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

Факультет компьютерных наук

Кафедра технологий обработки и защиты информации

Курсовая работа

по дисциплине: «Технологии программирования»

на тему: «Система хранения статистических данных погодных наблюдений за прошедшие годы, прогнозирования погоды на ближайшие дни и рекомендации одежды на эти дни»

Преподаватель _____В.С. Тарасов, ст. преподаватель __.__20__

Обучающийся _____А.М. Дракин, 4 курс, д/о

Обучающийся _____Т.С. Покушалова, 4 курс, д/о

Обучающийся _____И.С. Величко, 4 курс, д/о

Обучающийся _____А.С. Новиков, 4 курс, д/о

Воронеж

2023

Содержание

Содержание 2

Введение 4

1 Постановка задачи 5

2 Анализ предметной области 6

2.1 Предметная область 6

2.2 Основные возможности и требования к сайту 6

2.3 Обзор аналогов. 7

2.3.1 Статистика погоды Gismeteo 7

2.3.2 Статистика погоды Погодные сервисы 9

2.3.3 Статистика погоды Rp5 11

3 Графическое описание работы системы 13

3.1 Диаграмма IDEF0 13

3.2 Диаграмма прецедентов 13

3.3 Диаграмма активностей 14

3.4 Диаграмма последовательностей 16

3.5 Диаграмма состояний 18

3.6 Диаграмма классов 18

3.7 Диаграмма развертывания 19

4 Реализация 21

4.1 Средства реализации 21

4.2 Реализация серверной части 22

4.3 Реализация базы данных 23

4.4 Реализация прогноза одежды 24

4.5 Реализация клиентской части 27
4.6 Методология 28
5 Тестирование 30
5.1 Дымовое тестирование 30
5.2 Функциональное тестирование 31
Заключение 34
Список использованных источников 35

Введение

В современном мире практически все люди зависят от прогноза погоды. Из-за неё людям нередко приходится перестраивать свои планы или возвращаться домой, чтобы переодеться в подходящую одежду.

Однако существует значительное количество отраслей национальной экономики, деятельность которых была бы куда менее эффективной без знания наиболее вероятных условий атмосферы в будущем. Это авиация, морской флот, сельское хозяйство, энергетика. Грамотное использование гидрометеорологических прогнозов, безусловно, важно для обеспечения безопасности населения при таких стихийных явлениях, как наводнения, сильные грозы, смерчи (торнадо), экстремальные мороз и жара.

Безусловно не стоит забывать и о статистике погодных наблюдений. Она помогает составить прогноз погоды на будущие дни, определить наиболее удобное место для строительства посадочной полосы, а также возделывания сельскохозяйственного поля. Без статистических данных погоды не может обойтись и научная сфера деятельности человека.

Таким образом, целью нашей работы является создание такой системы, которая будет хранить статистику погодных наблюдений, составит прогнозы погоды на ближайшие дни, составит рекомендацию одежды, а также подчеркнет моменты, когда была и будет аномальная погода.

1 Постановка задачи

Целью данного курсового проекта является разработка самостоятельной системы хранения статистический наблюдений за погодой и прогнозирование её на ближайшие дни. В системе предусмотрена регистрация пользователя, после которой он получит доступ к составленной рекомендации одежды по часам на определённый день.

Данная система разделена на две части: сайт и управляющая система сервера. Сайт предназначен для работы авторизованных, неавторизованных пользователей и администраторов. Управляющая система сервера предназначена для:

- составления прогноза погоды;
- составления рекомендации одежды;
- хранения статистических данных и составления прогноза по аномальной погоде.

2 Анализ предметной области

2.1 Предметная область

Разработанный сайт должен представлять из себя удобный и понятный в использовании сервис, предоставляющий прогноз погоды, а также архив с погодой в определённые периоды. Он позволит пользователю просматривать как текущие прогнозы, так и погодные условия прошлых периодов. Из отличительных черт – введение рекомендуемой одежды по погодной ситуации.

Незарегистрированные пользователи смогут посмотреть статистику погоды за прошлые периоды, прогноз погоды на ближайшее время и предсказание одежды на день. Зарегистрированные поверх этих возможностей имеют доступ просматривать предсказание одежды по часам и просмотр аномальной погоды на определённые периоды, а также предсказание аномальной погоды на будущее. Отсюда можем сделать выводы, что авторизованные пользователи имеют больше возможностей на сайте.

Администраторы сайта имеют доступ к редактированию предсказаний погоды и одежды, изменению пользовательской информации и размещению рекламы.

Для реализации сайта потребуется выделить сущности, и их атрибуты:

- пользователь. Новый пользователь должен пройти регистрацию. Атрибуты данной сущности: id, имя, почта, пароль;
- администратор. Администраторы сайта, имеющие доступ к изменению публичной информации на сайте. Атрибуты данной сущности: id, имя, почта, пароль.

2.2 Основные возможности и требования к сайту

Можно выделить следующие функциональные возможности:

- наличие интерфейса;

— регистрация пользователей. Должна быть реализована регистрация и авторизация для пользователей сайта;
— реализация дополнительных возможностей для авторизованных пользователей;
— предсказание аномальной погоды на будущее и подбор рекомендуемой одежды;
— доступ к статистике погоды прошлых периодов. Пользователь сможет изучить архив с погодными условиями за прошедшие периоды.

С данным набором функций и задач сайт будет наиболее приближен к поставленным целям.

Выводы:

— проведён анализ предметной области;
— рассмотрена структура данных для создания сайта;
— выделены атрибуты для сущностей;
— выделены основные функциональные возможности и требования к сайту.

2.3 Обзор аналогов.

2.3.1 Статистика погоды Gismeteo

Gismeteo – это сайт, где можно посмотреть статистику погоды за определённый период времени в определённом населенном пункте страны. Содержит возможность оставлять отзывы. Интерфейс представлен на рисунке 1.

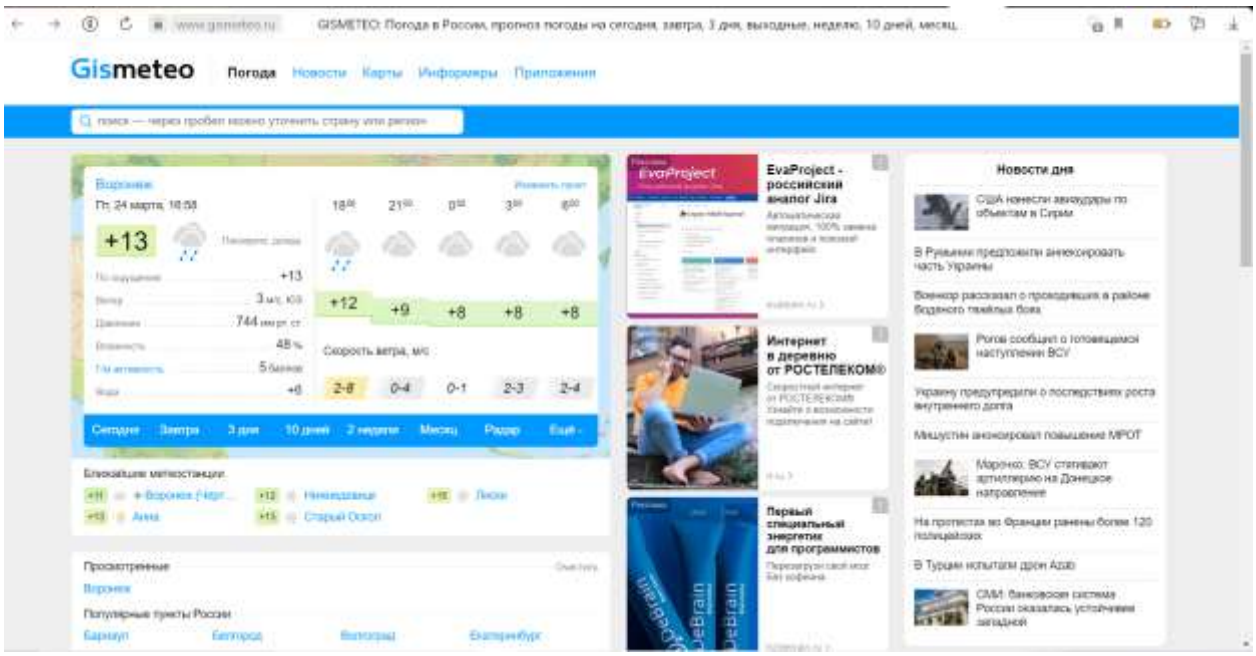

Рисунок 1 - Интерфейс сайта «Gismeteo»

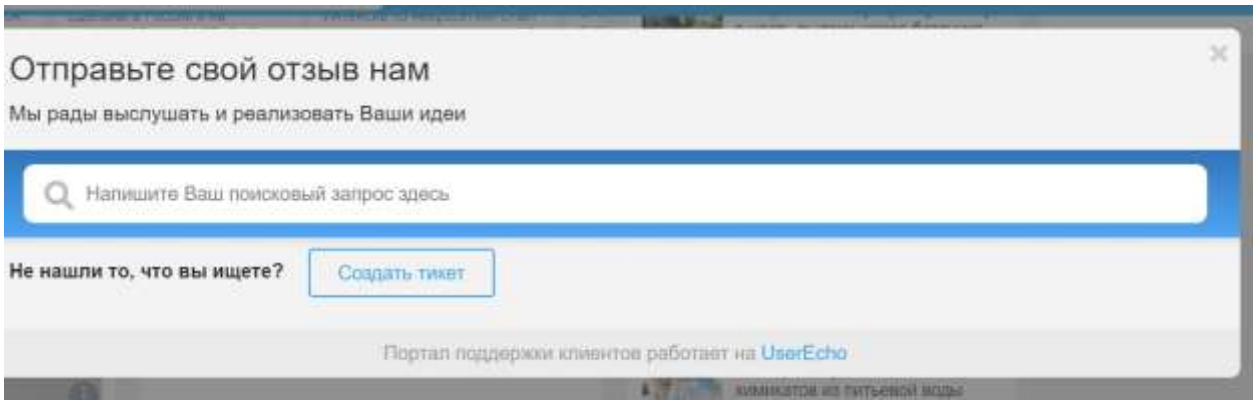

Рисунок 2 - Интерфейс отзывов

Рисунок 3 - Статистика погоды

Недостатками сайта «Gismeteo» являются:

— отсутствие рекомендаций одежды;

— большое количество рекламы на главной странице;

— малая наполненность статистики погоды.

2.3.2 Статистика погоды Погодные сервисы

Погодные сервисы – это сайт, где можно посмотреть статистику погоды за определённый период времени в определённом населенном пункте страны. Интерфейс представлен на рисунке 4.

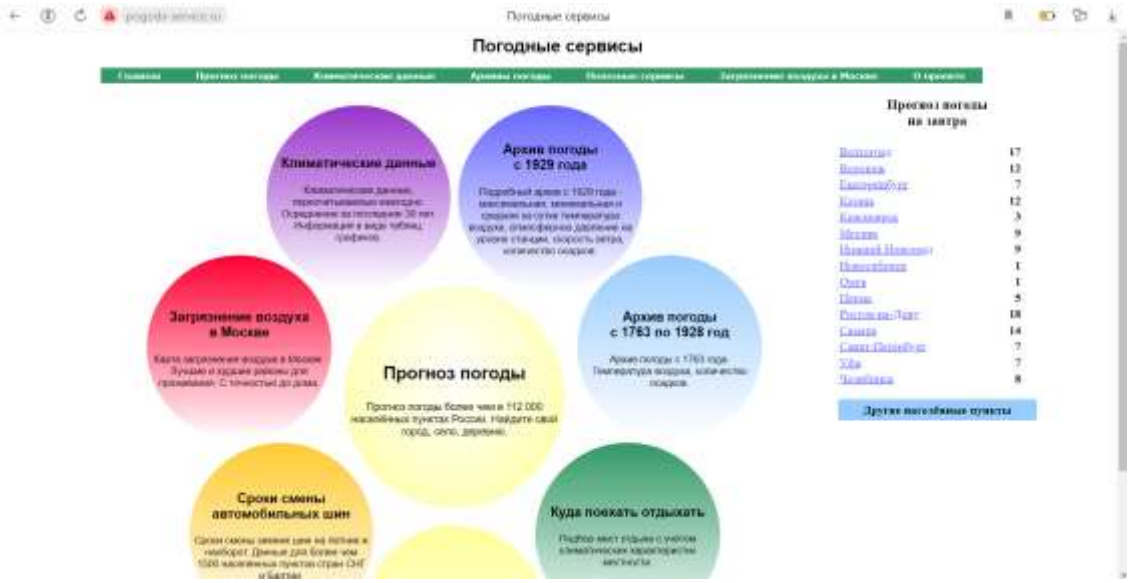

Рисунок 4 - Интерфейс сайта «Погодные сервисы»

Рисунок 5 - Прогноз погоды

Рисунок 6 - Архив погоды

Недостатками сайта «Погодные сервисы» являются:

— отсутствие рекомендаций одежды;

— скудное оформление сайта;

— не работающий архив погоды.

2.3.3 Статистика погоды Rp5

Rp5 – Сайт разработан и сопровождается компанией (ООО) "Расписание Погоды", Санкт-Петербург, Россия, с 2004 года. Компания имеет лицензию на деятельность в области гидрометеорологии и в смежных с ней областях.

Сайт представляет прогнозы погоды на ближайшие семь суток и информацию о фактической погоде, наблюдаемую на наземных станциях. Интерфейс представлен на рисунке 7.

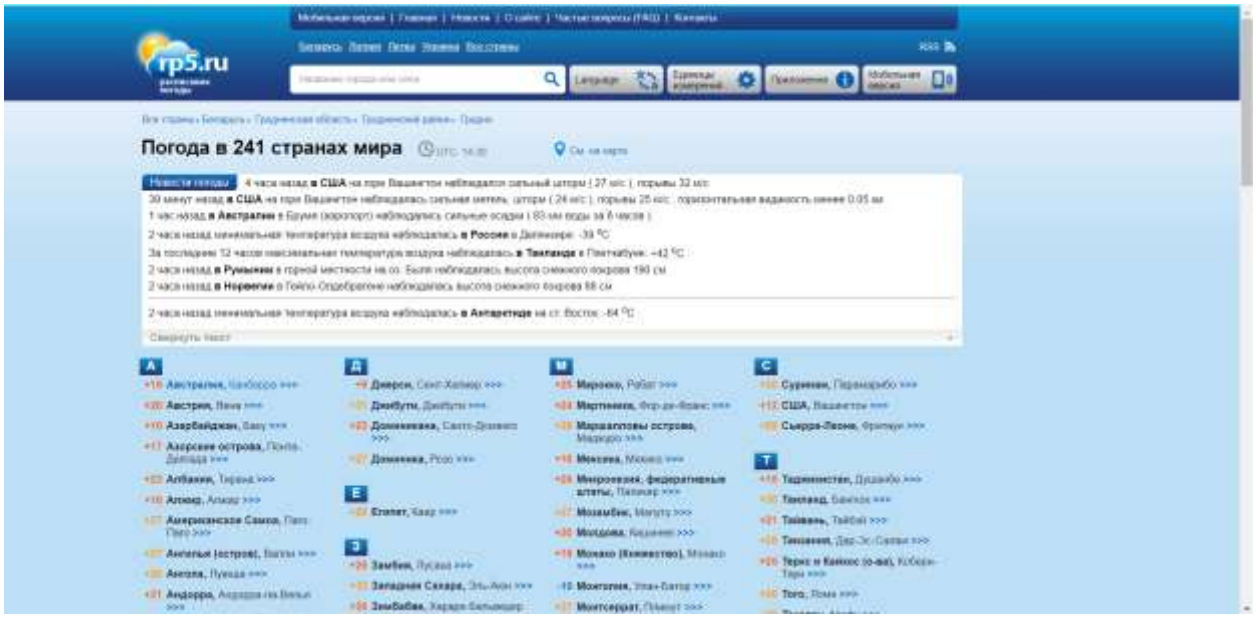

Рисунок 7 - Интерфейс сайта «Погодные сервисы»

Рисунок 8 - Прогноз погоды

Рисунок 9 - Архив погоды

Недостатками сайта «Rp5» являются:

— отсутствие рекомендаций одежды;

— переизбыток данных архива и статистики погоды для обычного пользователя.

3 Графическое описание работы системы

3.1 Диаграмма IDEF0

Рисунок 10 - Диаграмма IDEF0

Диаграмма IDEF0 позволяет описать систему на любом желаемом уровне детализации функционала и сформировать общее представление о её назначении.

3.2 Диаграмма прецедентов

Диаграмма вариантов использования (Use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. Она помогает понять требования к системе и определить её функциональные возможности.

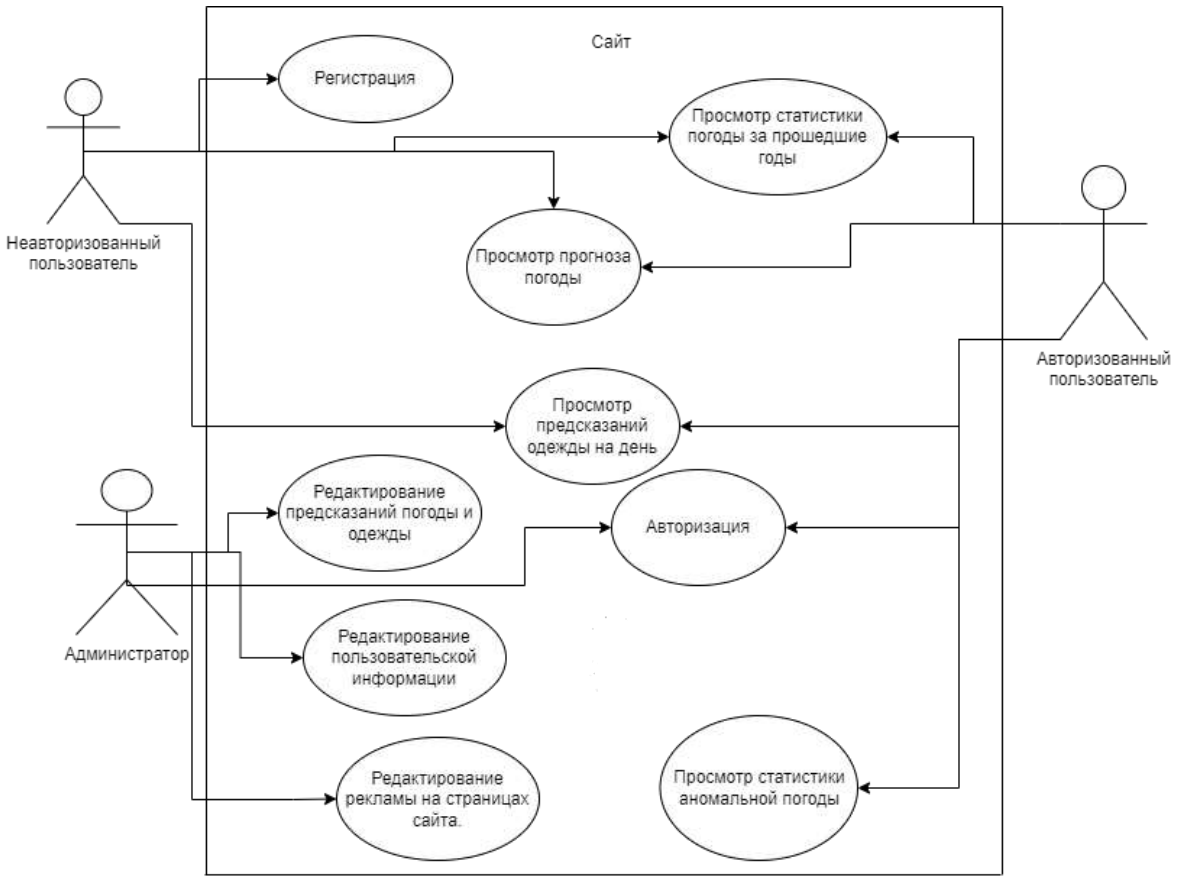

Рисунок 11 - Диаграмма прецедентов

Система предусматривает наличие трёх ролей:

- неавторизованный пользователь (гость) – человек, не имеющий учетной записи, но при желании способный зарегистрироваться, а затем авторизоваться, способный просматривать неполноценную информацию и данные, предоставляемые сайтом (просмотр прогноза и статистики погоды, рекомендаций одежды);
- авторизованный пользователь – авторизованный в системе человек, пользующийся дополнительными возможностями сайта (просмотр аномальной погоды);
- администратор – пользователь, у которого есть возможность редактирования прогноза погоды и рекламы.

3.3 Диаграмма активностей

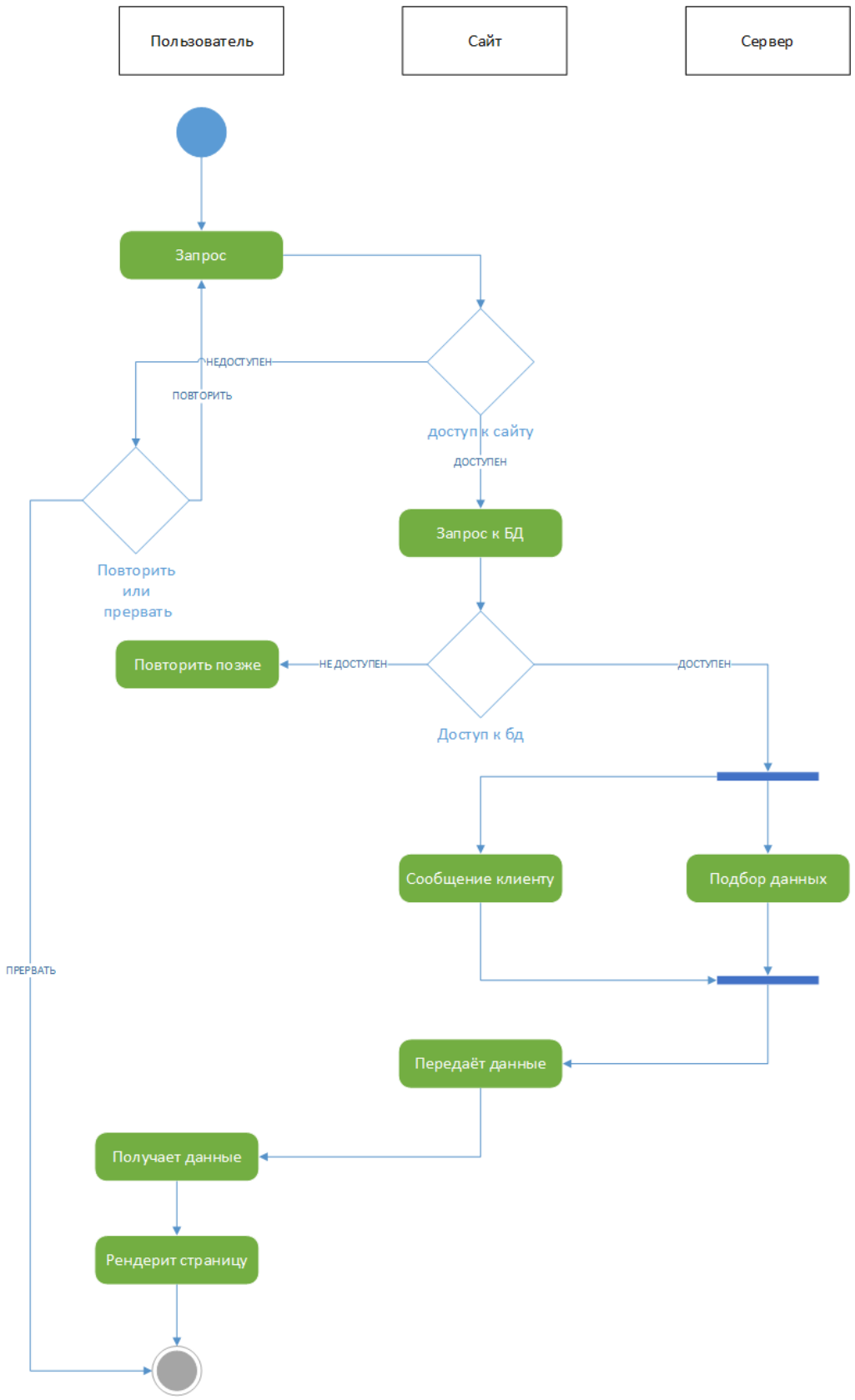

Рисунок 12 - Диаграмма активностей

На рисунке 12 приведена диаграмма активностей. Это поведенческая диаграмма, которая иллюстрирует поток деятельности через систему. На диаграмме можно видеть последовательность процедур, которые будут происходить при запросе пользователя.

3.4 Диаграмма последовательностей

Диаграмма последовательностей – это тип диаграммы, который позволяет описать взаимодействие между объектами в системе в виде последовательности операций, отображая порядок выполнения действий и обмена информацией между объектами во времени.

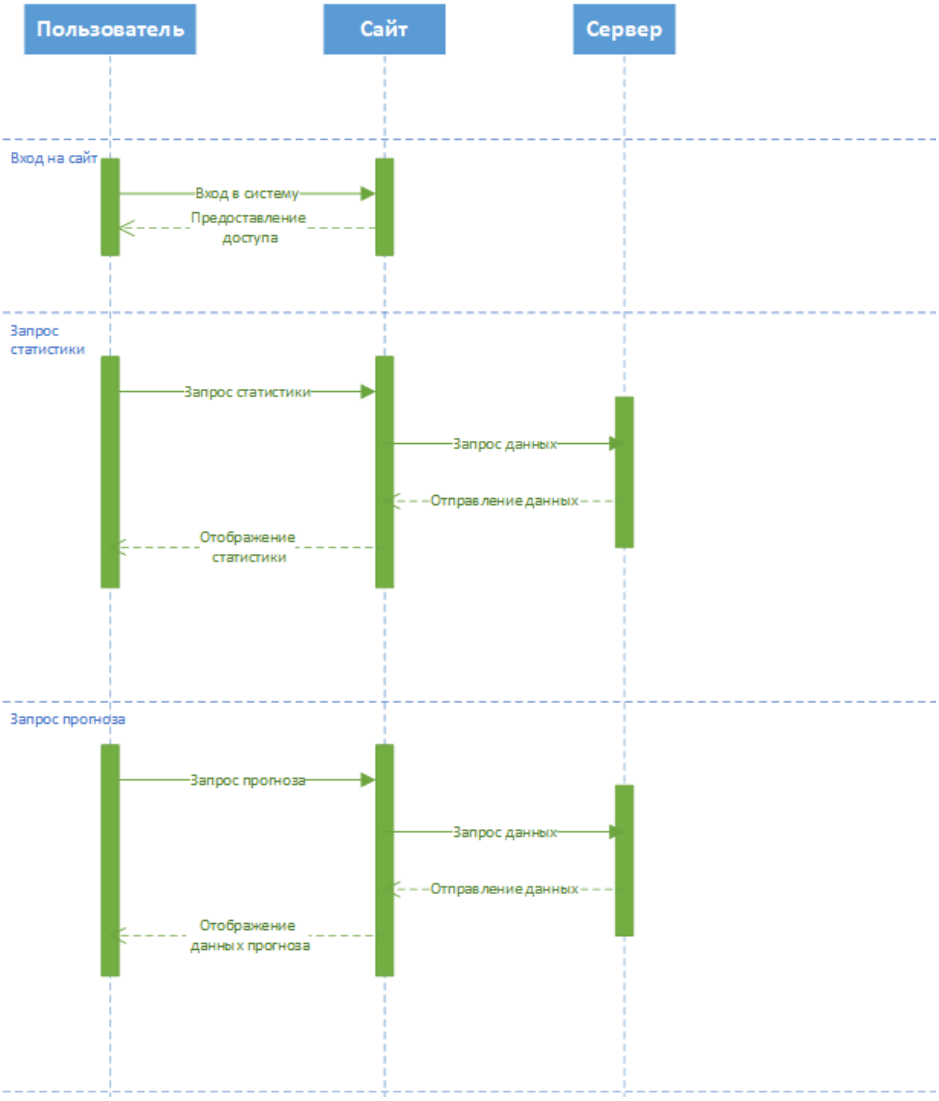

Рисунок 13 - Диаграмма последовательностей (часть 1)

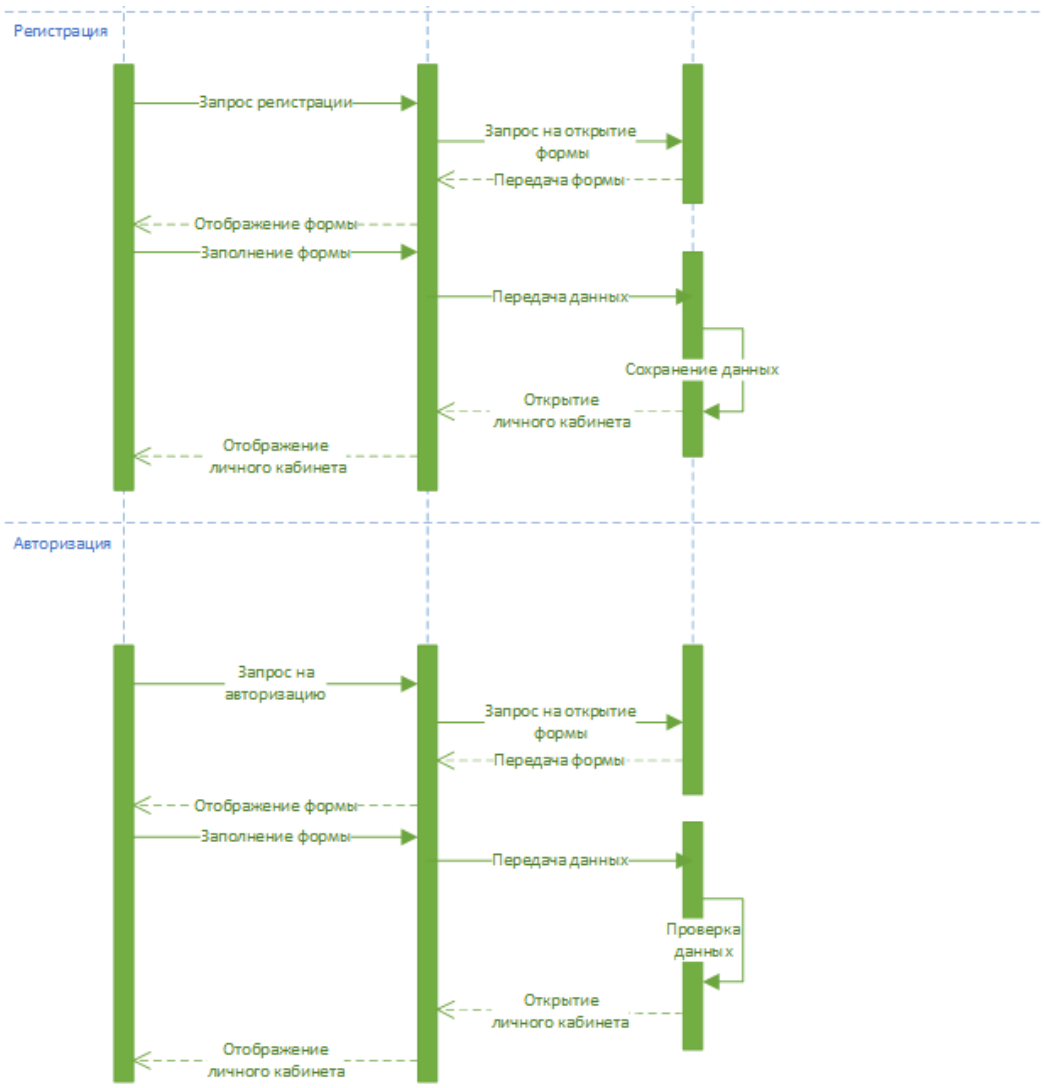

Рисунок 14 - Диаграмма последовательностей (часть 2)

При входе на сайт происходит предоставление доступа пользователю.

При запросе статистики соответствующие данные запрашиваются на сервере, после чего они отображаются для пользователя.

При запросе прогноза соответствующие данные запрашиваются на сервере, после чего они отображаются для пользователя.

При запросе на регистрацию происходит отображение формы регистрации, затем введённые данные передаются на сервер, после чего отображается личный кабинет.

При запросе на авторизацию происходит отображение формы, после чего введённые в неё данные отправляются на сервер, который осуществляет

их проверку. Если проверка успешно пройдена, то отображается личный кабинет.

3.5 Диаграмма состояний

Диаграмма состояний позволяет визуализировать и моделировать поведение объекта или системы в различных состояниях. Она позволяет описать все возможные состояния объекта, а также переходы между ними в ответ на определенные события.

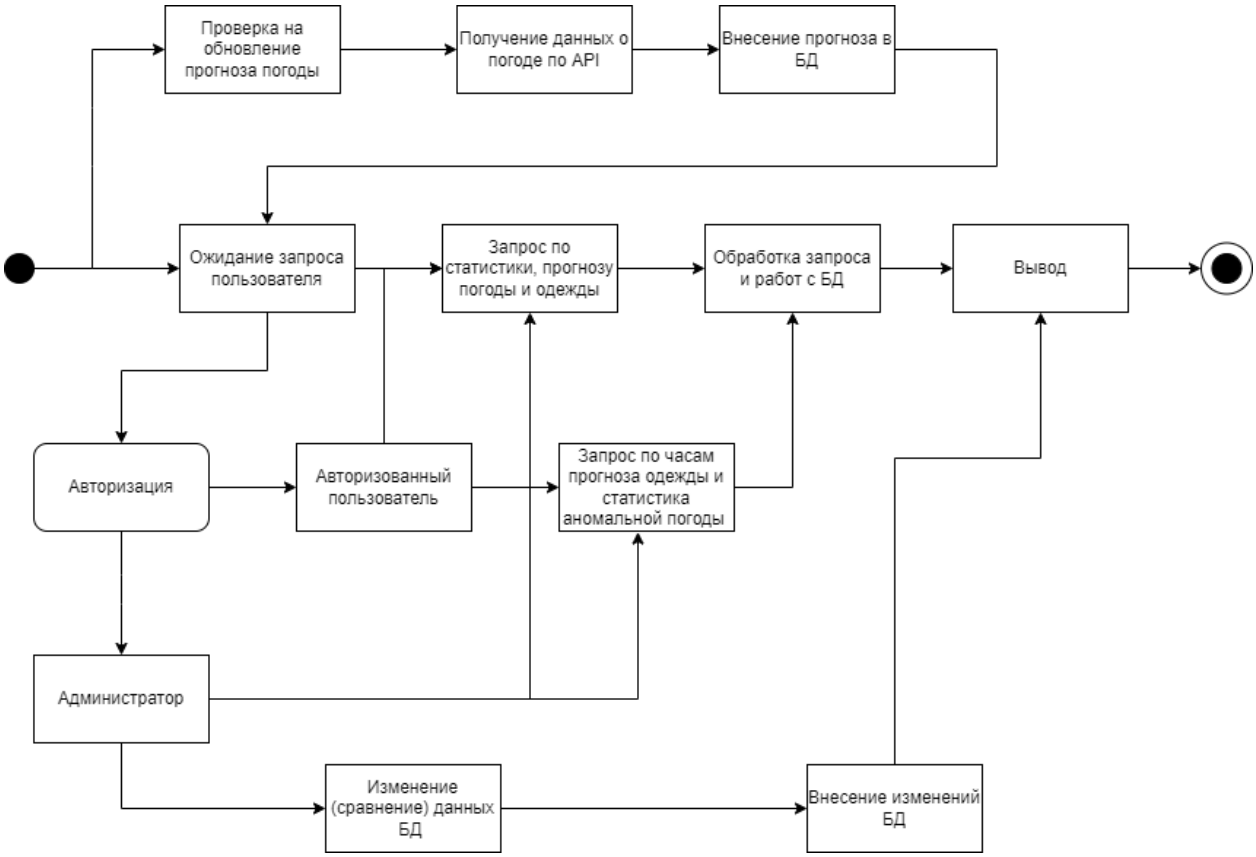

Рисунок 15 - Диаграмма состояний

3.6 Диаграмма классов

Диаграмма классов представляет описание структуры классов в системе и их взаимосвязи. Она отображает как статические аспекты системы, включая классы, атрибуты и методы, а также динамические аспекты, такие как связи между объектами и выполнение методов во время выполнения программы.

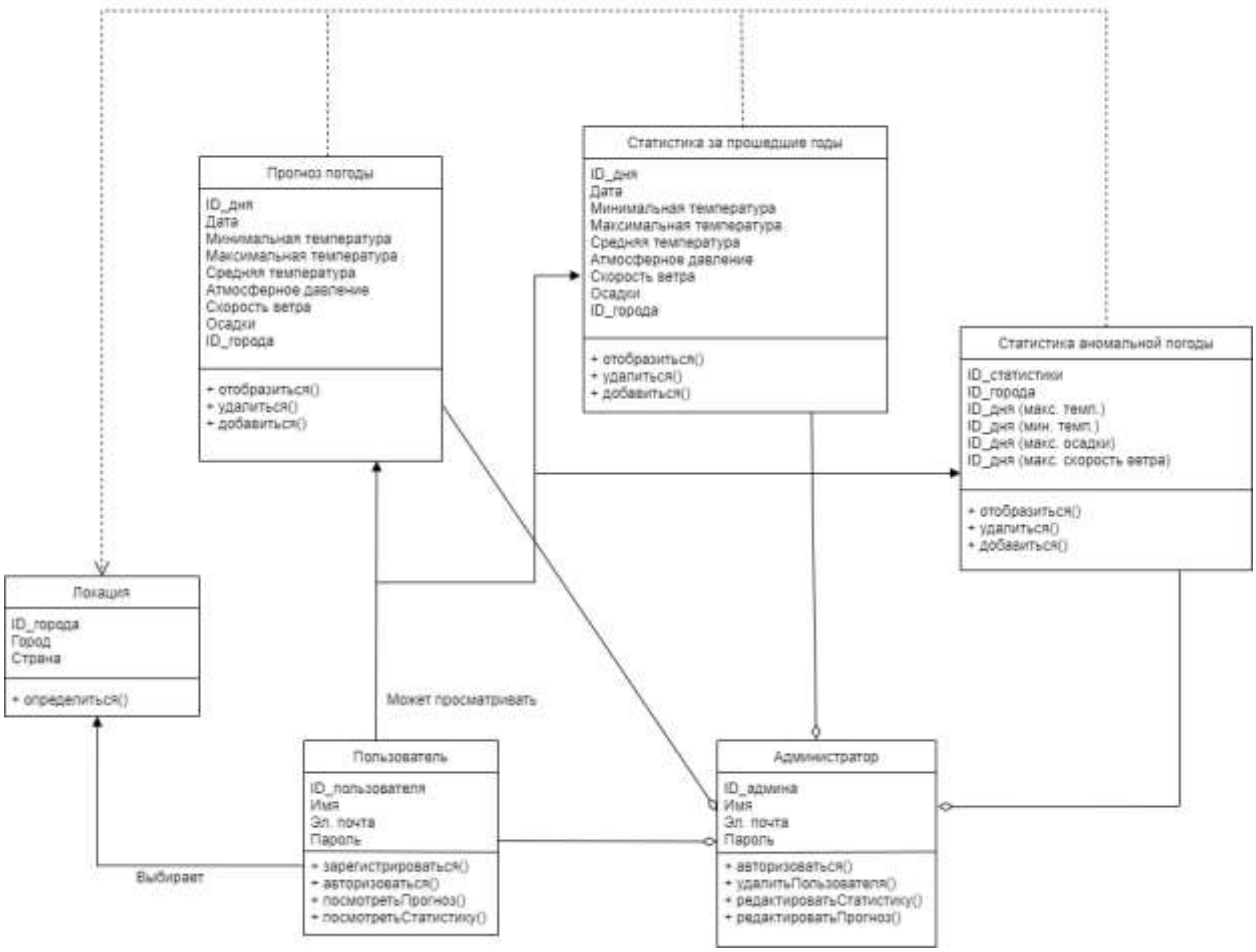

Рисунок 16 - Диаграмма классов

3.7 Диаграмма развертывания

Диаграммы развертывания обычно используются для визуализации физического аппаратного и программного обеспечения системы. Она является UML-диаграммой, которая показывает архитектуру исполнения системы, включая такие узлы, как аппаратные или программные среды исполнения, а также промежуточное программное обеспечение, соединяющее их.

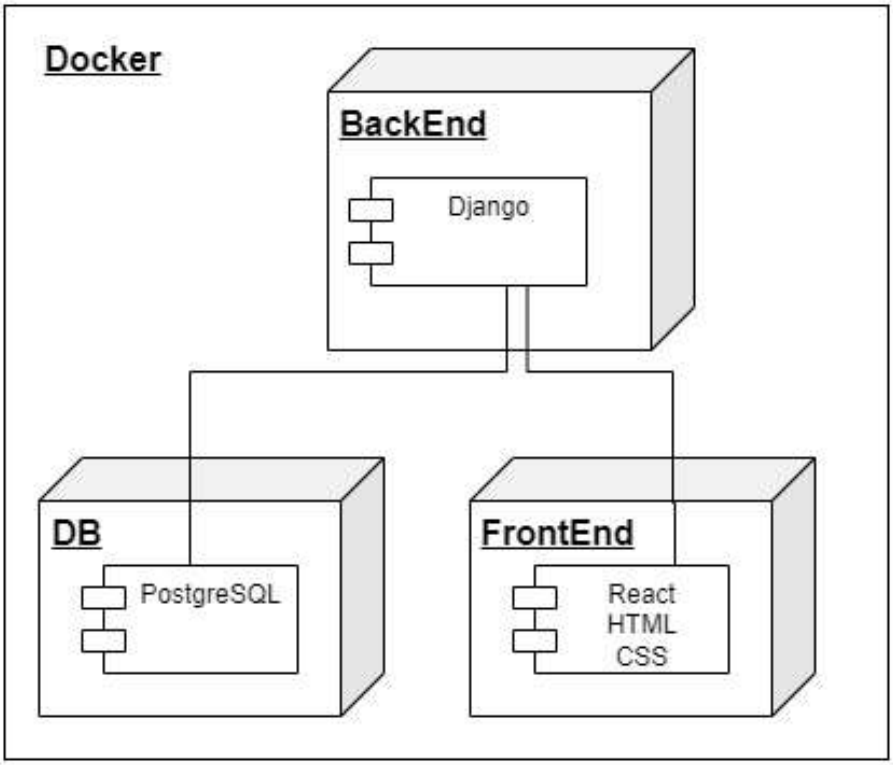

Рисунок 17 - Диаграмма развертывания

4 Реализация

4.1 Средства реализации

Для реализации серверной части нашего веб-приложения с архитектурой MVC (Model-View-Controller: модель-вид-контроллер) мы выбрали язык программирования Python и фреймворк Django. Python предоставляет нам простоту и выразительность кода, а Django обеспечивает эффективное управление запросами, маршрутизацию и взаимодействие с базой данных. В качестве инструмента развертки приложения будет использоваться Docker, который позволяет автоматизировать процесс развертывания и управления приложениями [4].

Для хранения и управления данными мы выбрали базу данных PostgreSQL. Эта выбор обусловлен ее надежностью, масштабируемостью и большим набором функций. Работа с базой данных будет осуществляться через удобное и интуитивно понятное средство администрирования pgAdmin4, что позволит нам удобно управлять данными, выполнять запросы и поддерживать целостность базы данных.

В клиентской части нашего веб-приложения мы будем использовать язык программирования JavaScript в сочетании с библиотекой React. JavaScript обеспечивает динамическое взаимодействие с пользователем и обработку событий на стороне клиента. Библиотека React позволяет нам создавать компоненты пользовательского интерфейса с использованием модульного и переиспользуемого подхода. Для разметки страниц и стилизации мы будем использовать HTML и CSS соответственно.

Таким образом, выбор языка программирования Python, фреймворка Django, базы данных PostgreSQL, средства администрирования фреймворка Django, а также языка программирования JavaScript с библиотекой React, HTML и CSS позволяет нам реализовать веб-приложение погоды [1]. Эти

средства реализации обеспечивают нам функциональность, эффективность и удобство взаимодействия с пользователем.

4.2 Реализация серверной части

Реализация серверной части веб-приложения для нашего сайта погоды включала использование базы данных PostgreSQL и фреймворка Django. Этот выбор технологий обеспечил нам надежную и масштабируемую инфраструктуру для хранения и управления данными.

Для документации разрабатываемого REST API будет использоваться Swagger, предоставляющий набор инструментов, который позволяет автоматически описывать API на основе его кода [5].

База данных PostgreSQL была выбрана из-за своей надежности, расширенных возможностей и хорошей поддержки для веб-приложений. Она позволяет нам хранить и структурировать информацию о пользователях, учетных записях, а также о погодных данных, полученных из внешних источников.

С помощью Django, мощного и гибкого фреймворка разработки веб-приложений на языке Python, мы создали серверную часть нашего приложения. Django предоставляет набор инструментов и функций для обработки запросов, маршрутизации, аутентификации и авторизации пользователей, а также для работы с базой данных.

При реализации прогноза погоды был использован сервис OpenWeather который предоставляет, хоть и более точный прогноз, но только на 5 дней вперёд. Пришлось смириться с невозможностью предоставления прогноза погоды на 10 дней и на месяц, как планировалось изначально: получения API прогнозов погоды с Gismeteo. Изначально планировалось получать прогноз погоды на месяц с данного сервиса, но позже выяснилось, что сервис перестал предоставлять прогноз погоды. И Token представленный в разделе для разработчиков не является рабочим.

Доступ к информации о погоде был реализован таким образом, что любой посетитель сайта может свободно просматривать актуальные погодные данные. Однако, зарегистрированным пользователям предоставляется дополнительный функционал и возможность доступа к архивным данным о погоде. Они могут просматривать предыдущие записи о погоде, а также использовать дополнительные функции, такие как рекомендации по выбору одежды в зависимости от погодных условий.

Таким образом, реализация серверной части веб-приложения на базе Django с использованием PostgreSQL базы данных обеспечила нам надежность, безопасность и функциональность нашего сайта погоды. Пользователи могут свободно просматривать текущую погоду, а зарегистрированные пользователи имеют доступ к более расширенному функционалу, включая архивные данные и рекомендации по одежде на основе погоды.

4.3 Реализация базы данных

Для реализации базы данных в нашем веб-приложении погоды мы выбрали PostgreSQL, одну из наиболее надежных и мощных систем управления базами данных. PostgreSQL предоставляет широкий набор возможностей, включая поддержку сложных запросов, транзакций, целостности данных и масштабируемости [3].

С использованием Django, популярного фреймворка Python, мы легко интегрировали PostgreSQL в наше приложение. Django предоставляет ORM (Object-Relational Mapping), что позволяет нам работать с базой данных через объекты и классы Python, а не писать прямые SQL-запросы. Это значительно упрощает и ускоряет разработку и обслуживание базы данных.

Мы определили несколько моделей данных в Django, которые соответствуют сущностям нашего приложения, таким как пользователи, погода и архивы погоды. Каждая модель имеет свои поля, определяющие типы

данных и ограничения. Например, модель пользователя может содержать поля для имени, адреса электронной почты и пароля.

С помощью миграций Django мы создали необходимые таблицы в базе данных. Миграции представляют собой скрипты, которые автоматически создают, изменяют или удаляют таблицы и столбцы в базе данных в соответствии с определенными моделями и изменениями в них.

В процессе разработки мы использовали pgAdmin4, удобную среду администрирования для PostgreSQL. Это позволило нам взаимодействовать с базой данных, выполнять запросы, просматривать и редактировать данные. Мы также использовали мощные возможности PostgreSQL, такие как индексы, чтобы обеспечить быстрый доступ к данным и оптимизировать их производительность.

Таким образом, с помощью PostgreSQL, Django ORM и инструментов администрирования мы реализовали надежную и эффективную базу данных для нашего веб-приложения погоды. Это позволяет нам хранить, управлять и извлекать данные о пользователях, погоде и архивах погоды, обеспечивая функциональность и надежность всего приложения.

4.4 Реализация прогноза одежды

Для реализации прогноза одежды был использован подход на основе нечёткой логики (Fuzzy Logic). Нечёткая логика позволяет обрабатывать концепцию частичной истинности, где истинность может находиться в промежутке между полностью истинным и полностью ложным. Этот подход учитывает неопределённость и нечёткость в данных, что делает прогноз более гибким и адаптивным.

В контексте прогноза одежды, нечёткая логика позволяет учесть различные факторы, такие как температура, скорость ветра и осадки. Вместо того, чтобы предоставлять жёсткие рекомендации, наша система прогнозирования использует нечёткие правила, основанные на входных

данных. Это позволяет учесть разнообразные варианты и условия, которые могут влиять на выбор одежды.

Такой подход даёт возможность более точно оценить соответствие выбранной одежды конкретным входным данным и предоставить пользователю наиболее релевантные рекомендации.

Для начала было проведено исследование и практическое наблюдение, какая одежда для какой погоды подходит. Естественно, невозможно выбрать всевозможные вариации одежды для рекомендации их в виду обильного количества различных моделей. Нами же были выбраны следующие предметы одежды и аксессуаров: ветровка, куртка (легкая), куртка (теплая), пальто, пуховик, рубашка водолазка, свитер, майка, футболка, шорты/юбка, брюки, джинсы, теплые брюки, кепка/панама, шапка (легкая), шапка (теплая), зонт дождевик. Из этих предметов были составлены 29 комбинаций одежды, каждая из которых может являться результатом работы функции.

В качестве влияющих факторов (состояний) выступают:

— температура в диапазонах: -30…-14, -14…-10, -10…-6, -6…-2, -2…2, 2…10, 14…18, 18…22, 22…26, 26…30;

— скорость ветра в диапазонах: 0…10 (слабый), 10…20 (сильный);

— осадки в диапазонах: 0, 0…15, 15…30.

Если значение влияющих факторов выходит за диапазон, выдается рекомендация оставаться дома, т.к. погодные условия неблагоприятны.

На основе этого была построена модель в среде MATLAB с помощью Toolbox Fuzzy Logic:

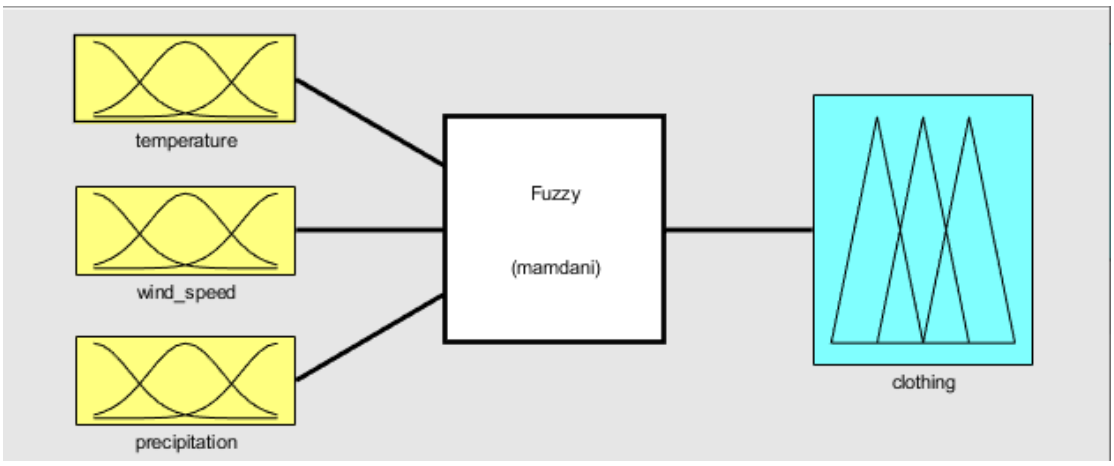

Рисунок 18 - Модель прогноза одежды

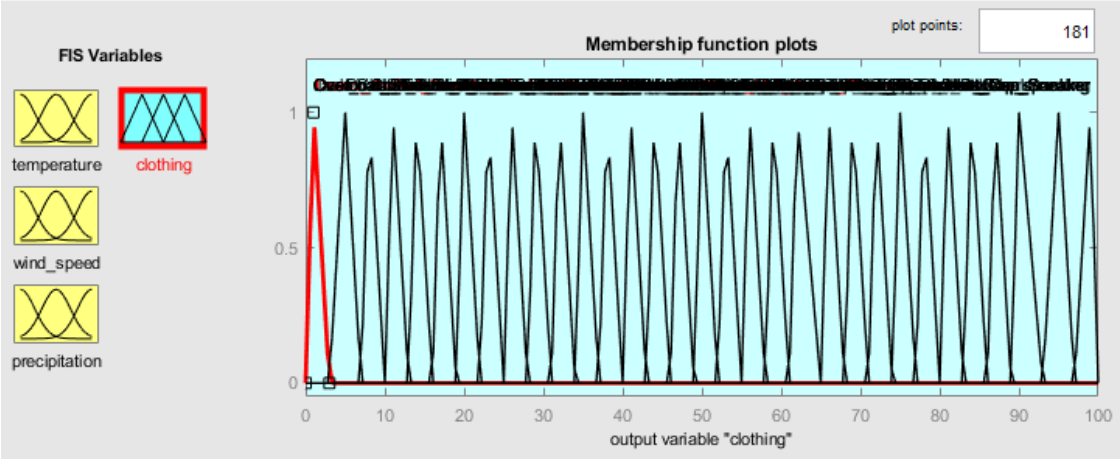

Рисунок 19 - График функции принадлежности для одежды

Для создания прогноза одежды был задан набор правил, на основе которых происходит рекомендация в соответствии с текущими условиями.

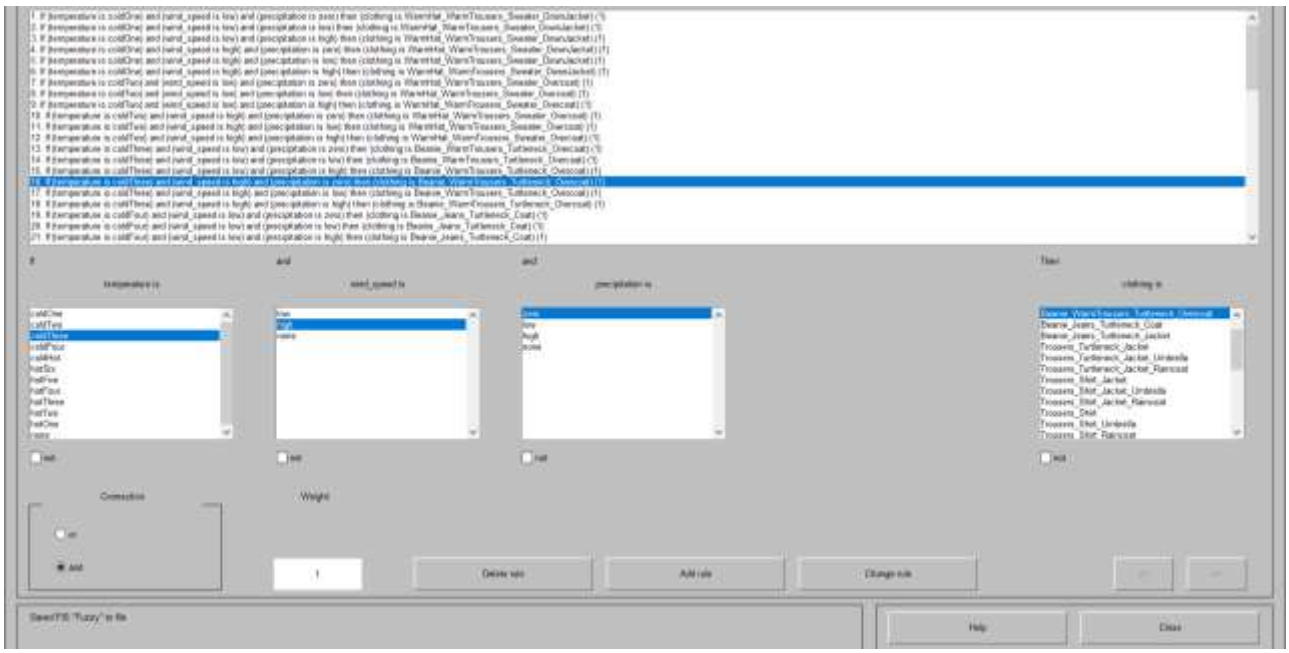

Рисунок 20 - Набор правил

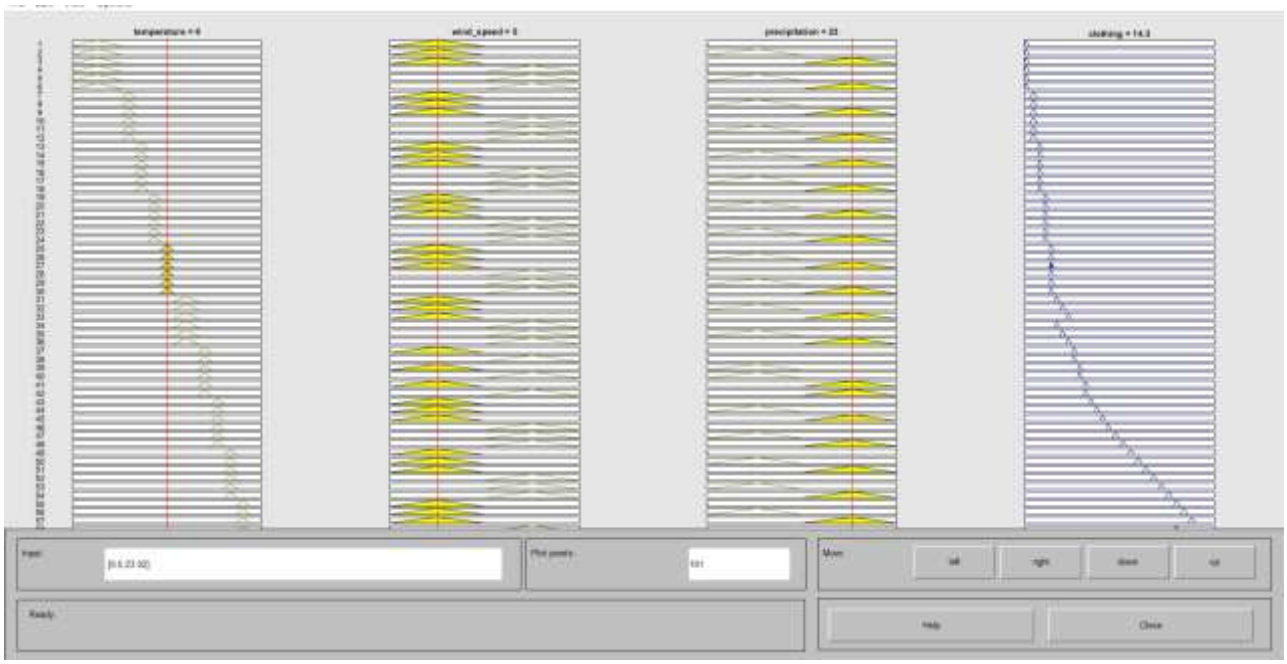

Рисунок 21 - Графическое представление набора правил

4.5 Реализация клиентской части

Для эффективной реализации основных сценариев веб-приложения команда разработчиков применила структурированный подход, разделяя клиентскую часть на отдельные страницы. Каждая из этих страниц является

результатом объединения языка программирования JavaScript, языка разметки HTML и инструментов, предоставляемых библиотекой React [2]. Это позволило создать функциональные и интуитивно понятные пользовательские интерфейсы, полностью соответствующие заданным требованиям.

Архитектура разработки была организована с учетом бизнес-логики проекта и применением модульного подхода, который способствует логическому группированию компонентов и логики взаимодействия. Компоненты размещены рядом друг с другом, что упрощает их обслуживание и расширение. Кроме того, благодаря использованию специального модуля для работы с файлами и их загрузки через файл index.js, все необходимые ресурсы и функции, утвержденные командой разработчиков дизайна, могут быть легко экспортированы для использования в проекте.

Реализация всех страниц веб-приложения была завершена практически в полном соответствии с желаемым дизайном и требованиями проекта. К сожалению, не для всех задумок хватило практических навыков и знаний.

Результатом труда команды является веб-приложение, которое обладает привлекательным и интуитивно понятным пользовательским интерфейсом. Все страницы приложения были протестированы и утверждены.

4.6 Методология

При реализации проекта была использована методология Whaterfall (каскадная модель). Следуя каскадной модели, разработчик переходит от одной стадии к другой строго последовательно. Сначала полностью завершается этап «определение требований», в результате чего получается список требований к ПО. После того как требования полностью определены, происходит переход к проектированию, в ходе которого создаются документы, подробно описывающие для программистов способ и план реализации указанных требований. После того как проектирование полностью выполнено, программистами выполняется реализация полученного проекта.

На следующей стадии процесса происходит интеграция отдельных компонентов, разрабатываемых программистами. После того как реализация и интеграция завершены, производится тестирование и отладка продукта; на этой стадии устраняются все недочёты, появившиеся на предыдущих стадиях разработки. После этого программный продукт внедряется и обеспечивается его поддержка — внесение новой функциональности и устранение ошибок.

5 Тестирование

При проведении тестирования разработанной системы были охвачены следующие типы тестов:

- дымовое тестирование;
- функциональные тесты (тестирование пользовательского интерфейса).

5.1 Дымовое тестирование

Дымовое тестирование выполняется для проверки основных функций и стабильности программного обеспечения после его обновления или внесения значительных изменений.

Цель дымового тестирования заключается в том, чтобы убедиться, что основные функции приложения или системы работают должным образом и не возникают критические ошибки после внесения изменений. Это важный этап тестирования, который проводится перед более подробными и глубокими тестами, чтобы выявить наиболее критические проблемы в короткие сроки.

При проведении дымового тестирования выполняются базовые сценарии использования или тестовые случаи, которые охватывают основные функции и ключевые пути в приложении. Тесты обычно являются автоматизированными и могут включать такие проверки, как запуск приложения, проверка отображения интерфейса, выполнение базовых операций и проверка результата.

В таблице 1 представлены результаты дымового тестирования для основных сценариев.

Таблица 1 - Результаты дымового тестирования

Тестовый сценарий	Результат теста
Регистрация	Пройден

Авторизация	Пройден
Просмотр главной страницы	Пройден
Просмотр страницы с “завтрашним” прогнозом погоды	Пройден
Просмотр страницы с прогнозом погоды на 5 дней	Пройден
Просмотр страницы статистики прошедшей погоды	Пройден
Просмотр страницы статистики аномальной погоды	Пройден
Просмотр страницы в админ панели для загрузки рекламы	Пройден
Просмотр страницы в админ панели для обработки данных пользователей	Пройден
Просмотр страницы личного кабинета пользователя	Пройден

5.2 Функциональное тестирование

Функциональное тестирование в виде тестирования пользовательского интерфейса (UI тестирование) является процессом проверки соответствия программного обеспечения его функциональным требованиям. Оно фокусируется на проверке работы системы с точки зрения ее внешнего поведения и взаимодействия с пользователем.

Цель функционального тестирования заключается в том, чтобы убедиться, что приложение или система работает в соответствии с заданными функциональными требованиями и ожиданиями пользователей. Это включает проверку правильности работы интерфейса, функциональности приложения,

обработку пользовательского ввода, отображение данных и выполнение заданных операций.

В таблице 2 представлена часть результатов тестирования пользовательского интерфейса. В ней отражены тестовые сценарии для неавторизованного пользователя.

Таблица 2 - Результаты UI-тестирования для неавторизованного пользователя в демо-режиме

Тестовый сценарий	Ожидаемый результат	Статус теста
Нажатие на кнопку «Завтра» на главной странице	Переход на страницу с прогнозом на завтра	Пройден
Нажатие на кнопку «На 5 дней» на главной странице	Переход на страницу с прогнозом на 5 дней	Пройден
Выбрать дату на главной странице	Переход на страницу с прогнозом на выбранную дату	Пройден
Нажатие на кнопку «Зарегистрируйтесь»	Переход на страницу регистрации	Пройден
Нажатие на кнопку «Войти»	Переход на главную страницу с авторизацией пользователя	Пройден
Нажать выбрать страну на странице «статистика прошедшей погоды»	Выпадает список всех стран	Пройден

Нажать выбрать город на странице «статистика прошедшей погоды»	Выпадает список всех городов страны	Пройден
Выбрать даты странице «статистика прошедшей погоды»	Выдаёт список прошедшей погоды в выбранном городе, стране и в выбранный период	Пройден
Нажать выбрать страну на странице «статистика аномальной погоды»	Выпадает список всех стран	Пройден
Нажать выбрать город на странице «статистика аномальной погоды»	Выпадает список всех городов страны	Пройден
Выбрать годы странице «аномальной прошедшей погоды»	Выдаёт список прошедшей погоды в выбранном городе, стране и в выбранный период	Пройден
Нажатие на ссылку «Уже есть аккаунт? Войдите» на странице регистрации	Переход на страницу авторизации	Пройден
Нажатие на ссылку «Еще нет аккаунта? Зарегистрируйтесь» на странице авторизации	Переход на страницу регистрации	Пройден

Заключение

В данной работе была представлена разработка системы, которая имеет целью обеспечить пользователей информацией о погодных условиях и помочь им принимать соответствующие решения. Современный мир все больше зависит от прогноза погоды, и данная система стремится облегчить жизнь людей, предоставляя им надежную и актуальную информацию о погоде.

Одним из ключевых аспектов системы является хранение статистики погодных наблюдений. Это позволяет проводить анализ и использовать исторические данные для составления прогнозов погоды на ближайшие дни. Точность и достоверность прогнозов являются важными факторами, которые влияют на принятие решений в различных отраслях экономики, таких как авиация, морской флот, сельское хозяйство и энергетика.

Система также предоставляет рекомендации по одежде, исходя из прогнозируемых погодных условий. Это позволяет пользователям быть готовыми к изменениям погоды и подобрать подходящую одежду для комфортного пребывания на улице. Кроме того, система обращает внимание на аномальные погодные явления.

В заключение, разработанная система является ценным инструментом, обеспечивающим информацией о погоде, помогающим принимать решения и адаптироваться к изменяющимся условиям. Она вносит вклад в безопасность и эффективность различных сфер деятельности, а также повышает качество жизни людей, обеспечивая им надежную информацию о погоде и соответствующие рекомендации.

Список использованных источников

1. Next.js Documentation: [Электронный ресурс]. — Режим доступа: https://nextjs.org/docs. — Заглавие с экрана. — (Дата обращения: 25.04.2023).

2. Что такое React и как его освоить? [Электронный ресурс]. — Режим доступа: https://academy.yandex.ru/journal/chto-takoe-react-i-kak-ego-osvoit. — Заглавие с экрана. — (Дата обращения: 29.04.2023).

3. Система управления объектно-реляционными базами данных PostgreSQL [Электронный ресурс]. — Режим доступа: https://web-creator.ru/technologies/webdev/postgresql. — Заглавие с экрана. — (Дата обращения: 2.05.2023).

4. Моуэт, Э. Использование Docker: Разработка и внедрение программного обеспечения при помощи технологии контейнеров. — Перевод с английского. — Москва: Издательство: «ДМК Пресс», 2019. - 156 с.

5. Тестирование API с помощью Swagger: особенности и преимущества [Электронный ресурс]. — Режим доступа: https://blog.ithillel.ua/ru/articles/api-testing-with-swagger. — Заглавие с экрана. — (Дата обращения: 14.05.2023).