

Фильтры



2-комн. квартира площадью 35.50 м² на 5/10 этаже

Просторная квартира в новом доме

Цена: 4 499 994 Р

[БОЛЬШЕ](#)

© HomeCity

Рис. 4.2. Страница с объявлениями с примененным фильтром

После применения фильтра, его можно сбросить. Также есть возможность использовать строку поиска, чтобы искать по описанию, либо по улице, на которой находится объект, как и показано на рис. 4.3.

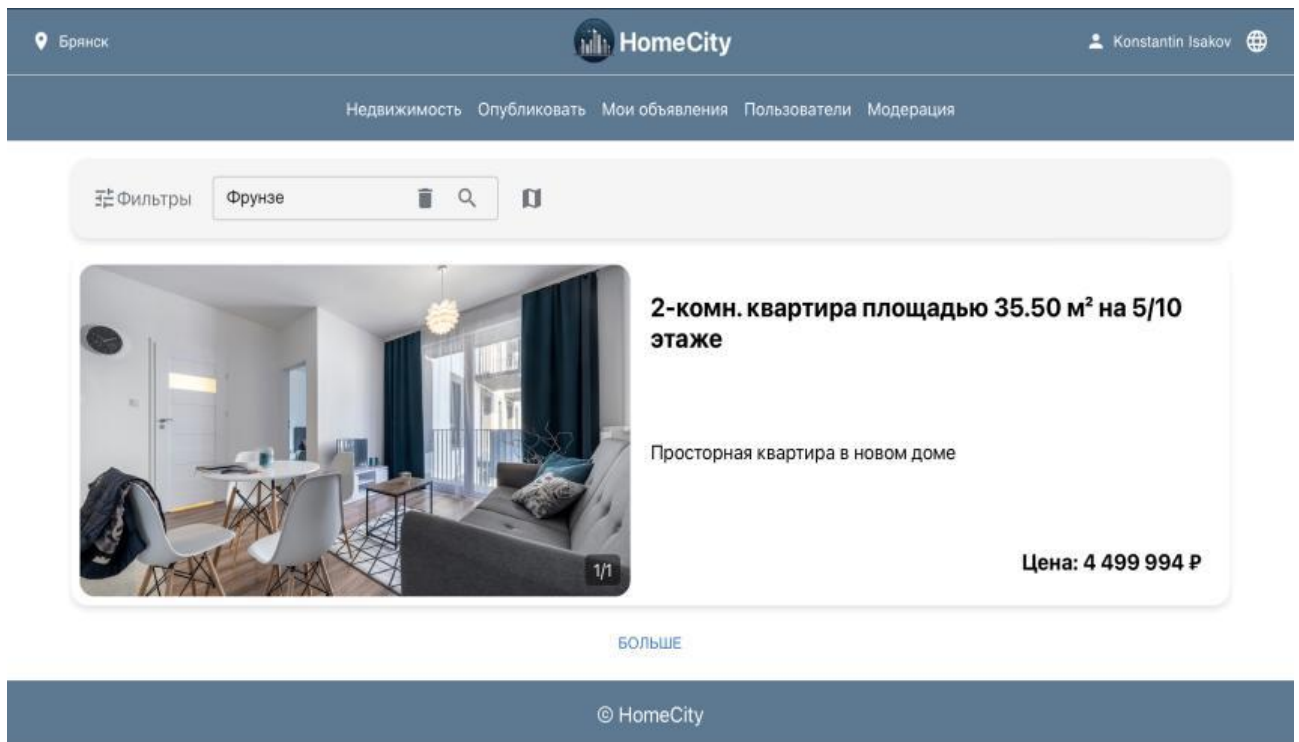


Рис. 4.3. Страница с объявлениями с применением поиска по улице

4.3.2. Проверка отображения на карте

Для перехода на страницу с отображением объектов на карте, необходимо нажать на пиктограмму карты, в строке с фильтром и поиском, как видно на рис. 4.3. На данной странице можно увидеть точки расположений объектов на интерактивной карте (рис. 4.4). Также на этой странице доступен фильтр и поиск, при выборе которых, отсеиваются ненужные точки на карте. На любую точку можно нажать и тогда снизу карты появится карточка с краткой информацией выбранной недвижимости, на которую можно также нажать и перейти к детальной информации об объекте

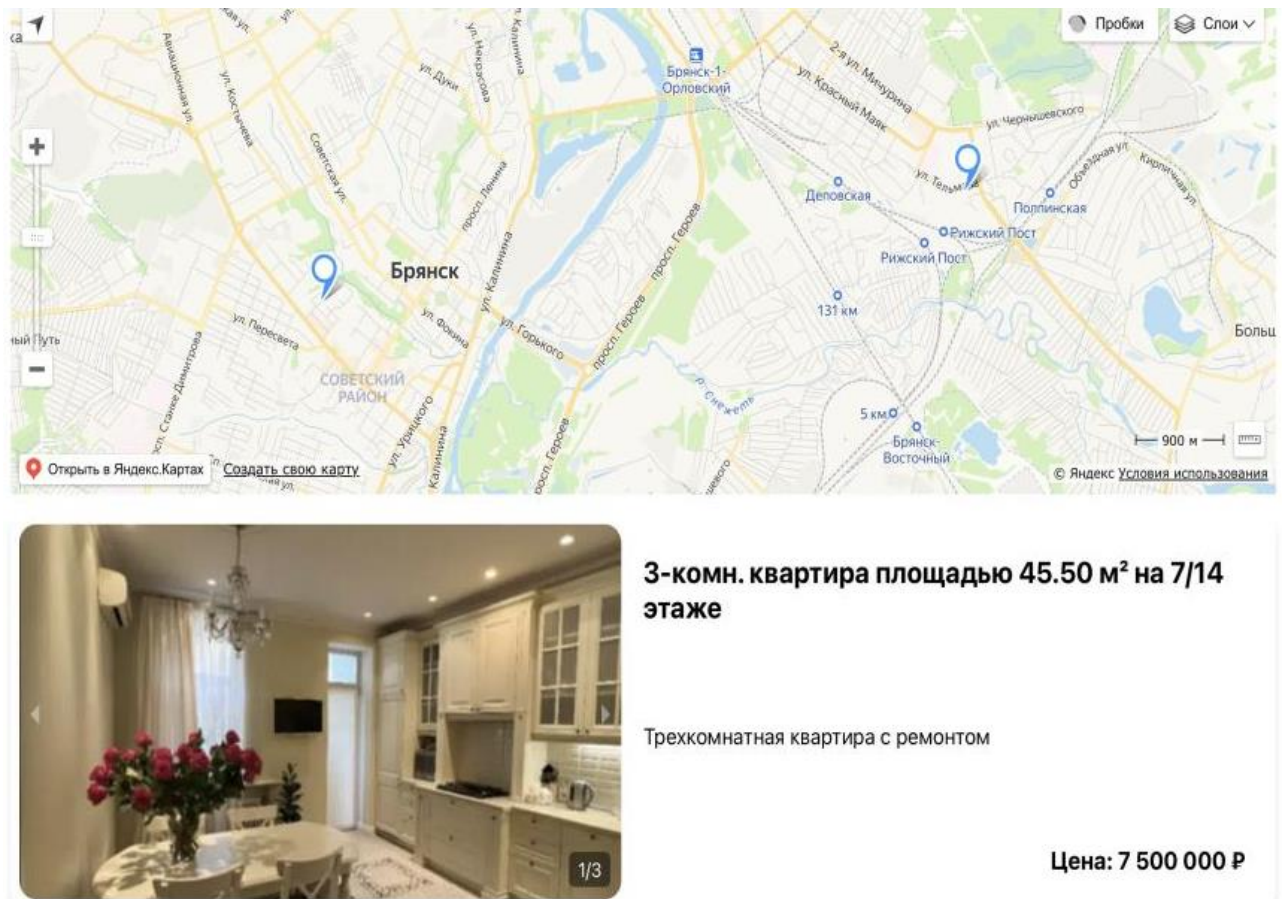



Рис. 4.4. Страница с отображением объектов на карте

4.3.3. Проверка страницы мои объявления

На данной странице отображаются объекты, которые размещал пользователь (рис. 4.5). Для доступа к этой странице, необходимо авторизоваться и после разместить свое объявление. На данной странице также присутствует фильтр и поиск, также как и отображение на карте.


Пользователь может нажать кнопку изменить, и перейти к редактированию информации об объекте, после нажатия кнопки сохранить, объявление отправится на модерацию. Нажав на кнопку со статусом, открывается диалоговое меню (рис. 4.6), в котором он может изменить статус объявления на нужный.



2-комн. квартира площадью 35.50 м² на 5/10 этаже

Просторная квартира в новом доме

ОДОБРЕНО ИЗМЕНИТЬ Цена: 4 499 994 Р



5-комн. квартира площадью 200.00 м² на /2 этаже

Элитная квартира

ОТКЛОНЕНО ИЗМЕНИТЬ Цена: 50 000 000 Р

Рис. 4.5. Страница мои объявления

Выбор категории

☐ Черновик

☒ Публикация

☐ Архив

☐ Удалено

ОТМЕНА ИЗМЕНИТЬ

Рис. 4.6. Диалоговое окно изменения статуса объявления

4.3.4. Проверка страницы модерации

На данной странице отображаются объявления, которые были отправлены на модерацию, здесь также работают все функции фильтрации и поиска объекта (рис. 4.7). На данной странице модератор может ознакомиться с объявлением, нажав на карточку объявления, после чего может нажать на кнопку статуса и поменять статус на одобрено или отклонено (рис. 4.8).

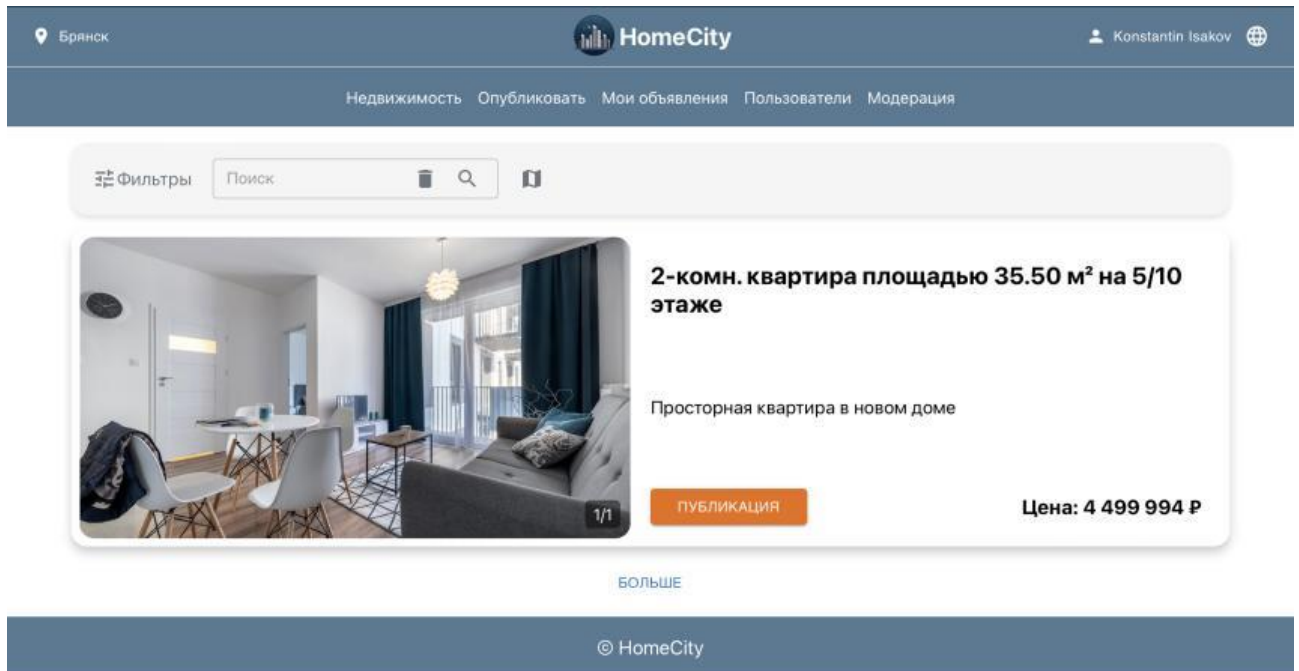


Рис. 4.7. Страница модерации

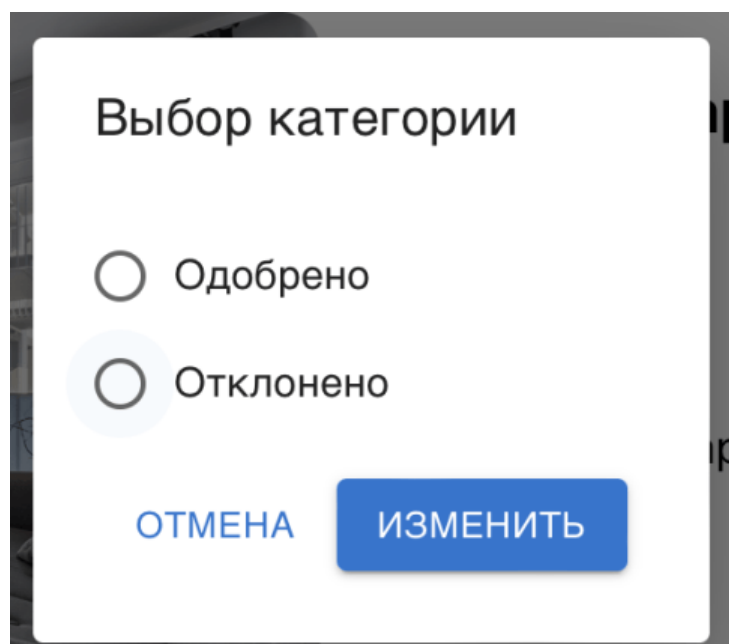


Рис. 4.8. Диалоговое окно изменения статуса объявления при модерации

5. ОРГАНИЗАЦИОННАЯ ЧАСТЬ

Тема дипломной работы связана с разработкой Веб-сервис для публикации и просмотра объявлений о недвижимости, представляет собой инновационное решение, которое облегчает процесс поиска и публикации информации о недвижимости.

Работа веб-разработчиков и других технических специалистов, занятых в этом проекте, требует высокой концентрации внимания и активации многих психологических функций. Все участники проекта подвержены воздействию различных вредных и опасных факторов производственной среды.

Особенности режима и характера работы, в сочетании с значительным умственным напряжением, могут привести к изменению функционального состояния центральной нервной системы у работников, а также нервно-мышечного аппарата рук при работе с клавиатурой. Нерациональное размещение и конструкция элементов рабочего места могут привести к необходимости поддержания неудовлетворительной рабочей позы. Длительное пребывание в таком дискомфорте приводит к увеличению напряжения мышц и может способствовать развитию общей усталости и снижению работоспособности. При длительной работе за экраном компьютера зрительный аппарат значительно напрягается, что может вызывать жалобы на головную боль, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, пояснице, в области шеи и рук. Особенностью опасностей, связанных с работой на компьютере, является то, что они невидимы, но оказывают сильное негативное воздействие на человека.

Выделим опасные и вредные факторы на рабочем месте, оборудованном компьютерной техникой, согласно СП 2.2.3670-20 "Санитарно-эпидемиологические требования к условиям труда" и СанПиН 1.2.2.3685-21 "Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания".

5.1. Опасные и вредные факторы

Работа программиста, участвующего в создании веб-сервиса для публикации и просмотра объявлений о недвижимости, также связана с определенными рисками и потенциально вредными условиями труда. В процессе деятельности этих специалистов следующие опасные и вредные производственные факторы могут оказывать влияние:

1. Физические:

- повышенные уровни электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения;
- повышенный уровень статического электричества;
- повышенные уровни запыленности воздуха рабочей зоны;
- пониженная или повышенная влажность и подвижность воздуха рабочей зоны;
- повышенный уровень шума;
- повышенный или пониженный уровень освещенности;
- повышенный уровень прямой и отраженной блескости;
- повышенный уровень ослепленности;
- неравномерность распределения яркости в поле зрения;
- повышенная яркость светового изображения;
- повышенный уровень пульсации светового потока;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.

2. Химические:

- повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола, формальдегида и полихлорированных бифенилов.

3. Психофизиологические:

- напряжение зрения и внимания;
- интеллектуальные и эмоциональные нагрузки;

- длительные статические нагрузки и монотонность труда;
- большой объем информации, обрабатываемой в единицу времени;
- нерациональная организация рабочего места.

4. Биологические:

- повышенное содержание в воздухе рабочей зоны микроорганизмов.

5.2. Микроклимат

Тепловые эмиссии – один из побочных эффектов работы компьютерного оборудования, что может вызвать повышение температуры и снижение относительной влажности в рабочей области. Важно обеспечить поддержание нормативных микроклиматических показателей в помещениях, где расположены компьютеры.

Так как основная задача при использовании разрабатываемого программного решения связана с работой на компьютере, и она влечет за собой нервно-эмоциональное напряжение, становится необходимым поддержание оптимальных параметров микроклимата для работ 1а и 1б (таблица 5.1). Для достижения этого рабочее пространство должно быть оборудовано системами отопления и кондиционирования воздуха.

Таблица 5.1

Оптимальные параметры микроклимата

Период года	Категория работ по уровню энергозатрат	Температура воздуха, (°C)	Относительная влажность	Скорость движения воздуха, (м/с)
Холодный (ниже +10°C)	1а (до 139)	22-24	40-60	0,1
	1б (140-174)	21-23		
Теплый (+10°C и выше)	1а (до 139)	23-25	40-60	0,1
	1б (140-174)	22-24		

5.3.Профилактическая гимнастика

Неправильное расположение и дизайн компонентов рабочего пространства могут заставить работника поддерживать неудобную позу в течение длительного времени. Продолжительное нахождение в такой некомфортной ситуации приводит к усиленному напряжению мышц, что может способствовать увеличению утомления и снижению эффективности труда. Поэтому, чтобы предотвратить возможные проблемы со здоровьем и поддерживать высокую эффективность труда, крайне важно внедрить регулярные профилактические гимнастические упражнения прямо на рабочем месте.

Для устранения статического и нервно-эмоционального напряжения рекомендуются обычные физические упражнения, которые нацелены на верхнюю часть туловища, такие как рывки руками, повороты и "рубка дров".

Для снятия напряжения в зрении рекомендуется проводить зрительную гимнастику. Небольшая продолжительность этой гимнастики (1 минута) при регулярном выполнении позволяет эффективно предотвратить утомление. Эффективность зрительной гимнастики объясняется периодическим переключением зрения с ближнего на дальнее при выполнении специальных упражнений, что способствует снятию напряжения с цилиарной мышцы глаза и активизации восстановительных процессов аккомодационного аппарата глаза, в результате чего функция зрения нормализуется. Кроме того, имеется специальное упражнение с меткой на стекле, предназначенное для тренировки и развития аккомодационной функции глаза.

Упражнения выполняются в положении сидя или стоя, с отвернутым взглядом от экрана и ритмичным дыханием, обеспечивая максимальную амплитуду движения глаз.

Упражнения для глаз:

1. Создать сильное напряжение глазных мышц, зажмурив их на некоторое время (счет 1-4), затем расслабить глазные мышцы, открыв глаза и посмотрев вдаль (счет 1-6). Повторить это упражнение 4-5 раз.

2. Фокусироваться на переносице и задерживать взгляд на ней (счет 1-4), не доводя глаза до усталости. Затем открыть глаза и направить взгляд вдаль (счет 1-6). Повторить это упражнение 4-5 раз.

3. Без поворота головы направить взгляд вправо и зафиксировать его (счет 1-4), затем направить взгляд вдаль прямо (счет 1-6). Аналогично выполнить упражнение, фиксируя взгляд влево, вверх и вниз. Повторить это упражнение 3-4 раза.

4. Быстро переместить взгляд по диагонали: справа вверх - слева вниз, затем прямо вдаль (счет 1-6); затем слева вверх - справа вниз и снова посмотреть вдаль (счет 1-6). Повторить 4-5 раз.

Таким образом, рекомендуется выполнять физические упражнения для снятия статического и нервно-эмоционального напряжения, особенно на верхнюю часть туловища, а также проводить зрительную гимнастику для снятия напряжения в зрении. Упражнения для глаз, описанные выше, выполняются сидя или стоя, отвернувшись от экрана, с ритмичным дыханием и максимальной амплитудой движения глаз.

Для снятия локального утомления можно выполнить следующие упражнения:

1. В исходном положении, встать на носки, руки поднять вверх и потянуться за ними. Затем опустить руки вниз, скрестить их перед грудью и наклонить голову вперед. Повторить данное движение 6-8 раз с быстрым темпом.

2. В стойке ноги врозь, руки вытянуть вперед. Сделать поворот туловища вправо, махнуть левой рукой вправо, а правой рукой назад за спину. Затем выполнить аналогичное движение в другую сторону. Упражнения выполняются с размахом и динамикой. Повторить 6-8 раз с быстрым темпом.

3. В исходном положении согнуть правую ногу вперед и, обхватив голень руками, притянуть ее к животу. Затем приставить ногу, руки поднять вверх-наружу и повторить то же самое с другой ногой. Повторить движение 6-8 раз с средним темпом.

Упражнение для улучшения кровообращения мозга включает следующие действия:

1. В исходном положении положить руки за голову, развести локти в стороны и наклонить голову назад. Затем протянуть локти вперед, опустить руки вниз и наклонить голову вперед. Повторить данное движение 4-6 раз с медленным темпом.

2. В стойке с ногами врозь, сжимать кулаки. Сделать мах левой рукой назад, а правой рукой вверх-назад. Затем поменять положение рук и выполнить встречные махи. Завершить движение рывком рук назад. Повторить это упражнение 6-8 раз с средним темпом.

3. Сидя на стуле, отвести голову назад и плавно наклонить ее назад. Затем наклонить голову вперед, не поднимая плечи. Повторить данное движение 4-6 раз с медленным темпом.

Данные упражнения оказывают механическое воздействие на стенки шейных кровеносных сосудов, повышая их эластичность. Они также стимулируют вестибулярный аппарат и способствуют расширению кровеносных сосудов головного мозга. Дыхательные упражнения, особенно через нос, влияют на кровенаполнение сосудов. Все это способствует усилению мозгового кровообращения, повышению его интенсивности и облегчению умственной деятельности.

5.4. Освещенность рабочего места

Рабочее пространство с недостаточным освещением или с излишним напряжением зрения может привести к деградации функций зрения, быстрой усталости глаз, снижению общей эффективности и производительности, а также увеличению количества ошибок.

При планировании рабочей зоны веб-разработчика ключевое внимание уделяется вопросам организации как искусственного, так и естественного освещения. Адекватное освещение не просто необходимо для выполнения задач, но и влияет на психологическое и физическое благополучие сотрудника.

Окна в помещениях, где находится компьютерное оборудование, идеально должны быть направлены на север и северо-восток. Рекомендуется располагать рабочие места так, чтобы мониторы смотрели боковой стороной к источникам света, и естественный свет в основном падал слева.

Световой поток на поверхности стола в рабочей зоне с документами должен составлять от 300 до 500 ЛК. Допускается использование локальных источников света для дополнительной освещенности документов, однако они не должны вызывать блики на экране или повышать уровень освещенности экрана свыше 300 ЛК.

Предпочтительными для искусственного освещения являются источники света, такие как люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ).

5.5. Расчет системы искусственного освещения

1. Тип лампы – люминесцентная.
2. Тип светильника – ПВЛ.
3. Освещенность – $E = 300$ лк.
4. Высота помещения – $H = 3,1$ м
5. Длина помещения – $a = 6$ м.
6. Ширина помещения – $b = 6$ м.
7. Рабочая высота – $h_p = 2,5$ м.

В зависимости от типа светильника необходимо выбирать коэффициент λ . Он определяет такое соотношение максимального расстояния между светильниками $L_{св(max)}$ и высотой их подвеса над рабочей поверхностью h_p , которое обеспечит равномерность освещения в помещении. Нужно определить максимальное расстояние между светильниками $L_{св(max)}$, м:

$$L_{св(max)} = \lambda \cdot h_p = 1,5 \cdot 2,5 = 3,75.$$

Нужно предельно расстояние $L_{l(max)}$ от стены до первого ряда светильников, м:

– при наличии рабочих мест у стены:

$$L_{l(max)} = (0,2 \dots 0,3) \cdot L_{св(max)} = 0,25 \cdot 3,75 = 0,9375.$$

Необходимо определить общее число рядов светильников (по ширине помещения):

$$n_{ш(min)} = \frac{b - 2 \cdot L_{l(max)}}{L_{св(max)}} + 1 = \frac{6 - 2 \cdot 0,9375}{3,75} + 1 = 2,1 \approx 2;$$

и число светильников в ряду (по длине помещения):

$$n_{д(min)} = \frac{a - 2 \cdot L_{l(max)}}{L_{св(max)}} + 1 = \frac{6 - 2 \cdot 0,9375}{3,75} + 1 = 2,1 \approx 2;$$

где a – длина, b – ширина помещения, для которого рассчитывается система освещения.

Полученные результаты необходимо округлить до ближайшего целого числа, после чего нужно определить общее расчётное минимальное количество светильников, которое надо разместить в помещении:

$$n_{общ(min)} = n_{ш(min)} \cdot n_{д(min)} = 2 \cdot 2 = 4;$$

$$S = a \cdot b = 6 \cdot 6 = 36.$$

По площади помещения S и высоте подвески светильника h_p нужно определить показатель помещения i :

$$i = \frac{S}{h_p \cdot (a + b)} = \frac{36}{2,5 \cdot (6 + 6)} = 1,2 \approx 1,25;$$

Необходимо найти значения коэффициентов отражения потолка ρ_n , стен ρ_c и полов $\rho_{пол}$ помещения, для которого рассчитывается осветительная установка.

$$\rho_n = 50\%;$$

$$\rho_c = 30\%;$$

$$\rho_{пол} = 10\%.$$

В зависимости от типа светильника и вида лампы нужно определить коэффициент использования светового потока η_u по показателю помещения i и коэффициентам отражения потолка ρ_n , стен ρ_c и полов $\rho_{пол}$.

$$\eta_u = 0,32.$$

Нужно определить коэффициент запаса k , учитывающий снижение уровня освещённости из-за неблагоприятных условий эксплуатации осветительной

установки: наличия дыма, копоти, пыли, повышенной концентрации химических веществ и т. д.; из-за старения и выхода из строя ламп.

$$k = 1,5.$$

Необходимо решить сколько источников света x будет в светильнике.

$$x = 3.$$

Далее назначить коэффициент z , характеризующий неравномерность освещённости (коэффициент отношения средней освещённости к максимальной):

$$z = 1,1 \text{ — для люминесцентных ламп.}$$

Необходимо рассчитать требуемый световой поток одной лампы:

$$\Phi_{\text{расч}} = \frac{E \cdot S \cdot k \cdot z}{\eta_{\text{и}} \cdot n_{\text{общ(min)}} \cdot x} = \frac{300 \cdot 36 \cdot 1,5 \cdot 1,1}{0,45 \cdot 4 \cdot 3} = 3300 \text{ лм.}$$

По рассчитанному световому потоку лампы $\Phi_{\text{расч}}$ нужно подобрать стандартную лампу со световым потоком $\Phi_{\text{табл}}$, значение которого близко к значению $\Phi_{\text{расч}}$ (желательно в пределах $-10\ldots+20\%$).

После выбора стандартных ламп нужно рассчитать число светильников, необходимых для обеспечения заданной освещённости E . Полученное число $n_{\text{расч}}$ округляют до ближайшего целого значения $n_{\text{пр}}$, при этом отклонение между принятым количеством светильников $n_{\text{пр}}$ и расчётным $n_{\text{расч}}$ допускается в пределах от -10 до $+20\%$.

$$\Phi_{1\text{табл}} = 3380;$$

$$n_{1\text{расч}} = \frac{E \cdot S \cdot k \cdot z}{\Phi_{\text{табл}} \cdot \eta_{\text{и}} \cdot x} = \frac{300 \cdot 36 \cdot 1,5 \cdot 1,1}{3380 \cdot 0,32 \cdot 3} = 5,49;$$

$$n_{1\text{пр}} = 6;$$

$$\Delta_1 = \frac{n_{\text{пр}} - n_{\text{расч}}}{n_{\text{расч}}} \cdot 100 = \frac{6 - 5,49}{5,49} \cdot 100 = 9,2\%.$$

Рассчитывают полную мощность проектируемой системы освещения, Вт:

$$N_1 = n_{\text{пр}} \cdot x \cdot N_{\text{лампы}} = 6 \cdot 3 \cdot 80 = 1440;$$

$$\Phi_{2\text{табл}} = 3390;$$

$$n_{2\text{расч}} = \frac{E \cdot S \cdot k \cdot z}{\Phi_{\text{табл}} \cdot \eta_{\text{и}} \cdot x} = \frac{300 \cdot 36 \cdot 1,5 \cdot 1,1}{3390 \cdot 0,32 \cdot 3} = 5,475;$$

$$n_{2\text{пр}} = 6;$$

$$\Delta_1 = \frac{n_{\text{пр}} - n_{\text{расч}}}{n_{\text{расч}}} \cdot 100 = \frac{5 - 5,475}{5,475} \cdot 100 = 9,59\%.$$

Рассчитывают полную мощность проектируемой системы освещения, Вт:

$$N_1 = n_{\text{пр}} \cdot \chi \cdot N_{\text{лампы}} = 5 \cdot 3 \cdot 65 = 1170.$$

$$n_p = 3; n_p \geq n_{\text{ш}(\min)}; n_{\text{св.р}} = 2; n_{\text{св.р}} \geq n_{\text{д}(\min)}; n_{\text{пр}} = 3 \cdot 2 = 6.$$

Таблица 5.2

Результаты расчётов параметров осветительной установки

№ п/п	Тип лампы	Световой поток лампы Φ , лм	Количество светильников		Отклонение $n_{\text{пр}}$ от $n_{\text{расч}}$, %	Мощность лампы, Вт	Полная мощность N, Вт
			расчётное $n_{\text{расч}}$	принятое $n_{\text{пр}}$			
1	ЛДЦ-80	3380	5,49	6	9,29	80	1440
2	ЛД-65	3390	5,475	6	9,59	65	1170

Вывод: оптимальным вариантом осветительной установки является установка с лампами ЛД-65, так как значение её полной потребляемой мощности меньше, чем у осветительной установки с лампами ЛДЦ-80.

Схема расположения светильников приведена на рис. 5.1.

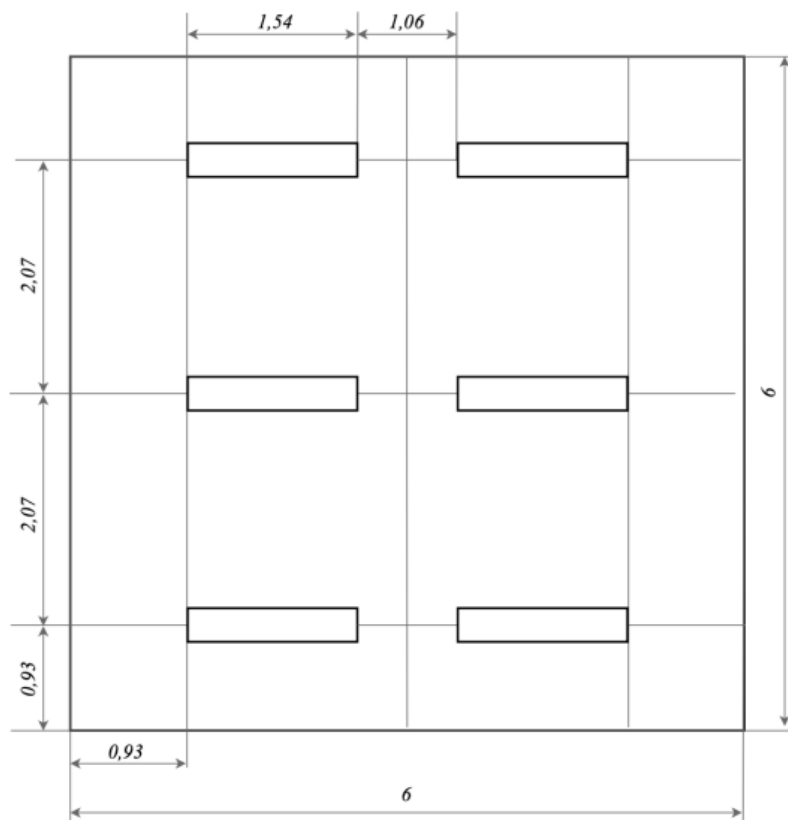


Рис. 5.1. Схема расположения светильников

5.6. Пожарная безопасность

Информация согласно СНиП 21-01-97 «Пожарная безопасность зданий и сооружений».

Потенциальные источники пожара могут включать: проводку, обнаженную из-за износа или повреждения; воспламенение изоляционных материалов силовых кабелей из-за искрения; небрежное обращение с открытым огнем; и неисправности в электрических устройствах.

Для поддержания безопасности от пожара, необходимо предпринимать профилактические меры, такие как: правильная эксплуатация оборудования; поддержание санитарной чистоты в помещениях; обеспечение доступности аварийного выхода; проведение инструктажей по пожарной безопасности; оборудование помещений огнетушителями; организация пожарных отрядов; соблюдение норм и правил пожарной безопасности при установке оборудования и освещения; рациональное размещение оборудования; запрет на курение в непредназначенных для этого местах; и регулярная проведение технических проверок и обслуживания.

В случае пожара, важно немедленно отключить все электрические системы, эвакуировать все персонал, за исключением тех, кто знает, как правильно использовать огнетушители, вызвать пожарных и встретить их для детального описания ситуации.

Для тушения возможного пожара в вычислительном центре, каждые 100 м² площади должны быть оборудованы одним углекислотным огнетушителем (типа ОУ-5 или ОУ-8) и порошковыми огнетушителями класса Е для тушения электрических устройств под напряжением до 1 кВт.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной работы был разработан веб-сервис для публикации и просмотра объявлений о недвижимости, который соответствует всем функциональным и техническим требованиям. В ходе исследования был проведен анализ требований, включающий обзор предметной области, ее моделирование и рассмотрение программ-аналогов. Была разработана архитектура программного комплекса, а также его модель данных, и проведено низкоуровневое проектирование.

Данный программный комплекс предназначен для облегчения процесса, поиска недвижимости и для размещения объявлений для поиска покупателей на объект. Он сочетает в себе достоинства рассмотренных программ-аналогов, позволяет эффективно искать и выбирать недвижимость в соответствии с требованиями пользователей, даже на иностранном языке.

В рамках работы были решены ряд задач, таких как создание интуитивно понятного интерфейса для упрощения процесса поиска и выбора недвижимости, использование принципов эргономики и дизайна для обеспечения удобства использования сервиса. Кроме того, была реализована функция мультиязычности, что делает сервис доступным для пользователей различных языковых групп.

В качестве перспектив дальнейшего развития данного программного продукта можно рассмотреть программную реализацию мобильного приложения, что позволит расширить его доступность и привлечь больше пользователей. Это значительно упростит процесс выбора подходящего объекта. Размещение платформы на стороннем хостинге способствует увеличению ее масштабируемости и доступности для пользователей.

В итоге, выполнение данной дипломной работы привело к созданию эффективного и удобного веб-приложения, способного удовлетворить потребности пользователей в поиске и размещении объявлений о недвижимости, а также предоставить им актуальные функциональные возможности.

СПИСОК ЛИТЕРАТУРЫ

1. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс, Д. Битти. – М.: Русская Редакция, БХВ-Петербург, 2018. – 736 с.
2. Гвоздева, Т.В. Проектирование информационных систем. Стандартизация: Учебное пособие / Т.В. Гвоздева, Б.А. Баллод. – СПб.: Лань, 2019. – 252 с.
3. Ларман К. Применение UML и шаблонов проектирования. – СПб.: Питер, 2018. – 736 с.
4. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. / Р. Мартин – СПб.: Питер, 2018. – 352 с.
5. Мартин Р. Чистый код: создание, анализ и рефакторинг. / Р. Мартин – СПб.: Питер, 2018. – 464 с.
6. Перлова, О.Н. Проектирование и разработка информационных систем: Учебник / О.Н. Перлова. - М.: Академия, 2018. – 272 с.
7. Веб-технологии для разработчиков. М.: MDN web docs – Режим доступа: <https://developer.mozilla.org/ru/docs/Web>.
8. React документация – Режим доступа: <https://ru.legacy.reactjs.org/>.
9. TypeScript документация – Режим доступа: <https://www.typescriptlang.org/docs/>.
10. Node.js документация – Режим доступа: <https://nodejs.org/en/docs>.
11. Express документация – Режим доступа: <https://expressjs.com/>.
12. PostgreSQL документация – Режим доступа: <https://www.postgresql.org/docs/>.
13. MDN Web API документация. Документация по веб-интерфейсам браузера – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/API>.
14. CSS-Tricks. Статьи, руководства и примеры по CSS и веб-разработке. – Режим доступа: <https://css-tricks.com/>.
15. React Router документация. Официальная документация React Router. – Режим доступа: <https://reactrouter.com/en/main>.
16. TypeScript Deep Dive. Подробное руководство по TypeScript – Режим доступа: <https://basarat.gitbook.io/typescript/>.

- 17.Multer документация. Официальная документация Multer (библиотека для загрузки файлов на сервер) – Режим доступа: <https://github.com/expressjs/multer>.
- 18.i18next. Библиотека для локализации веб-приложений через JSON файлы – Режим доступа: <https://www.i18next.com>.
- 19.React i18next – Интеграция i18next с React для локализации приложений. – Режим доступа: <https://react.i18next.com>.
- 20.Balsamiq – Режим доступа: <https://balsamiq.com/wireframes/desktop/>.
- 21.Postman – Режим доступа: <https://www.postman.com>.
- 22.PgAdmin – Режим доступа: <https://www.pgadmin.org>.
- 23.ЮКасса – Режим доступа: <https://yookassa.ru>.
- 24.Opera – Режим доступа: <https://www.opera.com/ru>.
- 25.MicrosoftEdge – Режим доступа: <https://www.microsoft.com/ru-ru/edge>.
- 26.Firefox – Режим доступа: <https://www.mozilla.org/ru/firefox/>.
- 27.GoogleChrome – Режим доступа: <https://www.google.com/intl/ru/chrome/>.
- 28.Яндекс браузер – Режим доступа: <https://browser.yandex.ru/>.
- 29.Nodemailer – Режим доступа: <https://nodemailer.com/about/>.
- 30.Dadata – Режим доступа: <https://dadata.ru/api/clean/>.

ПРИЛОЖЕНИЕ

Приложение А

Компонента карточки объявления

```
import React, { useState } from 'react';
..
interface ListingCardProps {
  listing: Listing;
  variant: string;
  onClick?: (id: number) => void;
  onChange?: (listing: Listing) => void;
  onRemove?: (id: number) => void;
  onEdit?: (listing: Listing) => void;
}
export const ListingCard: React.FC<ListingCardProps> = ({ listing,
variant, onClick, onChange, onRemove, onEdit }) => {
  const {t} = useTranslation();
  const title = t("listing_card.title", listing);
  const [imageIndex, setImageIndex] = useState(0);
  const [selectCategoryDialogVisible,
setSelectCategoryDialogVisible] = useState(false);

  const theme = useTheme();
  const colors: any = {
    draft: [theme.palette.primary.main,
theme.palette.primary.dark],
    checking: [theme.palette.warning.main,
theme.palette.warning.dark],
    approved: [theme.palette.success.main,
theme.palette.success.dark],
    rejected: [theme.palette.error.main,
theme.palette.error.dark],
    archived: ["#64748B", "#465364"]
  };
  const [c1, c2] = colors[listing.category] ||
[theme.palette.primary.main, theme.palette.primary.dark];
  const categoryStyle = {
    width: "156px",
    backgroundColor: c1,
    "&:hover": {
      backgroundColor: c2
    }
  };
};

return (
  <div
    className={styles.listingCard}
  >
    <TunedCarousel
      title={title}
      images={listing.images}
      onClick={() => onClick && onClick(listing.id)}
    >
```

```

        imageIndex={imageIndex}
        setImageIndex={setImageIndex}
      />
      <div className={styles.listingInfo} onClick={() =>
onClick && onClick(listing.id)}>
        <h3>{title}</h3>
        <p>{listing.description}</p>
        <div className={styles.priceContainer}>
          {(variant === "myListings" || variant ===
"moderation") && (
            <Button
              sx={categoryStyle}
              variant="contained"
              onClick={e => {
                e.stopPropagation();
setSelectCategoryDialogVisible(true);
              }}>
{t(`category_list.${listing.category}`)}
            </Button>
          )}
          {variant === "myListings" && (
            <Button
              variant="contained"
              onClick={e => {
                e.stopPropagation();
                onEdit && onEdit(listing);
              }}>
{t("change")}
            </Button>
          )}
          <p className={styles.price}>{t("price") + ": "
+ listing.price.toLocaleString()} ₪</p>
        </div>
      </div>
      {selectCategoryDialogVisible && (
        <SelectCategoryDialog
          variant={variant}
          listing={listing}
          onChange={listing => {
            setSelectCategoryDialogVisible(false);
            onChange && onChange(listing);
          }}
          onRemove={id => {
            setSelectCategoryDialogVisible(false);
            onRemove && onRemove(id);
          }}
          onCancel={() =>
setSelectCategoryDialogVisible(false)}
        />
      )}
    </div>
  );
};

```


Компонента searchBar

```

import React, { useState } from 'react';
..
interface SearchBarProps {
  filter: any;
  onFilterAppy: (filter: any) => void;
  FilterDialog?: React.FC<FilterDialogProps>;
  RightControls?: React.FC
}
export const SearchBar: React.FC<SearchBarProps> = ({ filter,
onFilterAppy, FilterDialog, RightControls }) => {
  const {t} = useTranslation();
  const [query, setQuery] = useState(filter.query || "");
  const [filterDialogVisible, setFilterDialogVisible] =
useState(false);

  const handleSearch = () => onFilterAppy(query === "" ?
pickBy(filter, (value, key) => key !== "query") : {...filter,
query});
  const extra = pickBy(filter, (value, key) => key !== "query");
  const extraIsEmpty = isEmpty(extra);
  return (
    <div className={styles.searchBar}>
      <div className={styles.container}>
        {FilterDialog !== undefined && (
          <Badge color="primary" variant="dot"
invisible={extraIsEmpty}>
            <IconButton
              size="small"
              color="primary"
              onClick={() =>
setFilterDialogVisible(true)}
              className={styles.filterIcon}
              disableRipple
            >
              <TuneIcon />
              <span>{t("search_bar.filters")}</span>
            </IconButton>
          </Badge>
        )}
        <TextField
          size="small"
          placeholder={t("search_bar.search_placeholder")}!}
          value={query}
          onChange={(e:
React.ChangeEvent<HTMLInputElement>) => setQuery(e.target.value)}
          onKeyDown={e => {
            if (e.key === "Enter") {
              handleSearch();
            }
          }}
        </TextField>
      </div>
    </div>
  );
}

```

```

        InputProps={{
            endAdornment: (
                <>
                    <InputAdornment position="end">
                        <IconButton
                            size="small"
                            color="primary"
                            onClick={() => {
                                setQuery("");
                            }}
                        />
                    </InputAdornment>
                </>
            )
        }}
    />
    {RightControls !== undefined && <RightControls />}
    {filterDialogVisible && FilterDialog !== undefined
    && (
        <FilterDialog
            filter={filter}
            onAccept={filter => {
                setFilterDialogVisible(false);
                onFilterAppy(query === "" ?
pickBy(filter, (value, key) => key !== "query") : {...filter,
query});
            }}
            onClose={() =>
setFilterDialogVisible(false)}
        />
    )}
</div>
</div>
);
};

```