

МИНОБРНАУКИ РОССИИ ФЕДЕРАЛЬНОЕ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ» (ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра информационных систем и технологий

Сайт учёта статистики погодных наблюдений «Weather  
Statistics»

Курсовая работа по дисциплине  
«Технологии программирования»

09.03.02 Информационные системы и технологии  
Информационные системы телекоммуникаций

Преподаватель \_\_\_\_\_ В.С. Тарасов, ст. преподаватель \_\_\_\_20\_\_

Обучающийся \_\_\_\_\_ Е.Б. Тимошинов, 3 курс, 2 группа, о/о

Обучающийся \_\_\_\_\_ А.Г. Самсонова, 3 курс, 2 группа, о/о

Обучающийся \_\_\_\_\_ С.В. Долгих, 3 курс, 2 группа, о/о

Воронеж 2023

## Содержание

Содержание.....	2
Введение.....	4
1 Постановка задачи.....	5
1.1 Требования к разрабатываемой системе .....	5
1.1.1 Функциональные требования .....	5
1.1.2 Нефункциональные требования .....	5
1.2 Требования к архитектуре.....	6
1.3 Задачи, решаемые в процессе разработки .....	6
2 Анализ предметной области .....	8
2.1 Терминология (гlossарий) предметной области .....	8
2.2 Обзор аналогов.....	9
2.2.1 Яндекс. Погода.....	9
2.2.2 GISMETEO .....	10
2.2.3 World Weather.....	11
2.3 Диаграммы, иллюстрирующие работу системы.....	12
2.3.1 Диаграмма прецедентов (Use case) .....	12
2.3.2 Диаграмма последовательности (Sequence diagram).....	14
2.3.3 Диаграмма состояний (Statechart diagram) .....	15
2.3.4 Диаграмма активностей (Activity diagram) .....	16
2.3.5 Диаграмма сотрудничества (Collaboration diagram).....	17
2.3.6 Диаграмма развёртывания (Deployment diagram).....	19
2.3.7 Диаграмма IDEF0.....	20
3 Реализация.....	22
3.1 Средства реализации.....	22

3.1.1 Средства реализации серверной части приложения .....	22
3.1.2 Средства реализации клиентской части приложения .....	22
3.2 Реализация серверной части приложения (backend) .....	23
3.2.1 Схема базы данных .....	23
3.2.2 Архитектура серверной части приложения .....	26
3.3 Реализация клиентской части приложения (frontend).....	27
3.4 Графический интерфейс .....	27
4 Статистика использования приложения .....	33
5 Тестирование .....	35
5.1 Дымовое тестирование .....	35
5.2 UI тестирование.....	36
5.3 Юзабилити тестирование .....	38
Заключение .....	40
Список использованной литературы.....	41

## **Введение**

В настоящее время интернет позволяет нам получать доступ к большому количеству информации, не выходя из дома. В этом контексте сайты для рассмотрения актуальной погоды играют важную роль, предоставляя пользователям возможность ознакомиться с ожидаемыми в ближайшее время погодными условиями.

Удобство таких сайтов состоит в том, что пользователи могут узнать, какая на улице погода, не выглядывая в окно. Узнав, сколько на улице градусов, идёт ли дождь, всегда можно подобрать наиболее подходящий комплект одежды.

В данной курсовой работе мы рассмотрим создание сайта учёта статистики погодных наблюдений «Weather Statistics», который будет предоставлять пользователям подробную информацию не только об актуальной погоде, но и о погоде прошлых лет.

## **1 Постановка задачи**

Данный проект предназначен для просмотра статистических данных об актуальной погоде, а также о погоде прошлых лет.

Целью данного проекта является разработка сайта учёта статистики погодных наблюдений.

### **1.1 Требования к разрабатываемой системе**

#### **1.1.1 Функциональные требования**

К разрабатываемому приложению выдвигаются следующие функциональные требования для пользователя:

- Возможность просмотра усреднённых данных о погоде в указанный день или месяц.
- Ввод интересующих погодных условий и получение списка ближайших дней, когда возможна такая погода, опираясь на статистику погоды прошлых лет.
- Возможность расписать список дел на неделю, с учётом погодных условий.
- Возможность скачивания таблицы с данными о погоде за указанный день/неделю/месяц

К разрабатываемому приложению выдвигаются следующие функциональные требования для администратора:

- Добавлять, удалять и редактировать данные о погодных условиях.

#### **1.1.2 Нефункциональные требования**

К разрабатываемому приложению выдвигаются следующие нефункциональные требования:

- Интерфейс сайта должен быть удобным и интуитивно понятным для пользователей различных возрастных групп.
- Сайт должен работать быстро и отвечать на запросы пользователей в течение нескольких секунд.
- Приложение должно использовать современные технологии и инструменты разработки, обеспечивающие высокую производительность и стабильность работы сайта.

## **1.2 Требования к архитектуре**

Список требований к архитектуре:

- Приложение должно быть построено на клиент-серверной архитектуре с использованием протоколов HTTP/HTTPS.
- Для хранения информации необходимо использовать реляционную базу данных, обеспечивающую высокую производительность и надежность.
- Клиентская часть приложения должна быть написана с использованием современных технологий front-end разработки, таких как HTML и CSS.
- Серверная часть приложения должна быть написана с использованием современных технологий back-end разработки, таких как фреймворк Spring и система управления базами данных MySQL.

## **1.3 Задачи, решаемые в процессе разработки**

В процессе разработки сайта учёта статистики погодных наблюдений будут решаться следующие задачи:

- Анализ предметной области: необходимо изучить особенности работы и требования пользователей к сайтам просмотра погодных условий.
- Проектирование базы данных: учитывая полученные требования, необходимо разработать структуру базы данных, которая будет использоваться при просмотре сайта.
- Разработка серверной части приложения: на этом этапе необходимо разработать серверную часть приложения, которая будет отвечать за обработку запросов клиента и взаимодействие с базой данных. Для этого используется фреймворк Spring.
- Разработка клиентской части приложения: клиентская часть приложения должна быть написана с использованием современных технологий front-end разработки, таких как HTML и CSS. Здесь необходимо разработать интерфейс пользователя, который будет удобен и понятен для любого пользователя.
- Тестирование и отладка: на этом этапе производится тестирование и отладка работы сайта для соотнесения с требованиями, определёнными в начале проекта.

## **2 Анализ предметной области**

### **2.1 Терминология (гlossарий) предметной области**

**Веб-приложение** — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

**Клиент (клиентская сторона)** – сайт, который предоставляет пользователю взаимодействовать со всей системой.

**Сервер (серверная часть)** – компьютер, обслуживающий другие устройства (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.

**Фреймворк** – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. [1]

**Backend** – логика работы сайта, внутренняя часть продукта, которая находится на сервере и скрыта от пользователя.

**Frontend** – презентационная часть информационной или программной системы, ее пользовательский интерфейс и связанные с ним компоненты.

**GitHub** – веб-сервис для хостинга IT-проектов и их совместной разработки.

**Авторизированный пользователь** — пользователь, прошедший авторизацию в системе.

**Неавторизированный пользователь** — пользователь, не прошедший авторизацию или не зарегистрированный в системе.



## 2.2 Обзор аналогов

Существует огромное количество сайтов для просмотра актуальных погодных условий, которые имеют свои преимущества и недостатки. Наиболее популярными и понятными являются «Яндекс. Погода», «GISMETEО» и «World Weather», особенности которых необходимо рассмотреть более подробно.

### 2.2.1 Яндекс. Погода

«Яндекс. Погода» — одна из наиболее популярных и точных платформ для прогнозирования погоды в России. [2]

Интерфейс приложения представлен на Рисунке 1.

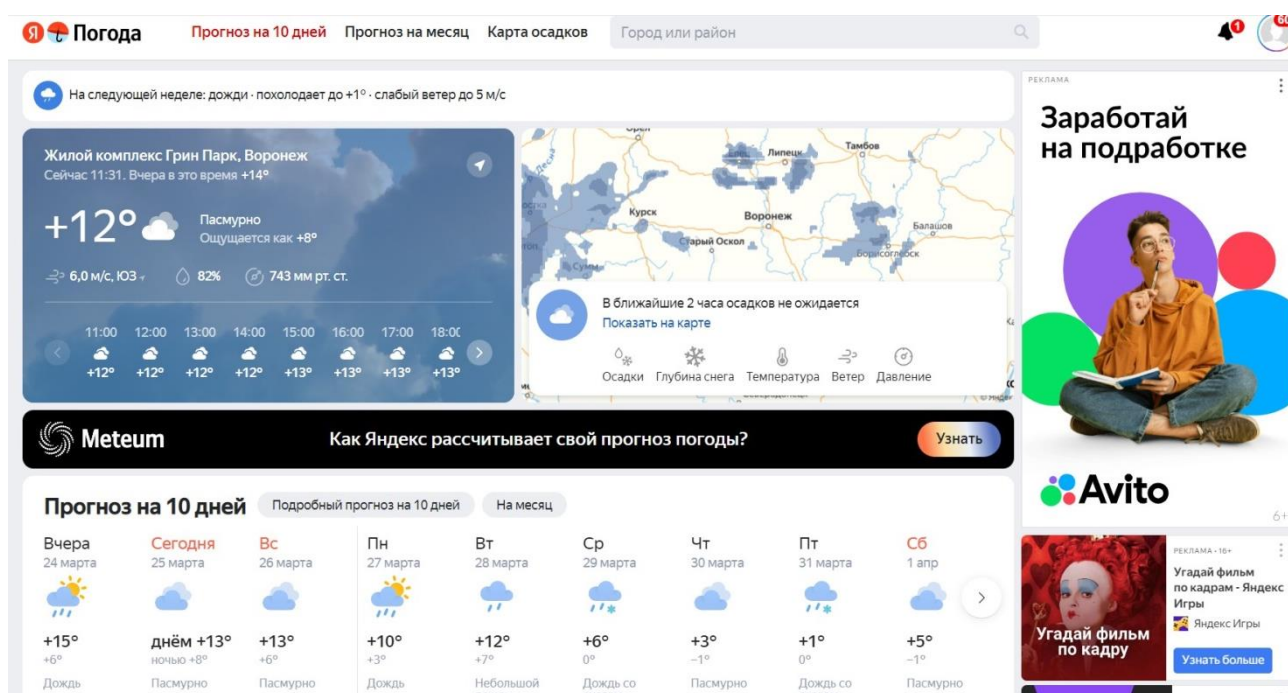


Рисунок 1 - Интерфейс сайта «Яндекс.Погода»

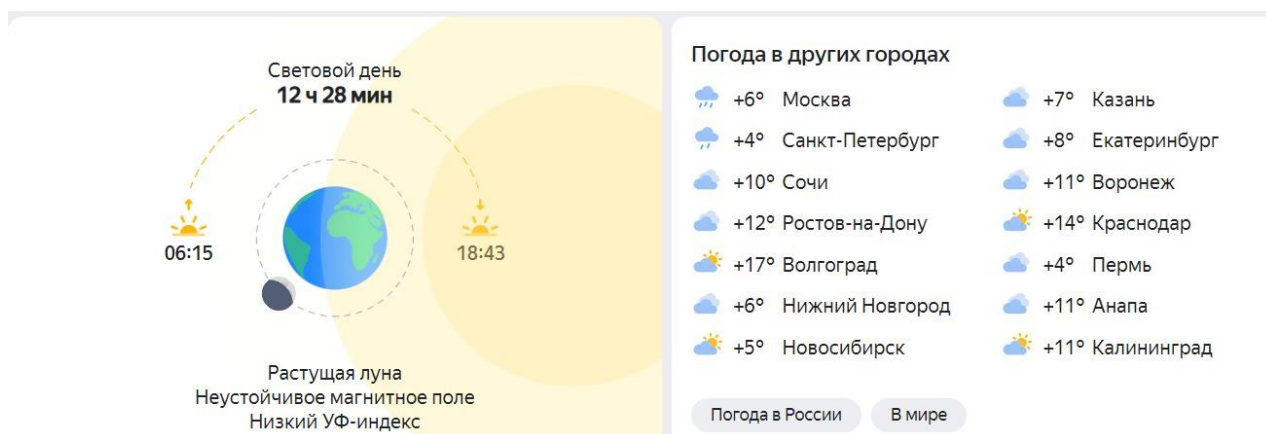


Рисунок 2 - Продолжение видимого экрана сайта «Яндекс.Погода»

Яндекс.Погода обладает следующим рядом преимуществ:

- Наглядное представление погодных условий (осадков) прямо на карте.
- Есть возможность посмотреть погоду не только в регионах России, но и во всем мире.
- Сайт может с точностью определить местонахождение пользователя (не город, а район или жилой комплекс) и предоставить информацию о погодных условиях в этом месте.

И в свою очередь следующим рядом недостатков:

- На сайте представлено большое количество рекламы.
- Нет возможности построить статистику на основе прошлых лет.

### 2.2.2 GISMETEO

GISMETEO — это метеорологический ресурс, которым ежедневно пользуются миллионы людей по всему миру. При помощи сайта можно узнать текущую погоду, а также получить прогноз температуры, осадков и других необходимых параметров в любой точке Земли. [3]

Интерфейс приложения представлен на Рисунке 3.

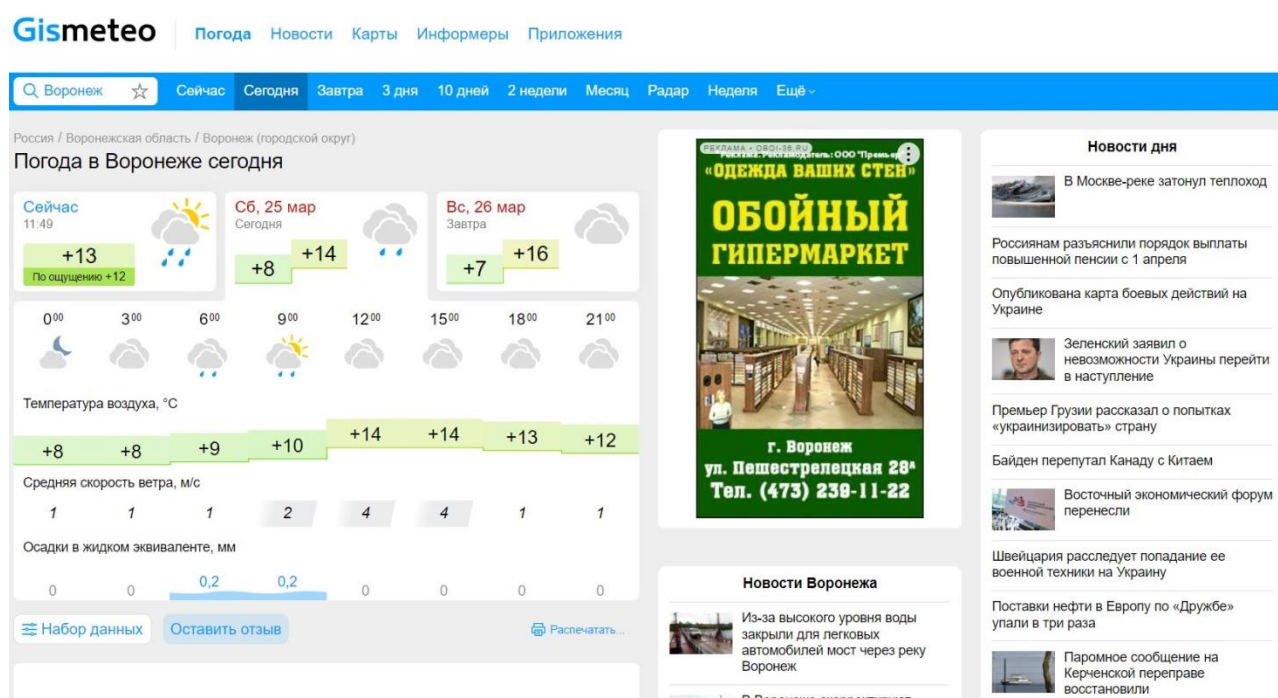


Рисунок 3 - Интерфейс сайта «GISMETEO»

GISMETEO обладает следующим рядом преимуществ:

- Возможен выбор данных, которые будут отражены в прогнозе.
- Возможно распечатать таблицу с актуальной погодой.
- Присутствует раздел с новостями о погоде в разных регионах.

И в свою очередь следующим рядом недостатков:

- Информация на этом сайте бывает неточной.
- Большое количество рекламы.

### 2.2.3 World Weather

World Weather – это уникальный Интернет-ресурс, предоставляющий своим пользователям погодные данные в удобной для чтения и анализа форме. [4]

Интерфейс приложения представлен на Рисунке 4.

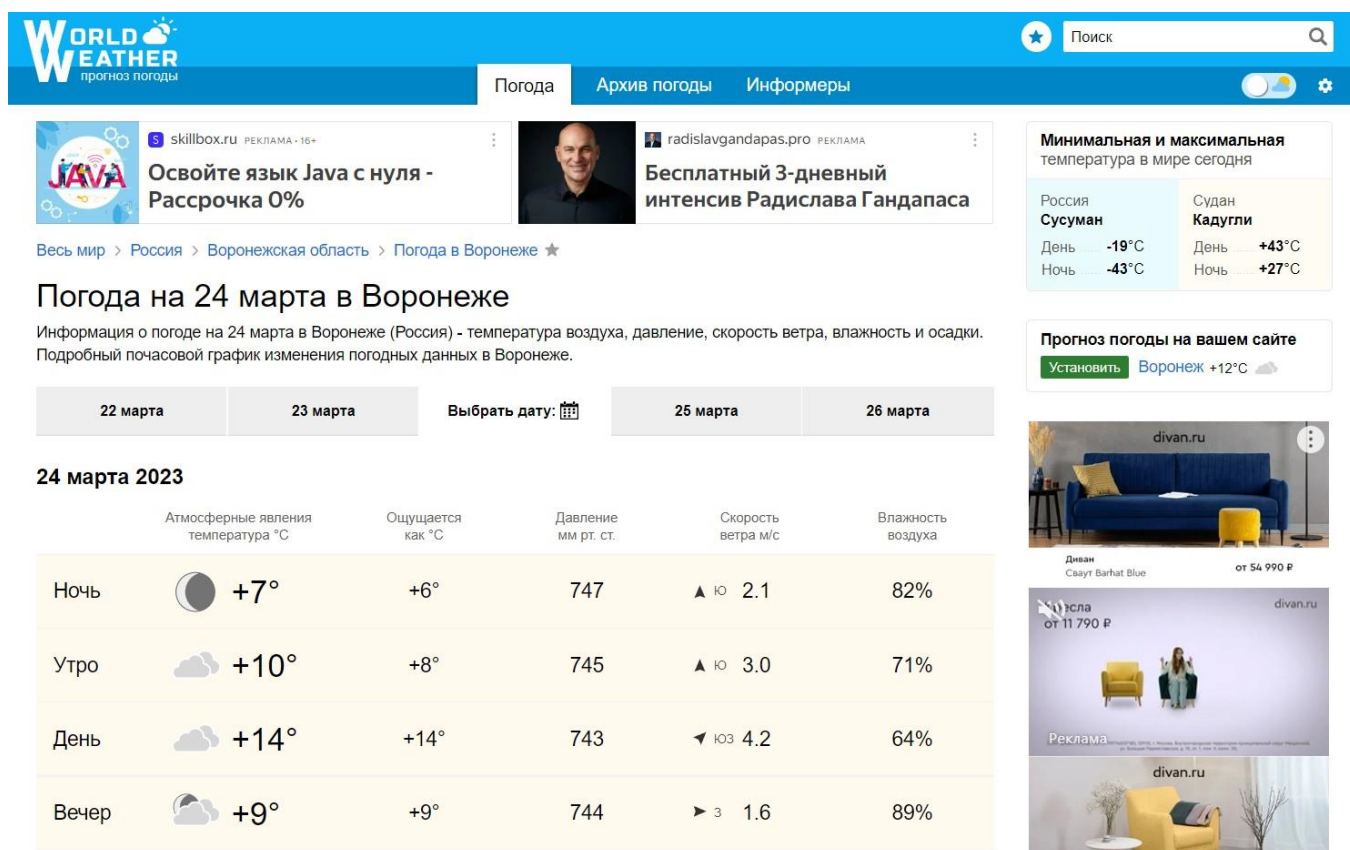


Рисунок 4 - Интерфейс сайта «World Weather»

World Weather обладает следующим рядом преимуществ:

- Возможен просмотр прогноза погоды на любую прошедшую дату.
- Существует биометрический прогноз с рекомендациями по слежению за здоровьем.
- Представлены графики с почасовым изменением температуры и направления ветра.

И в свою очередь следующим рядом недостатков:

- Не указана длительность светового дня.

## 2.3 Диаграммы, иллюстрирующие работу системы

### 2.3.1 Диаграмма прецедентов (Use case)

Диаграмма прецедентов (Use case) в общем виде представлена на Рисунке 5. В данной системе существует две возможные роли: пользователь и администратор.

Пользователь может:

- Просматривать погоду.
- Планировать дела в соответствии с прогнозом погоды.
- Вводить данные погоды для поиска ближайших дат с указанными показателями.
- Настраивать главную страницу.

Администратор может:

- Авторизоваться.
- Входить в свой профиль.
- Настраивать/изменять данные в БД.

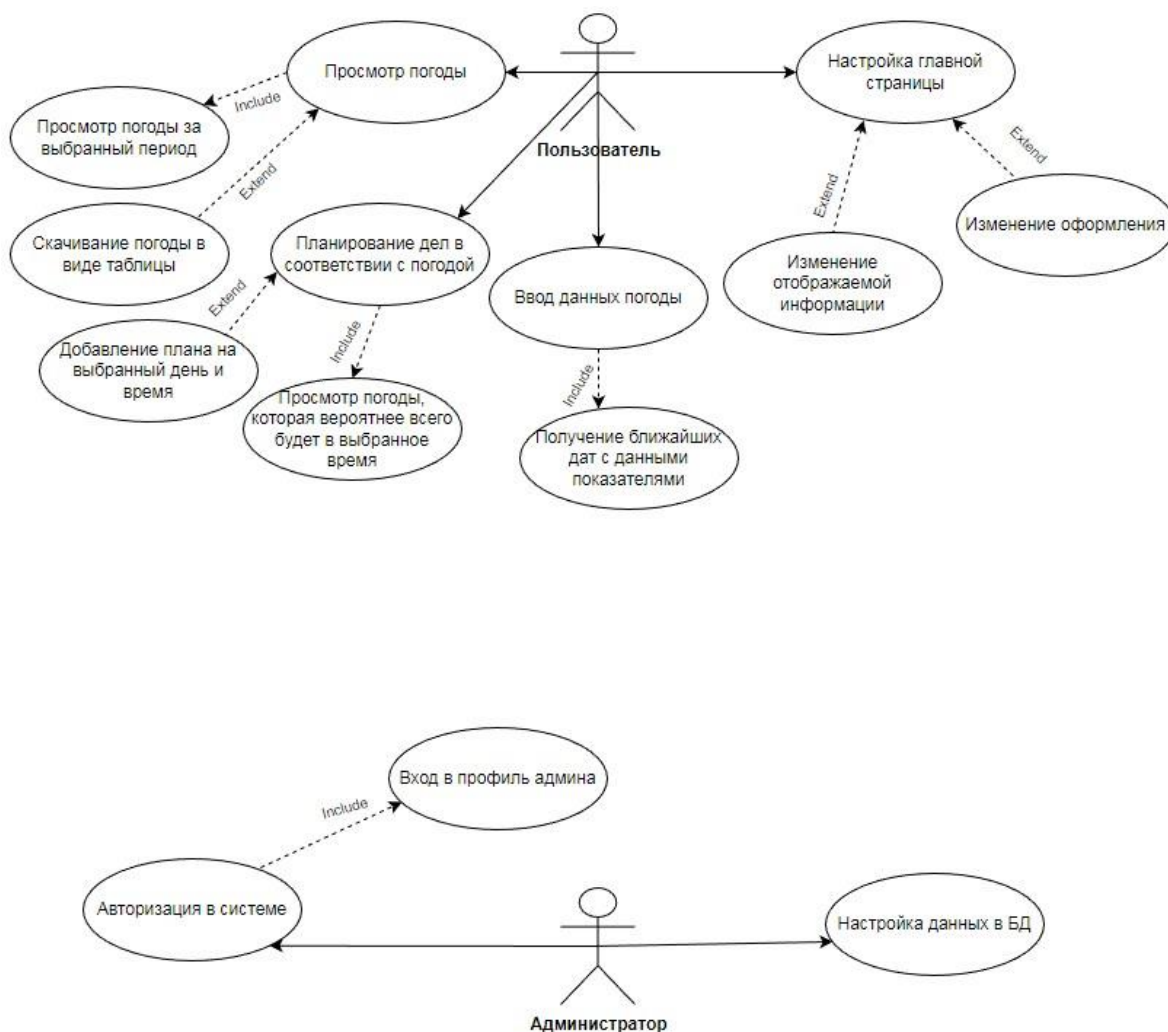


Рисунок 5 - Диаграмма прецедентов (Use case) в общем виде

### 2.3.2 Диаграмма последовательности (Sequence diagram)

Существует также диаграмма последовательностей (Рисунок 6), на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта и взаимодействие актеров информационной системы в рамках прецедента.



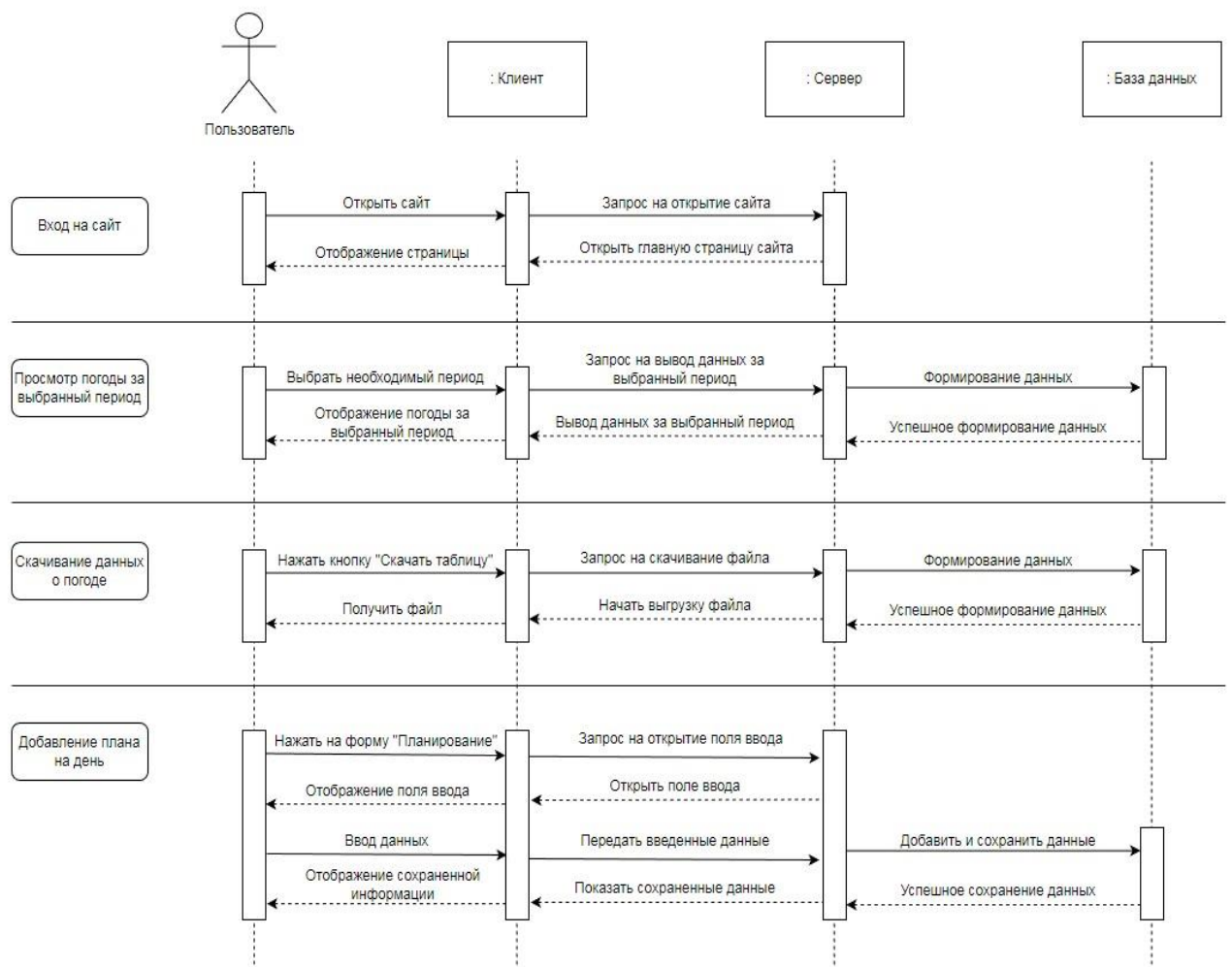


Рисунок 6 - Диаграмма последовательности

### 2.3.3 Диаграмма состояний (Statechart diagram)

Диаграмма состояний (Рисунок 7) отражает внутренние состояния объекта в течение его жизненного цикла от момента создания до разрушения. На данной диаграмме рассмотрены состояния от момента входа в систему до полного выхода из нее.

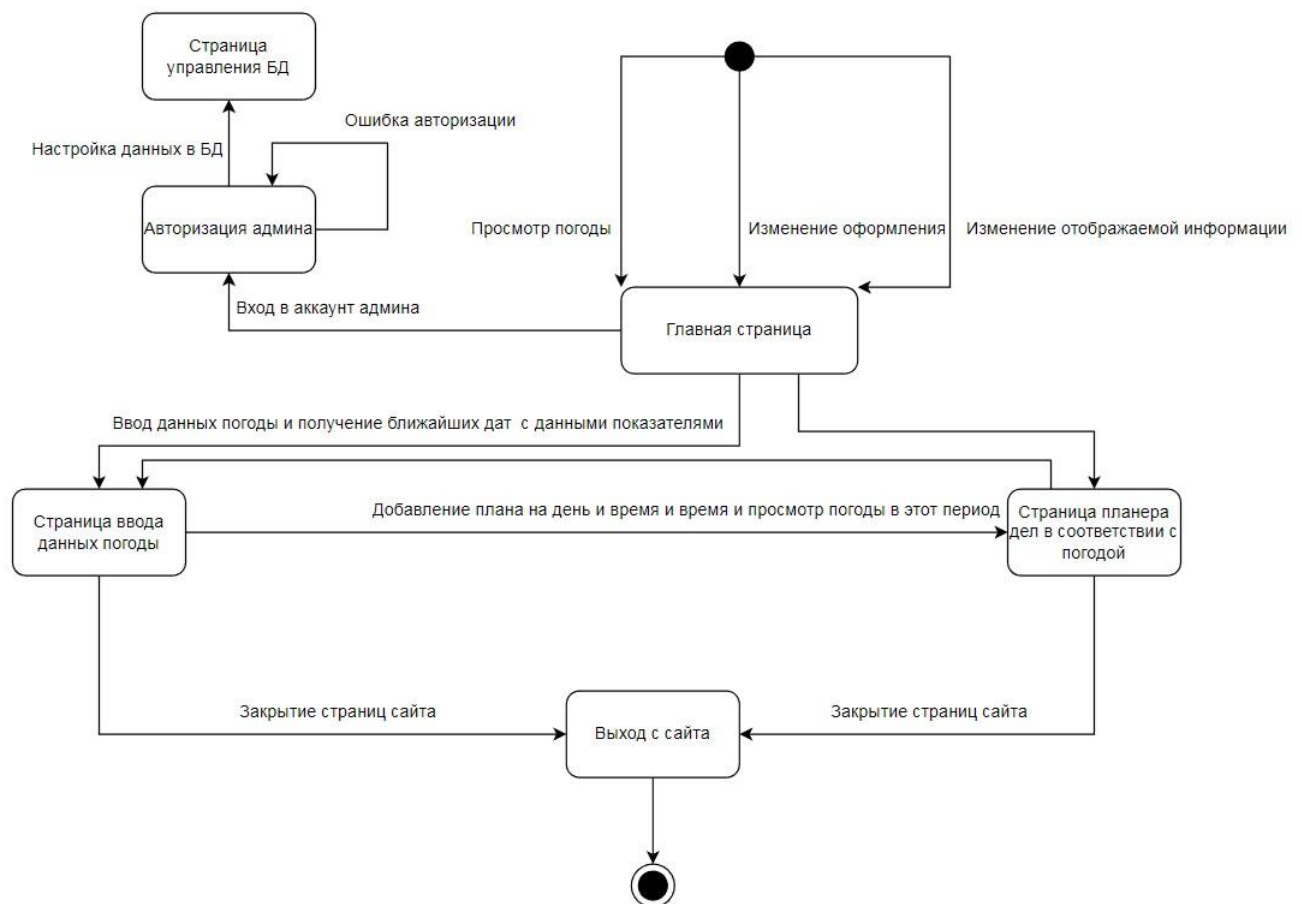


Рисунок 7 - Диаграмма состояний

### 2.3.4 Диаграмма активностей (Activity diagram)

Диаграмма активности (Рисунок 8) представляет собой диаграмму, на которой показаны действия, состояния которых описаны на диаграмме состояний. Она описывает действия системы или людей, выполняющих действия, и последовательный поток этих действий.

В данном случае рассмотрен путь действий пользователя.

Диаграмма показывает, что пользователь, находясь в неавторизованной зоне системы не может заходить на свой профиль, добавлять товары в избранное и комментировать продукцию.



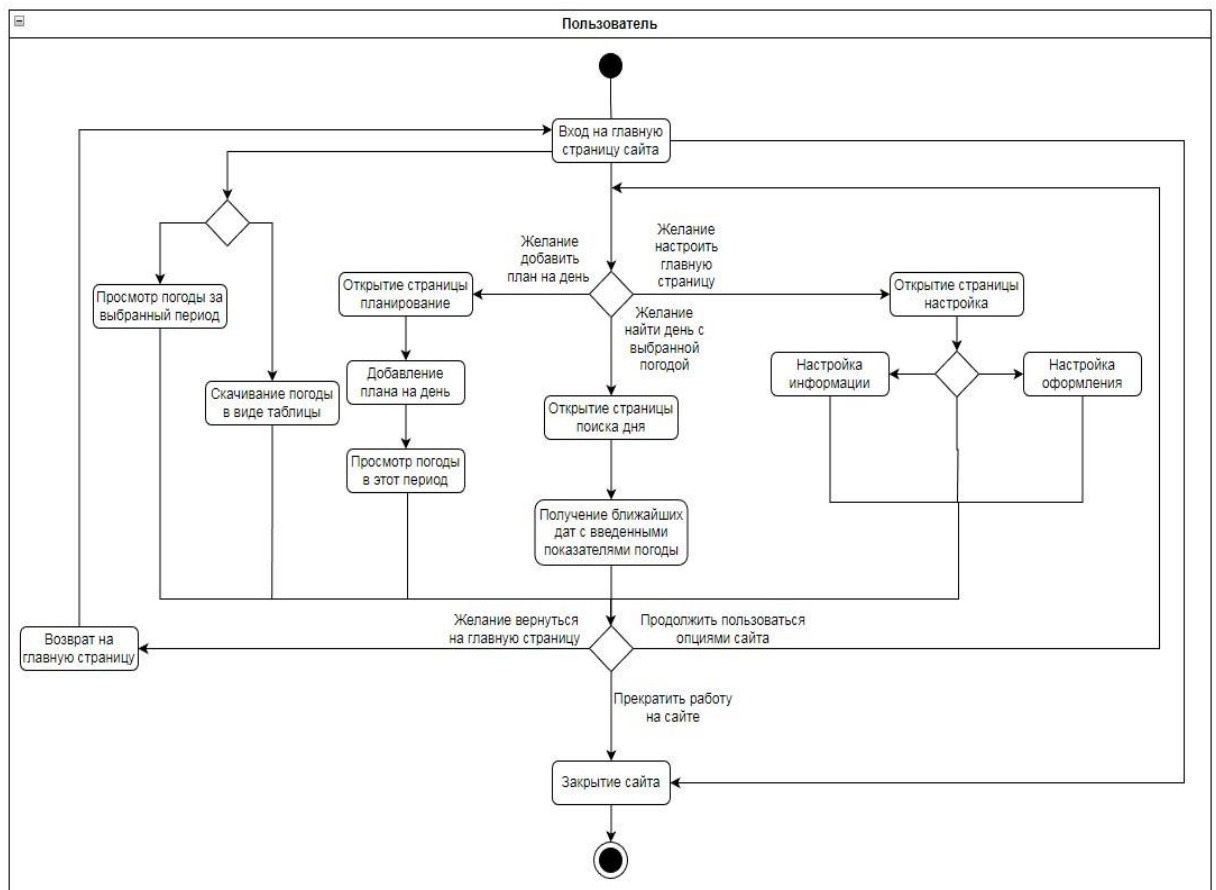


Рисунок 8 - Диаграмма активности

### 2.3.5 Диаграмма сотрудничества (Collaboration diagram)

Диаграмма сотрудничества (Рисунки 9-11 — это вид диаграммы взаимодействия, в котором основное внимание сосредоточено на структуре взаимосвязей объектов, принимающих и отправляющих сообщения.

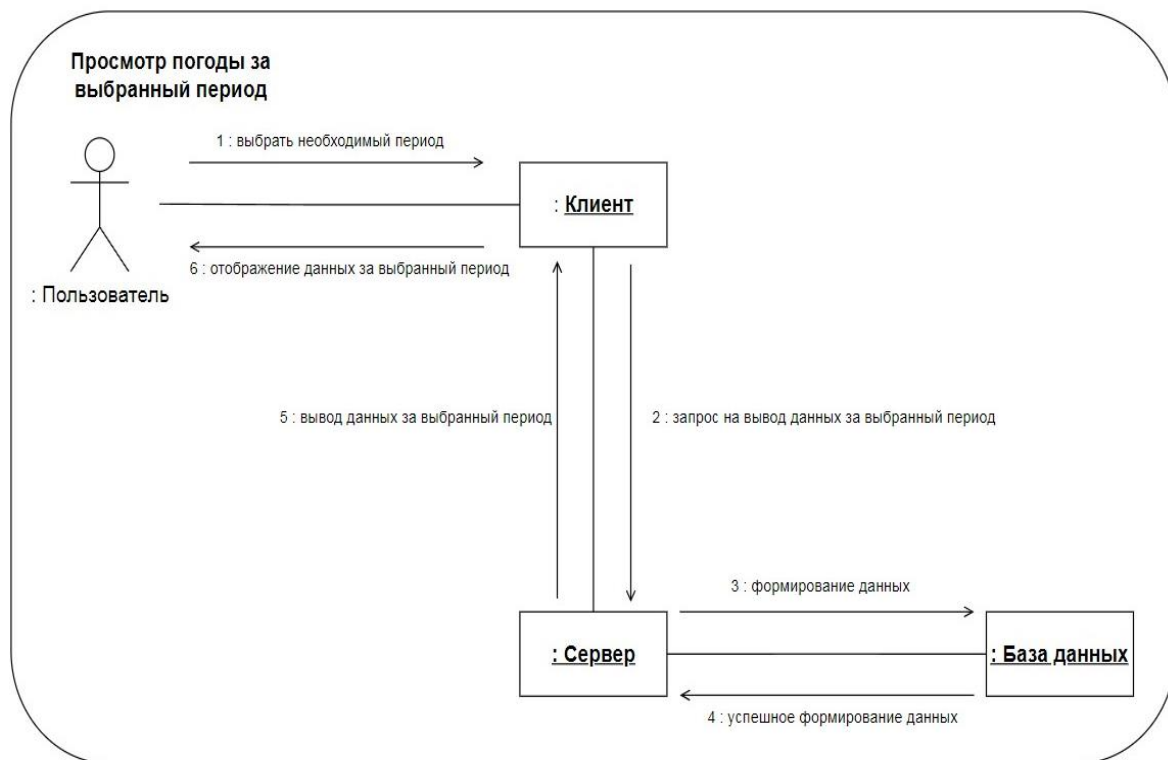


Рисунок 9 - Диаграмма сотрудничества при просмотре погоды за выбранный период

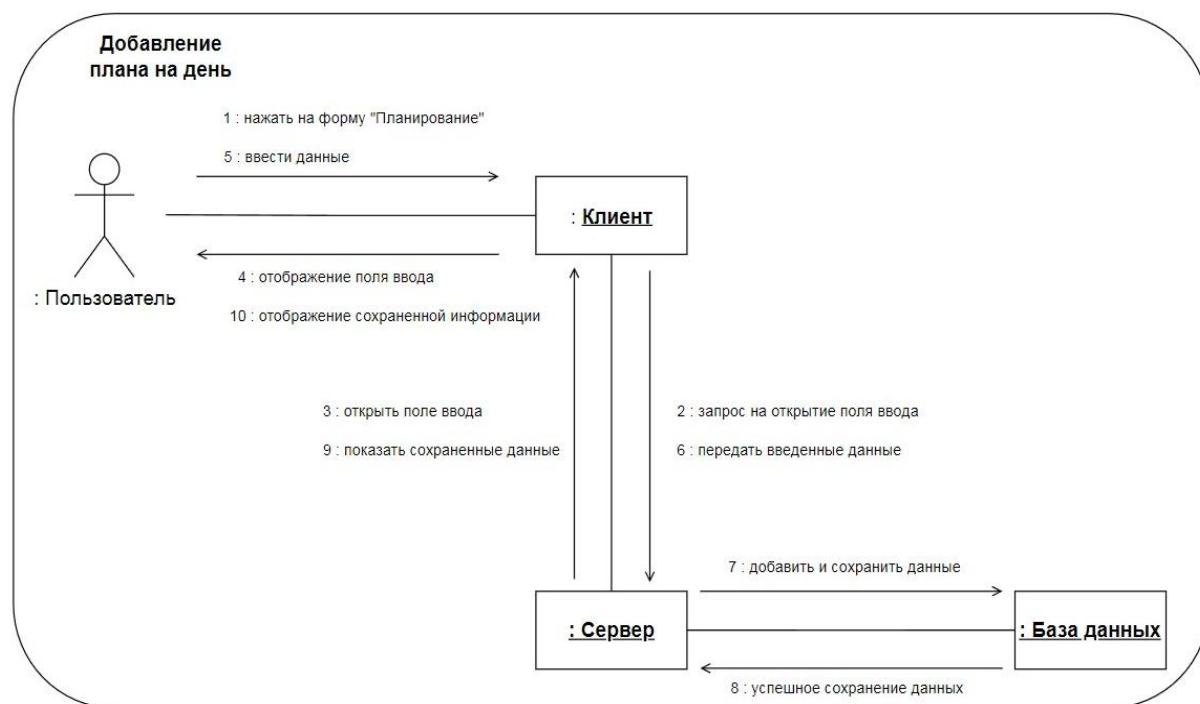


Рисунок 10 - Диаграмма сотрудничества при планировании дня

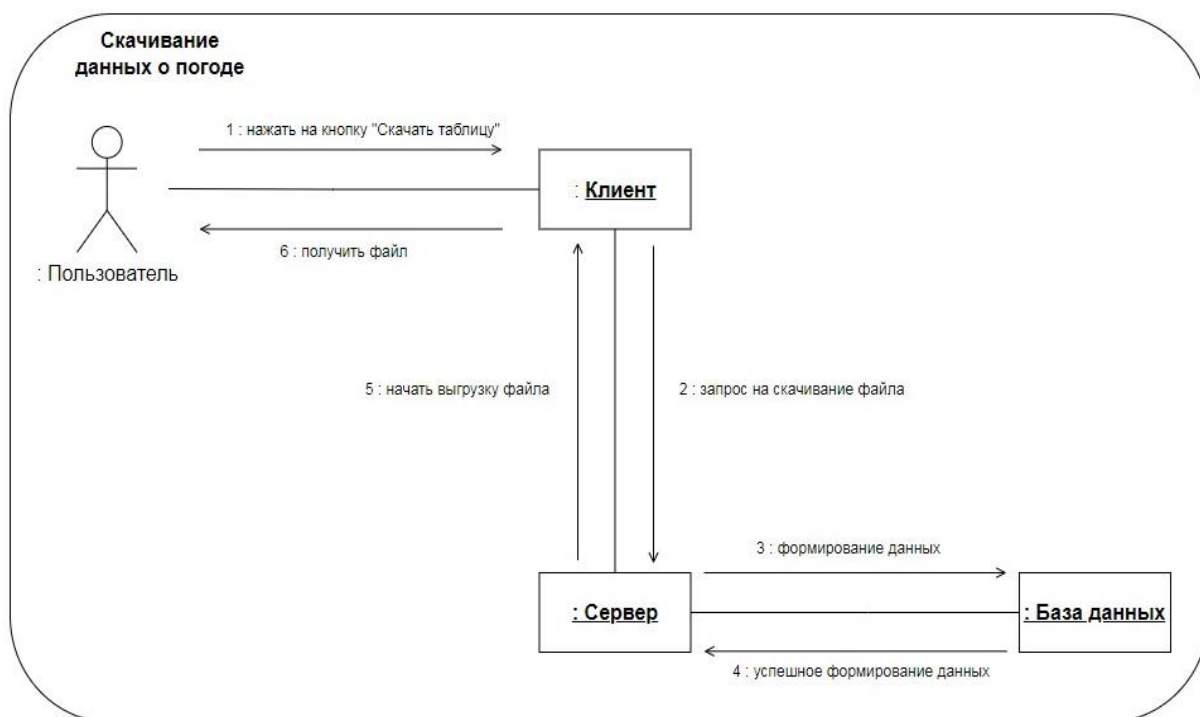


Рисунок 11 - Диаграмма сотрудничества при скачивании таблиц с данными о погоде

### 2.3.6 Диаграмма развертывания (Deployment diagram)

Диаграмма развертывания (Рисунок 12) предназначена для представления общей конфигурации или топологии распределенной программной системы.

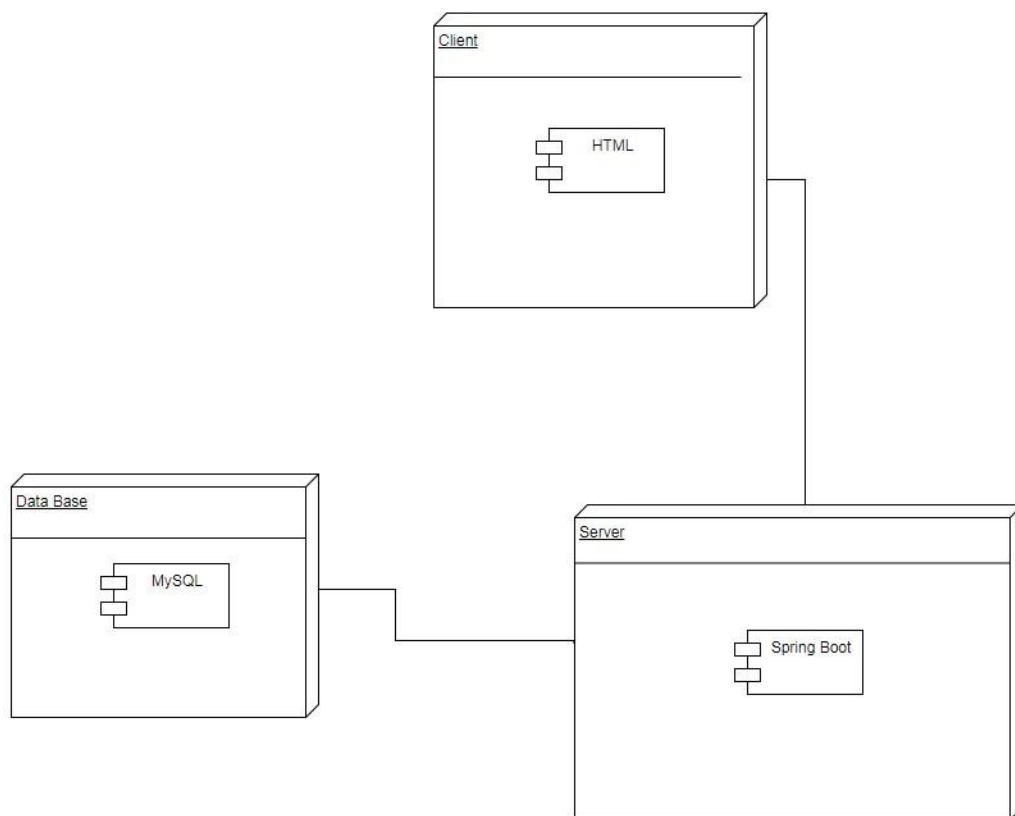


Рисунок 12 - Диаграмма развёртывания

### 2.3.7 Диаграмма IDEF0

IDEF0 используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальные объекты, связывающие эти функции.

На рисунке 13 представлена контекстная диаграмма системы. На вход системе поступает пользователь. Работу системы регулируют данные о погоде прошлых лет. Как ресурсы, необходимые для работы системы, в неё поступают администратор и сайт. На выходе системы мы имеем удовлетворённого пользователя.

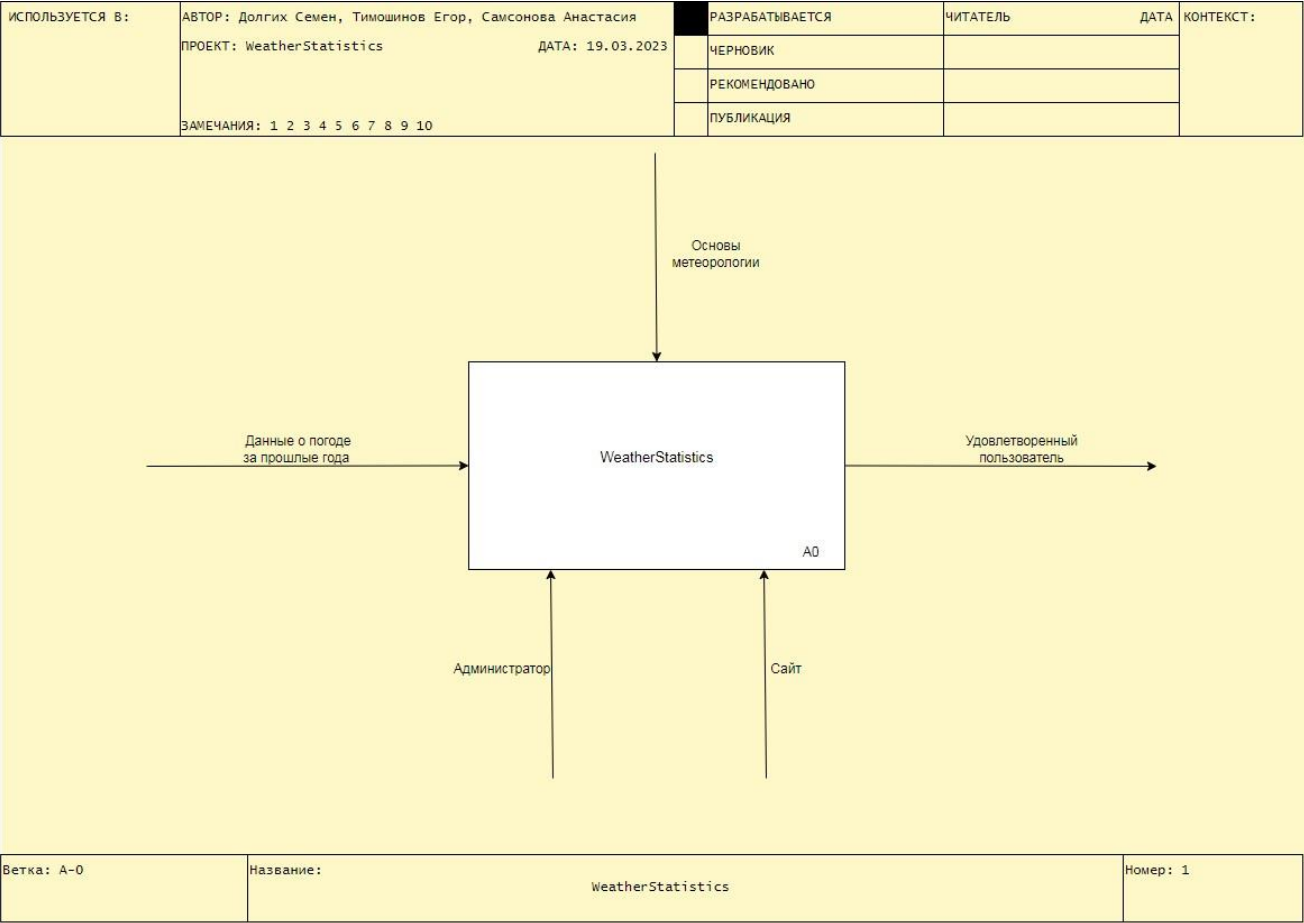


Рисунок 13 - Диаграмма IDEF0

## **3 Реализация**

### **3.1 Средства реализации**

#### **3.1.1 Средства реализации серверной части приложения**

Для разработки серверной (backend) части приложения был выбран следующий стек технологий:

- Фреймворк Spring - универсальный фреймворк с открытым исходным кодом для Java-платформы. Здесь используется возможность разработки сайта как набора слабосвязанных (loose-coupled) компонентов. Таким образом, администратор может управлять транзакциями независимо от основной логики взаимодействия с базой данных. [5]
- MySQL - свободная реляционная система управления базами данных. В основном MySQL используется для работы со средними и небольшими проектами. Она имеет открытый исходный код, поэтому любой разработчик может с ним ознакомиться. [6]
- Swagger - инструмент для документирования и тестирования API. Он позволяет создавать интерактивную документацию для вебсервисов, что упрощает их использование и интеграцию. Swagger автоматически генерирует документацию на основе аннотаций и комментариев в коде, что позволяет разработчикам сосредоточиться на написании логики приложения, а не на создании и поддержке документации. Благодаря Swagger, разработчики могут изучить доступные параметры, модели данных и примеры запросов и ответов.

#### **3.1.2 Средства реализации клиентской части приложения**

Для разработки клиентской (frontend) части приложения был выбран следующий стек технологий:

- HTML (HyperText Markup Language) — это язык разметки, используемый для создания веб-страниц. С помощью HTML разработчики определяют структуру и содержание веб-страниц, позволяя браузеру правильно интерпретировать и отображать контент для пользователей. HTML использует теги для форматирования текста, вставки изображений, оформления списков, аудио и видео, ссылок и многого другого.
- CSS (Cascading Style Sheets) — это язык стилей, используемый для задания внешнего вида веб-страниц. CSS позволяет разработчикам отделить визуальное представление веб-сайта от содержания, тем самым обеспечивая большую гибкость и управляемость веб-ресурсов. С помощью CSS можно задавать шрифты, цвета, расположение, размеры элементов, оформление фона и др. CSS является основным инструментом для создания визуального дизайна веб-сайтов и позволяет создавать уникальные дизайны и согласовывать внешний вид контента на всем сайте.

## **3.2 Реализация серверной части приложения (backend)**

### **3.2.1 Схема базы данных**

На Рисунке 14 представлена ER - диаграмма базы данных веб-приложения.

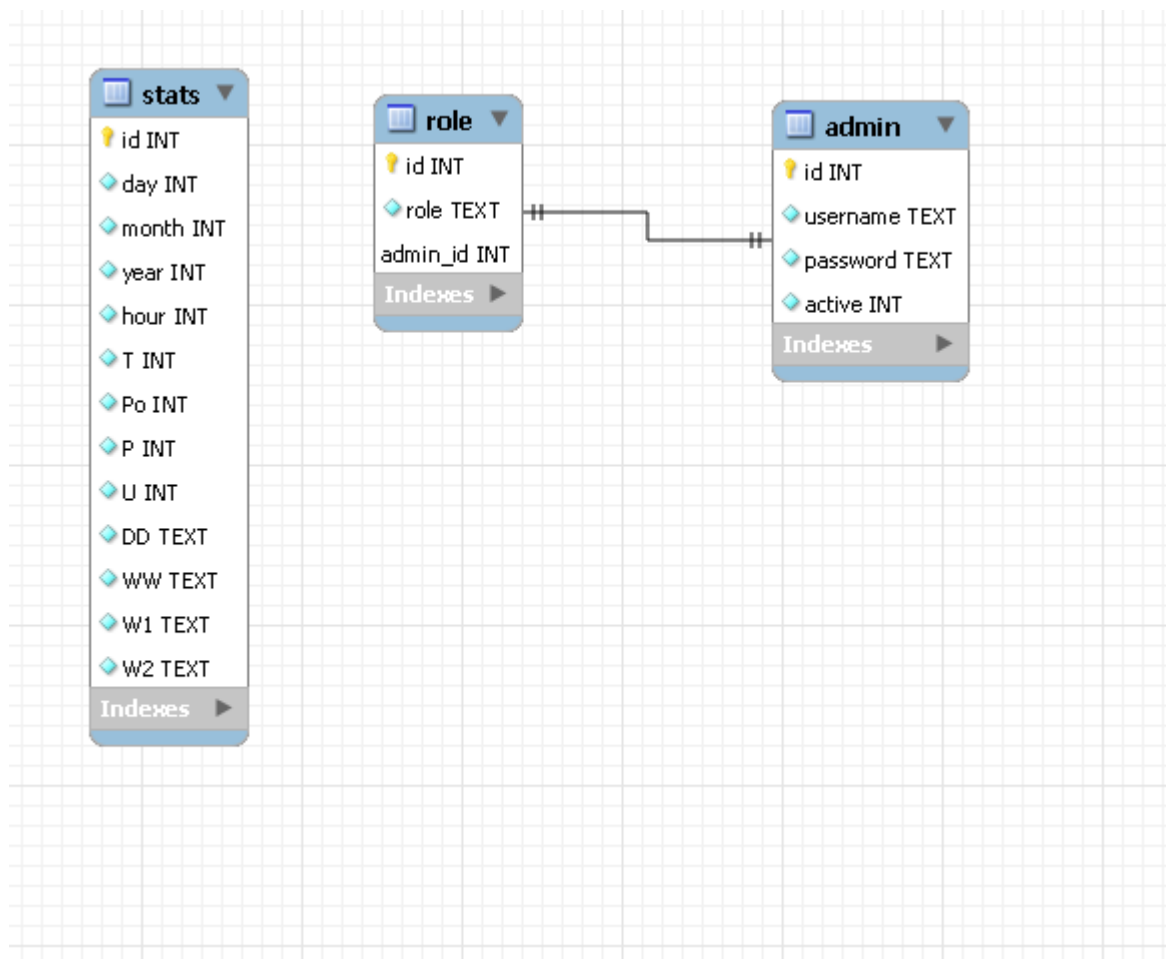


Рисунок 14 - ER-диаграмма базы данных веб приложения

В базе данных представлены 3 сущности:

- stats
- role
- admin

В таблице stats хранятся данные о погодных условиях с февраля 2005 года по март 2023 года.

Таблица имеет следующие атрибуты:

- “id”. Главный ключ. Домен: целые числа. Уникальный id каждой записи в таблице. Назначается автоматически при помощи СУБД.



- “day”. Домен: целые числа. Номер дня, в который была сделана запись.
- “month”. Домен: целые числа. Номер месяца, в который была сделана запись.
- “hour”. Домен: целые числа. Час, в который была сделана запись.
- “year”. Домен: целые числа. Год, в который была сделана запись.
- “T”. Домен: целые числа. Температура.
- “Po”. Домен: целые числа. Атмосферное давление на уровне станции.
- “P”. Домен: целые числа. Атмосферное давление, приведенное к среднему уровню моря.
- “U”. Домен: целые числа. Относительная влажность.
- “DD”. Домен: строки. Направление ветра.
- “WW”. Домен: строки. Погода наблюдаемая в начале наблюдения.
- “W1”. Домен: строки. Погода наблюдаемая в середине наблюдения.
- “W2”. Домен: строки. Погода наблюдаемая в конце наблюдения

Таблица “admin” содержит информацию об администраторах сайта.

Таблица имеет следующие атрибуты:

- “id”. Домен: целые числа. Уникальный id каждой записи в таблице.  
Назначается автоматически при помощи СУБД.
- “username”. Домен: строки. Логин администратора.
- “password”. Домен: строки. Пароль администратора.
- “active”. Домен: целые числа. Статус активности админа.

Таблица “role” содержит информацию о ролях администраторов.

Таблица имеет следующие атрибуты:

- “id”. Домен: целые числа. Уникальный id каждого администратора из таблицы “admin”.
- “role”. Домен: строки. Тип роли администратора.

### **3.2.2 Архитектура серверной части приложения**

В рамках данного приложения серверная часть отвечает за обработку запросов от клиентов, управление базой данных и предоставление необходимой функциональности для работы с таблицей статистики погоды.

Основные компоненты серверной части приложения включают следующие элементы:

- модели данных: в этой части приложения определены модели данных, описывающие сущности, такие как статистика погоды и администратор.
- контроллеры: контроллеры отвечают за обработку запросов от клиентов и взаимодействие с моделями данных. Они содержат функции для создания, чтения, обновления и удаления данных, а также другие операции, связанные с логикой приложения;
- база данных: серверная часть приложения взаимодействует с базой данных для хранения и извлечения данных. В серверной части приложения содержатся файлы, связанные с настройкой базы данных и файлы миграции, которые определяют структуру базы данных и её обновления;
- другие служебные файлы: это служебные файлы, такие как файлы конфигурации, файлы для обработки аутентификации и авторизации

администраторов или файлы для настройки среды выполнения серверной части приложения.

### 3.3 Реализация клиентской части приложения (frontend)

Клиентская часть отвечает за интерфейс пользователя и взаимодействие с серверной частью приложения для отображения данных и выполнения операций. Клиентская часть состоит из mustache-файлов, которые отвечают за отображение моделей полученных при помощи запросов, посылаемых из форм.

### 3.4 Графический интерфейс

На рисунках 15 - 25 представлен удобный и интуитивно понятный пользователю интерфейс сайта.

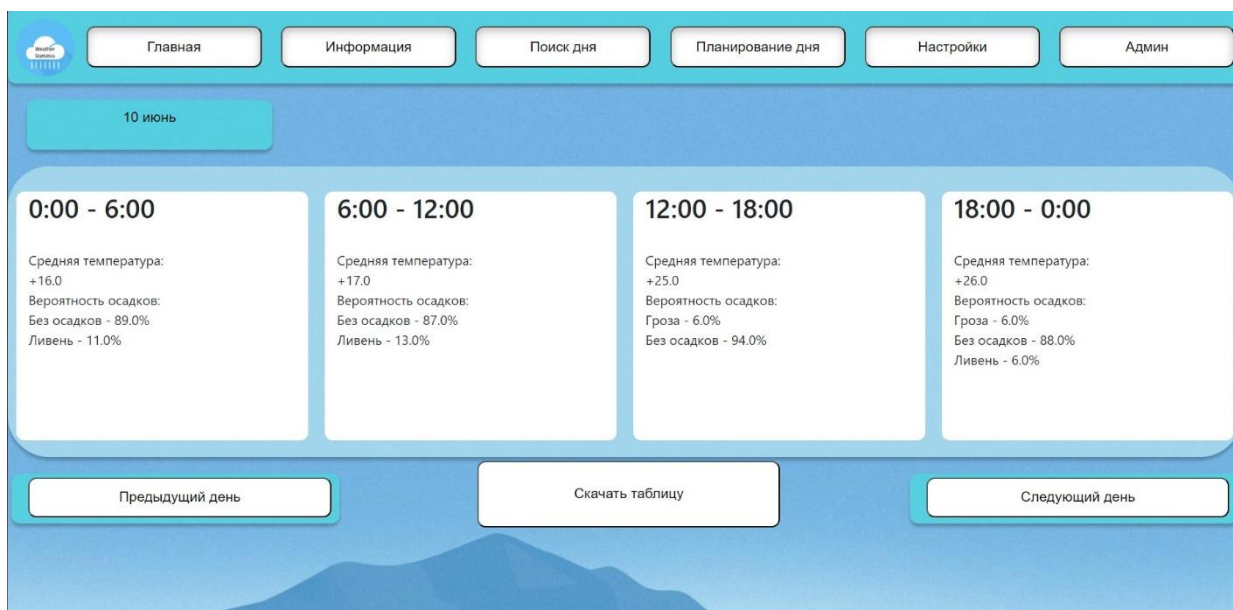


Рисунок 15 - Главная страница

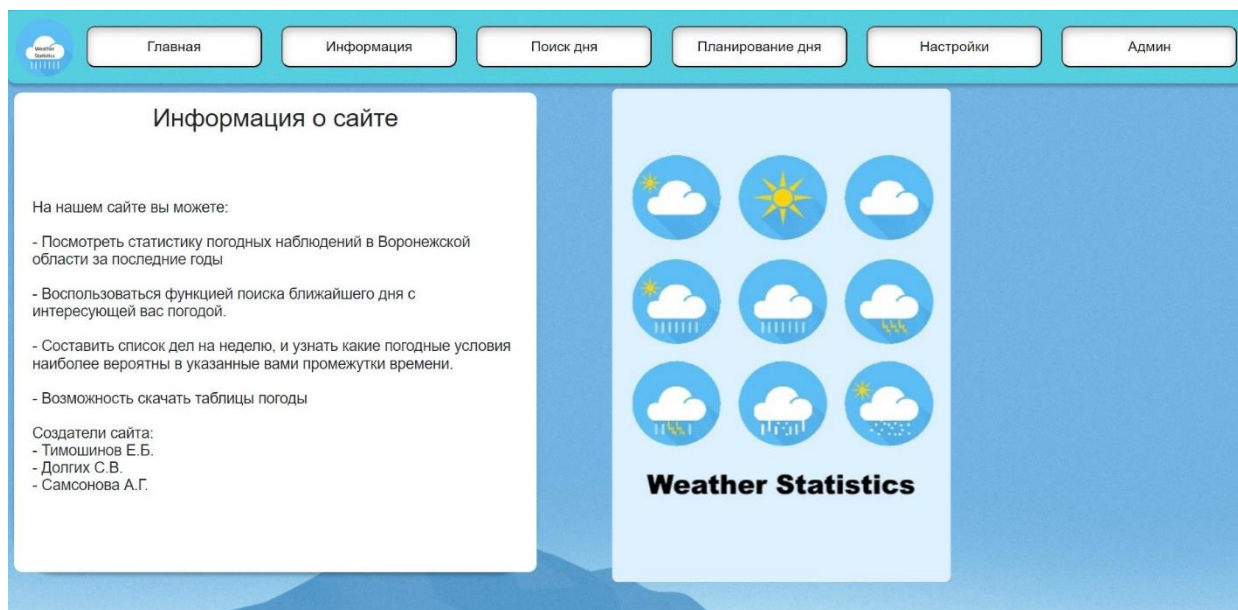


Рисунок 16 - Информация о сайте

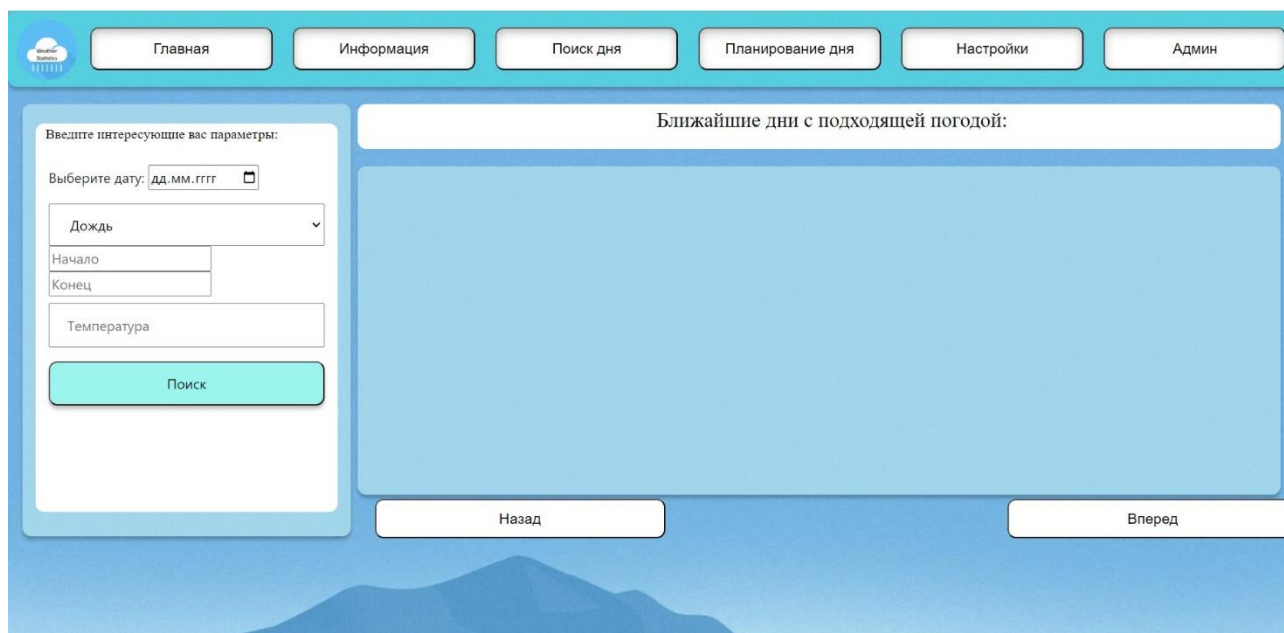


Рисунок 17 - Ввод данных для поиска ближайших дней с указанными погодными условиями

Введите интересующие вас параметры:

Выберите дату:

Дождь ☐

Начало

Конец

Температура

Ближайшие дни с подходящей погодой:

Дата	Средняя температура в течении дня	Вероятность осадков
12 августа 12:00 - 18:00	+25.0	Гроза - 3.0% Без осадков - 95.0% Ливень - 3.0%
19 августа 12:00 - 18:00	+19.0	Гроза - 2.0% Без осадков - 88.0% Дождь - 1.0% Ливень - 8.0%
21 августа 12:00 - 18:00	+22.0	Гроза - 1.0% Без осадков - 92.0% Ливень - 7.0%

Рисунок 18 - Результат подобранных дней с выбранными погодными условиями

Выберите дату:

Рисунок 19 - Редактирование данных для планирования дня (часть 1)

Рисунок 20 - Редактирование данных для планирования дня (часть 2)

Рисунок 21 - Отредактированное планирование дня



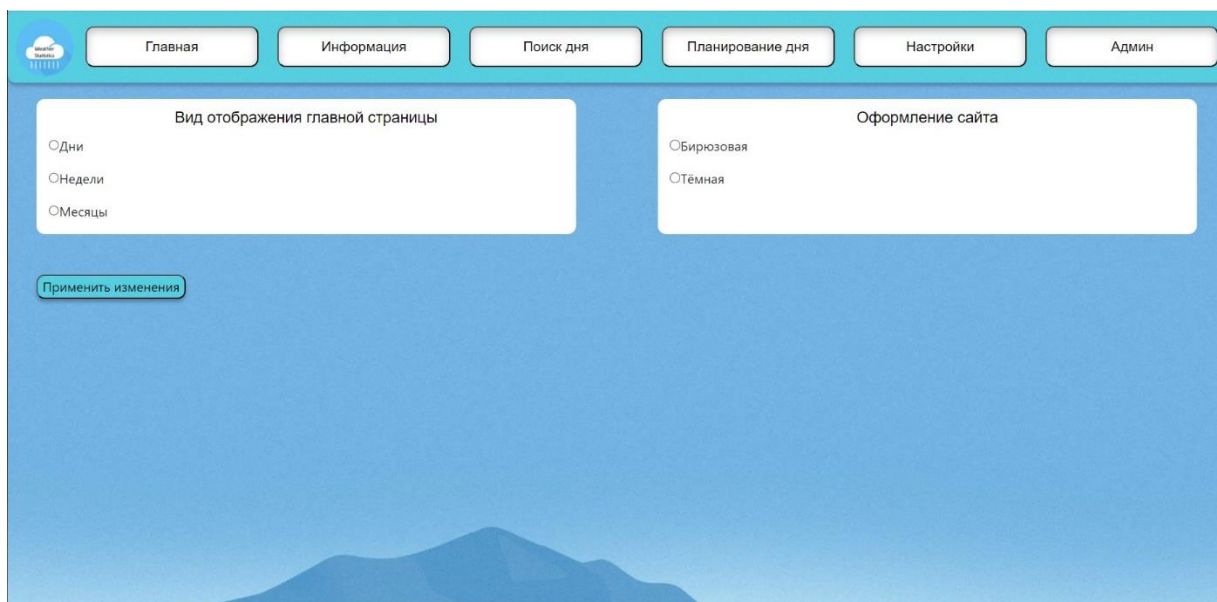


Рисунок 22 - Страница «Настройки»

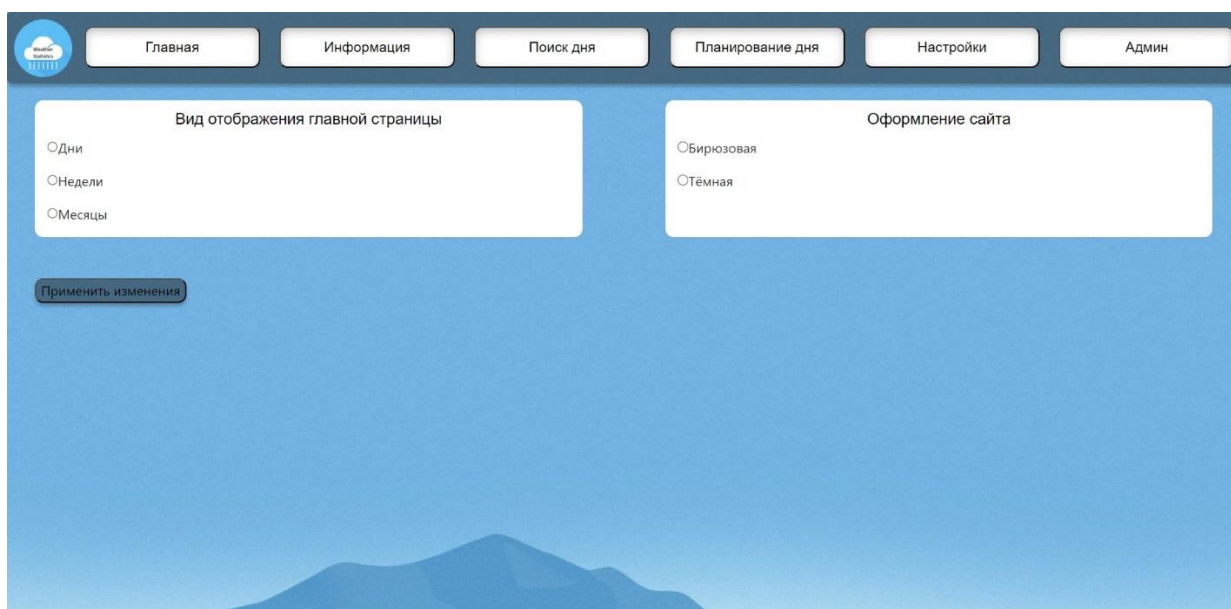


Рисунок 23 - Тёмная версия интерфейса

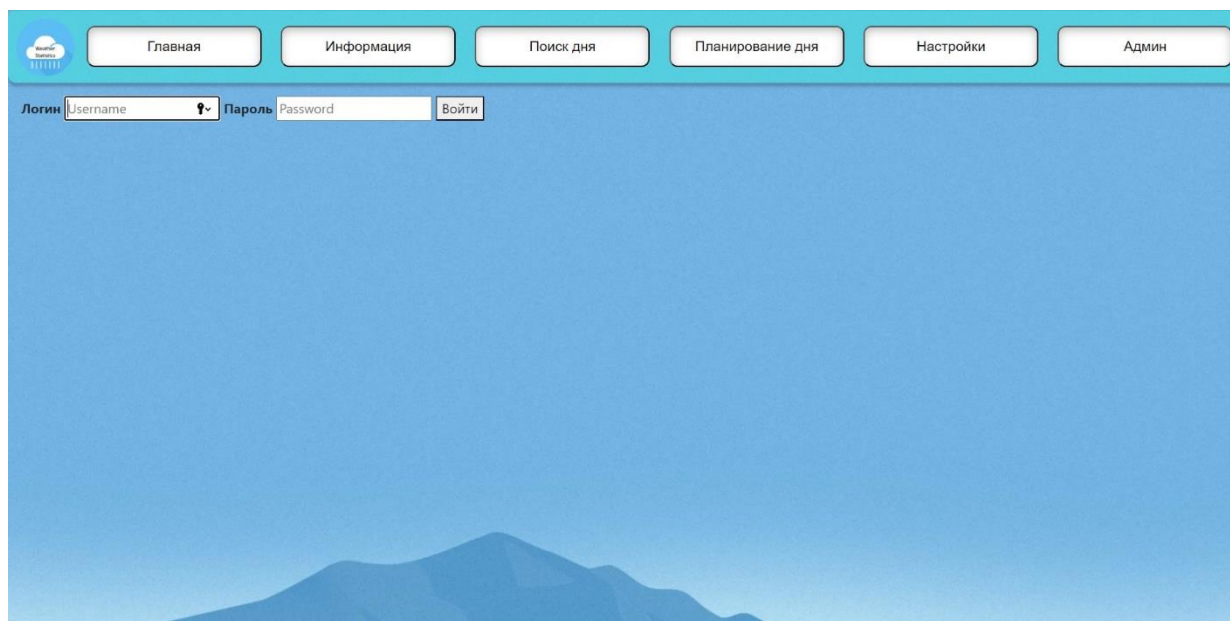


Рисунок 24 - Страница администратора до входа

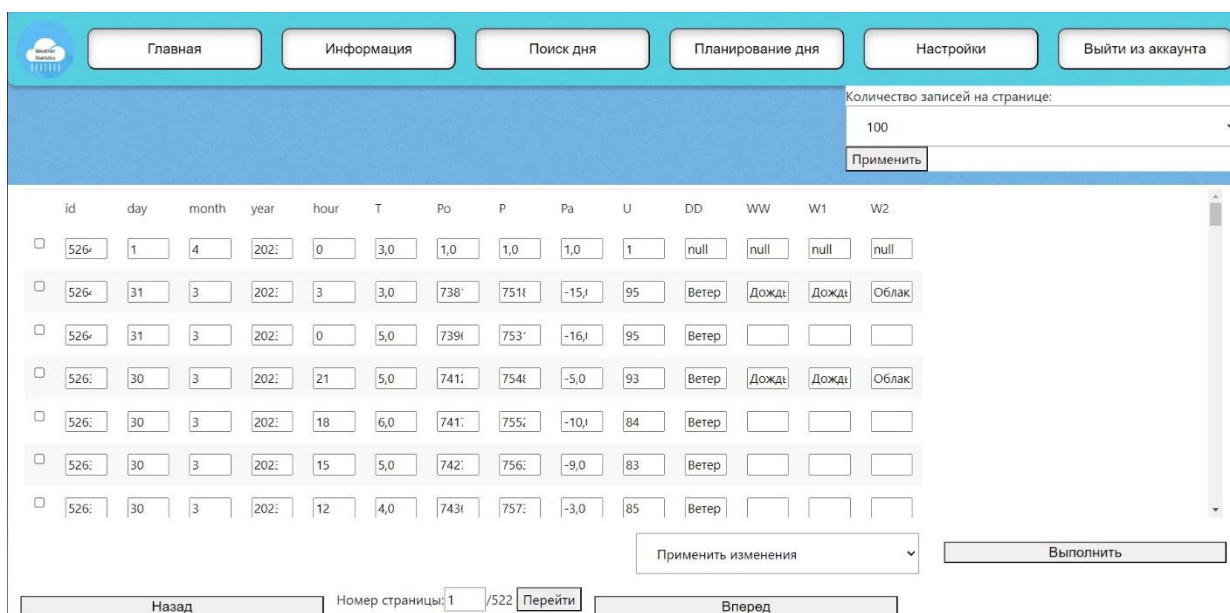


Рисунок 25 - Страница администратора после входа



#### 4 Статистика использования приложения

В качестве системы для сбора данных об использовании приложения пользователями выступает сервис «Яндекс.Метрика», так как позволяет наиболее быстро и легко настроить метрики для веб-приложений, обладает понятным интерфейсом и доступным руководством по использованию.

На Рисунках 26, 27 показаны данные, собранные с сервиса «Яндекс.Метрика».

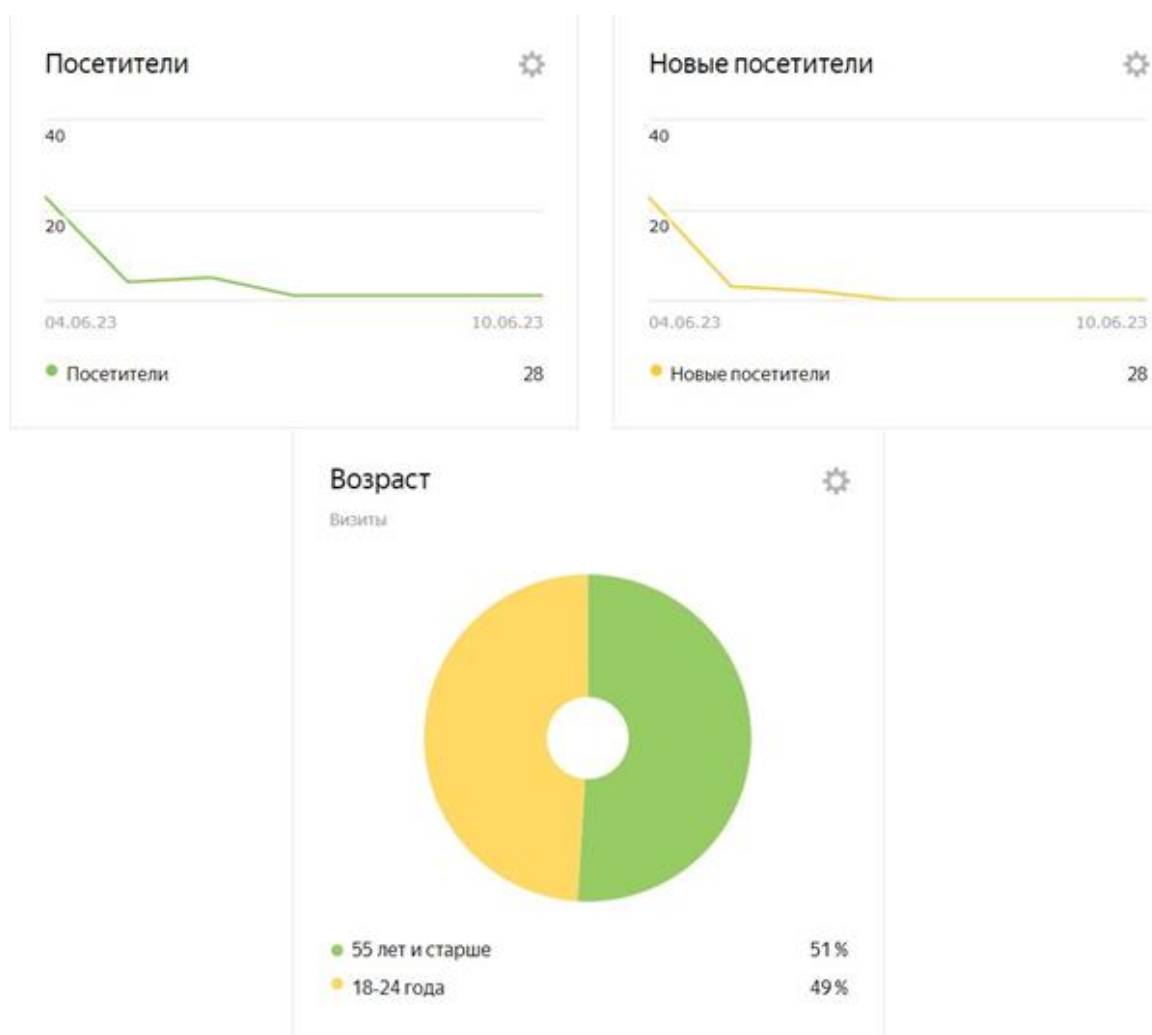


Рисунок 26 - Данные о посетителях сайта

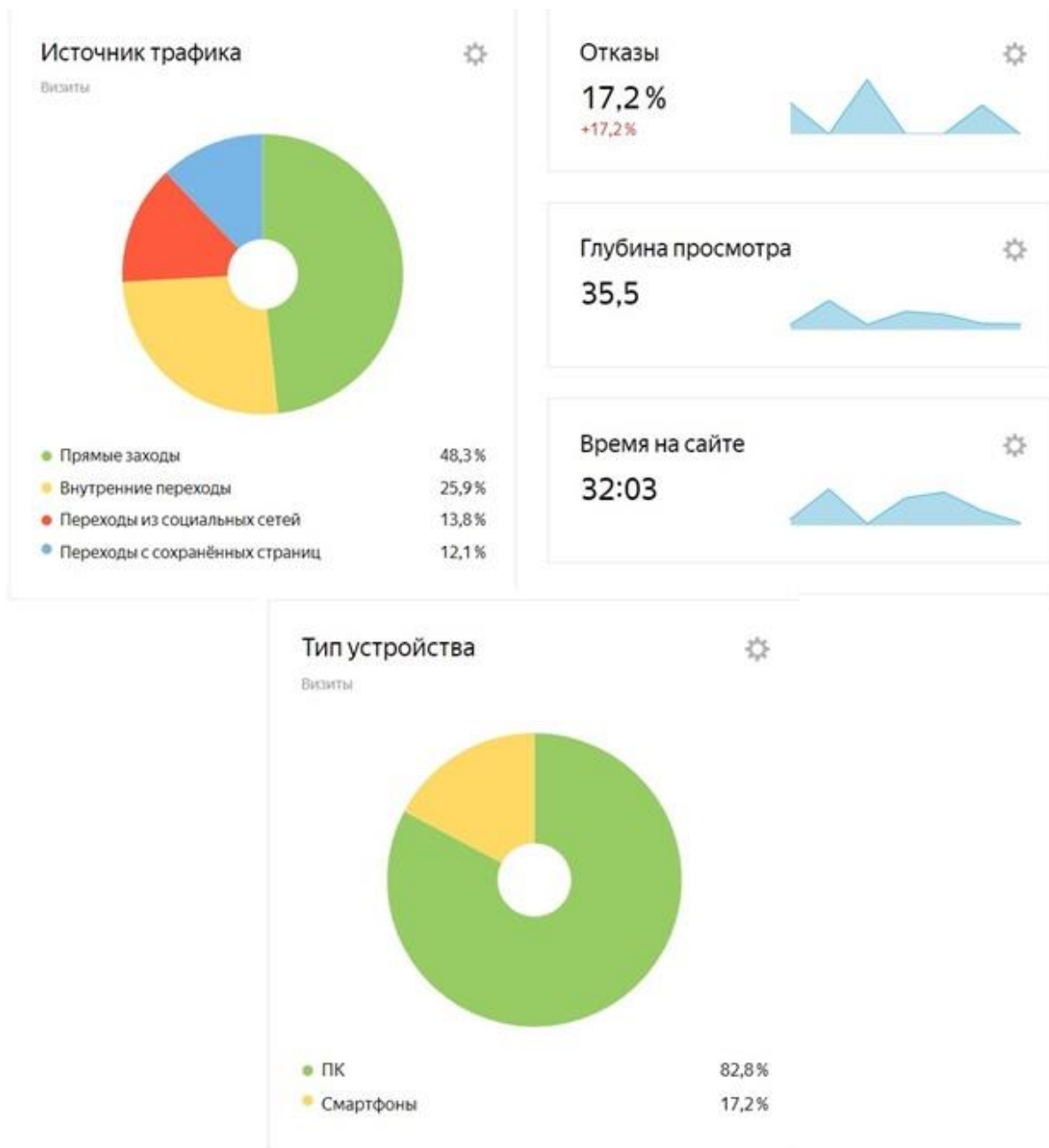


Рисунок 27 - Данные о просмотрах сайта

## 5 Тестирование

После реализации всех задач, был произведен запланированный набор тестов:

- Дымовое тестирование.
- UI тестирование.
- Юзабилити тестирование.

### 5.1 Дымовое тестирование

Для данного тестирования необходимо проверить работоспособность на следующие основные сценарии:

- Просмотр погоды на день/ неделю/ месяц.
- Поиск дня с заданными параметрами.
- Составление плана на день.
- Изменение оформления сайта.
- Авторизация администратора.
- Редактирование данных о погоде со страницы администратора.

Дымовые тестирования были проведены ручным способом в следующих браузерах: Yandex версии 23.5.1.714, Google Chrome версии 111.0.5563.64, с включенным WI-FI для связи с Back-end частью сайта.

Результаты тестирования приведены в Таблице 1.

Таблица 1 - Результаты дымового тестирования

Сценарий	Результат
Просмотр погоды на день/ неделю/ месяц	Пройден
Поиск дня с заданными параметрами	Пройден

Составление плана на день	Пройден
Изменение оформления сайта	Пройден
Авторизация администратора	Пройден
Редактирование данных о погоде со страницы администратора	Пройден

По результатам дымового тестирования было установлено, что сайт проходит все основные сценарии.

## 5.2 UI тестирование

Для данного вида тестирования было произведено 18 тестов, результат которых приведен в Таблице 2.

Таблица 2 - Результаты UI тестирования

Шаги теста	Ожидаемый результат	Результат
Нажатие на кнопку «Следующий день»	Открылась страница с данными о погоде на следующий день	Пройден
Нажатие на кнопку «Предыдущий день»	Открылась страница с данными о погоде на предыдущий день	Пройден
Нажатие на кнопку «Скачать таблицу»	На компьютер загрузился файл (таблица) с данными о погоде	Пройден
Нажатие на кнопку «Информация»	Открылась страница с данными о сайте	Пройден
1. Нажатие на кнопку «Поиск дня» 2. В поля вводятся интересующие данные 3. Нажатие на кнопку «Поиск» 4. Нажатие на кнопку «Вперёд» 5. Нажатие на кнопку «Назад»	1. Открылась страница «Поиск дня» 3. Открылась страница с ближайшими днями по введённым данным 4. Открылись следующие 3 дня 5. Открылись предыдущие 3 дня	Пройден
1. Нажатие на кнопку «Планирование дня»	1. Открылась страница «Планирование дня»	Пройден

<ul style="list-style-type: none"> <li>2. Ввод корректных данных о дате</li> <li>3. Нажатие на кнопку «+»</li> <li>4. Ввод плана на день</li> <li>5. Нажатие на кнопку «+»</li> <li>6. Нажатие на кнопку «Показать результаты»</li> </ul>	<ul style="list-style-type: none"> <li>3. Создалась заметка для выбранного дня</li> <li>5. Заметка дополнилась введенным планом</li> <li>6. Открылось окно с планом дня и соответствующей этому дню погодой</li> </ul>	
<ul style="list-style-type: none"> <li>1. Нажатие на кнопку «Настройки»</li> <li>2. Выбор вида главной страницы (дни/недели/месяцы)</li> <li>3. Нажатие на кнопку «Применить изменения»</li> <li>4. Нажатие на кнопку «Главная страница»</li> </ul>	<ul style="list-style-type: none"> <li>1. Открылась страница «Настройки»</li> <li>3. Сохранились изменения</li> <li>4. Открылась главная страница в виде: день/неделя/ месяц</li> </ul>	Пройден
<ul style="list-style-type: none"> <li>1. Нажатие на кнопку «Настройки»</li> <li>2. Выбор тёмной темы</li> <li>3. Нажатие кнопки «Применить изменения»</li> <li>4. Выбор бирюзовой темы</li> <li>5. Нажатие кнопки «Применить изменения»</li> </ul>	<ul style="list-style-type: none"> <li>1. Открылась страница «Настройки»</li> <li>3. Отобразилась тёмная тема</li> <li>5. Отобразилась бирюзовая тема</li> </ul>	Пройден
<ul style="list-style-type: none"> <li>1. Нажатие кнопки «Админ»</li> <li>2. Ввод корректных данных для входа</li> <li>3. Нажатие кнопки «Войти»</li> </ul>	<ul style="list-style-type: none"> <li>1. Открылась страница «Админ»</li> <li>3. Открылась главная страница с авторизованным администратором</li> </ul>	Пройден
<ul style="list-style-type: none"> <li>1. Нажатие кнопки «Админ»</li> <li>2. Ввод корректных данных</li> <li>3. Нажатие кнопки «Применить изменения/ удалить»</li> </ul>	<ul style="list-style-type: none"> <li>1. Открылась страница «Админ»</li> <li>4. Сохранились отредактированные данные о погоде / удалилась строка с данными о погоде на</li> </ul>	Пройден

выбранное» 4. Нажатие кнопки «Выполнить» 5. Выбрать количество отображаемых записей на странице 6. Нажатие кнопки «Применить»	выбранный день 6. Количество записей увеличилось по выбранному числу	
--	---	--

По результатам UI тестов было установлено, что сайт проходит все тесты.

### 5.3 Юзабилити тестирование

Для проведения юзабилити тестирования было отобрано 3 человека, ранее не использовавших сайт. Для данного вида тестирования необходимо проверить следующие сценарии взаимодействия пользователя с сайтом:

- Просмотр погоды на день/ неделю/ месяц.
- Поиск дня с заданными параметрами.
- Составление плана на день.
- Изменение оформления сайта.

Результаты тестов представлены в таблице 3.

Таблица 3 - Результаты юзабилити тестирования

Сценарий	Пользователь 1	Пользователь 2	Пользователь 3
Просмотр погоды на день/ неделю/ месяц	Пройден	Пройден	Пройден
Поиск дня с заданными параметрами	Пройден	Пройден	Пройден
Составление плана на день	Пройден	Пройден	Пройден

Изменение оформления сайта	Пройден	Пройден	Пройден
----------------------------	---------	---------	---------

По результатам юзабилити тестирования было установлено, что сайт проходит все основные сценарии.

## Заключение

В результате работы был разработан сайт учёта статистики погодных наблюдений «Weather Statistics».

Реализация сайта производилась на основе выбранных технологий front-end (HTML и CSS) и back-end (Фреймворк «Spring» и СУБД «MySQL») разработки. Во время разработки сайта происходила поэтапная реализация представленных требований относительно функциональной части приложения. Также был разработан удобный и интуитивно понятный интерфейс сайта.

В процессе разработки сайта были произведены:

- Анализ предметной области.
- Проектирование базы данных.
- Разработка серверной части и клиентской части приложения.
- Тестирование и отладка.

Таким образом, при реализации сайта были выполнены все требования, представленные на старте работы.



## **Список использованных источников**

1. Фреймворк как программная платформа, Классификация и виды фреймворков (framework) [Электронный ресурс]. – Режим доступа: URL: <https://intellect.icu/frejmwork-kak-programmnaya-platforma-klassifikatsiya-i-vidy-frejmworkov-framework-9515>. – (Дата обращения: 15.03.2023).
2. Прогноз погоды в Воронеже на 10 дней [Электронный ресурс]. – Режим доступа: URL: <https://yandex.ru/pogoda/voronezh?via=hnav&lat=51.759598&lon=39.182407>. – (Дата обращения: 21.03.2023).
3. GISMETEO: Погода в Воронеже на 3 дня [Электронный ресурс]. – Режим доступа: URL: <https://www.gismeteo.ru/weather-voronezh-5026/3-days/>. – (Дата обращения: 21.03.2023).
4. Погода в мире на неделю, 10 и 14 дней. Погода в странах мира [Электронный ресурс]. – Режим доступа: URL: <https://world-weather.ru/pogoda/>. – (Дата обращения: 21.03.2023).
5. Уолс К. Spring в действии / Уолс К. - М.: ДМК Пресс, 2013. – 752 с.
6. MySQL: что это за сервер базы данных, пример [Электронный ресурс]. – Режим доступа: URL: <https://blog.skillfactory.ru/glossary/mysql/>. – (Дата обращения: 22.03.2023)