

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Веб-приложение для просмотра погоды
«WeatherApp»

Курсовой проект по дисциплине Технологии программирования
09.03.02 Информационные системы и технологии
Программная инженерия в информационных системах

6 семестр 2022/2023 учебного года

Обучающийся _____ Н.В. Щербаков, 3 курс

Руководитель _____ И.В. Клейменов, ассистент

Руководитель _____ В.С. Тарасов, ст. Преподаватель __.__.20__

Воронеж 2023

СОДЕРЖАНИЕ

Введение	3
1 Постановка задач.....	3
1.1 Требования к разрабатываемой системе	4
1.1.1 Функциональные требования	4
1.1.2 Технические требования	6
1.2 Требования к интерфейсу	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	10
2.2.1 Гисметеo.....	10
2.2.2 Яндекс.Погода.....	11
2.3 Моделирование системы	12
2.3.1 Диаграмма прецедентов	12
2.3.2 Диаграмма активностей	13
2.3.3 Диаграмма сотрудничества	13
2.3.4 Диаграмма объектов	14
2.4 Границы проекта	14
3 Реализация	16
3.1 Средства реализации.....	16
3.2 Реализация серверной части веб-приложения.....	17
3.3 Реализация клиентской части веб-приложения	17
4 Тестирование	19
4.1 Ручное UI–тестирование.....	19
4.2 Юзабилити–тестирование.....	21
Заключение	25
Список используемых источников	26

Введение

В современном мире, где доступ к информации играет важную роль в повседневной жизни, приложения о погоде становятся все более популярными и востребованными. Они предоставляют пользователю актуальные данные о погодных условиях, помогая ему принимать осознанные решения и планировать свои действия. Однако, чтобы удовлетворить все потребности пользователей, требуется разработка инновационных веб-приложений, способных предоставлять более широкий спектр информации о погоде.

Целью данной курсовой работы является разработка веб-приложения о погоде с расширенным функционалом, включающим просмотр почасовой погоды, и гороскопа. Веб-приложение будет предоставлять пользователю не только основные данные о текущей температуре и погодных условиях, но и детальный почасовой прогноз, позволяющий лучше планировать свои действия на протяжении дня.

В дополнение к информации о погоде, разработанное приложение будет предлагать гороскоп на каждый день, основанный на знаке зодиака пользователя. Это позволит пользователю получать рекомендации и советы, связанные с погодными условиями и настроением, чтобы лучше подготовиться к своим планам и событиям.

Данная курсовая работа включает в себя анализ требований пользователей, проектирование архитектуры веб-приложения, разработку интерфейса, реализацию функционала. В конечном итоге, разработанное веб-приложение будет предоставлять пользователю удобный и информативный опыт, связанный с погодой, прогнозами, гороскопом.

Используя современные технологии разработки веб-приложений, эта работа стремится предложить инновационное решение для удовлетворения потребностей пользователей в области получения информации о погоде и связанных с ней аспектах.

1 Постановка задач

Целью данного проекта является обеспечение пользователей точной и полезной информацией о погоде, чтобы помочь им принимать информированные решения и планировать свои действия.

Основными задачами проекта являются создание максимально комфортного и удобного веб-приложения о погоде, которое будет понятно всем. Важно реализовать функцию гороскопа, потому что обычно на сайтах с прогнозом погоды данная функция отсутствует. Наш проект рассчитан на массового потребителя, поэтому добавления гороскопа придаст нашему приложению элемент развлечения. Пользователи смогут соотносить прогноз погоды с своим гороскопом и составлять более конкретные планы на день.

Также важно реализовать почасовой прогноз погоды, чтобы люди были заранее предупреждены о неблагоприятных погодных условиях и времени, когда вернется хорошая погода. Данная функция очень полезна для соблюдения безопасности.

Для достижения поставленных целей необходимо иметь представление о разрабатываемой системе, представленное необходимыми UML-диаграммами и разработанным дизайном веб-приложения.

Система будет состоять из двух основных компонентов: backend (серверная) часть и frontend (клиентская) часть, которые будут взаимодействовать посредством REST API.

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

К разрабатываемому приложению выдвинуты следующие требования:

- разделение пользователей на авторизованного неавторизованного
- обеспечение возможности просмотра почасовой погоды;
- обеспечение просмотра погоды для любого города путём ввода наименования города в поиске;
- обеспечение возможности просмотра гороскопа для авторизованного пользователя;
- обеспечение возможности авторизации пользователя;

— обеспечение возможности просмотра рекомендаций по погоде для всех пользователей (авторизованного и неавторизованного).

1.1.2 Технические требования

Программный продукт должен обеспечить:

- Авторизацию пользователей;
- Удобное пользование не только с ПК, но и с телефона;
- хранение данных в БД.

1.2 Требования к интерфейсу

Для всех экранов приложения необходимо использовать одну цветовую гамму и стиль, чтобы создать единообразный внешний вид. Все текстовые элементы должны быть хорошо читаемыми, а элементы управления должны иметь одинаковый стиль и размер, чтобы они привлекали внимание на фоне остального содержимого экранов.

Интерфейс приложения должен содержать только необходимую информацию для пользователя. Информация должна быть размещена на соответствующих местах, где она имеет актуальность и значение. Основные элементы управления должны быть хорошо видимыми и легко обнаруживаемыми для пользователей.

В целом, главная задача интерфейса - предоставить пользователю только необходимую информацию и сделать ее доступной и удобной для восприятия. При этом все элементы интерфейса должны быть согласованы по стилю и размеру, чтобы обеспечить единообразный и легко узнаваемый вид приложения на всех экранах.

1.3 Задачи, решаемые в процессе разработки

Перед проектом были поставлены следующие задачи:

- анализ предметной области;
- анализ аналогов;
- написание технического задания;
- описание разрабатываемой системы UML диаграммами;
- разработка функциональных возможностей сайта;

- создание макета дизайна и его реализация;
- реализация интерфейса;
- описание процесса разработки и результата.

2 Анализ предметной области

2.1 Терминология (гlossарий) предметной области

Веб-приложение - программа, которая работает в браузере и используется для выполнения различных задач через Интернет.

Пользователь - человек, который использует веб-приложение для получения информации о погоде, просмотра гороскопа. Текущая погода - совокупность погодных условий (температура, влажность, скорость ветра, давление, состояние погоды), актуальных на данный момент времени.

Прогноз погоды - предсказание погодных условий на ближайшие дни, основанное на данных с погодных станций и моделях прогнозирования погоды.

API погодных сервисов - программный интерфейс, предоставляемый сервисами погоды (например, OpenWeatherMap, Weather API), который позволяет получать данные о погоде.

Гороскоп - прогноз или предсказание, основанное на положении звезд и планет в определенное время.

Знак зодиака - одна из двенадцати равных частей зодиакального круга, в которую входит Солнце в течение года.

Front-end – клиентская часть приложения. Отвечает за получение информации с программно-аппаратной части и отображение ее на устройстве пользователя.

Back-end – программно-аппаратная часть приложения. Отвечает за функционирование внутренней части приложения.

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

Фреймворк – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

REST API - архитектурный стиль интерфейсов, основанный на принципах HTTP, который обеспечивает возможность взаимодействия между клиентом и сервером.

JavaScript - язык программирования, используемый для создания интерактивных элементов на веб-страницах.

HTML - язык разметки для создания структуры веб-страниц.

MongoDB - NoSQL база данных, используемая для хранения данных в формате документов.

Bootstrap - набор инструментов для создания дизайна сайта и веб-приложений.

HTTPS - протокол безопасной передачи данных, который использует шифрование для обеспечения безопасности информации.

2.2 Обзор аналогов

2.2.1 Гисметео

"Gismeteo" — это популярный веб-сайт и сервис прогноза погоды, ориентированный на русскоязычную аудиторию. Gismeteo предоставляет подробные прогнозы погоды для множества городов по всему миру, включая информацию о температуре, влажности, скорости и направлении ветра, давлении, фазах Луны и другую связанную с погодой информацию.



Рисунок 1 - Интерфейс сайта «Гисметео»

Недостатками сайта Гисметео являются:

- Мобильный опыт. Возможно, мобильная версия сайта Гисметео не всегда оптимально адаптирована для использования на мобильных устройствах. Некоторые пользователи могут столкнуться с проблемами отображения или неудобством при использовании сайта на мобильных устройствах.

— Ограниченная информация. Некоторые пользователи отмечают, что Гисметео может предоставлять ограниченную информацию о погоде, особенно по сравнению с другими сервисами. Например, почасовой прогноз погоды может быть недостаточно подробным или отсутствовать в некоторых регионах.

2.2.2 Яндекс.Погода

Яндекс.Погода - это сервис, предоставляемый Яндексом, для получения информации о погоде. Он предоставляет актуальные данные о текущей погоде, прогноз на несколько дней вперед и почасовой прогноз. Сервис доступен через веб-интерфейс и мобильные приложения. Яндекс.Погода также предлагает функции по поиску погоды для различных городов.

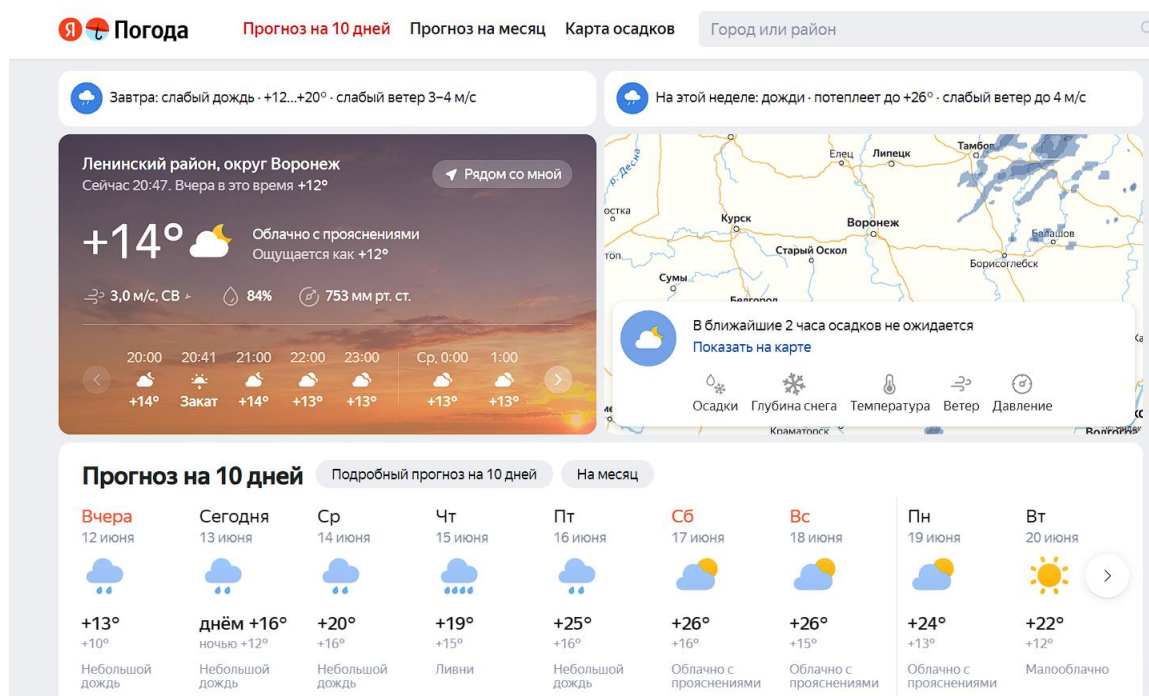


Рисунок 2 - Интерфейс сайта «Яндекс.Погода»

Недостатками сайта Яндекс.Погода являются:

- наличие большого количества рекламы;
- Неудобный интерфейс.

2.3 Моделирование системы

2.3.1 Диаграмма прецедентов

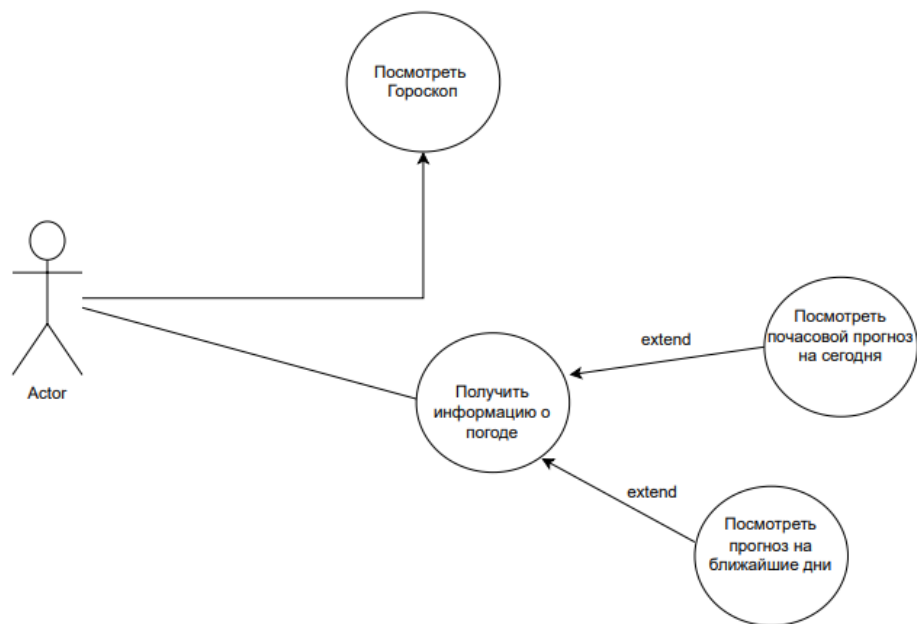


Рисунок 3 - Диаграмма прецедентов

2.3.2 Диаграмма активностей

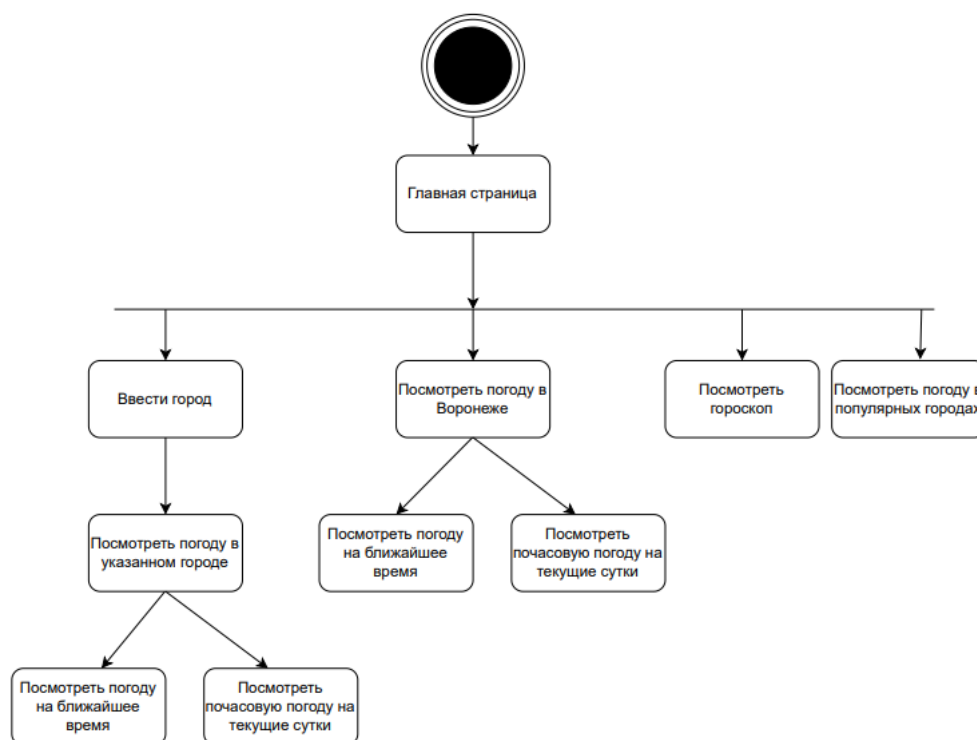


Рисунок 4 - Диаграмма активностей

2.3.3 Диаграмма сотрудничества

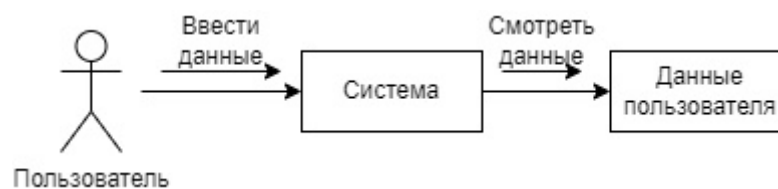


Рисунок 5 - Диаграмма сотрудничества

2.3.4 Диаграмма объектов

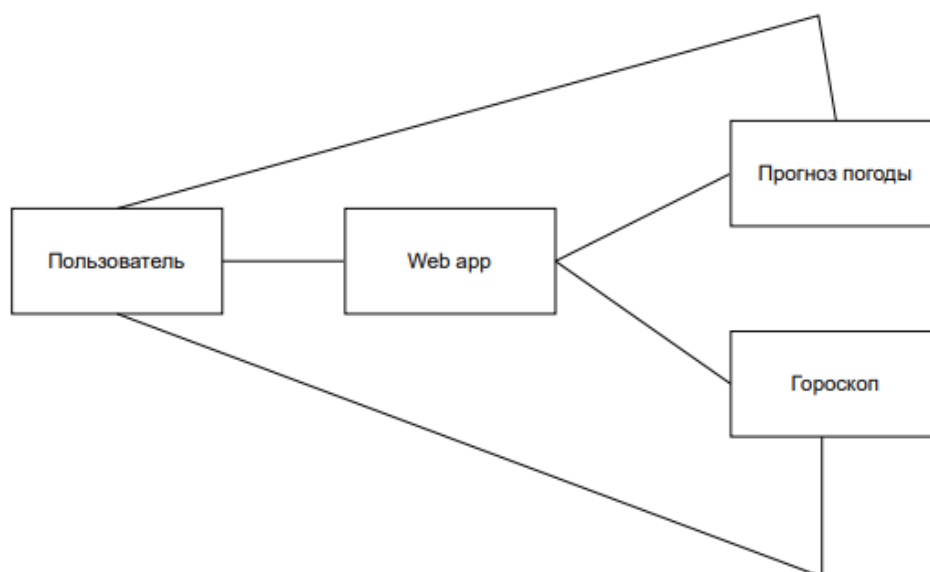


Рисунок 6 - Диаграмма объектов

2.4 Границы проекта

Основными границами проекта являются:

- Реализация функционала для просмотра актуальной погоды на текущий момент и прогноза на несколько дней вперед, включая температуру, осадки, влажность, ветер и другие показатели.
- Возможность просмотра почасового прогноза погоды для более точного планирования.
- Интеграция гороскопа, предоставляющего информацию о знаке зодиака и предсказаниях для каждого дня.
- Согласование с пользователями ограничений и возможностей приложения в рамках их потребностей и ожиданий.
- Ограниченный бюджет и сроки разработки, требующие эффективного управления ресурсами для успешного завершения проекта.

- Стабильное и надежное взаимодействие с погодным API для получения актуальной информации о погоде.
- Разработка удобного и интуитивно понятного пользовательского интерфейса с четким отображением информации о погоде, графическими элементами и элементами управления.
- Возможность масштабирования и расширения функционала приложения для будущих улучшений и добавления новых возможностей.

3 Реализация

3.1 Средства реализации

Веб-приложение имеет архитектуру, соответствующую шаблону клиент-серверного приложения и разделяется на back-end и front-end посредством REST API.

Данная архитектура веб-приложения соотносится с основными требованиями к проекту, а именно:

- отображение текущей погоды. Приложение должно показывать текущую погоду с температурой. Информация должна быть ясной и понятной для пользователей.
- почасовой прогноз погоды. Пользователи должны иметь возможность просматривать прогноз погоды на несколько ближайших часов. Приложение должно отображать температуру, возможные осадки.
- гороскоп. Пользователи могут заинтересоваться астрологическими прогнозами. Приложение должно предоставлять гороскоп для каждого знака зодиака. Гороскоп должен быть обновляемым и содержать соответствующую информацию для каждого дня.
- интуитивный пользовательский интерфейс. Приложение должно иметь простой и интуитивно понятный пользовательский интерфейс. Пользователи должны легко находить необходимую информацию о погоде, гороскопе.
- актуальность данных. Приложение должно обновлять информацию о погоде, гороскопе. Обновления должны происходить регулярно, чтобы предоставлять пользователям актуальную информацию.

Для реализации серверной части были выбраны следующие технологии:

- язык программирования JavaScript;
- Node JS
- MongoDB.

Для реализации клиентской части были выбраны следующие технологии:

- язык программирования JavaScript;
- язык разметки HTML.
- фреймворк Bootstrap;

3.2 Реализация серверной части веб-приложения

Для осуществления основных сценариев веб-приложения: просмотра погоды, гороскопа, был разработан запросы, которые предоставляют информацию о погоде и гороскопе.

Сервер хранит в себе данные о гороскопе по знаку зодиака, сохраняет гороскоп в базе данных.

Также, для удобства работы с сервером подключен swagger, в котором видны все ссылки.

3.3 Реализация клиентской части веб-приложения

Для реализации основных сценариев веб-приложения, клиентская часть разработки делится постранично. Каждая страница описывается языком программирования JavaScript, языком разметки HTML.

Архитектура разработки была организована согласно бизнес-логике проекта на основании модульного подхода, по которому все компоненты и логика находятся рядом друг с другом.

Все страницы веб-приложения были реализованы и представлены командой разработчиков в соответствии с заявленным дизайном и соответствующими правками, внесенными в процессе разработки системы.

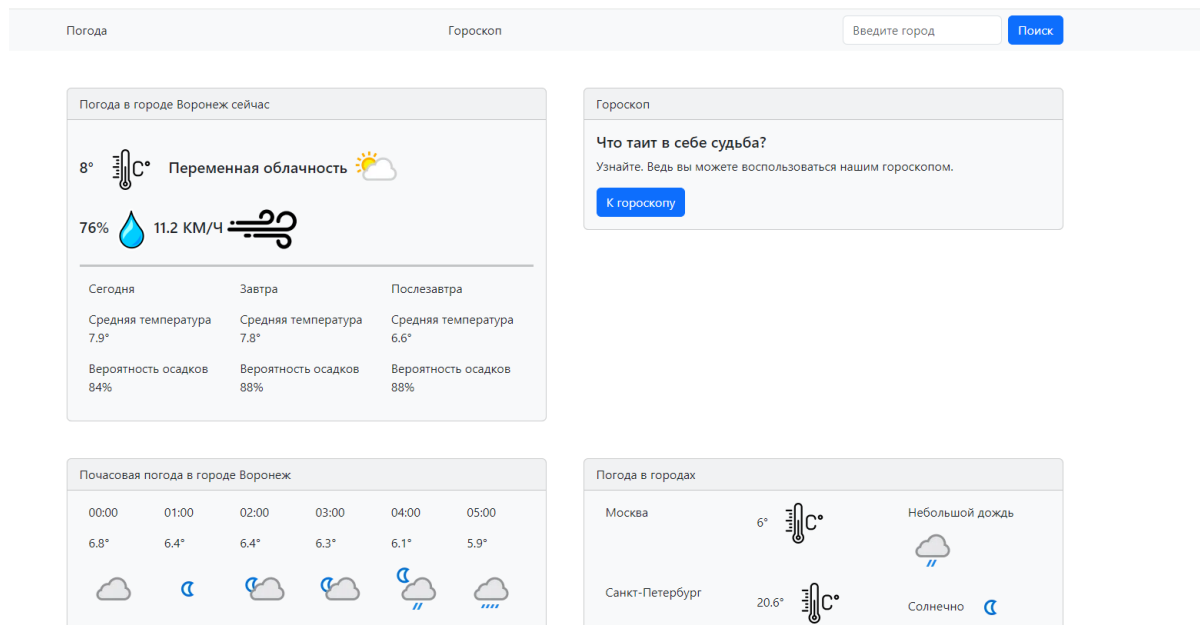


Рисунок -1 Интерфейс погоды

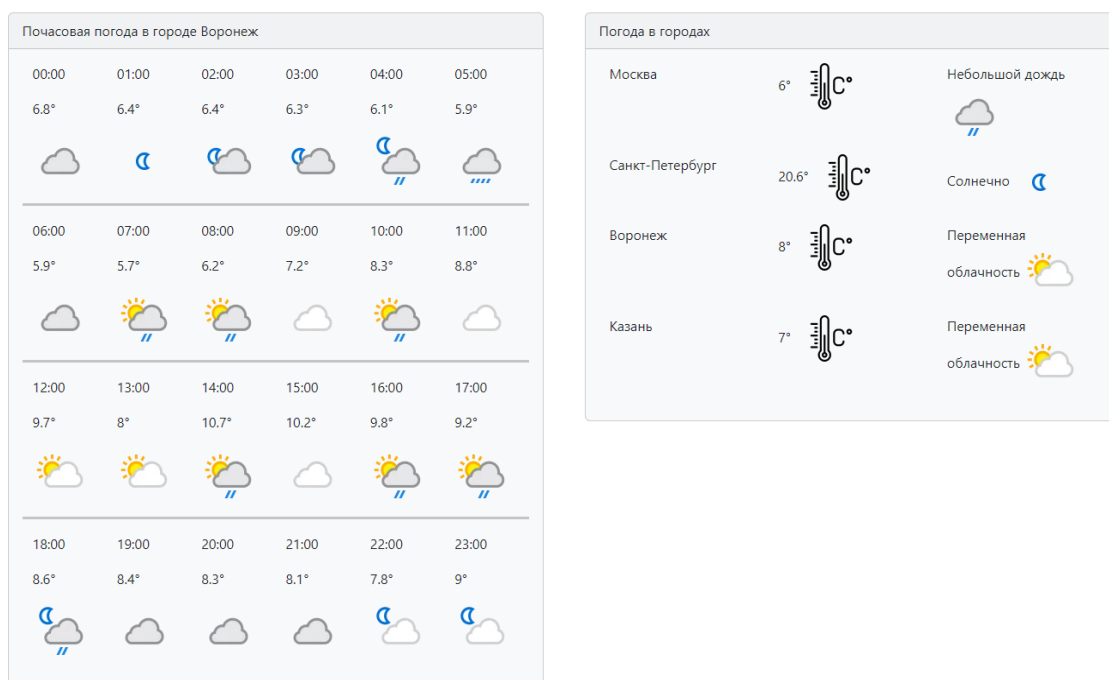


Рисунок -2 Интерфейс погоды

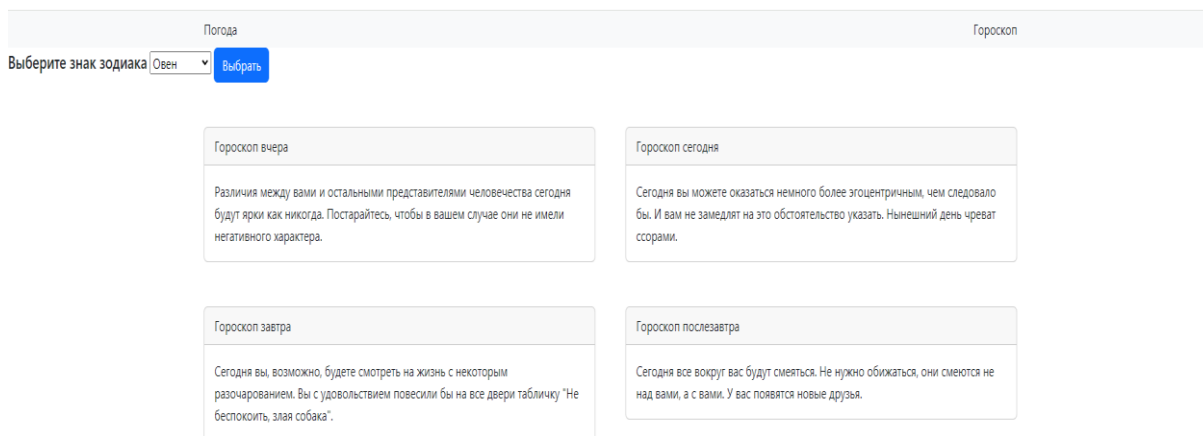


Рисунок -3 Интерфейс гороскопа

4 Тестирование

4.1 Ручное UI–тестирование

UI-тестирование — этап комплексного тестирования программного обеспечения

В выборе варианта UI–тестирования мы остановились на ручном, так как, данный вариант является наиболее эффективным, поскольку обеспечивает наилучшее покрытие тестами.

В работе были рассмотрены следующие аспекты данного тестирования:

- визуальные характеристики — проверка адаптивности;
- композиция — проверка корректного отображения компонентов на странице;
- взаимодействие — обрабатываются ли события так, как задумано
- доступность — проверка корректного расположения компонентов на странице и возможность правильного взаимодействия с ними;
- пользовательские потоки — проверка полного прохода по сценарию.

Для каждого аспекта (кейса) тестирования были введены критерии успешности прохождения UI–тестирования для той или иной страницы веб-приложения:

- 0 – тестируемая страница полностью не прошла тестирование;
- 1 – тестируемая страница частично не прошла тестирование;
- 2 – тестируемая страница полностью прошла тестирование.

Тестирование веб–страниц для сценария основных функций веб-приложения приведено в соответствии с таблицей 1.

Таблица 1 -Тестирование основных функций веб-приложения

Кейс	Страница погоды
Визуальные характеристики	1
Композиция	2
Взаимодействие	2
Доступность	2

Тестирование веб–страниц для сценария использования гороскопа приведено в соответствии с таблицей 2

Таблица 2 -Тестирование основных функций веб-приложения

Кейс	Страница гороскопа
Визуальные характеристики	1
Композиция	2
Взаимодействие	2
Доступность	2

4.2 Юзабилити–тестирование

Юзабилити–тестирование является исследованием, выполняемым с целью определения, удобен ли некоторый искусственный объект для его предполагаемого применения.

Объектом тестирования является веб–приложение просмотра прогноза погоды с возможностью ознакомления с гороскопом:

- порога вхождения;
- прохождение сценариев;

- степени ориентированности в веб-приложении во время прохождения сценариев.

Первым сценарием для тестирования:

- сценарий 1 «Основные функции веб-приложения»;

Для каждой единицы информации были введены критерии успешности выполнения того или иного задания:

- 0 – респондент не справился с заданием;
- 1 – респондент справился с заданием со значительными затруднениями;
- 2 – респондент справился с заданием и перешел на следующий шаг.

Респонденты, отобранные для тестирования, были разделены по возрасту и полу. Тем самым в исследовании приняли участие:

- двое молодых людей возраста двадцати лет;
- двое молодых девушек возраста двадцати лет;
- двое мужчин с средним возрастом в 40 лет;
- двое женщин с средним возрастом в 40 лет.

Данное распределение для тестирования было выбрано с целью выявления преимуществ и недостатков в уже разработанном веб-приложении при работе с ним разных лиц пользователей.

Результаты юзабилити-тестирования переведены в соответствии с таблицей 1

Таблица 3 -Результаты юзабилити–тестирования

Как Вас зовут?	Ваш пол?	Сколько Вам лет?	Использовал и ли вы раньше приложения для просмотра погоды?	Если Вы ответили утвердительно на предыдущий вопрос, оцените удовлетворенность использованием такого приложения:
Андрей	Мужской	18-25 лет	Да	1
Егор	Мужской	18-25 лет	Да	1
Дарья	Женский	18-25 лет	Да	0
Елизавета	Женский	18-25 лет	Да	1
Евгений	Мужской	40+ лет	Нет	—
Сергей	Мужской	40+ лет	Да	1
Ольга	Женский	40+ лет	Нет	2
Мария	Женский	40+ лет	Да	2

Таблица 4 -Результаты юзабилити–тестирования

Сценарий 1 «Основные функции веб-приложения»
2
2
2
2
2
2
2

Таблица 5 -Результаты юзабилити–тестирования

Как Вас зовут?	Оцените общее впечатление от работы с веб-приложением погода:
Андрей	2
Егор	2
Дарья	2
Елизавета	2
Евгений	2
Сергей	2
Ольга	2
Мария	2

Заключение

В ходе выполнения курсового проекта командой было разработано веб-приложение погода с возможностью, соответствующее поставленным перед проектом задачам.

В начале разработки был проведен анализ предметной области, определены основные требования к разрабатываемой системе, определены основные сценарии веб-приложения.

По результатам разработки проводился ряд тестов с целью проверки работоспособности системы.

В процессе работы были реализованы следующие задачи проекта:

- обеспечение возможности просмотра почасовой погоды;
- обеспечение возможности ввода города и просмотра погоды по выбранному городу;
- обеспечение возможности просмотра гороскопа;

Список используемых источников

1. Eloquent JavaScript - Marijn Haverbeke [Электронный ресурс]. Режим доступа: <https://eloquentjavascript.net/> - Заглавие с экрана. (Дата обращения: 26.05.2023).
2. OpenWeatherMap API: API Documentation [Электронный ресурс]. Режим доступа: [https:// openweathermap.org/api](https://openweathermap.org/api) - Заглавие с экрана. (Дата обращения: 27.05.2023).
3. JavaScript [Электронный ресурс]. Режим доступа: <https://javascript.com> - Заглавие с экрана. - (Дата обращения: 26.04.2023).
4. MongoDB [Электронный ресурс]. Режим доступа: <https://mongodb.com> – Заглавие с экрана. – (Дата обращения: 31.05.2023)
5. Документация JavaScript-библиотеки React.js [Электронный ресурс]. Режим доступа: <https://ru.react.js.org/docs/getting-started.html>
6. Основы HTML [Электронный ресурс] Режим доступа: <https://html5book.ru/osnovy-html/>
7. Юзабилити-тестирование [Электронный ресурс]. Режим доступа: <https://lpgenerator.ru/blog/2016/07/28/yuzabiliti-testirovanie-poshagovaya-instrukciya-na-primere-yelp/>
8. ГОСТ 7.32-2001 [Электронный ресурс]. Режим доступа: https://kpfu.ru/portal/docs/F1867381138/gost7_32_2001.pdf