Министерство науки и высшего образования РФ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Федеральное государственное автономное образовательное учреждение высшего образования **«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»** | | | | | | | | |
| Институт космических и информационных технологий | | | | | | | | |
| Кафедра вычислительной техники | | | | | | | | |
|  | | | | | | | УТВЕРЖДАЮ Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_О.В. Непомнящий "\_\_\_"\_\_\_\_\_\_\_\_\_\_2023 г. | |
| Бакалаврская работа | | | | | | | | |
| Сайт «Поиск пропавших животных» | | | | | | | | |
| 09.03.01 Информатика и вычислительная техника | | | | | | | | |
|  | | | | | | | | |
| Руководитель |  |  |  |  |  | старший преподаватель | | А.П. Яблонский |
|  |  | подпись |  | дата |  | должность, ученая степень | |  |
| Выпускник |  |  |  |  |  |  | | Д.А. Гуськов |
|  |  | подпись |  | дата |  |  | |  |
| Нормоконтролёр |  |  |  |  |  |  | | А.П. Яблонский |
|  |  | подпись |  | дата |  |  | |  |
| Красноярск 2023 | | | | | | | | |

Содержание

[Содержание 2](#_Toc138593527)

[Реферат 4](#_Toc138593528)

[Введение 5](#_Toc138593529)

[1. Анализ предметной области 6](#_Toc138593530)

[1.1 Анализ существующих аналогов 6](#_Toc138593531)

[1.2 Анализ целевой аудитории 11](#_Toc138593532)

[1.3 Функциональные требования 12](#_Toc138593533)

[1.4 Описание функций 15](#_Toc138593534)

[1.5 Средства разработки 19](#_Toc138593535)

[2 Устройство сайта 22](#_Toc138593536)

[2.1 Разработка интерфейса 22](#_Toc138593537)

[2.2 Разработка базы данных 25](#_Toc138593538)

[3 Реализация функций 31](#_Toc138593539)

[3.1 Авторизация 31](#_Toc138593540)

[3.2 Регистрация 33](#_Toc138593541)

[3.3 Создание объявления 34](#_Toc138593542)

[3.4 Просмотр объявлений 36](#_Toc138593543)

[3.5 Добавление комментария к объявлению 38](#_Toc138593544)

[3.6 Удаление объявления 39](#_Toc138593545)

[3.7 Модерация 40](#_Toc138593546)

[3.8 Администрирование 42](#_Toc138593547)

[3.9 Модальные окна сайта 44](#_Toc138593548)

[3.10 Помощь 45](#_Toc138593549)

[3.11 Тестирование 46](#_Toc138593550)

[Заключение 51](#_Toc138593551)

[Список использованных источников 52](#_Toc138593552)

[Приложения A – E 54](#_Toc138593553)

Реферат

Выпускная квалификационная работа по теме «Сайт по поиску пропавших животных» содержит 58 страниц текстового документа, 47 рисунков и 12 использованных источников.

САЙТ ПО ПОИСКУ ПРОПАВШИХ ЖИВОТНЫХ.

Объект исследования – сайт по поиску пропавших животных.

Цель работы: разработать сайт по поиску пропавших животных.

В первой главе проведён анализ существующих аналогов, анализ целевой аудитории, функциональные требования и выявление ролей.

Во второй главе приведена разработка интерфейса и базы данных.

В третьей главе приведена реализация функций проекта и их тестирование.

В результате работы над дипломной работой был разработан сайт по поиску пропавших животных.

Введение

В данной выпускной квалификационной работе разработан «Сайт по поиску пропавших животных». Этот сайт предоставляет информацию о питомцах, которые были найдены, потеряны или нуждаются в новом доме. Таким образом, он помогает придать огласку о пропаже животного, ускоряя процесс поиска. Реализованы функции поиска по типу публикации и виду животного, создания и просмотра объявлений и возможности оставлять комментарии к ним, а также просмотра ветеринарных клиник на карте.

Целью выпускной квалификационной работы является создание интуитивно понятного и доступного веб-ресурса, который позволит владельцам пропавших животных быстро сообщать о случаях их исчезновения, а также связаться с другими пользователями, волонтерами и специализированными организациями, чтобы максимально увеличить шансы на успешный поиск и спасение потерянных питомцев.

Был проведен анализ существующих аналогов и изучены потребности пользователей, что позволило создать наиболее подходящую концепцию сайта и предоставить удобный интерфейс взаимодействия, который будет способствовать объединению усилий сообщества в решении проблемы с пропажами животных.

Для достижения поставленной цели были определены следующие задачи:

1. Разработать макет сайта;
2. Разработать возможность оставлять комментарии;
3. Добавить карту с ветеринарными клиниками на сайт;
4. Разработать возможность поиска объявлений по критериям;
5. Разработать функционал для администрирования сайта.

# 1. Анализ предметной области

Проводя анализ конкурентов, было отмечено, что специализированного под город Красноярск сайта с подобной проблематикой не существует. Таким образом основными аналогами становятся сообщества в социальных сетях и всероссийские сайты. Основным минусом сообществ являются отсутствие фильтра-поиска, что приводит к затрудненному поиску нужного объявления и отсутствие какой-либо структуры и макета объявления, что так же затрудняет навигацию.

## Анализ существующих аналогов

Из проектов с аналогичным функционалом были выделены:

* «Poiskzverei.ru»;
* «Pet911.ru»;
* «Foundpets.ru».

Данные веб-ресурсы также предлагают функционал поиска пропавших животных и ниже будут рассмотрены их достоинства и недостатки.

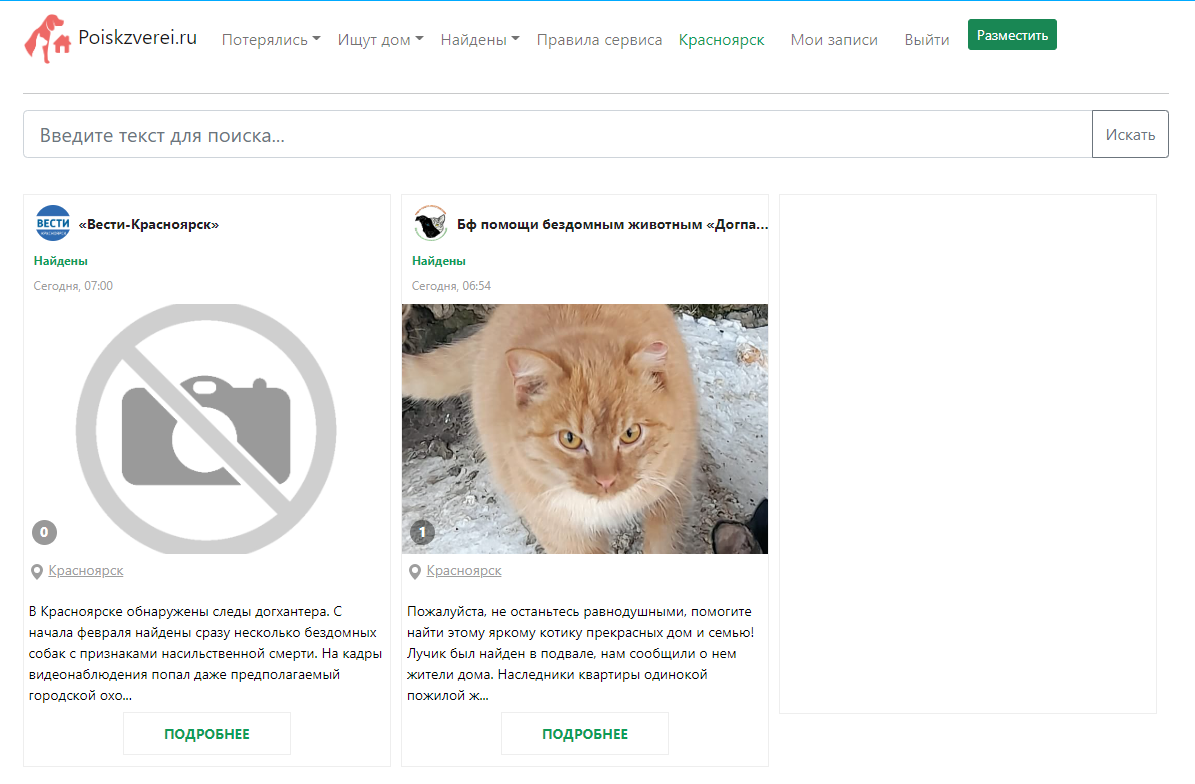
1. Сайт Poiskzverei.ru.

Poiskzverei.ru [1] работает уже более 5 лет, за это время было спасено огромное количество пропавших животных.

Плюсом данной системы является простота в эксплуатации, ведь авторизация здесь происходит через сайт «vk.com» и написать объявление не доставляет большого труда. Также стоит отметить, что данный сайт предназначен для использования по всей России.

Минусом данной системы является отсутствие фильтрации поиска объявлений по типу объявления и виду питомца и связи с поддержкой.

Главное меню сайта можно увидеть на рисунке 1.



1. – Главное меню сайта
2. Сайт Pet911.ru.

Веб-сайт Pet911.ru [2] работает с 2017 года, и по данным самого сайта была оказана помощь более шестидесяти четырем тысячам питомцам.

Плюсы данного сайта:

* хороший и удобный дизайн сайта;
* связь с ветеринарными клиниками и автоматическое распространение информации в соцсетях;
* ответы на самые задаваемые вопросы;
* размещение объявления без регистрации.

Из минусов можно отметить маленький размер кнопок и шрифта.

Основные разделы сайта можно увидеть на рисунках 2 и 3.

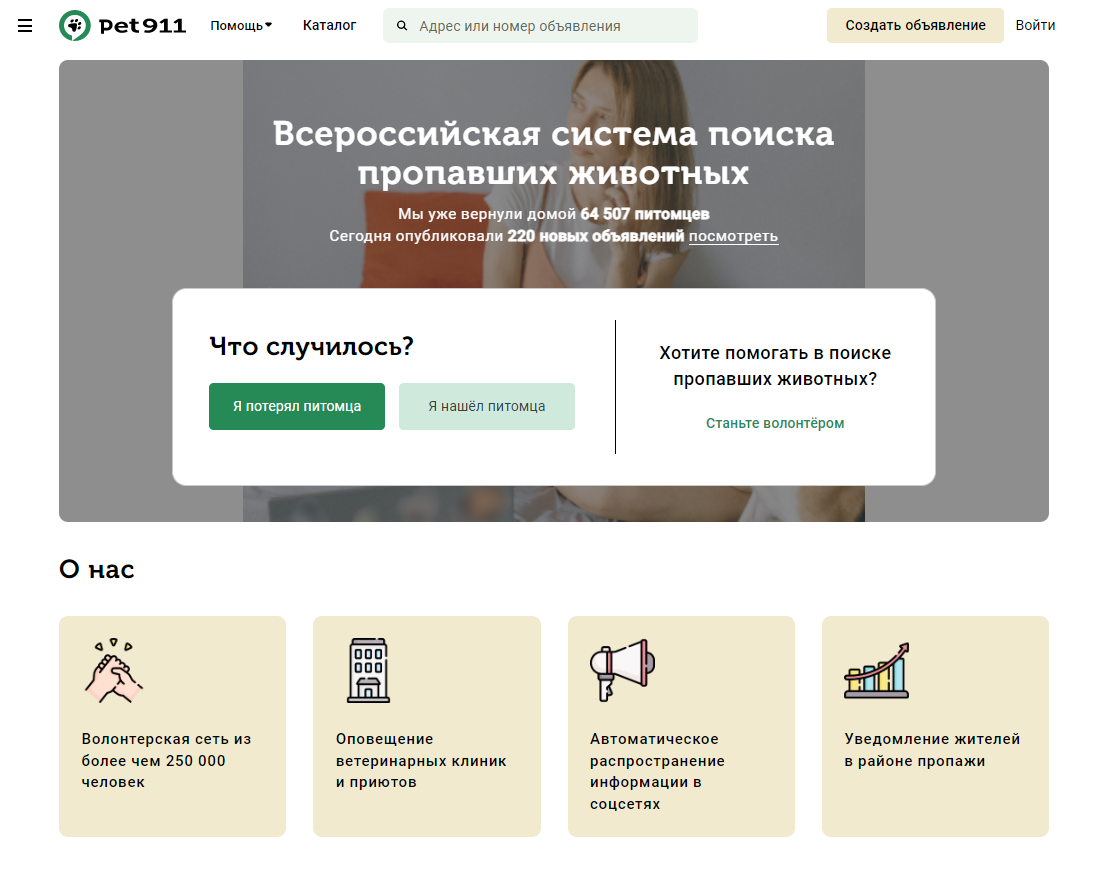


Рисунок 2 – Главная страница сайта



Рисунок 3 – Раздел подачи объявления

1. Сайт Foundpets.ru

Foundpets.ru [3] работает с 2017 года, на главной странице располагается информация об объявлениях, поданных за сегодня и за 7 дней, недавние публикации и кнопка подачи собственного объявления.

Плюсы данного сайта:

* уместный и практичный дизайн сайта;
* фильтры поиска по городу и радиусу;
* наличие отдельной вкладки «завершенные поиски», что является большим плюсом, ведь эта дает больше информации человеку, который на протяжении долгого времени не может отыскать своего питомца;
* наличие вкладки «идеи», где можно оставлять свои пожелания и мнение по данному ресурсу.

Из минусов можно отметить отсутствие возможности удаления своего объявления без отправки на сайт комментария, так как сайт предполагает, что это может привести к неким ошибкам. Также было замечено, что даже при написании комментария появляется окно с ошибкой, которое гласит: «Не удалось закрыть объявление. Необходимо оставить комментарий к объявлению перед завершением поиска».

Основные разделы сайта можно увидеть на рисунках 4 и 5.

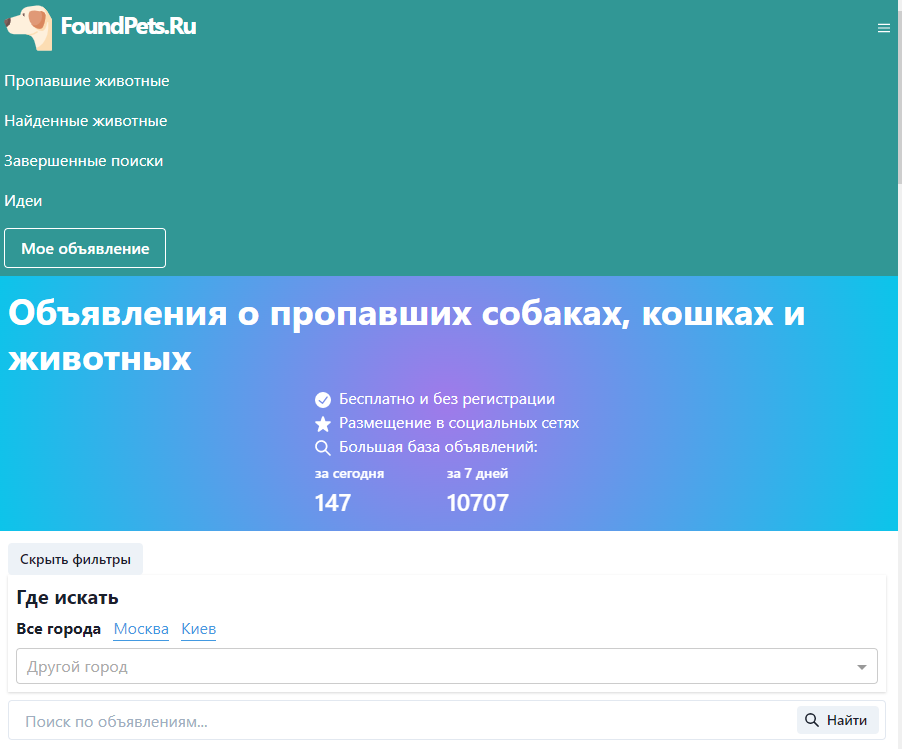


Рисунок 4 - Главная страница сайта

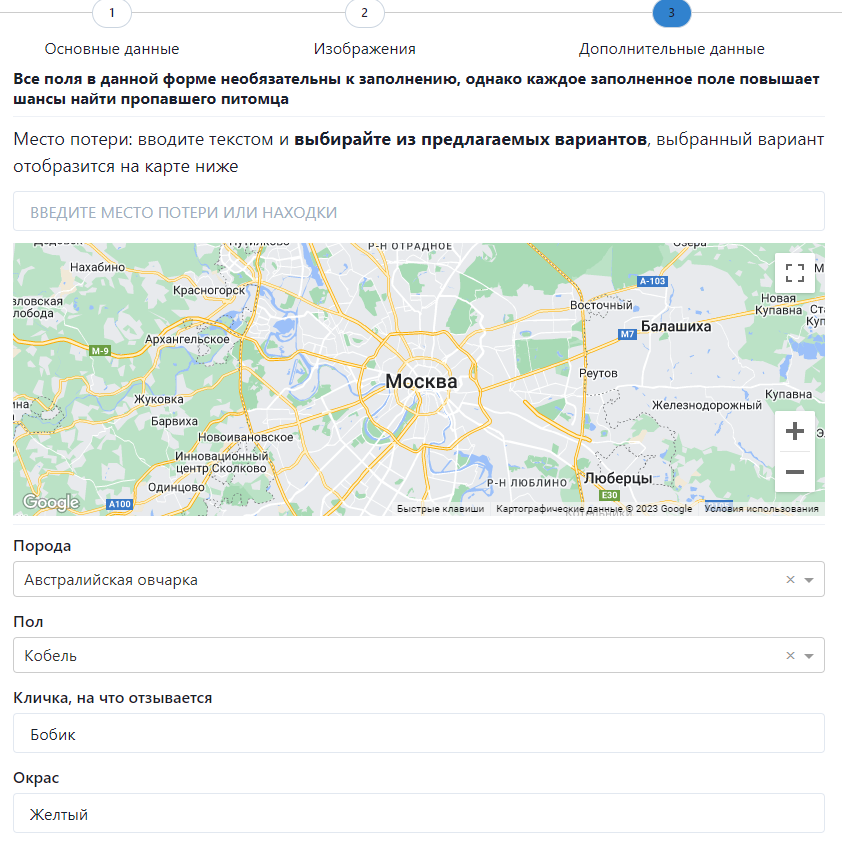


Рисунок 5 – Карта с указанием местоположения питомца при создании объявления

После обзора и анализа похожих по функционалу сайтов, была составлена таблица 1 с основными критериями сравнения.

1. – Таблица сравнения аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | «poiskzverei.ru» | «pet911.ru» | «FoundPets.ru» |
| Привлекательный дизайн | + | + | + |
| Список объявлений на главной странице | + | - | + |
| Связь с ветеринарными клиниками | - | + | - |
| Понятный интерфейс | - | - | - |
| Фильтрация поиска по критериям | + | + | - |
| Наличие обратной связи | - | + | + |
| Возможность авторизации/регистрации | + | + | - |
| Возможность взаимодействия пользователей между собой в комментариях под объявлениями | - | + | + |
| Раздел для связи с администратором или модератором | - | + | + |

## Анализ целевой аудитории

Для разработки данного сайта необходимо провести аналитику целевой аудитории и понять потребности пользователей, так как это необходимо при создании дизайна и определению функциональных возможностей проекта.

В городе Красноярск был проведен опрос [4], на основе которого были выделены следующие сегменты пользователей, которым может быть интересна информация на сайте:

* люди, которые нашли питомца и помогают отыскать для него своего хозяина;
* люди, которые хотят изучить информацию о потерянных животных и помочь им найти дом;
* люди, которые потеряли домашнего питомца и хотят его найти;
* люди, которые хотят отдать питомца в добрые руки.

Определив сегменты пользователей, можно сделать вывод о том, что функциональные требования должны подходить под потребности пользователей, поэтому он должен иметь удобный дизайн с заметными кнопками на главной странице, а также необходимо разработать список постов, поиск по категориям объявлений и их упрощенное создание.

## Функциональные требования

В результате анализа существующих аналогов и целевой аудитории, рассмотрения функционала подобных систем и для отображения функциональных требований к веб-сайту были построены диаграммы вариантов использования [5], представленные на рисунках 6 – 9.

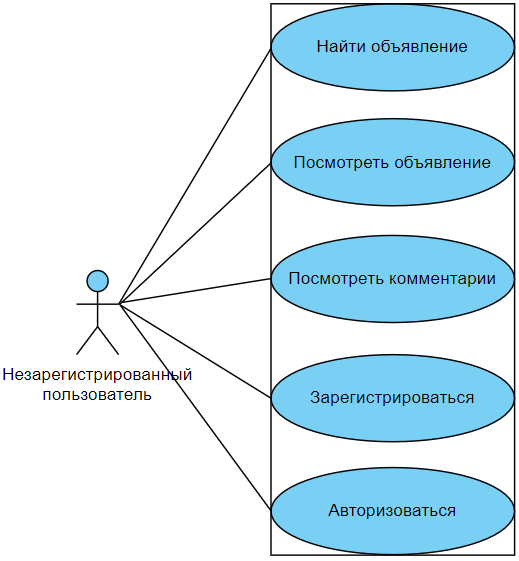


Рисунок 6 - Диаграмма вариантов использования сущности «Незарегистрированный пользователь»

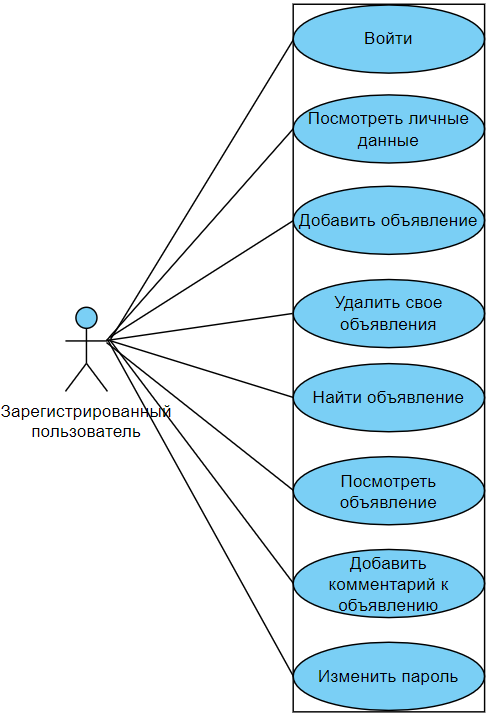


Рисунок 7 - Диаграмма вариантов использования сущности «Зарегистрированный пользователь»

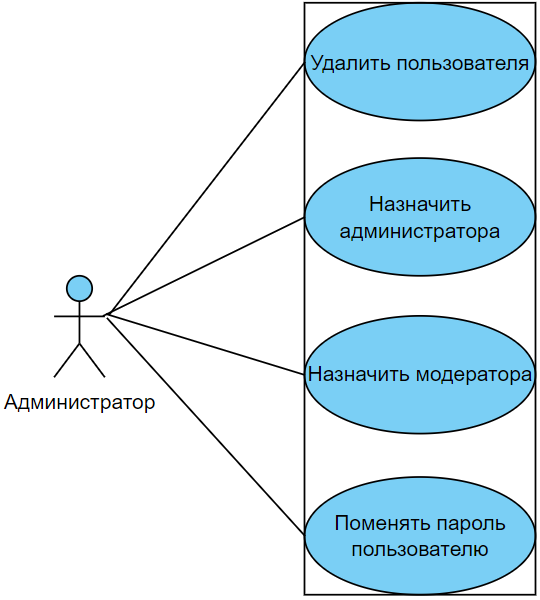


Рисунок 8 - Диаграмма вариантов использования сущности «Администратор»

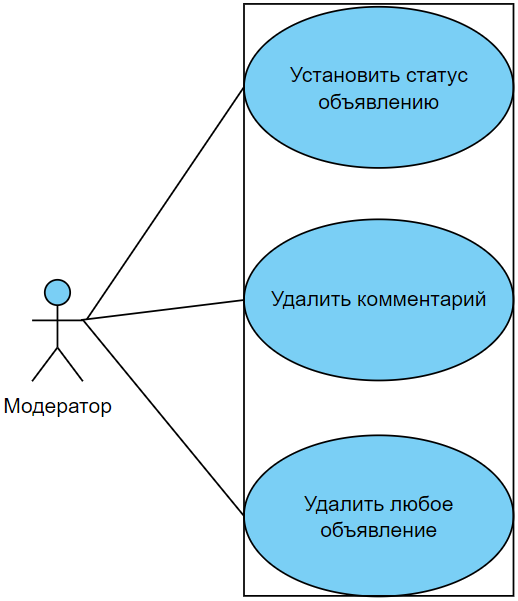


Рисунок 9 – Диаграмма вариантов использования сущности «Модератор»

Данные диаграммы описывают, какие функции должны быть реализованы в разрабатываемом сайте.

Исходя из вышеперечисленных диаграмм использования, можно заметить, что в разрабатываемом проекте имеются 4 сущности: «Неавторизованный пользователь», «Авторизованный пользователь», «Администратор» и «Модератор».

Для роли «Незарегистрированный пользователь» система предоставляет следующие возможности: регистрация, авторизация, просмотр комментариев и объявлений и поиск объявлений по категориям.

Для роли «Зарегистрированный пользователь» система предоставляет следующие возможности: просмотр личного кабинета, добавление и удаление своего объявления, добавление комментариев, изменение своего пароля, а также все возможности незарегистрированного пользователя.

Для роли «Администратор» система предоставляет следующие возможности: удаление пользователя, изменение пароля пользователю, а также назначение модераторов и администраторов. Сущностей «Администратор» может быть несколько, в зависимости от нагрузки на сайт.

Для роли «Модератор» система предоставляет следующие возможности: установка статуса объявлениям, удаление объявлений на сайте, а также удаление комментариев. Сущностей «Модератор» может быть неограниченное количество, в зависимости от нагрузки на сайт.

## Описание функций

В данном подразделе будет описана работа основных функций сайта по поиску пропавших животных, а именно:

1. Функция «Регистрация»;

Выполняет регистрацию пользователя

Входные данные: введенные с клавиатуры логин, электронная почта и пароль.

Выходные данные: переход в личный кабинет.

* вводится имя пользователя, электронная почта и пароль;
* электронная почта не занята, то выполняется пункт 3;
* логин и пароль сохраняются в БД;
* при успешной регистрации переход на страницу авторизации.

1. Функция «Авторизация»;

Выполняет авторизацию пользователя.

Входные данные: введенные с клавиатуры логина и пароля пользователя.

Выходные данные: переход в личный кабинет.

* считывается имя пользователя и пароль;
* если поле пустое, то выполняется пункт 3, если данные введены, то пункт 4;
* выводится сообщение об ошибке;
* подключение к БД, выборка логинов и проверка пароля, если данные не совпадают, то выполняется пункт 3, если совпали – 5;
* происходит проверка пароля, если выполнено успешно, то выполняется пункт 6;
* открытие основной формы.

1. Функция «Добавление объявления»;

Данная функция реализует добавление новых объявлений в таблицу «Объявление».

Входные данные: введенные данные с клавиатуры в поля: «Кличка», «Дополнительная информация о питомце», «Категория», «Вид животного» «Добавить фотографию».

Выходные данные: новая запись в таблице.

* считываем данные, введенные с клавиатуры;
* если имеются не заполненные поля, тогда выводится предупреждающее сообщение и выполняется пункт 3;
* производится запись введённых значений в поля таблицы.

1. Функция «Удаление объявления»

Данная функция реализует удаление записей из таблиц.

Входные данные: Название объявления.

Выходные данные: Преобразованная таблица.

* выбирается название объявления;
* переводим курсор на кнопку «Удалить»;
* выводится преобразованная таблица.

1. Функция «Поиск объявления»;

Данная функция реализует поиск объявления в таблице «Объявление».

Входные данные: выбранные пользователем «Тип» и «Категория» животного.

Выходные данные: запись таблицы.

* считываем данные, выбранных критериев;
* производится поиск в таблице с заданными значениями;
* вывод записей с заданными значениями.

1. Функция «Загрузка фотографии»;

Данная функция реализует загрузку фотографии в таблицу «Объявления».

Входные данные: имя изображения, тип изображения.

Выходные данные: запись в базу данных.

* выбранный файл;
* в случае некорректных данных, выводится предупреждающее сообщение;
* производится локальное сохранение картинки в подпапку с годом, месяцем и числом. Далее происходит добавление пути к этому файлу в базу данных.

1. Функция «Помощь»;

Данная функция реализует отправку формы на почту администратору пользователем.

Входные данные: заполненные поля «Проблема» и «Email».

Выходные данные: новое письмо на почте администратора.

* загрузка формы;
* заполнение пользователем полей «Помощь» и «Email»;
* производится отправка формы по почте.

1. Функция «Изменение своего пароля»;

Данная функция предназначена для изменения своего пароля или восстановления администратором пароля пользователя в случае его потери.

Входные данные: старый и новый пароль пользователя.

Выходные данные: изменение пароля в базе данных.

* открытие формы смены пароля;
* вводятся данные в поля «Старый пароль», «Новый пароль» и «Подтвердите новый пароль»;
* подтверждение смены пароля
* в случае неверных данных повторяется пункт 2
* смена пароля аккаунта.

1. Функция «Назначение статуса объявлению»;

Данная функция предназначена для установки статуса объявлению.

Входные данные: списки объявлений «Посты на одобрение», «Посты публичные» и «Посты архивные».

Выходные данные: изменение статуса объявлению.

* вход в меню менеджера;
* выбор нового статуса объявлению;
* изменение статуса объявления и его запись в базу данных.

1. Функция «Назначение роли пользователю»;

Данная функция предназначена для назначение администратором новой роли пользователю.

Входные данные: список всех пользователей и их ролей.

Выходные данные: изменение роли пользователя.

* вход в меню администратора;
* выбор новой роли пользователя;
* изменение роли пользователя и ее запись в базу данных.

1. Функция «Удаление пользователя»;

Данная функция предназначена для удаления аккаунта пользователя.

Входные данные: список всех зарегистрированных пользователей.

Выходные данные: удаление из базы данных аккаунта пользователя.

* вход в меню администратора;
* выбор нужного аккаунта;
* удаление данных о пользователе из базы данных.

1. Функция «Удаление комментария»;

Данная функция реализует удаление модератором комментария под объявлением.

Входные данные: выбранный комментарий.

Выходные данные: удаление комментария из базы данных.

* выбран нужный комментарий
* производится удаление комментария из базы данных.

1. Функция «Изменение пароля пользователю»

Данная функция предназначена для изменения администратором пароля пользователя.

Входные данные: выбранный пользователь.

Выходные данные: измененный пароль пользователя.

* выбран нужный пользователь;
* открытие формы смены пароля пользователю;
* ввод нового пароля;
* в случае ввода некорректных данных выполняется пункт 3;
* изменение пароля в базе данных.

## Средства разработки

Анализировав функциональные требования и проведя аналитику сайтов аналогов и аудитории, было решено выбрать следующие инструменты для реализации данного веб-сайта:

1. Язык программирования Python;
2. Библиотека Django;
3. Библиотека Pillow;
4. Библиотека Smtplib;
5. База данных SQLite3.

Комбинация этих инструментов обеспечивает гибкую разработку веб-сайта по поиску пропавших животных, так как Python [6] – язык, обеспечивающий быстрое прототипирование сайтов, так как он имеет готовый набор фреймворков и вспомогательных библиотек, упрощающих разработку веб-приложений. При увеличении нагрузки, система легко поддаётся масштабированию и позволяет расширять приложение как в ширину, за счёт микросервисной архитектуры, так и в глубину, за счёт оптимизации кодовой базы и распараллеливания критических задач.

В качестве основного веб фреймворка была выбрана библиотека Django [7], так как данная библиотека уже включает в себя полный цикл разработки веб-приложений и обеспечивает необходимые API для удобства разработки. Функции шаблонов, включающие в себя встроенную административную панель, систему аутентификации, авторизации, ускоряют процесс разработки. В библиотеке Django имеется встроенная ORM, которая обеспечивает взаимодействие между веб-сайтом и базой данных.

В качестве базы данных была выбрана SQLite [8], так как она по умолчанию поддерживается ORM, встроенной в Django. Необходимости хранения больших файлов в проекте нет и SQLite подходит под реализуемый веб-сайт в связи с тем, что все ключевые поля хранятся в текстовом виде, за исключение картинок, которые хранятся в структурированных каталогах на сервере. Файл базы данных SQLite3 является переносимым, что означает возможность легко перемещать базу данных между различными окружениями и платформами без необходимости в дополнительной настройке.

Для обеспечения формализации и загрузки картинок используется библиотека Pillow. Она обеспечивает поддержку различных форматов изображений, включая JPEG, PNG, GIF, BMP и многие другие.

Для реализации отправки писем на почту администратора использована встроенная в Django библиотека Smtplib [9]. Она обеспечивает безопасное подключение к почтовому SMTP-серверу с использованием его адреса и защищенного порта 587.

# Устройство сайта

## Разработка интерфейса

Диаграмма потока экранов разрабатываемого сайта по поиску пропавших животных представлена в приложении А.

На главной странице сайта присутствует 4 блока:

* шапка сайта с основными навигационными кнопками:

1. Главная страница;
2. Поиск по критериям;
3. О нас;
4. Контакты и адреса;
5. Архив;
6. Войти в существующий аккаунт/Зарегистрироваться (Неавторизованный пользователь);
7. Мой профиль/Выйти (Авторизованный пользователь).

* список объявлений;
* создать объявление;
* кнопка «Помощь»;
* последние объявления.

В шаблоне base.html [10] присутствуют следующие блоки кода:

* шапка сайта с основными навигационными кнопками;
* задний фон сайта;
* кнопка «Помощь», с особой стилизацией [11];
* кнопки «Менеджмент» и «Меню Админа», предназначенные для настройки конфигураций пользователей и их объявлений, а также кнопки для регистрации и просмотра личного кабинета;
* подвал сайта со ссылками на социальные сети администратора для дополнительной связи, идей и предложений по улучшению веб-сервиса;
* модальное окно и скрипт для окна с подтверждением при выходе из аккаунта.

Листинг модального окна и скрипта приведен в приложении Б.

В шаблоне home.html присутствуют следующие блоки кода:

* название сайта и логотип «BobikFinder», с использованием эффекта тени и шрифта Comic Sans MS;
* разделы для просмотра и создания объявлений;
* раздел с последними объявлениями на сайте.

Листинг раздела с последними объявлениями приведен в приложении В.

Файл models.py служит для описания моделей, которые представляют таблицы в базе данных. Он задает их структуру, поля и связи между ними.

В данном файле присутствуют следующие модели:

1. Category. Модель для категорий питомцев. Содержит поле name для названия категории;
2. Type\_of\_animal. Модель для типов животных. Содержит поле name для названия типа животного;
3. Status. Модель для статусов объявлений. Содержит поле name для названия статуса;
4. Posts. Модель для объявлений о питомцах. Содержит поля, такие как id\_user, title, slug, additional\_information, photo, time\_create, time\_update, category, type\_of\_animal и status, которые представляют соответствующие атрибуты объявления;
5. Comments. Модель для комментариев к объявлениям. Содержит поля id\_user, id\_pet, content, time\_create и user, которые представляют соответствующие атрибуты комментария;
6. Role. Модель для ролей пользователей. Содержит поле name для названия роли;
7. List\_role. Модель для связи между ролями и пользователями. Содержит поля role и user, которые представляют связи между ролью и пользователем.

В приложении Г приведен листинг файла models.py с описанием модели объявления.

Файл views.py служит для определения представлений и функций-обработчиков для различных манипуляций для http-запросов. Данный файл содержит следующие импорты и определения представлений:

* импорты модулей и классов;
* определение функций-обработчиков.

Из основных можно выделить следующие определения:

1. Функция «Add\_pet»;

Данная функция обрабатывает запрос на добавление нового объекта «pet».

1. Функция «Delete\_post»;

Данная функция удаляет указанный пост пользователя из базы данных.

1. Функция «Menadje»;

Данная функция отвечает за управление публикациями пользователей. Она обрабатывает запросы модератора, позволяя изменять статус записей между «проверка», «публично» и «архив», а также выводит списки всех записей.

1. Функция «Delete\_user»;

Данная функция выполняет удаление пользователя из системы. Она обрабатывает запросы администратора, исключает указанного пользователя из базы данных и связанных с ним данных, таких как объявления и комментарии.

1. Функция «Аrchive\_old\_posts»;

Данная функция выполняет архивацию старых записей путем обновления их статуса на «архивный». Она определяет записи, время последнего обновления которых превышает месяц, меняет их статус и перемещает в архивную категорию.

В приложении Д приведет листинг файла views.py с функциями «Delete\_post» и «Аrchive\_old\_posts».

В приложении Е представлен код файла contact.html. Данный шаблон предназначен для отображения интерактивной карты с адресами и номерами ветеринарных клиник в городе Красноярск. Для этого используется JavaScript-код, который подключает API «Яндекс.Карт» и конструктор карты с заданными параметрами.

## Разработка базы данных

Для разработки приложения с возможностью контроля задач и анализа дальнейшей разработки веб-сайта требуется разработать базу данных.

База данных будет состоять из нескольких таблиц:

Таблица users\_category – данная таблица предназначена для хранения данных о категории объявления. Таблицы представлена на рисунке 10.

* ID – id категории;
* Name – наименование категории (пропал/нашел/в добрые руки).

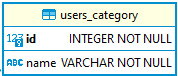


Рисунок 10 – Таблица с категориями объявлений

Таблица users\_type\_of\_animal – данная таблица предназначена для хранения данных о виде животного (кошка, собака, ёжик, другое и т.д.). Таблица представлена на рисунке 11.

* ID – id самого вида;
* Name – наименование вида.

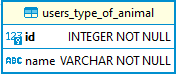


Рисунок 11 – Таблица с видами животных

Таблица auth\_user – данная таблица предназначена для хранения данных о пользователе (логин, пароль, почта и т.д.). Таблица представлена на рисунке 12.

* ID – id пользователя;
* Password – пароль пользователя;
* Last\_login – время последнего входа на сайт;
* Email – почта пользователя;
* Date\_joined – дата регистрации;
* Username – имя пользователя.

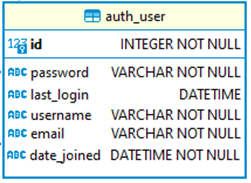


Рисунок 12 – Таблица с информацией о пользователе

Таблица users\_posts – данная таблица предназначена для хранения данных об объявлениях на сайте. Таблица представлена на рисунке 13.

* ID – id поста;
* Id\_user – id пользователя поста;
* Title – имя животного в объявлении;
* Additional\_information– дополнительная информация и питомце в объявлении;
* Photo – путь к фотографии в локальных файлах;
* Time\_create – время создания поста;
* Time\_update – время обновления поста;
* Category\_id – id категории объявления (пропал/нашел/в добрые руки);
* Type\_of\_animal\_id – id вида животного в объявлении (кошка/собака/ёжик и т.д.);
* Status\_id – id статуса объявления (проверка/публично/архив).

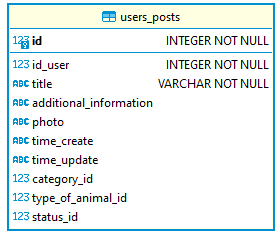


Рисунок 13 – Таблица с данными об объявлении

Таблица comments – данная таблица предназначена для хранения данных о всех комментариев на сайте и показана на рисунке 14.

* ID – уникальный аутентификатор комментария;
* Id\_user – id пользователя, оставившего комментарий;
* Id\_post – id объявления, под которым оставили комментарий;
* Content – текст данного комментарий;
* Time\_create – время создания комментария;

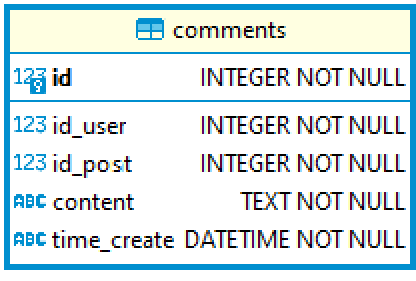


Рисунок 14 – Таблица с комментариями

Таблица users\_list\_role – данная таблица предназначена для хранения информации о ролях всех пользователей и показана на рисунке 15.

* ID – id записи в таблице;
* Role\_id – id роли (администратор, модератор, обычный пользователь);
* User\_id – id пользователя из таблицы auth\_user;

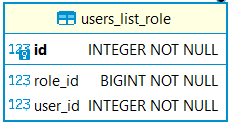


Рисунок 15 – Таблица ролей пользователя

Таблица users\_role – данная таблица предназначена для хранения названий ролей (строка «Name» - администратор/модератор/обычный пользователь) и id роли Диаграмма представлена на рисунке 16.

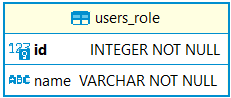


Рисунок 16 – Таблица названий ролей

В базе данных присутствует тип связи один к одному между таблицами auth\_user и users\_list\_role. Такой тип связи выбран во избежание конфликтов между стандартным функционалом библиотеки Django и функционалом, созданным в данном проекте. В связи с этим, было решено составить новую таблицу users\_list\_role и связь между данной таблицей и auth\_user осуществить как один к одному. В строке user\_id записывается id пользователя из таблицы auth\_user, а в строке role\_id записывается сама роль, взятая из таблицы users\_role.

По итогу составления таблиц получается логически связанная диаграмма базы данных [12], представленная на рисунке 17.

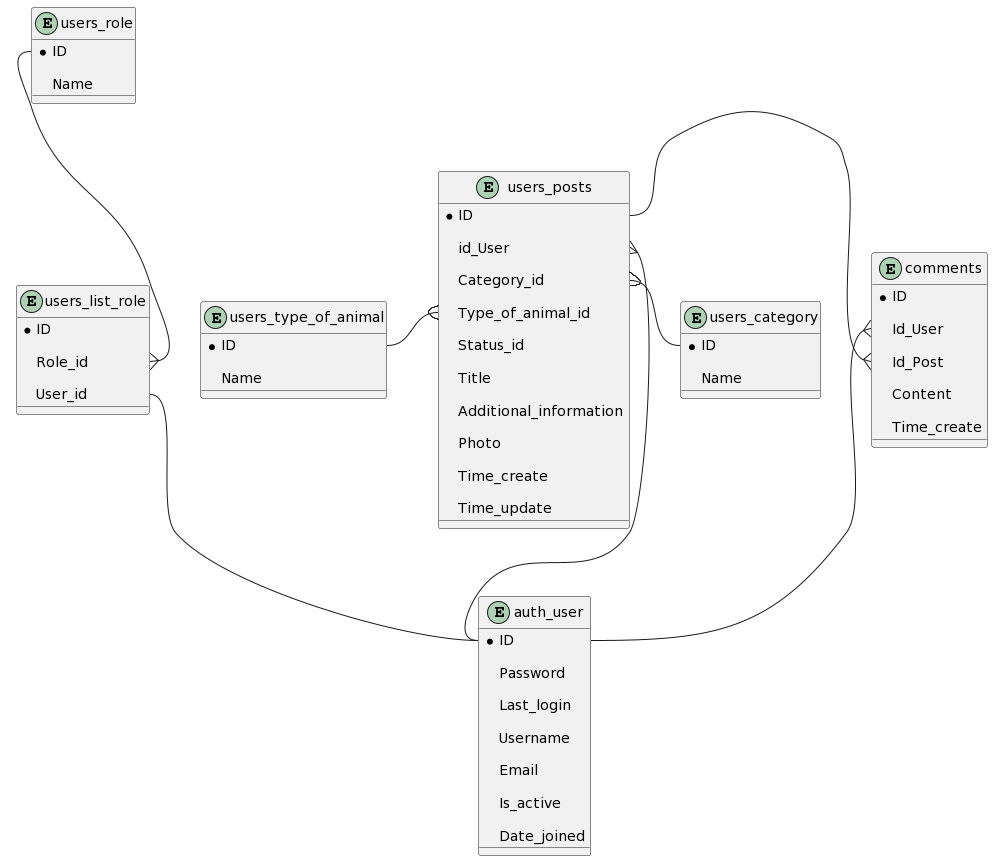


Рисунок 17 – Диаграмма сущностей базы данных

# Реализация функций

В данном разделе описана реализация основных функций веб-сайта по поиску пропавших животных, а именно:

1. «Авторизация»;
2. «Регистрация»;
3. «Просмотр объявлений»;
4. «Добавление/удаления объявления»;
5. «Поиск объявления»;
6. «Загрузка фотографии»;
7. «Добавление/удаление комментария»;
8. «Поиск ветеринарных клиник на карте»;
9. «Изменение статуса объявлений»;
10. «Помощь».

## Авторизация

Пользователь находится в начальном окне авторизации, которое изображено на рисунке 18.

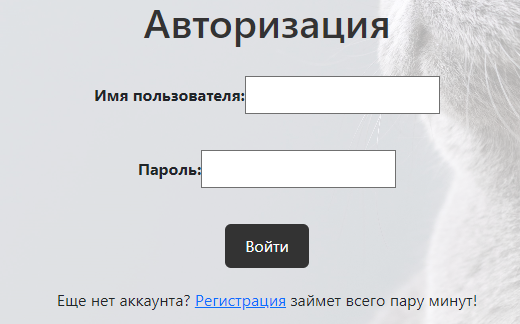


Рисунок 18 – Окно «Авторизация»

Основная последовательность:

* пользователь вводит свой логин и пароль;
* нажимает кнопку войти.

На верхней панели появляется кнопка «Мой профиль», где пользователь может увидеть свои данные.

Альтернативная последовательность: введены неправильные данные.

Появляется уведомление о неправильности введенных данных.

На рисунках 19 и 20 показан личный кабинет пользователя и кнопки профиля и выхода с аккаунта.

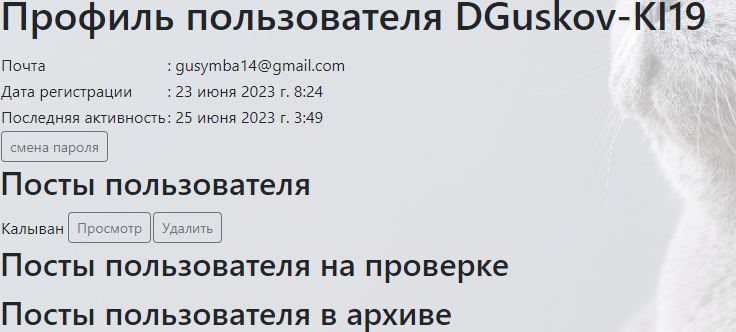


Рисунок 19 – Личный кабинет пользователя

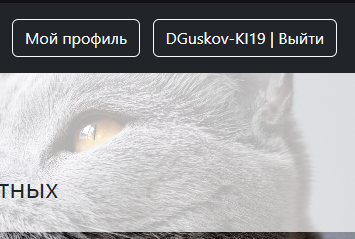


Рисунок 20 – Кнопки входа в свой профиль и выхода из аккаунта

## Регистрация

Предусловие: регистрацию может проходить только неавторизованный пользователь.

Основная последовательность:

* пользователь вводит свои данные: логин, email, пароль и повтор пароля;
* нажимает кнопку зарегистрироваться.

Постусловие: появляется возможность входа в слой личный кабинет и выхода из профиля.

Альтернативная последовательность: неправильный ввод данных

Постусловие: сохраняется основная последовательность, но появляются уведомления о неправильности ввода данных. На рисунках 21 и 22 изображены окна регистрации и уведомление о неправильности ввода данных.

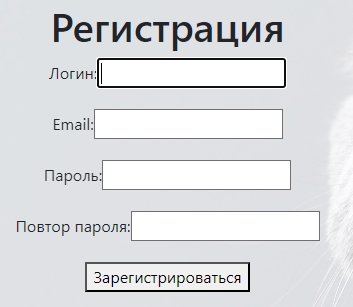


Рисунок 21 – Вкладка регистрации

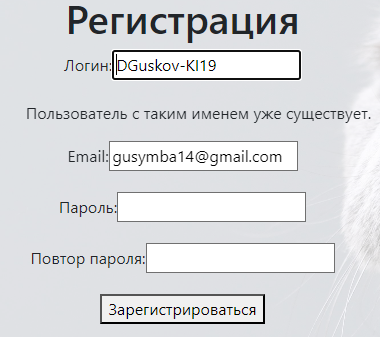


Рисунок 22 – Уведомления при неправильности вводимых данных

## Создание объявления

Предусловие: пользователь должен войти в свой аккаунт.

Основная последовательность:

* пользователь нажимает кнопку «Создать объявление» на главной странице;
* попадает на страницу «Новое объявление»;
* вводит в поля информацию о найденном/потерянном питомце;
* нажимает на кнопку «Разместить объявление».

Постусловие: на сайте во вкладке «Список объявлений» появляется созданный пользователем пост.

Альтернативная последовательность: пользователь не авторизован или не все поля заполнены.

Постусловие: повторение основной последовательности до ввода текста на всех полях, либо переход на страницу авторизации.

На рисунках 23 и 24 показаны окна с кнопкой создания объявления, страница с добавлением нового объявления, поля для ввода информации о питомце, кнопка «Разместить объявление» и оповещение, что объявление будет доступно на сайте 30 суток, после чего оно будет перемещено в архив.

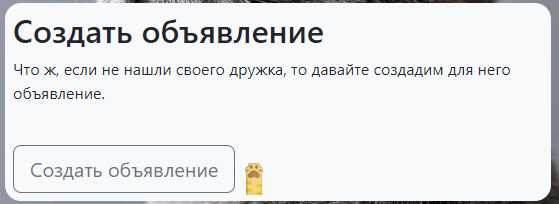


Рисунок 23 – Раздел создания объявления и кнопка «Создание объявления»

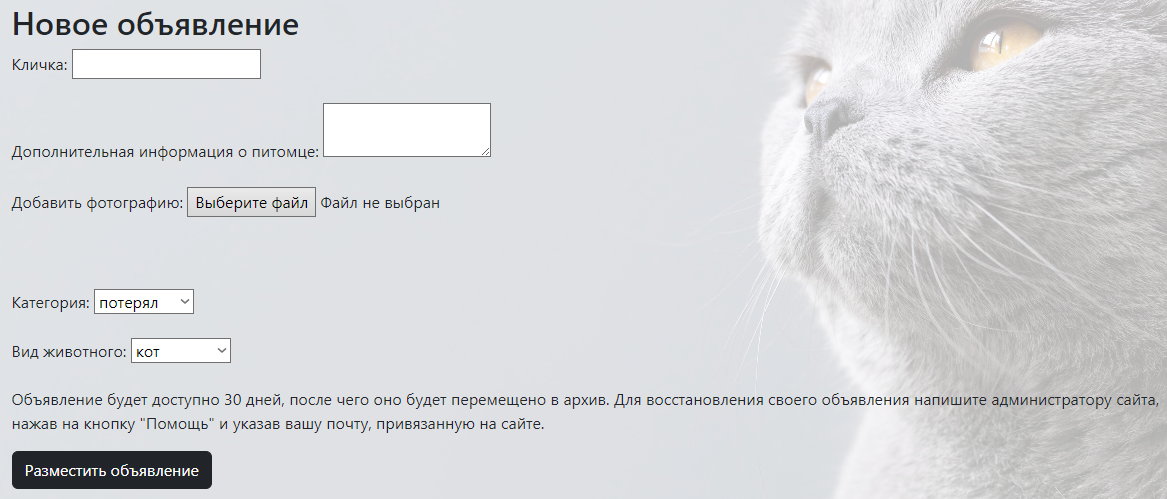


Рисунок 24 – Раздел ввода данных при создании нового объявления

## Просмотр объявлений

Предусловие: может осуществлять как авторизованный, так и неавторизованный пользователь, также пользователь находится на главной странице.

Основная последовательность:

* пользователь нажимает на кнопку «Смотреть объявления»;
* переходит на страницу со списком объявлений;
* нажимает на кнопку «Смотреть» на объявлении и просматривает выбранное объявление.

Постусловие: открывается выбранное пользователем объявление. На рисунках 25 – 27 изображена кнопка «Смотреть объявления», окно со списком объявлений, а также окно выбранного пользователем объявления.

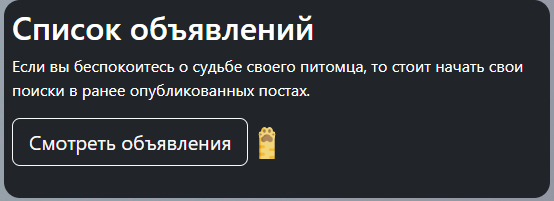


Рисунок 25 – Раздел для входа в список объявлений

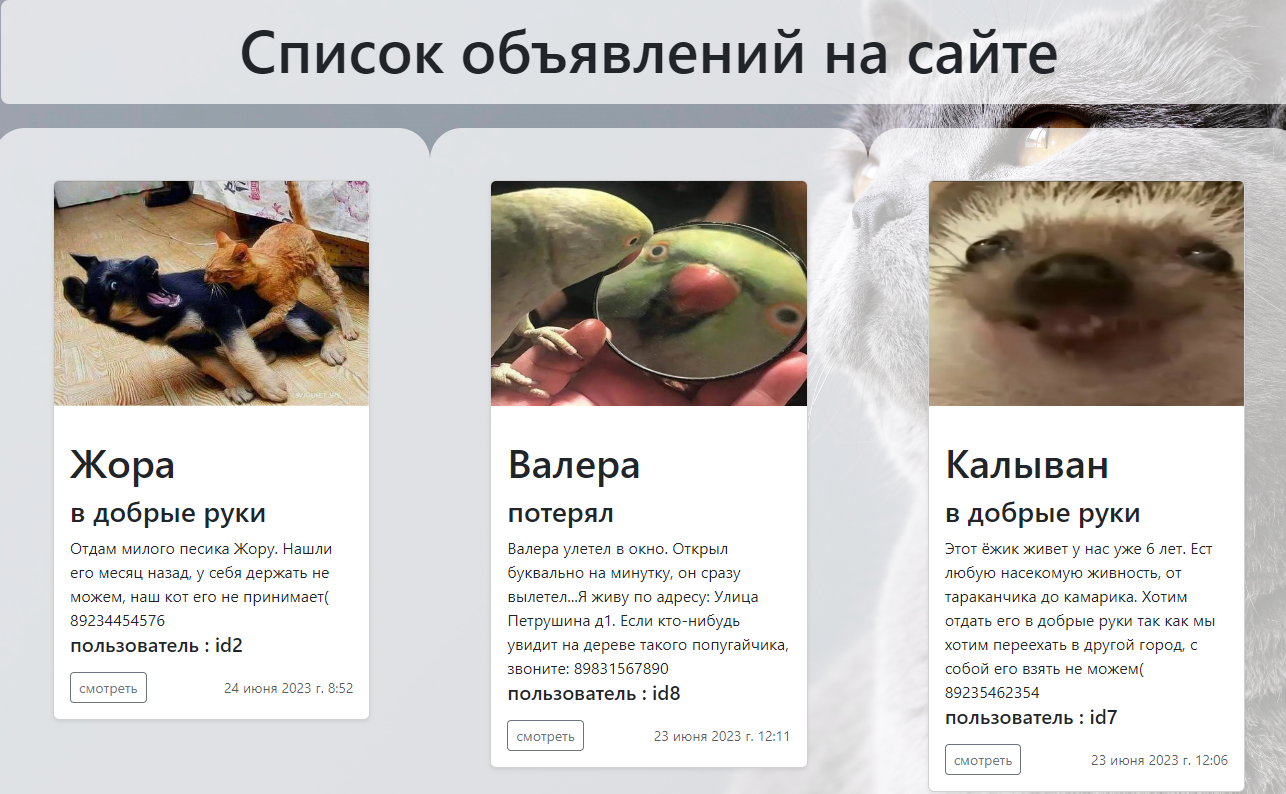


Рисунок 26 – Список всех объявлений на сайте



Рисунок 27 – Просмотр объявления при нажатии кнопки «Смотреть»

## Добавление комментария к объявлению

Предусловие: оставлять комментарий может только авторизованный пользователь, также он должен находиться во вкладке с выбранным им объявлением.

Основная последовательность:

* пользователь нажимает на поле ввода текста в блоке «Комментарии»;
* нажимает на кнопку «Добавить коммент».

Постусловие: пользователь видит оставленный пользователем комментарий в блоке «Комментарии» к объявлению. На рисунке 28 изображено окно «Комментарии», кнопка «Добавить коммент», поле для ввода текста комментария и последний комментарий к объявлению.



Рисунок 28 – Раздел «Комментарии» с полем для ввода и кнопкой для добавления комментария

## Удаление объявления

Предусловие: «Администратор» может удалять любое объявление на сайте, а «Авторизованный пользователь» – только свои, также пользователь должен находиться в личном кабинете.

Основная последовательность: пользователь нажимает на кнопку «Удалить» в поле «Посты пользователя».

Постусловие: пост полностью удаляется со всех баз данных сайта. На рисунке 29 изображена кнопка «Удалить» в поле «Посты пользователя» в личном кабинете.

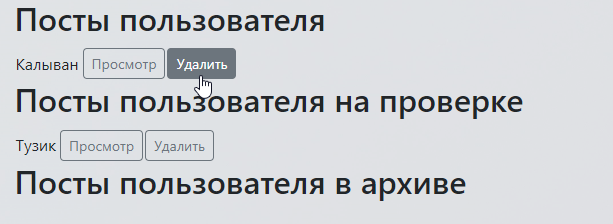


Рисунок 29 – Кнопка «Удалить» объявление в личном кабинете

## Модерация

Предусловие: воспользоваться данной функцией может только модератор.

Основная последовательность: модератор нажимает на кнопку Менеджмент в шапке сайта.

Постусловие: Модератор изменяет статусы объявлениям – архивное, публичные и находящиеся на одобрении.

На рисунках 30 и 31 изображена вкладка «Менеджмент» с объявлениями и их статусами, а также кнопка удалить комментарий.

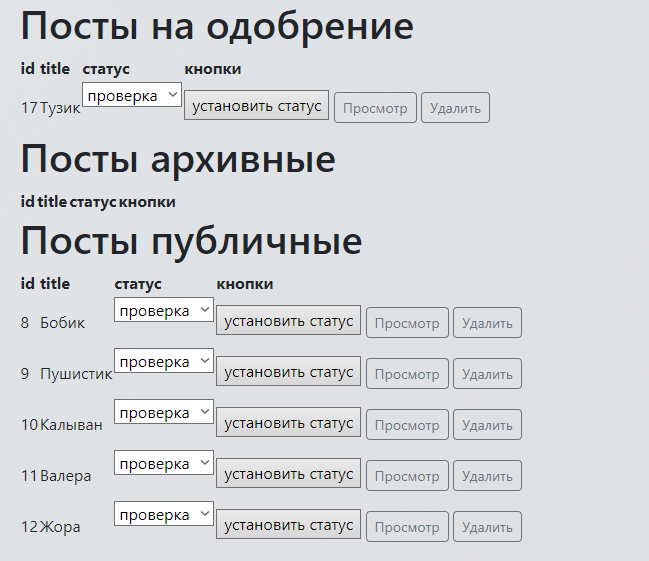


Рисунок 30 – Меню модератора

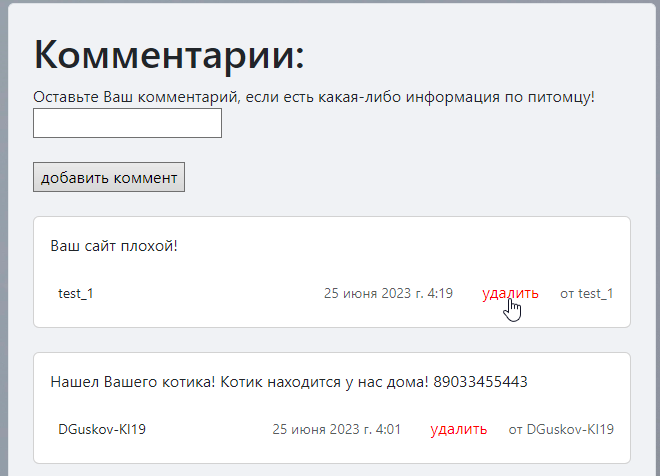


Рисунок 31 – Функция удаление комментария модератором

## Администрирование

Предусловие: функция администрирования включает в себя удаление пользователя, а также смена пароля пользователю.

Основная последовательность: администратор входит в «Меню Админа», где в дальнейшем происходит смена пароля или удаление пользователя.

Постусловие: администратор изменяет пароль пользователю, либо удаляет его из базы данных. На рисунке 32 - 34 изображена вкладка «Меню Админа» с основными функциями администрирования, смена пароля пользователю, а также смена пароля в личном кабинете.

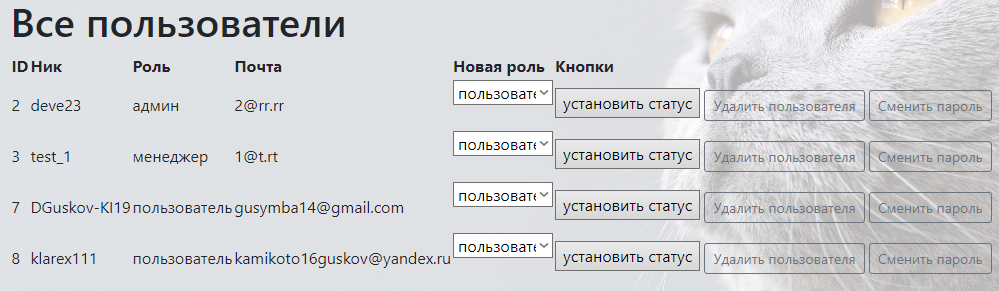


Рисунок 32 – Меню администратора

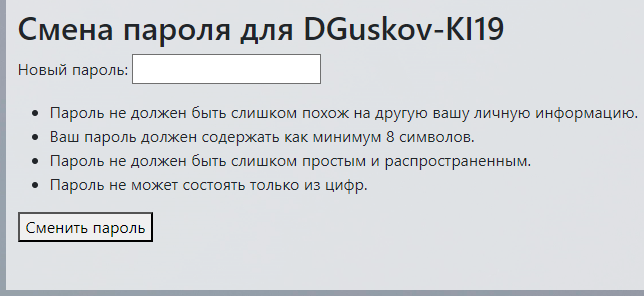


Рисунок 33 – Изменение пароля пользователю в «Меню Админа»

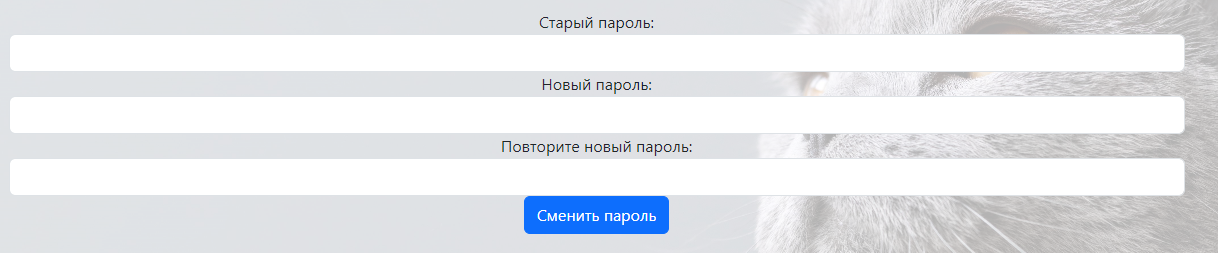


Рисунок 34 – Изменение пароля в личном кабинете пользователя

## Модальные окна сайта

На сайте подобного типа решено использовать модальные окна с оповещениями и подтверждениями действий пользователя и администратора. Важной частью разработки веб-сайта является создание модальных окон с подтверждениями или оповещениями пользователя. Реализованы 3 модальных окна с оповещениями и подтверждениями действий пользователей.

Модальное окно с подтверждением выхода из аккаунта изображено на рисунке 35.

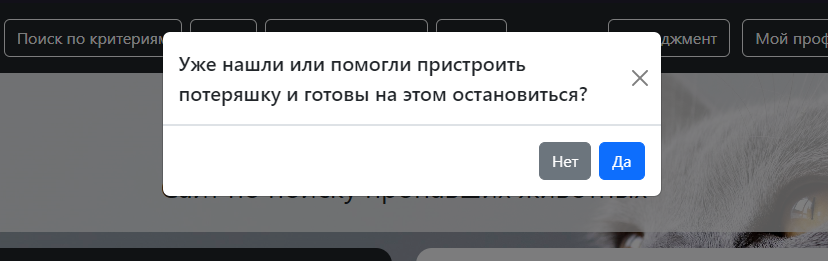


Рисунок 35 – модальное окно с подтверждением выхода

Модальное окно с оповещением при создании объявления изображено на рисунке 36.

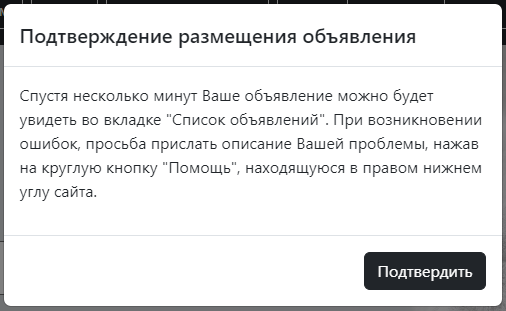


Рисунок 36 – Оповещение пользователя при создании объявления

Модальное окно с подтверждением удаления аккаунта в меню администратора изображено на рисунке 37.

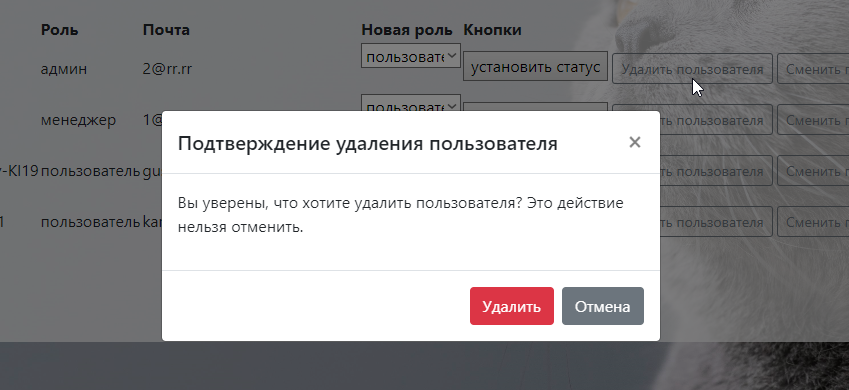


Рисунок 37 – Подтверждение удаления пользователя в меню администратора

## Помощь

Кнопка помощь показана на рисунке 38. Она предназначена для связи с администратором по почте. При нажатии на нее, открывается форма, при заполнении которой присылается письмо на почту администратору сайта. Вкладка помощь с формой записи возникшей проблемы и капчей, для устранения спама, показана на рисунке 39.



Рисунок 38 – Кнопка помощь на главной странице сайта

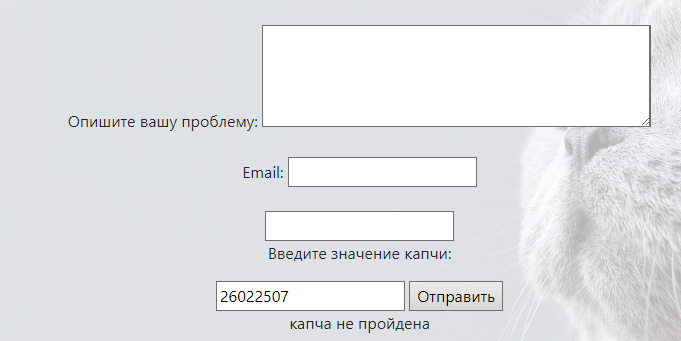


Рисунок 29 – Форма записи проблемы и ее отправки на почту администратора

## Тестирование

Тестирование проводилось вручную. Например, для проверки корректной регистрации в системе выполнялись следующие действия:

* вводился русский язык в раздел «Email»;
* не вводился символ @ в раздел «Email»;
* какое-то из полей оставалось пустым;
* вводился уже существующий логин;
* вводился похожий на логин пароль;
* вводился пароль, состоящий только из цифр;
* вводился пробел в раздел «Логин»;
* вводился пароль менее 8 символов.

Аналогичное тестирование было проведено для всех прецедентов, описанных в первом разделе.

Была протестирована функция добавления старого объявления в архив. Изменяя дату на сервере на 31 день вперед, объявление меняло статус из «Публичное» в «Архивное»

Также была протестирована отправка сообщения на почту. Результаты данных тестов изображены на рисунках и 40 и 46.

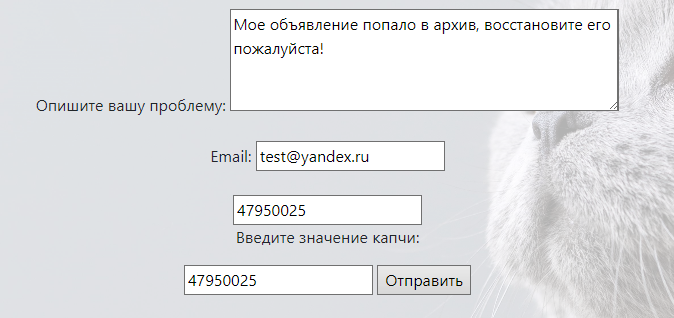


Рисунок 40 – Тестирование отправки формы на почту

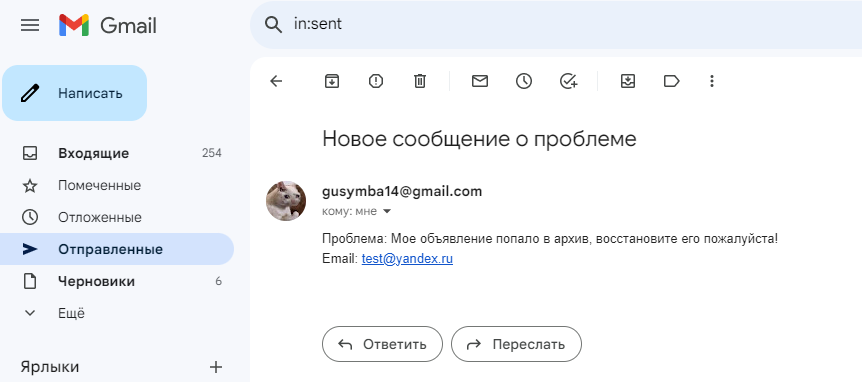


Рисунок 41 – Присланное с сайта тестовое сообщение

На рисунках 42 - 50 показаны действия для проверки и тестирования ввода данных при регистрации и тестирование выбора правильного формата файла.

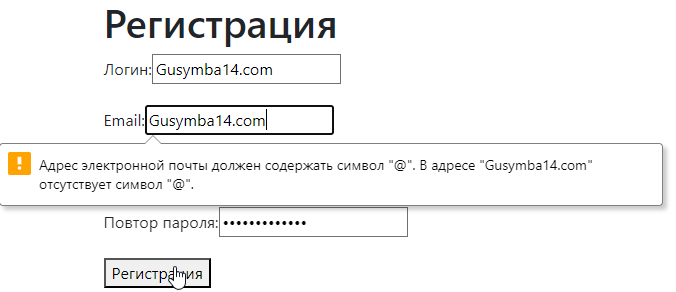


Рисунок 42 – Тестирование правильности ввода почты

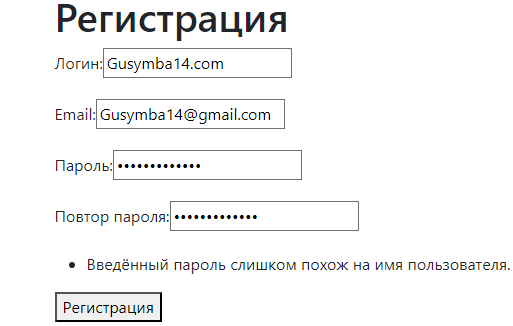


Рисунок 43 – Проверка пароля на схожесть с логином

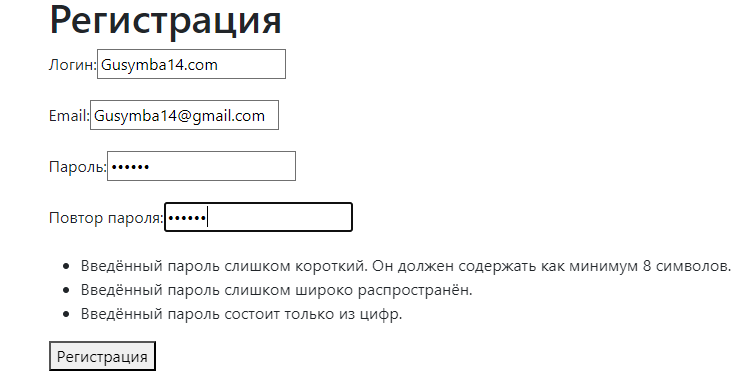


Рисунок 44 – Проверка защищенности регистрации, при вводе простого пароля

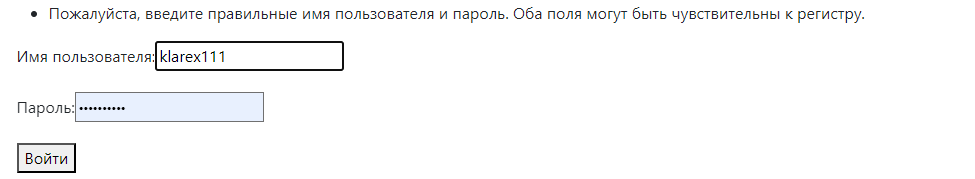


Рисунок 45 – Проверка входа в аккаунт, при попытке ввода неправильного пароля



Рисунок 46 – Проверка правильности выбора формата файла

Во избежание спама на почту администратора, было решено использовать простую капчу. На рисунке 47 показана капча и тестирование ее работы.

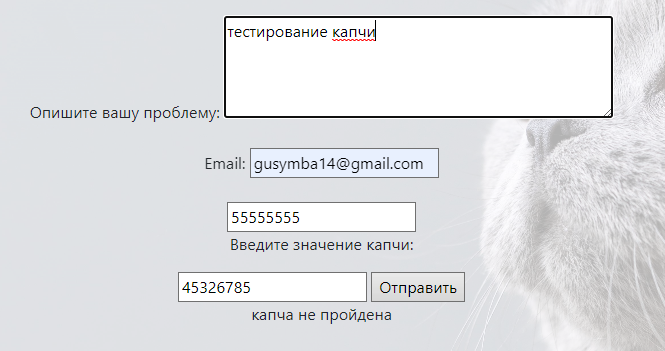


Рисунок 47 – Тестирование капчи

Заключение

В выпускной квалификационной работе разработан «Сайт по поиску пропавших животных», предоставляющий пользователям объявления о найденных и потерянных животных.

Были достигнуты следующие результаты:

1. Проведен анализ предметной области;
2. Разработан макет сайта с удобным и понятным интерфейсом;
3. Реализованы функции по поиску, просмотру, удалению, добавлению объявлений, а также функция просмотра ветеринарных клиник на интерактивной карте;
4. Реализована функция комментирования записей для взаимодействия между пользователями на сайте;
5. Разработан функционал администратора и модератора для конфигурации настроек пользователей и их объявлений.

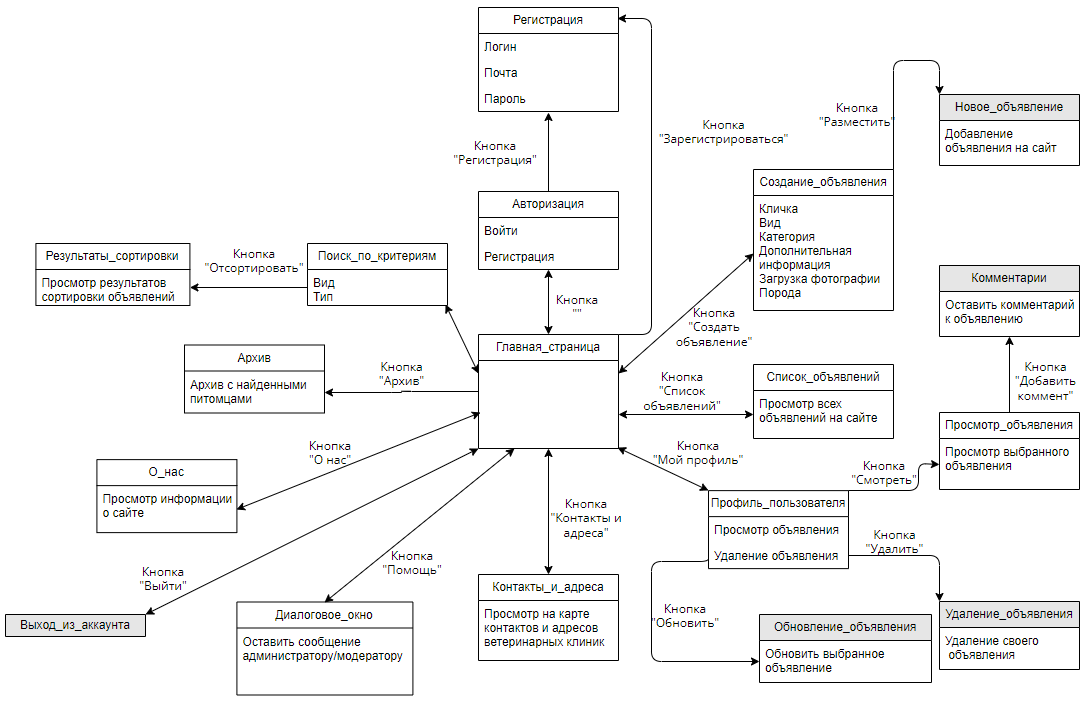
Созданный веб-сайт обладает интуитивно понятным и удобным интерфейсом. Основная цель разработки сайта "Поиск пропавших животных" была успешно достигнута, а его задачи были полностью выполнены. Результатом этой работы является удобный ресурс, который предоставляет пользователям эффективное решение их проблемы. Пользователи могут легко обращаться к сайту и использовать его функционал, чтобы найти пропавших животных и решить свои задачи по поиску домашних животных.

Список использованных источников

1. Сайт «POISKZVEREI.RU»: официальный сайт. – Красноярск, 2017. – URL: <https://poiskzverei.ru> (дата обращения: 06.04.2023).
2. Сайт «PET911.RU»: официальный сайт. – Красноярск, 2017. – URL: <https://pet911.ru> (дата обращения: 06.04.2023).
3. Сайт «FOUNDPETS.RU»: официальный сайт. – Красноярск, 2017. – URL: <https://foundpets.ru> (дата обращения: 06.04.2023).
4. Сайт социологических опросов V-project / V-project – URL: <https://v-project-online.ru/news/priyuty-dlya-zhivotnykh/> (дата обращения 13.05.2023).
5. Диаграммы вариантов использования: редакция от 5 июля 2021 // Хабр : сайт-статейник. – URL: <https://habr.com/ru/post/566218> (дата обращения: 06.04.2023).
6. The official home of the Python Programming Language // Python : официальный сайт – URL: <https://www.python.org> (дата обращения: 06.04.2023).
7. The web framework for perfectionists with deadlines | Django // Django : официальный сайт. – URL: <https://www.djangoproject.com> (дата обращения: 06.04.2023).
8. SQLite Home Page: редакция от 2 апреля 2023 // SQLite: официальный сайт. – URL: <https://www3.sqlite.org/index.html> (дата обращения: 06.04.2023).
9. Сайт библиотек Python / smtplib — SMTP protocol client – URL: <https://docs.python.org/3/library/smtplib.html> (дата обращения 10.04.2023).
10. Язык HTML: что это такое и как он работает : редакция от 19 августа 2022 // SkillBox: сайт-статейник. – URL: <https://skillbox.ru/media/code/chto_takoe_html> (дата обращения: 06.05.2023).
11. Что такое Bootstrap и зачем он нужен: редакция от 5 ноября 2021 // ИТШЕФ: сайт-статейник. – URL: <https://itchief.ru/bootstrap/introduction> (дата обращения: 06.05.2023).
12. Инструмент с открытым исходным кодом, использующий простые текстовые описания для рисования UML диаграмм / Plantuml. – URL: <https://plantuml.com/ru/> (дата обращения: 13.05.2023).

Приложение А

Диаграмма потока экранов



Приложение Б

Листинг файла base.html

<!-- Модальное окно для подтверждения выхода -->

<div class="modal fade" id="confirmationModal" tabindex="-1" aria-labelledby="confirmationModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="confirmationModalLabel">Уже нашли или помогли пристроить потеряшку и готовы на этом остановиться?</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Нет</button>

<a href="/users/logout/">

<button type="button" class="btn btn-primary">Да</button>

</a>

</div>

</div>

</div>

</div>

<!-- Скрипт для окна подтверждения -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

<script>

document.getElementById('logoutButton').addEventListener('click', function() {

var confirmationModal = new bootstrap.Modal(document.getElementById('confirmationModal'));

confirmationModal.show();

});

</script>

Приложение В

Листинг файла home.html

<h1 style="background-color:#212529;border-radius: 5px; margin-top:10px ;color :#fff;padding: 5px; width:100% ;" >Последние объявления</h1>

<div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">

{% for person in persons %}

<div class="p-5 mb-5 rounded-5 bg-overlay">

<div class="container-fluid py-1" style="opacity: 1;">

<div class="col">

<div class="card shadow-sm">

<img src="../media/{{ person.photo }}" class="bd-placeholder-img card-img-top" width="100%" height="225">

<div class="card-body">

<p class="card-text"><h1>{{ person.title }}</h1>

<h3>{{ person.category }}</h3>

{{ person.content }}<h5 >пользователь : id{{ person.id\_user }}</h5></p>

<div class="d-flex justify-content-between align-items-center">

<div class="btn-group">

<a href="/users/get\_post/{{ person.id }}"><button type="button" class="btn btn-sm btn-outline-secondary">смотреть</button></a>

</div>

<small class="text-body-secondary">{{ person.time\_create }}</small>

</div>

</div>

</div>

</div>

</div>

</div>

{% endfor %}

</div>

Приложение г

Листинг файла models.py

#Модель объявления

class posts(models.Model):

id\_user = models.IntegerField()

title = models.CharField(max\_length=255, verbose\_name="Заголовок")

slug = models.SlugField(max\_length=255, unique=True, db\_index=True, verbose\_name="URL")

additional\_information = models.TextField(blank=True, verbose\_name="Текст статьи")

photo = models.ImageField(upload\_to="photos/%Y/%m/%d/", verbose\_name="Фото")

time\_create = models.DateTimeField(auto\_now\_add=True, verbose\_name="Время создания")

time\_update = models.DateTimeField(auto\_now=True, verbose\_name="Время изменения")

category = models.ForeignKey('Category', on\_delete=models.PROTECT, verbose\_name="Категории")

type\_of\_animal = models.ForeignKey('type\_of\_animal', on\_delete=models.PROTECT, verbose\_name="вид")

status = models.ForeignKey('status', on\_delete=models.PROTECT, verbose\_name="status")

def \_\_str\_\_(self):

return self.title

def get\_absolute\_url(self):

return reverse('post', kwargs={'post\_slug': self.slug})

class Meta:

verbose\_name = 'pet'

verbose\_name\_plural = 'pet'

ordering = ['id']

Приложение д

Листинг файла views.py

@login\_required(login\_url='/users/login')

def delete\_post(request, id):

# Получаем URL страницы, с которой пришел запрос

previous\_page = request.META.get('HTTP\_REFERER')

try:

petp = pet.objects.get(id=id)

petp.delete()

if previous\_page:

return redirect(previous\_page) # Перенаправляем пользователя на предыдущую страницу

except Exception as e:

print(e)

pass

# Если предыдущая страница не определена, перенаправляем пользователя на страницу профиля

return redirect('profile')

def archive\_old\_posts():

# Вычисляем время месяц назад от текущего времени

month\_ago = timezone.now() - timedelta(days=30)

# Получаем все записи, у которых время обновления больше месяца назад

old\_posts = posts.objects.filter(time\_update\_\_lt=month\_ago)

# Обновляем статус на 'архивный' для всех найденных записей

status\_id = status.objects.get(name='архив').id

old\_posts.update(status\_id=status\_id)

Приложение Е

Листинг файла contact.html

{% extends 'base.html' %}

{% block name %}Контакты{% endblock %}

{% block content %}

{% load static %}

<link rel="stylesheet" href="{% static 'css/custom-styles.css' %}">

<div class="bg-overlay">

<div class="container-fluid py-5" >

<h1 class="display-5 fw-bold text-center">Контакты и адреса ветеринарных клиник в городе Красноярск</h1>

<div id="map" style="width: 100%; height: 20px;"></div>

<script src="https://api-maps.yandex.ru/2.1/?apikey=ae232ec8-e420-4b57-bedf-dde5537783fc&lang=ru\_RU" type="text/javascript"></script>

<script type="text/javascript" charset="utf-8" async src="https://api-maps.yandex.ru/services/constructor/1.0/js/?um=constructor%3Ae11 ba3e9062c50ed3f185e40c6737cd1075c4a7eb4a04adc0eb9f692f775ad78&amp;width=100%&amp;height=400&amp;lang=ru\_RU&amp;scroll=true"></script>

</div>

</div>

{% endblock %}