



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практической работе №1

по дисциплине «Тестирование и верификация ПО»

Выполнили:

Студенты группы ИКБО-25-22

Петрушенко А.Д.

Гринь Я.Ю.

Ракитин В.А.

Костин М.Н.

Проверил:

Ассистент

Овчинникова М.А.

2024 г.

СОДЕРЖАНИЕ

1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ	3
2 ОШИБКИ, ВНЕСЕННЫЕ В ПО	4
3. ТЕХНИЧЕСКОЕ ЗАДАНИЕ ДРУГОЙ КОМАНДЫ	5
5. ОШИБКИ.....	11
6. ОПИСАНИЕ ОШИБОК В ПО.....	11
7 ЗАКЛЮЧЕНИЕ	16

1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Создание приложения на Java с использованием следующих элементов GUI:

- Текстовые поля и области ввода текста;
- Менеджеры компоновки компонентов;
- Слушатель мыши;
- Меню.

Text Fields - текстовое поле или поля для ввода текста (можно ввести только одну строку). Примерами текстовых полей являются поля для ввода логина и пароля, например, используемые, при входе в электронную почту.

Добавление меню в программе Java происходит несложно. Java определяет три компонента для обработки:

- JMenuBar: который представляет собой компонент, который содержит меню.
- JMenu: который представляет меню элементов для выбора.
- JMenuItem: представляет собой элемент, который можно кликнуть из меню.

2 ОШИБКИ, ВНЕСЕННЫЕ В ПО

Мы отразили ошибки, которые внесли в комментариях к определенным строчкам кода.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Main extends JFrame{
    int milan;
    int madrid;
    JButton but1 = new JButton("AC Milan");
    JButton but2 = new JButton(""); //Real Madrid нет надписи
    JLabel lbl1 = new JLabel("Result: 0 X 0");
    JLabel lbl2 = new JLabel("Last Scorer: N/A");
    JLabel lbl3 = new JLabel("Winner: DRAW");
    public Main(){
        super("Result");
        setLayout((LayoutManager) null);
        but1.setBounds(12, 12, 150, 100); //(12, 12, 100, 100) кнопка перекрывает текстовый элемент
        but2.setBounds(250,120,100,100); //(250,12,100,100); кнопка не на месте
        lbl1.setBounds(112, 12,100,100);
        lbl2.setBounds(112,30,150,100);
        lbl3.setBounds(112,50,150,100);
        //add(but1); кнопка не добавлена
        add(but2);
        add(lbl1);
        add(lbl2);
        add(lbl3);
        but1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
                try {
                    Main.this.milan+=2; //+=1 не верный счет
                    Main.this.lbl1.setText("Result: "+Main.this.milan+" X "+Main.this.madrid);
                    Main.this.lbl2.setText("Last Scorer: AC Milan");
                    if(Main.this.milan > Main.this.madrid){
                        Main.this.lbl3.setText("Winner: AC Milan");
                    }
                    if (Main.this.milan == Main.this.madrid){
                        Main.this.lbl3.setText("Winner: DRAW");
                    }
                }
                catch (Exception e){
                }
            }
        });
        but2.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
                try {
                    ++Main.this.madrid;
                    Main.this.lbl1.setText("Result: "+Main.this.milan+" X "+Main.this.madrid);
                    Main.this.lbl2.setText("Last Scorer: Real Madrid");
                    if(Main.this.milan < Main.this.madrid){
                        Main.this.lbl3.setText("Winner: Real Madrid");
                    }
                    if (Main.this.milan == Main.this.madrid){
                        Main.this.lbl3.setText("Winner: DRAW");
                    }
                }
                catch (Exception e){}
            }
        });
    }
}
```

```

    }
    });
    setSize(50,200); //(400,200) маленький размер окна
}
public static void main(String[] args){
    Main w = new Main();
    w.setVisible(true);
    w.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}

```

3. ТЕХНИЧЕСКОЕ ЗАДАНИЕ ДРУГОЙ КОМАНДЫ

1. Общие сведения

Сайт <https://cozy-clafoutis-50d295.netlify.app/addListing> представляет собой одностраничное веб-приложение (SPA) с современным и минималистичным интерфейсом. Сайт создан для предоставления информации о продуктах и услугах, таких как сдача жилья и аренда автомобилей, а также для демонстрации пользовательского интерфейса, выполненного в современном стиле с использованием технологий HTML, CSS и JavaScript.

Основные элементы интерфейса сайта включают:

- Главное меню навигации позволяет пользователю быстро перемещаться между основными разделами сайта;
- Блок с информацией о продуктах или услугах содержит описания и изображения предлагаемых товаров или услуг, а также основные характеристики и цены;
- Форма обратной связи или заказа позволяет пользователю оставить свои контактные данные для обратной связи или оформить заказ;
- Футер сайта содержит контактные данные, ссылки на социальные сети и дополнительную информацию о компании.

Сайт создан с целью предоставления пользователям информации о товарах или услугах компании и облегчения процесса взаимодействия с потенциальными клиентами через простую и интуитивно понятную навигацию. Разработка автоматизированной системы для данного сайта

предполагает улучшение пользовательского опыта, автоматизацию обработки данных и повышение общей производительности сайта.

2. Цели и назначение создания автоматизированной системы

Целями создания автоматизированной системы являются:

- Предоставление информации;
- Взаимодействие с клиентами;
- Продвижение и реклама;
- Автоматизация процессов;
- Улучшение пользовательского опыта;
- Сдача жилья в аренду пользователям через удобный интерфейс;
- Предоставление возможности аренды автомобилей через онлайн-сервис;
- Обеспечение прозрачности и удобства для арендаторов и арендодателей;
- Интеграция с платежными системами для удобной оплаты аренды жилья и автомобилей;
- Поддержка бронирования и управления арендой в реальном времени.

Назначение сайта заключается в предоставлении удобного и понятного инструмента для пользователей, позволяющего легко находить нужную информацию, совершать покупки или заказы, а также общаться с представителями компании.

3. Характеристика объектов автоматизации

Объектами автоматизации являются процессы, связанные с арендой жилья и автомобилей. Система автоматизирует взаимодействие между арендаторами и арендодателями, обеспечивая удобный интерфейс для поиска, бронирования и оплаты услуг.

Основные объекты автоматизации включают:

- Каталог предложений — система автоматизирует добавление, редактирование и удаление объявлений по аренде жилья и автомобилей. Включает управление описаниями, фотографиями, доступностью и ценами.
- Процесс бронирования и аренды — автоматизация процедуры поиска и бронирования жилья и автомобилей пользователями. Система предоставляет пользователям возможность выбора дат, типов арендуемого объекта и других параметров.
- Платежные операции — автоматизация приема и обработки платежей через интеграцию с популярными платежными системами, обеспечивая безопасные транзакции и управление финансами.
- Управление пользователями — система автоматизирует процесс регистрации и авторизации пользователей, предоставляя арендаторам и арендодателям личные кабинеты для управления своими бронированиями, предложениями и отзывами.
- Система уведомлений — автоматизация отправки уведомлений и напоминаний пользователям о предстоящих бронированиях, изменениях в их статусе и специальных предложениях.
- Аналитика и отчеты — система собирает данные о бронированиях, транзакциях и поведении пользователей, предоставляя администраторам и арендодателям отчеты для анализа и принятия решений.

4. Требования к автоматизированной системе

1. Функциональные требования:

- Возможность создания и редактирования объявлений аренды жилья и автомобилей;
- Интерактивный поиск и фильтрация предложений по различным параметрам (цена, местоположение, тип жилья, марка автомобиля и т.д.);
- Поддержка бронирования с выбором дат, сроков аренды и дополнительных услуг;
- Интеграция с платежными системами для безопасной онлайн-оплаты аренды;

- Личные кабинеты для арендаторов и арендодателей с возможностью управления бронированиями и объявлениями;
- Система отзывов и рейтингов для повышения доверия пользователей;
- Автоматическая отправка уведомлений и напоминаний пользователям о бронированиях и новых предложениях.

2. Нефункциональные требования:

- Высокая производительность и быстрое время отклика при работе с большим объемом данных;
- Адаптивный дизайн для корректного отображения на всех типах устройств (ПК, планшеты, смартфоны);
- Обеспечение безопасности данных пользователей, включая шифрование личной информации и защиту от атак;
- Масштабируемость для поддержки увеличения числа пользователей и объявлений;
- Легкость в сопровождении и поддержке системы.

5. Состав и содержание работ по созданию автоматизированной системы

1. Сбор и анализ требований заказчика и пользователей, исследование аналогичных сервисов и определение ключевых функций;
2. Разработка архитектуры системы (фронтенд, бэкенд, база данных), создание прототипов пользовательского интерфейса и схем базы данных;
3. Разработка клиентской части (фронтенд) с использованием современных фреймворков (например, React или Vue.js) и с реализацией API для взаимодействия с клиентом и базой данных, интеграция с платежными системами и настройка безопасности;
4. Проведение модульного, интеграционного и функционального тестирования;
5. Установка системы на сервере, настройка окружения и домена, конфигурация SSL-сертификатов и обеспечение защиты данных;

6. Обучение пользователей и администраторов, техническая поддержка и обновление системы.

6. Порядок разработки автоматизированной системы

1. Определение требований, согласование с заказчиком, формирование проектной команды;
2. Разработка технической документации, прототипов интерфейсов и архитектуры системы;
3. Поэтапная реализация модулей системы и их интеграция в единое целое;
4. Поэтапное тестирование системы, выявление и исправление ошибок;
5. Установка системы на сервере, настройка и подготовка к эксплуатации;
6. Передача системы заказчику, обучение, поддержка и сопровождение.

7. Порядок контроля и приемки автоматизированной системы

1. Проверка соответствия реализованных функций требованиям ТЗ;
2. Оценка производительности системы под высокой нагрузкой;
3. Проверка системы на уязвимости и их устранение;
4. Проверка системы заказчиком и конечными пользователями;
5. Устранение всех выявленных недостатков и ошибок;
6. Подписание акта приемки системы в эксплуатацию при условии выполнения всех требований.

8. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие

Подготовка объекта автоматизации к вводу системы в действие включает несколько ключевых этапов. Во-первых, необходимо развернуть серверное окружение, которое включает установку и настройку серверного ПО и баз данных, а также настройку доменного имени и SSL-сертификатов для обеспечения безопасного соединения. Следующим шагом является подготовка данных — импорт начальных данных в систему и проверка их корректности. Также важно провести обучение персонала, которое включает обучение администраторов и пользователей работе с системой, ее

функционалом, процессами бронирования и управления объявлениями. Кроме того, следует настроить резервное копирование данных и разработать план восстановления системы в случае сбоев.

9. Требования к документированию

Документирование должно включать несколько ключевых разделов. Во-первых, пользовательская документация должна содержать руководство пользователя, описывающее функционал сайта, инструкции по бронированию, аренде и оплате, а также рекомендации по использованию всех доступных функций. Техническая документация должна включать описание архитектуры системы, API и программного кода, чтобы обеспечить понимание и поддержку системы другими разработчиками. Необходимо также подготовить инструкции по развертыванию системы, которые будут содержать все шаги по установке и настройке на сервере. Руководства по безопасности должны описывать меры защиты данных пользователей, а также план реагирования на возможные инциденты и атаки. Наконец, отчеты по тестированию должны содержать результаты функционального, нагрузочного и безопасности тестирования, что поможет выявить и устранить возможные ошибки и уязвимости до ввода системы в эксплуатацию.

10. Источники разработки

- Техническое задание (ТЗ);
- Методические рекомендации и стандарты разработки ПО;
- Проектная документация;
- Научные статьи и исследования;
- Анализ пользовательских требований и исследований рынка;
- Примеры аналогичных проектов и открытые репозитории;
- Обратная связь и результаты тестирования.

Ссылка на сайт: <https://66db0711c84f8d82e39adbade--rad-dieffenbachia-67aef6.netlify.app/>

5. ОШИБКИ

1. Не работает чекбокс с типом квартир по комнатам (проверяется нажатием)
2. левая граница цены не вводится (текст не доступен для ввода при нажатии)
3. правая граница цены не вводится (текст не доступен для ввода при нажатии)
4. объявления не загружаются (замечается при заходе на страницу по кнопке показать объявления)
5. картинки на главной странице не загружаются (видно при заходе на стартовую страницу)

6. ОПИСАНИЕ ОШИБОК В ПО

1. Не прокликается кнопка при выборе количества комнат в квартире. Аналогично при выборе размера коттеджа и типа авто (рисунки 1-3).

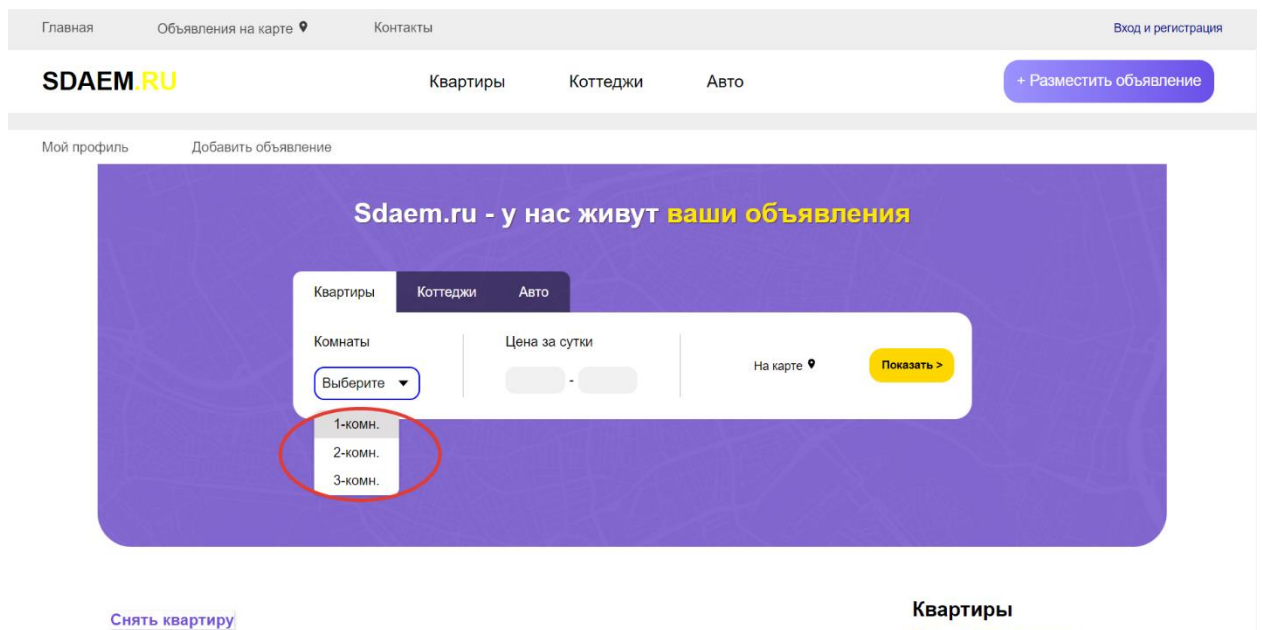
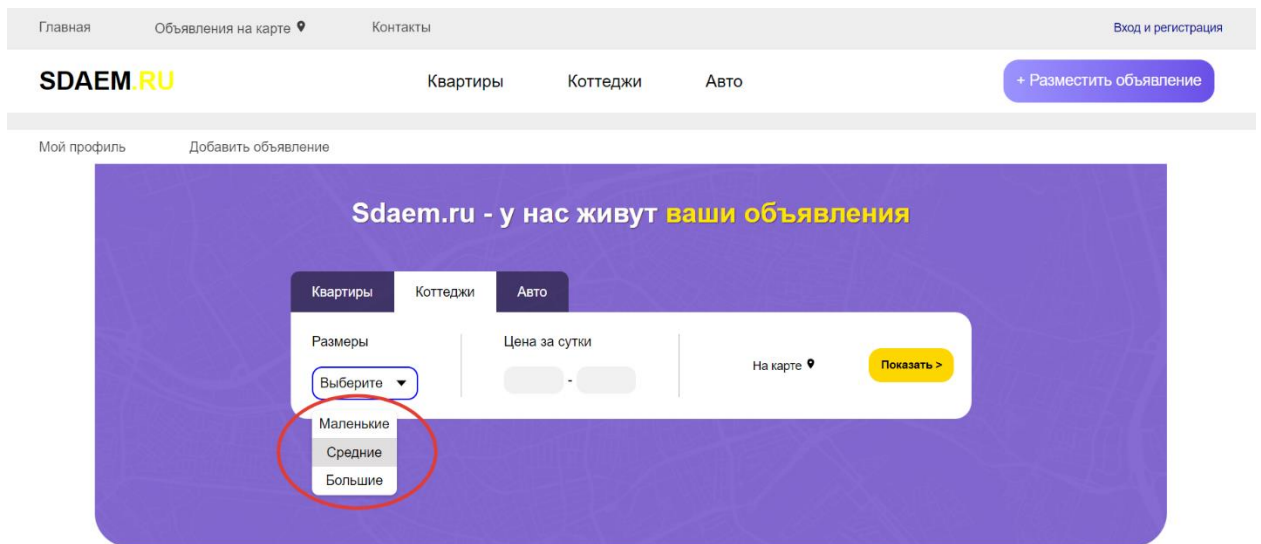


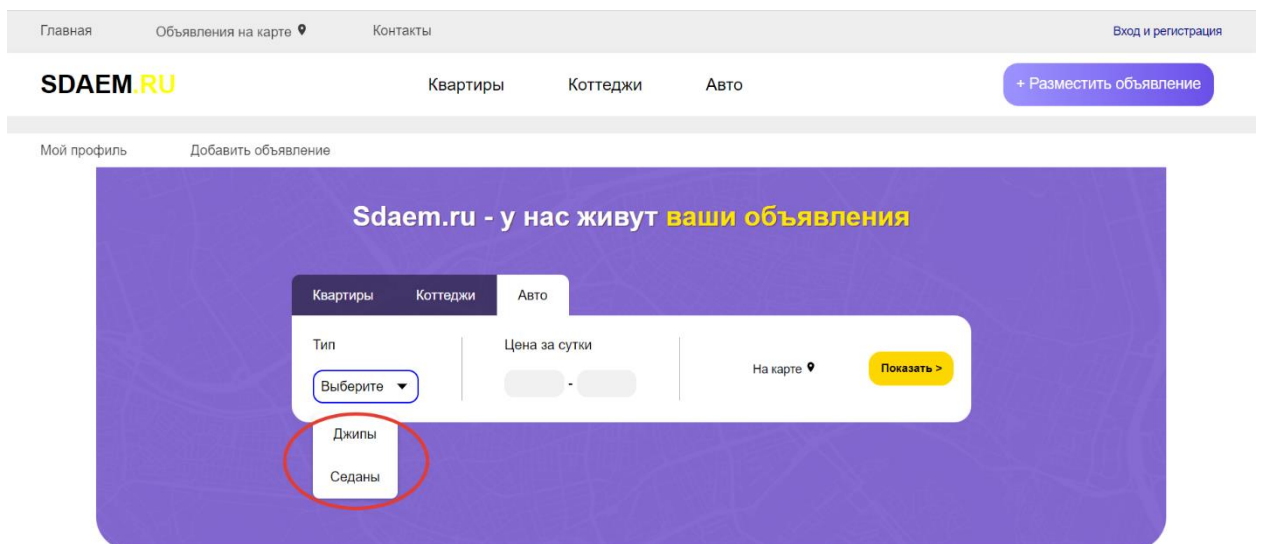
Рисунок 1 – Выбор количества комнат в квартире



[Снять квартиру](#)

[Квартыры](#)

Рисунок 2 – Выбор размера коттеджа

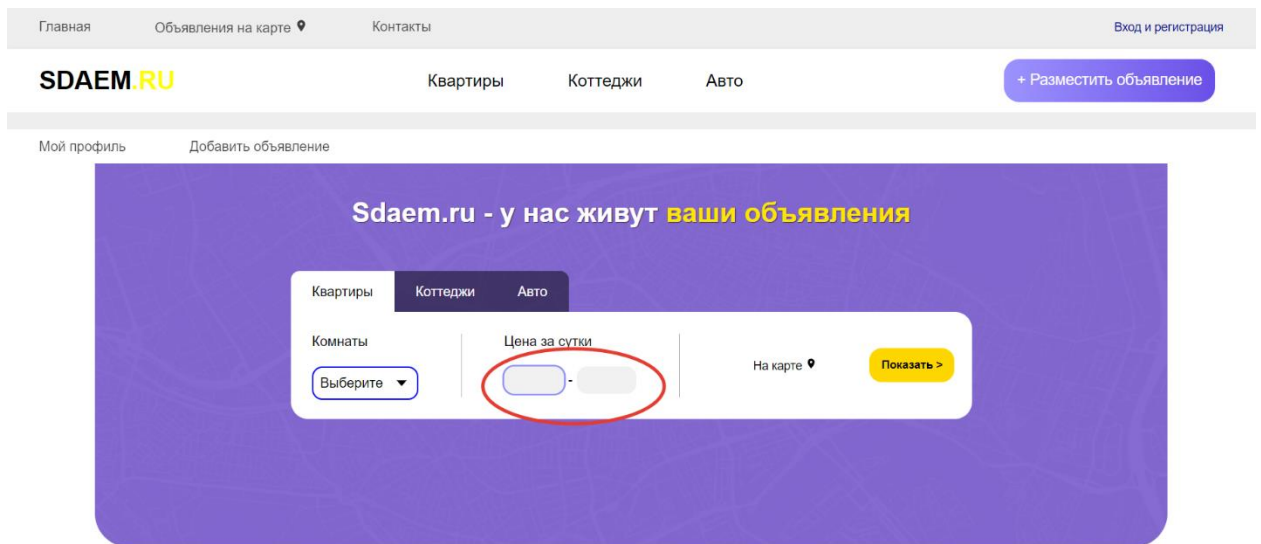


[Снять квартиру](#)

[Квартыры](#)

Рисунок 3 – Выбор типа авто

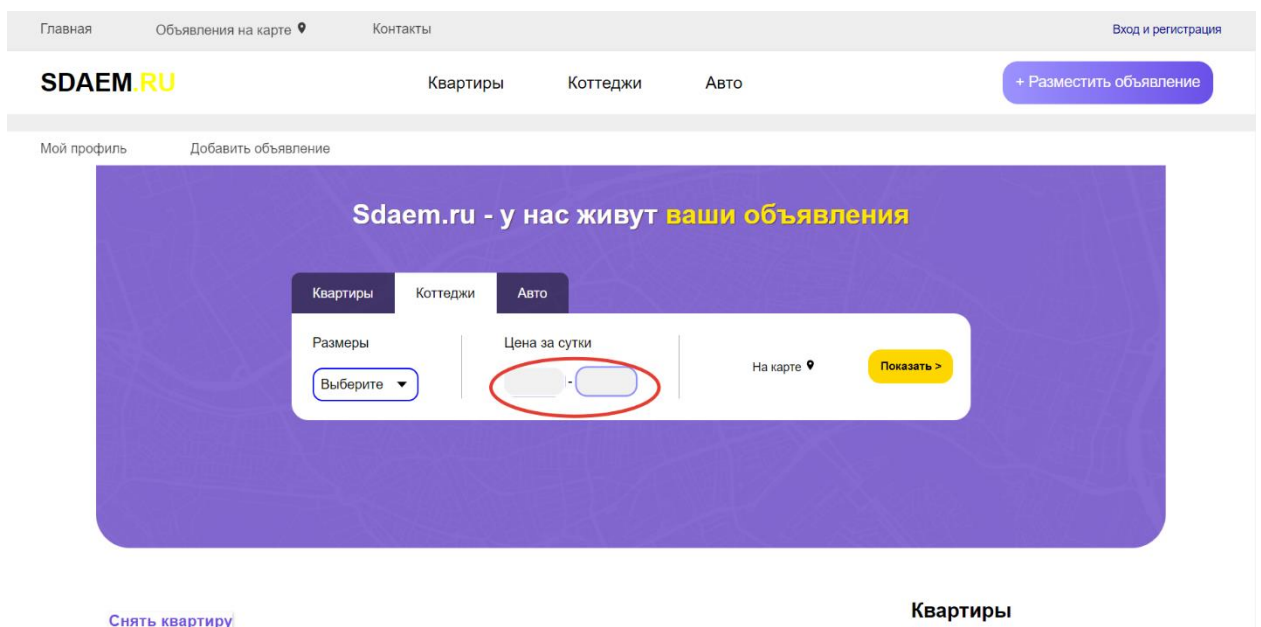
2. Нет возможности ввода цены квартиры за сутки. Аналогично с коттеджами и авто (рисунки 4-6).



[Снять квартиру](#)

[Квартыры](#)

Рисунок 4 – Ввод цены квартиры за сутки



[Снять квартиру](#)

[Квартыры](#)

Рисунок 5 – Ввод цены коттеджа за сутки

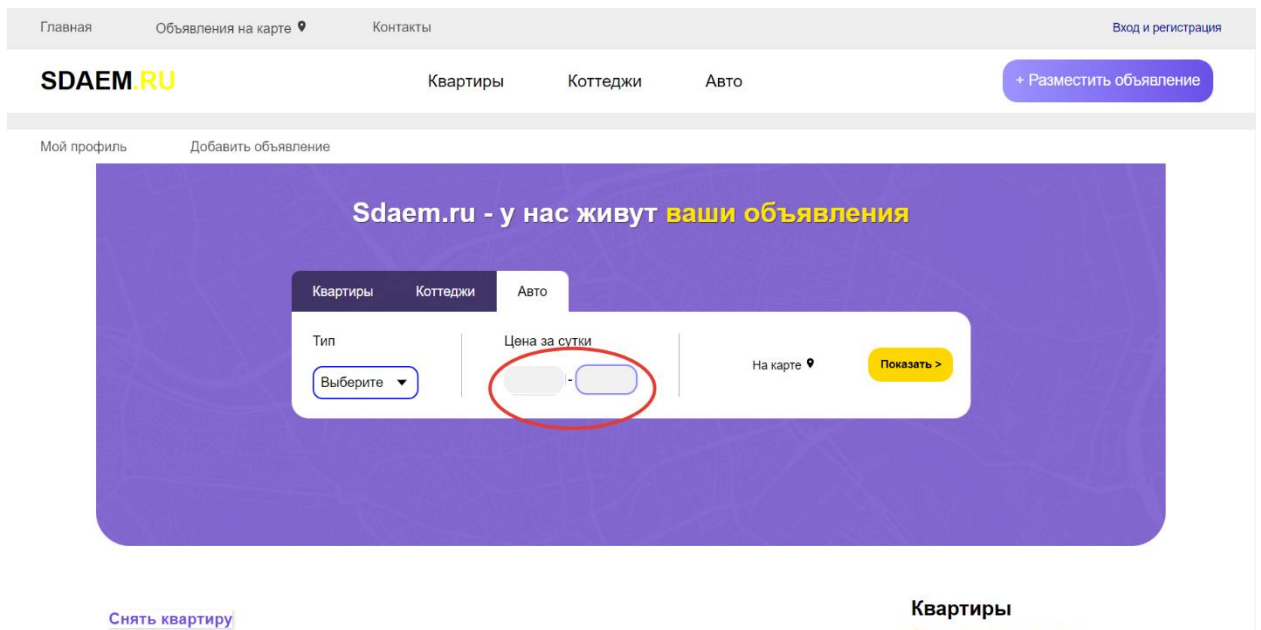


Рисунок 6 – Ввод цены авто за сутки

3. Не рабочие кнопки (рисунок 7).

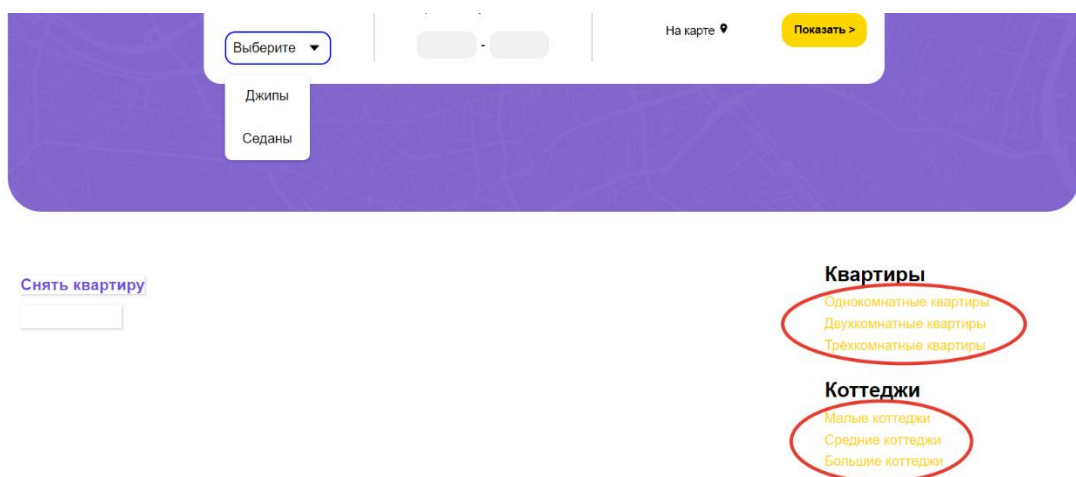


Рисунок 7

4. Не работают кнопки «Главная», «Объявления на карте», «Контакты», «Вход и регистрация» (рисунок 8).

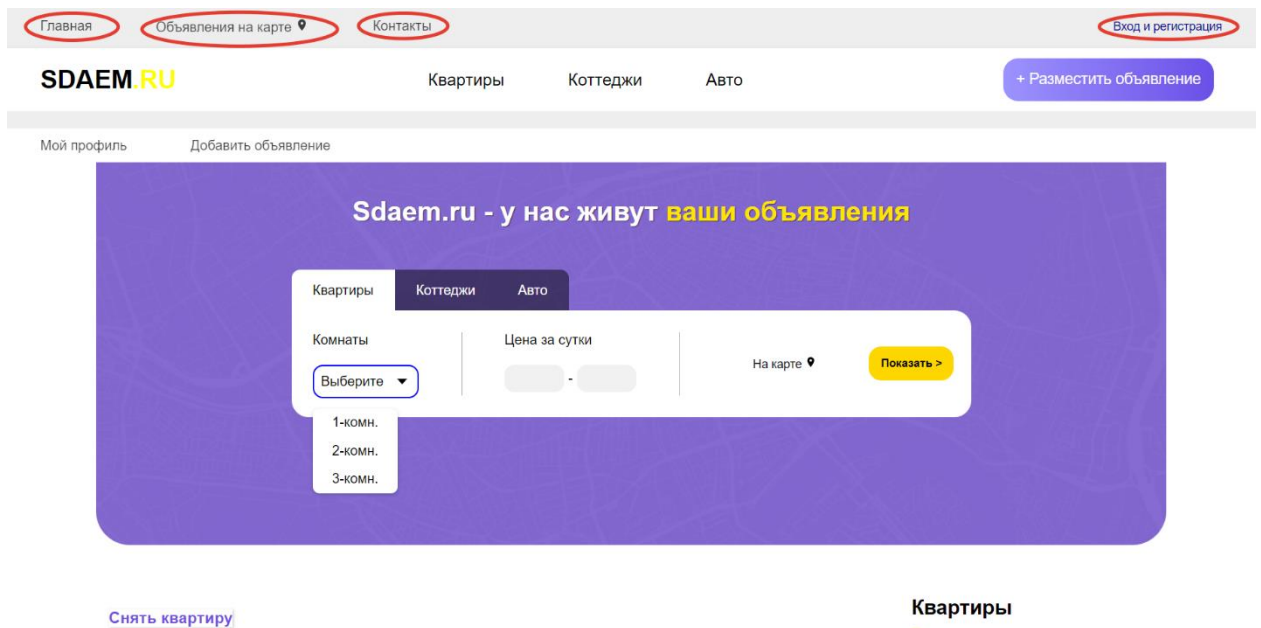


Рисунок 8

5. Не пропадает окно с выдаваемым адресом после её нажатия (рисунок 9).

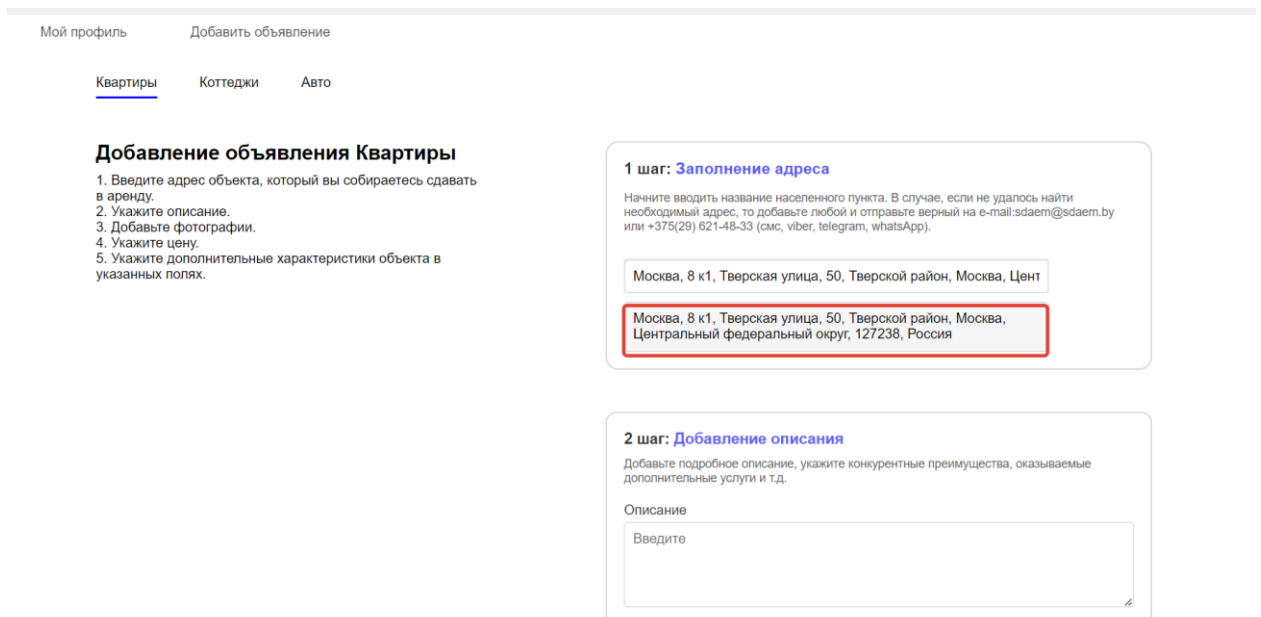


Рисунок 9 – Не пропадает окно с адресом

6. При перезагрузке сайта вылезает ошибка (рисунок 10).

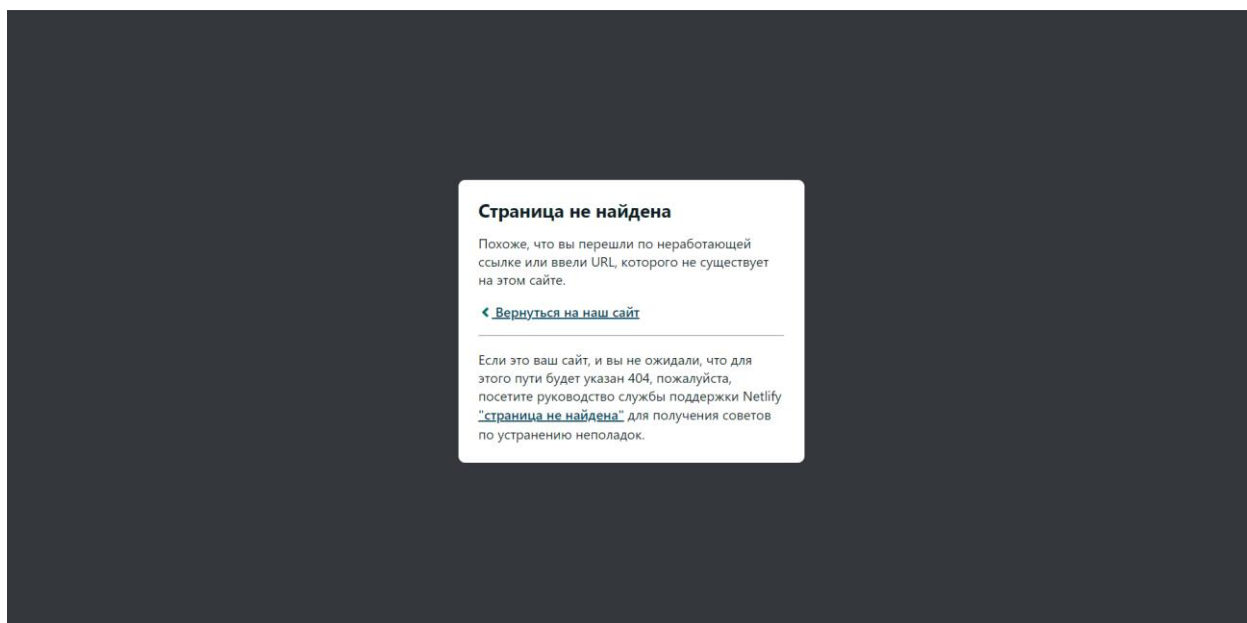


Рисунок 10

7. Бесконечная загрузка информации о добавленном мною объявлении (рисунок 11).

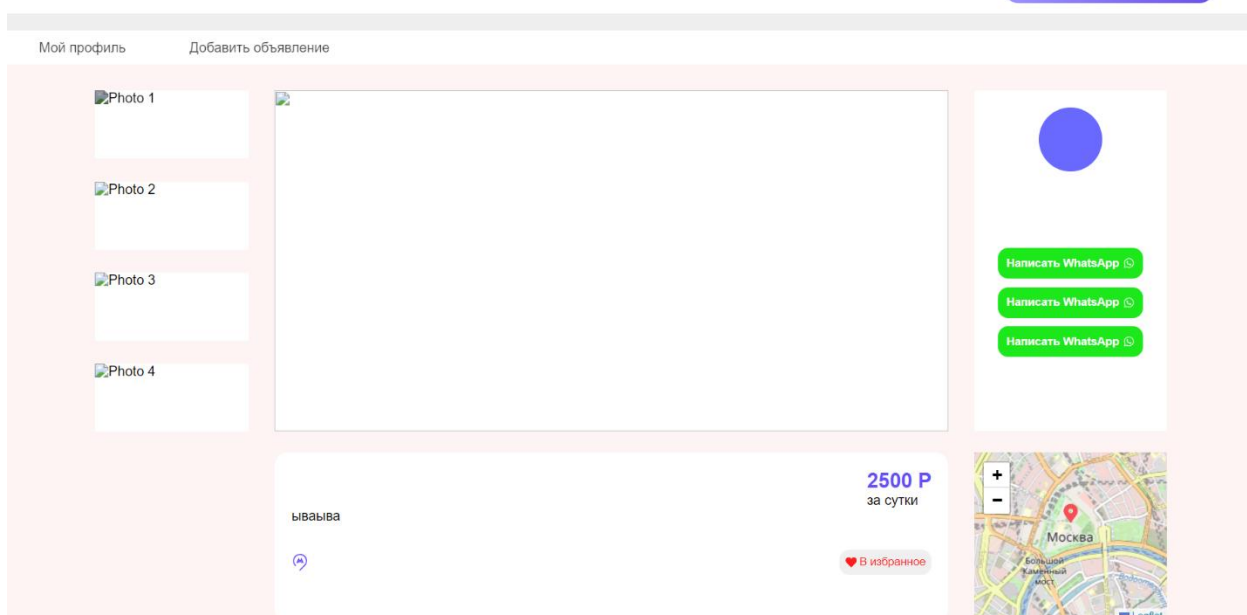


Рисунок 11

7 ЗАКЛЮЧЕНИЕ

Во время выполнения практической работы была осуществлена проверка соответствия программной документации требованиям ГОСТ, а также проведено тестирование программного продукта на наличие ошибок.