|  |  |
| --- | --- |
| **­­** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ \_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ДОМАШНЕМУ ЗАДАНИЮ***

***НА ТЕМУ:***

***Прогноз погоды. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_ИУ5-51Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_Ларкин Б. В.\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_А.И. Канев\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

# АННОТАЦИЯ

Расчётно-пояснительная содержит 26 страниц. С приложениями объем составляет 43 страницы. Работа включает в себя 8 диаграмм и 26 изображений системы. В процессе выполнения было использовано 8 источников.

Объектом разработки является система для прогнозирования погоды по введенным значениям в виде предсказаний.

Цель работы заключается в создании набора программного обеспечения и сопутствующих веб-сервисов для оформления предсказаний погоды.

В работе была разработана архитектура веб сервиса, разработан интерфейс для общения с данным веб сервисом, разработан и развернут веб сервер, нативное приложение и прогрессивное веб приложение, способные общаться с данным веб сервисом.

Пояснительная записка содержит 2 приложения.

**СОДЕРЖАНИЕ**

[АННОТАЦИЯ 2](#_Toc185715032)

[ВВЕДЕНИЕ 4](#_Toc185715033)

[1 ПРЕДМЕТНАЯ ОБЛАСТЬ 6](#_Toc185715034)

[2 АРХИТЕКТУРА 9](#_Toc185715035)

[3 АЛГОРИТМЫ 13](#_Toc185715036)

[4 ОПИСАНИЕ ИНТЕРФЕЙСА 15](#_Toc185715037)

[ЗАКЛЮЧЕНИЕ 22](#_Toc185715038)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23](#_Toc185715039)

[ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ 24](#_Toc185715040)

[1. Введение 25](#_Toc185715041)

[2. Назначение разработки 25](#_Toc185715042)

[3. Стадии и этапы разработки 25](#_Toc185715043)

[4. Требования к функциональным характеристикам 26](#_Toc185715044)

[5. Требования к составу и параметрам технических средств 31](#_Toc185715045)

[6. Требования к информационной и программной совместимости 31](#_Toc185715046)

[ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ 34](#_Toc185715047)

# ВВЕДЕНИЕ

В современном мире технологии предсказания погоды глубоко проникли как в повседневные жизни людей, так и в более научные, промышленные области. В связи с этим актуальна идея разработки систем прогнозирования с универсальным и удобным, дружелюбным пользователю интерфейсом. Сайты корпораций представляют удобный и быстрый доступ пользователей к прогнозированию погоды, однако, они не позволяют пользователям самим вводить данные измерений счетчиков для последующего предсказания. Такой функционал, очевидно, может быть необходим при отсутствии актуальных данных по некоторой местности, при недоступности средств передачи данных, или при необходимости получения точных значений. С этим многие существующие системы не справятся, в отличие от разрабатываемой мной.

Целью данной системы является автоматизация процессов подсчета предсказаний, включая управление операциями внесения и изменения измерений, а также поддержку различных видов прогнозов. Система должна предоставлять удобный интерфейс для метеорологов, позволяя им создавать заявки на предсказания, отслеживать их статус и получать актуальную информацию о доступных прогнозах. Кроме того, система должна обеспечивать эффективное управление предсказаниями со стороны синоптиков, включая обработку предсказаний, контроль за состоянием предсказаний и поддержку различных видов прогнозов.

Назначение системы заключается в обеспечении удобного и безопасного взаимодействия между метеорологами и синоптиками. Метеорологи по собственным измерениям смогут создавать заявки на предсказания погоды. Синоптики получат возможность просматривать список всех заявок, обрабатывать их и контролировать выполнение расчетов. Кроме того, система будет предоставлять актуальную информацию о различных видах прогнозов.

Такая система позволит реализовать системный подход к управлению предсказаниями в приложении, обеспечивая удобство, прозрачность и высокую эффективность работы. Метеорологи получат быстрый и безопасный доступ к вычислительным услугам, а система прогнозирования сможет эффективно управлять своими ресурсами и обеспечивать высокий уровень обслуживания.

Нефункциональные требования к разрабатываемой системе:

1. Должна поддерживать кроссплатформенность.
2. Интерфейс системы и текст ошибок должны быть русифицированы.

В ходе работы необходимо выполнить следующие задачи:

1. Разработать дизайн приложения в Figma на основе yandex.weather. Ознакомиться с разработкой бэкенда с использованием фреймворка Gin для Go.
2. Разработать структуру и создать базу данных PostgreSQL, подключить её к бэкенду.
3. Создать веб-сервис со всей итоговой бизнес-логикой, кроме авторизации, в бэкенде системы для использования его в SPA.
4. Добавить авторизацию, использовать Redis для хранения сессий, а также внедрить Swagger в веб-сервис.
5. Разработать базовый интерфейс приложения для гостя на React.
6. Внедрить менеджер состояний Redux Toolkit для хранения значений фильтров, добавить адаптивность и PWA.
7. Завершить разработку интерфейса пользователя в React, использовать для обращений к методам веб-сервиса Axios.
8. Реализовать React интерфейс издатель, внедрить Real-time web.
9. Разработать десктопное приложение на Tauri.
10. Развернуть приложение на GitHub Pages.
11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код проекта.

# 1 ПРЕДМЕТНАЯ ОБЛАСТЬ

Метеорологу предоставляется возможность выбирать прогнозы, тем самым, формируя черновик предсказания. После добавления в операцию, как минимум одного прогноза, метеоролог может перейти к его редактированию. Для успешной реализации предсказания требуется указать величину окна расчета, количество предсказаний, и измерения для каждого из прогнозов. При необходимости из операции можно удалить ненужный прогноз. Более того, на этапе редактирования предсказание может быть полностью удалено его владельцем.

После того, как метеоролог заполнил данные в предсказании и закончил выбор интересующих его прогнозов, он может сформировать запрос на предсказание, т.е. отправить его в работу. После формирования предсказание доступно только для просмотра информации о нем, редактирование невозможно. Метеоролог может отслеживать изменение статусов всех своих заявок.

Синоптик отслеживает запросы, формируемые метеорологами в системе. Он может фильтровать их по имени владельца, статусу и дате оформления.

Источник для информации о системе по оформлению предсказаний представлен в [1], откуда была взята основная информация и дизайн системы.

Если предсказание одобрено продавцом, то система подсчитывает результат и завершает его, в случае невозможности реализации – отклоняет. После завершения или отклонения заявки предсказания, система не может изменить его статус. Дополнительная информация о разработке и внедрении таких сервисов может быть найдена в [2].

Бизнес-процесс создания, оформления и завершения или отклонения заявки на предсказание описан в нотации BPMN2.0 и представлен на рисунке 1.

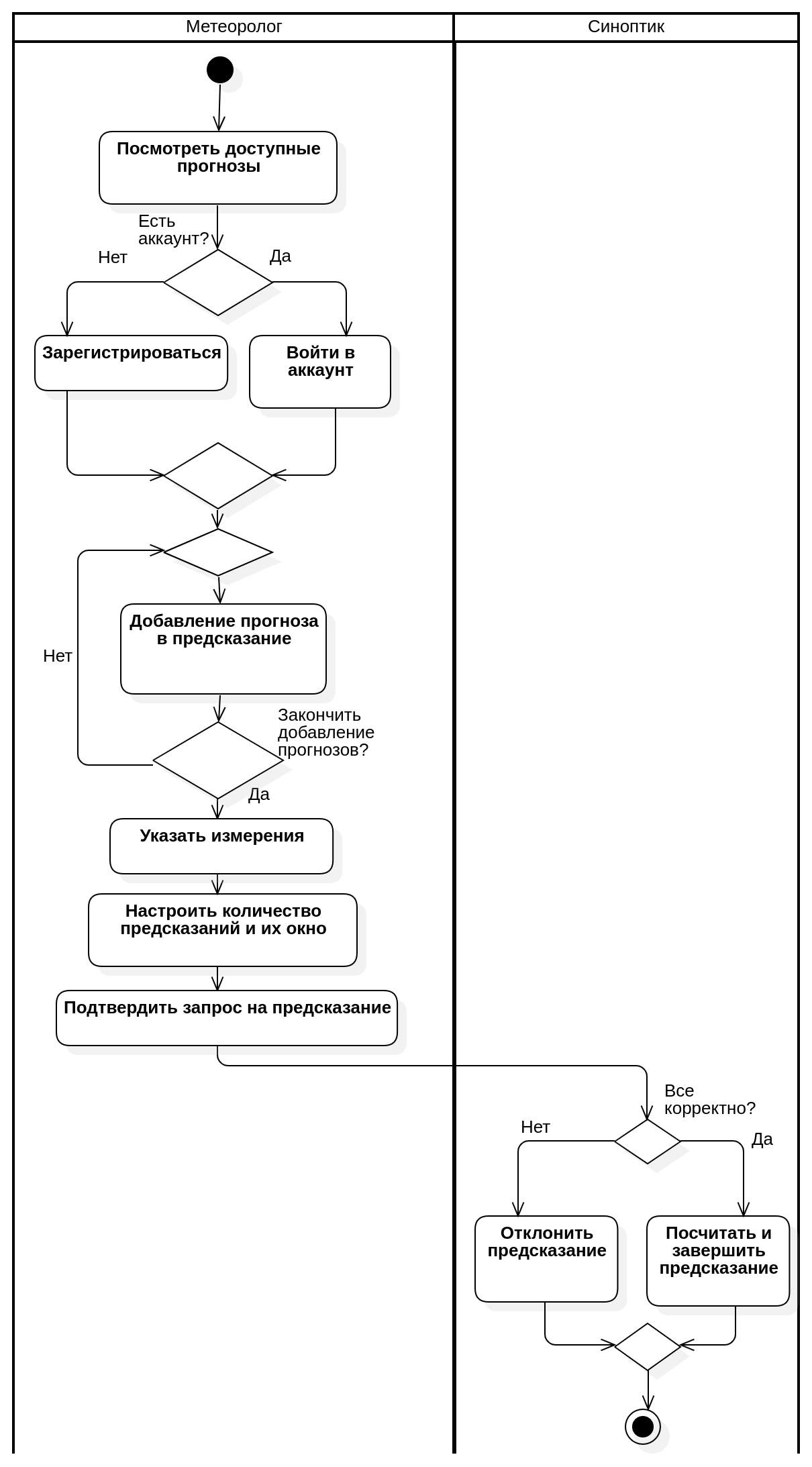
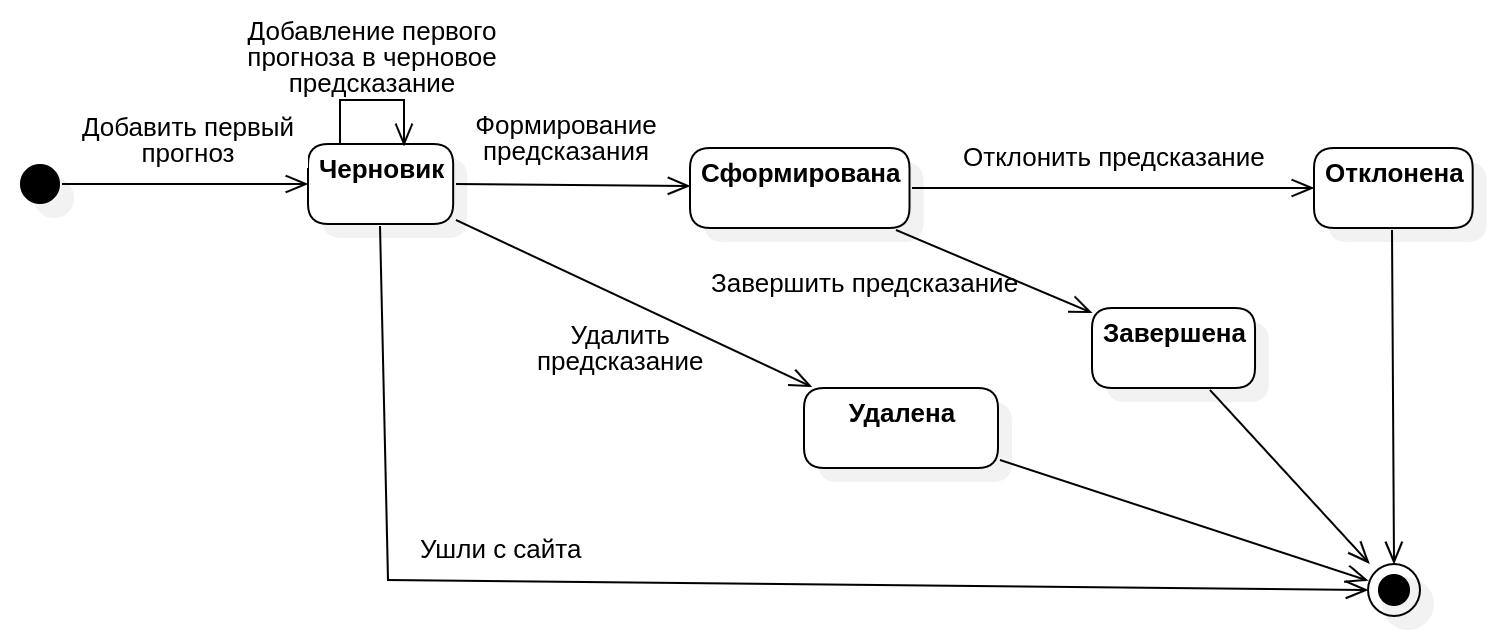


Рисунок *1* — Диаграмма взаимодействия в BPMN2.0

Диаграмма состояний приведена на рисунке 2.

Рисунок *2* — Диаграмма состояний заказов

Синоптик может редактировать список услуг, а именно: изменять сведения об уже имеющемся в системе прогнозе, добавлять новые и удалять устаревшие. При необходимости синоптик может создать и оформить предсказание.

На рисунке 3 представлена диаграмма прецендентов детально описывающая функции пользователей с различными ролями.

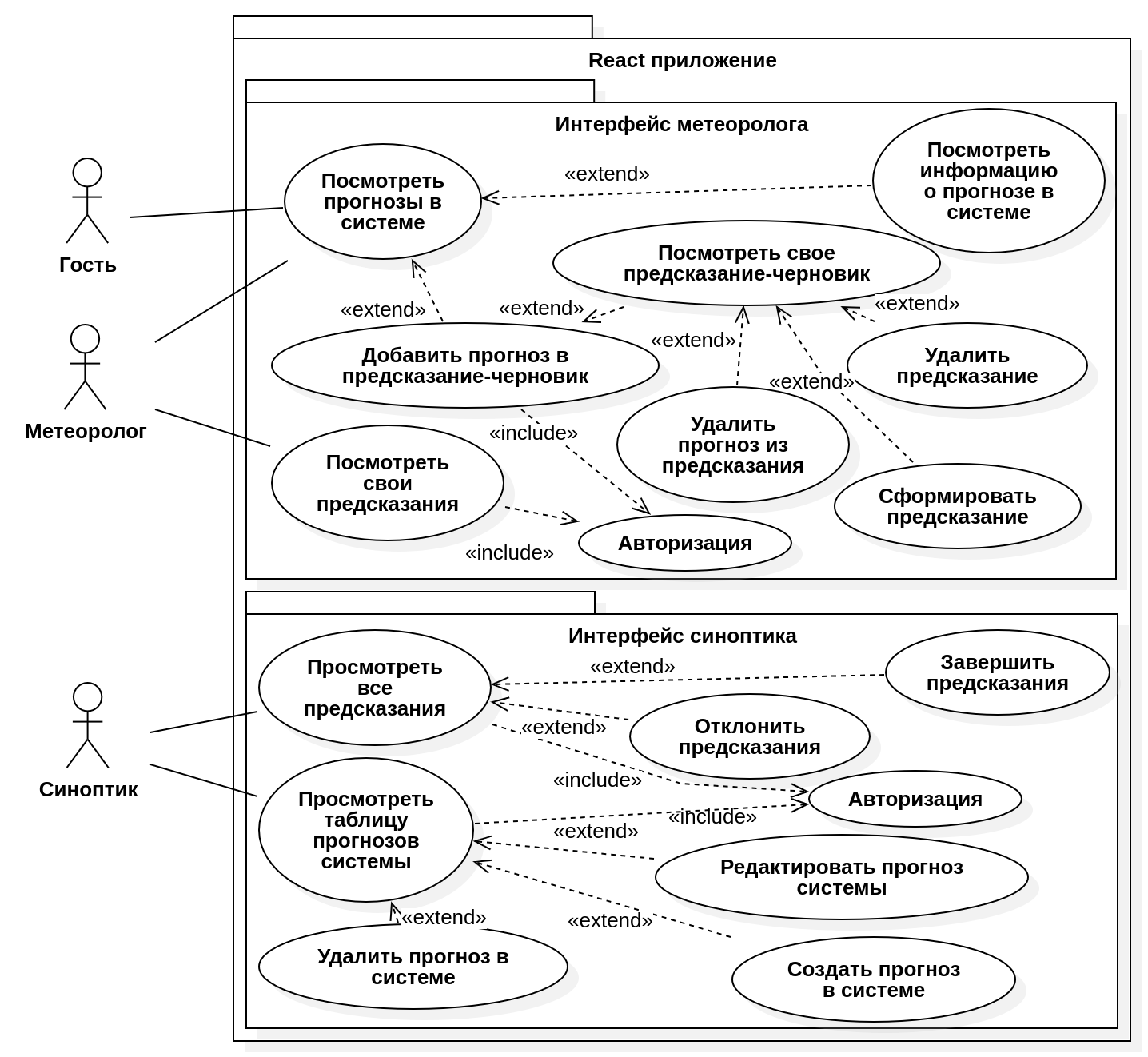


Рисунок 3 — Диаграмма прецендентов

# 2 АРХИТЕКТУРА

Архитектура системы представлена на диаграмме развертывания, изображенной на рисунке 4.

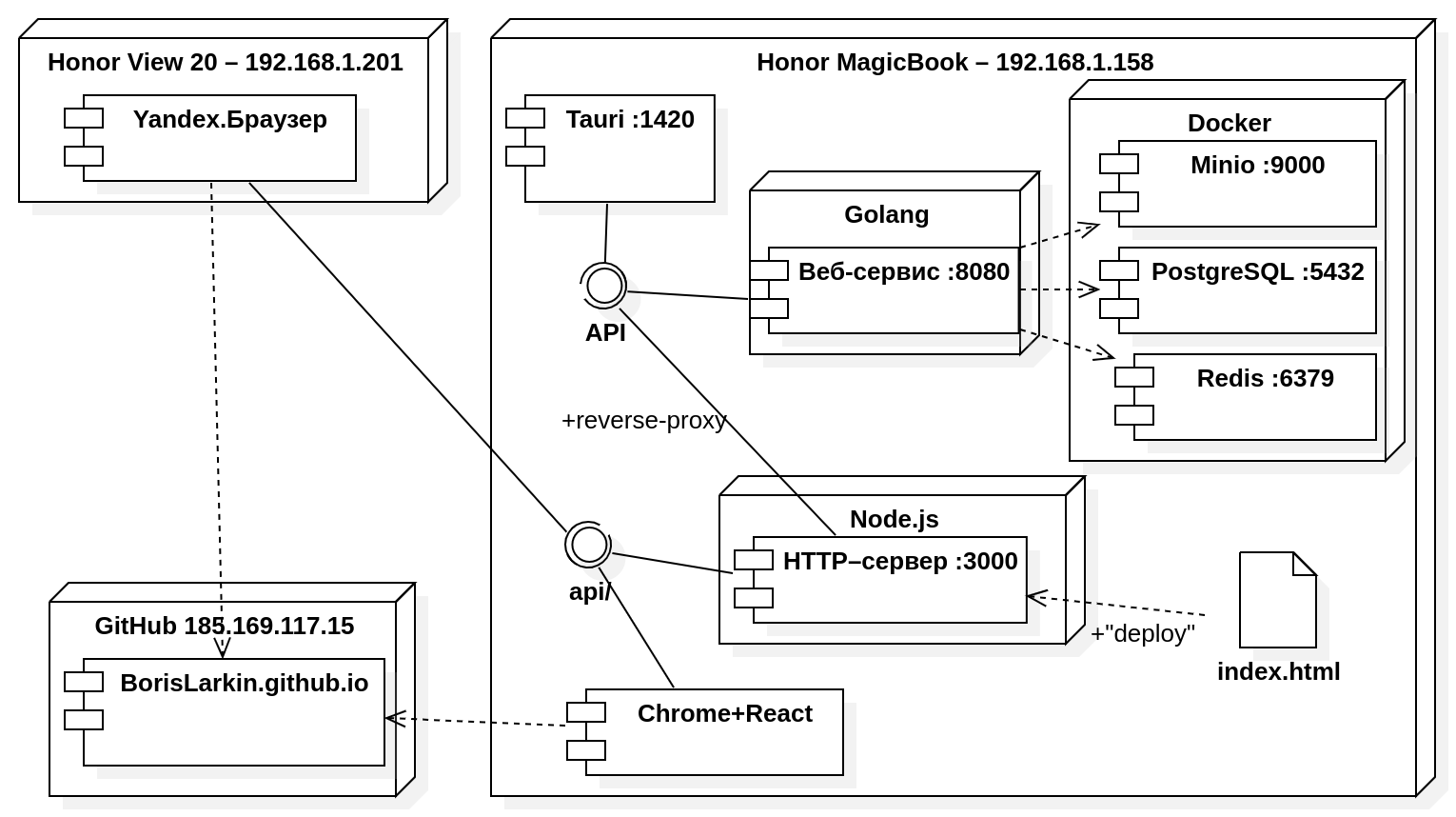


Рисунок 4 — Диаграмма развертывания

Веб-сервис, реализованный на фреймворке Gin for Golang [3] связан с серверами Minio [4], Redis [5] и СУБД PostgreSQL [6]. В Redis осуществляется хранение активных сессий пользователей, а при выходе пользователя из системы – их удаление.

В качестве языка программирования был выбран Golang [7], отличающийся популярностью и мощностью исполняемых на нем бекендов. Использование фреймворка Gin обусловлено широким спектром его библиотек для решения специализированных задач, например, для упрощения работы с API. Связь с БД производится через GORM. Веб-сервис на Golang является общим для веб-сервера и для десктопного приложения Tauri [9].

Согласно с текущей политикой импортозамещения, для хранения данных была выбрана СУБД PostgreSQL. Выбранная СУБД является стандартом индустрии.

Структура данных приведена на ER-диаграмме (рисунок 5). Модель услуг системы представляет собой набор полей, характеризующих прогнозы. Сведения о прогнозах хранятся в таблице *Прогнозы*. Данные о предсказаниях хранятся в таблице *Предсказания*. Для хранения в одном предсказании нескольких прогнозов используется промежуточная таблица *м-м Предсказания\_прогнозы*, с помощью которой реализуется связь «многие ко многим». Данные о пользователях системы хранятся в таблице *Пользователи*.

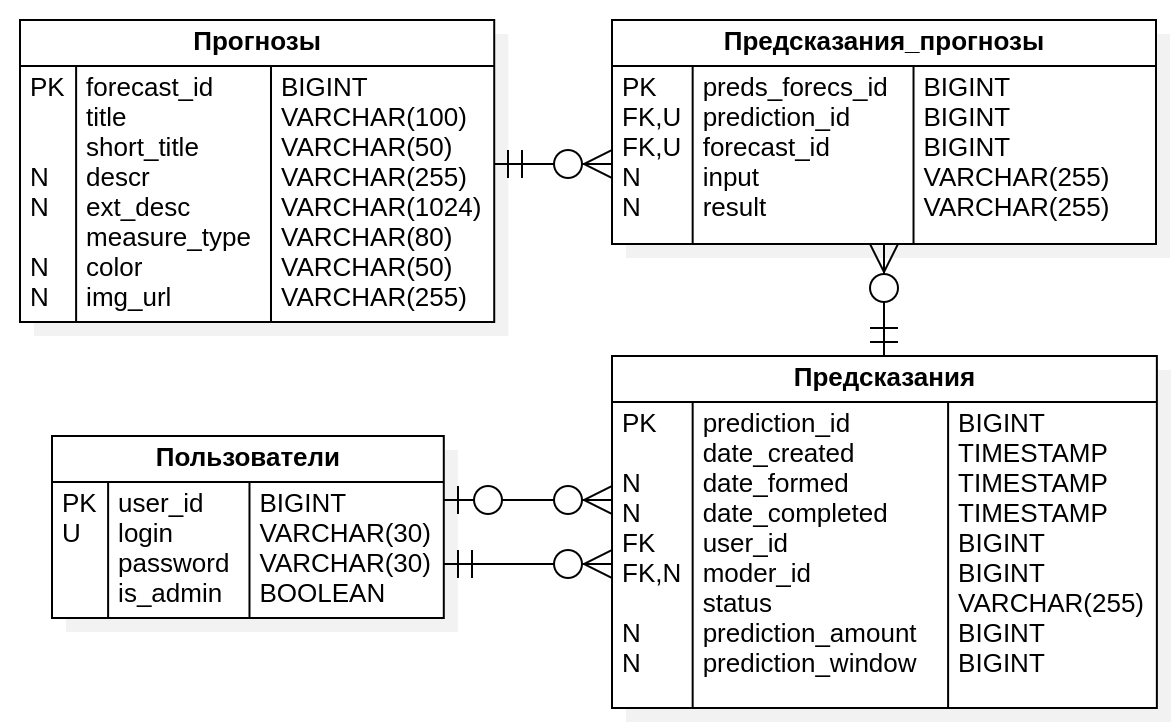


Рисунок 5 — ER-диаграмма

Устройство бэкенда приложения разработанной системы отражено на диаграмме классов бэкенда (рисунок 6). Пользователи взаимодействуют с доменами, которые, в свою очередь, связаны с моделями. Модели имеют связи с таблицами в базе данных.

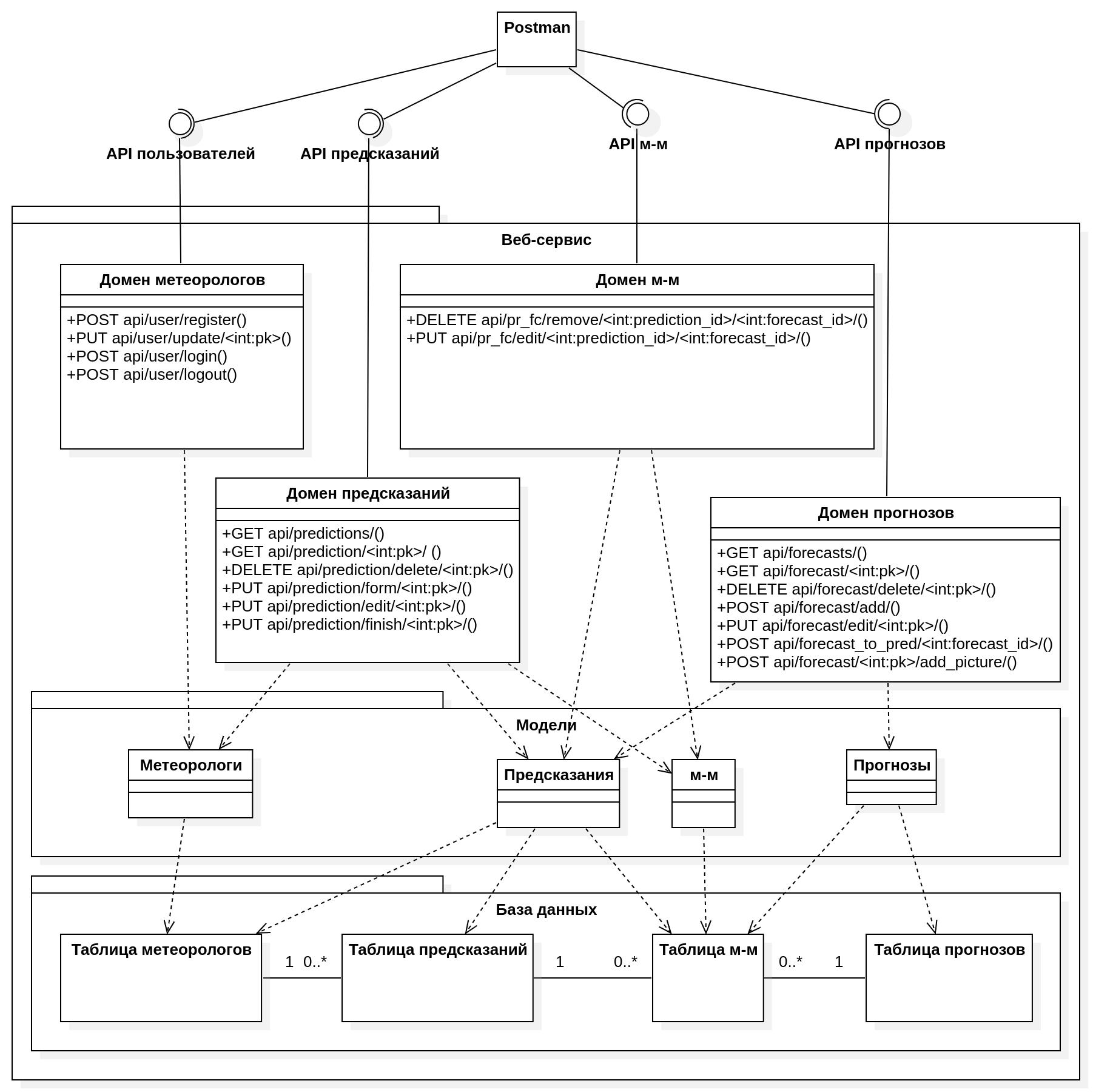


Рисунок 6 — Диаграмма классов бэкенда

Диаграмма классов фронтенда визуализирует связь фронтенда и бэкенда (рисунок 7). Каждая страница связана с API, используемым для взаимодействия с данными на соответствующей страницы. Фронтенд реализован с помощью фреймворка React [10].

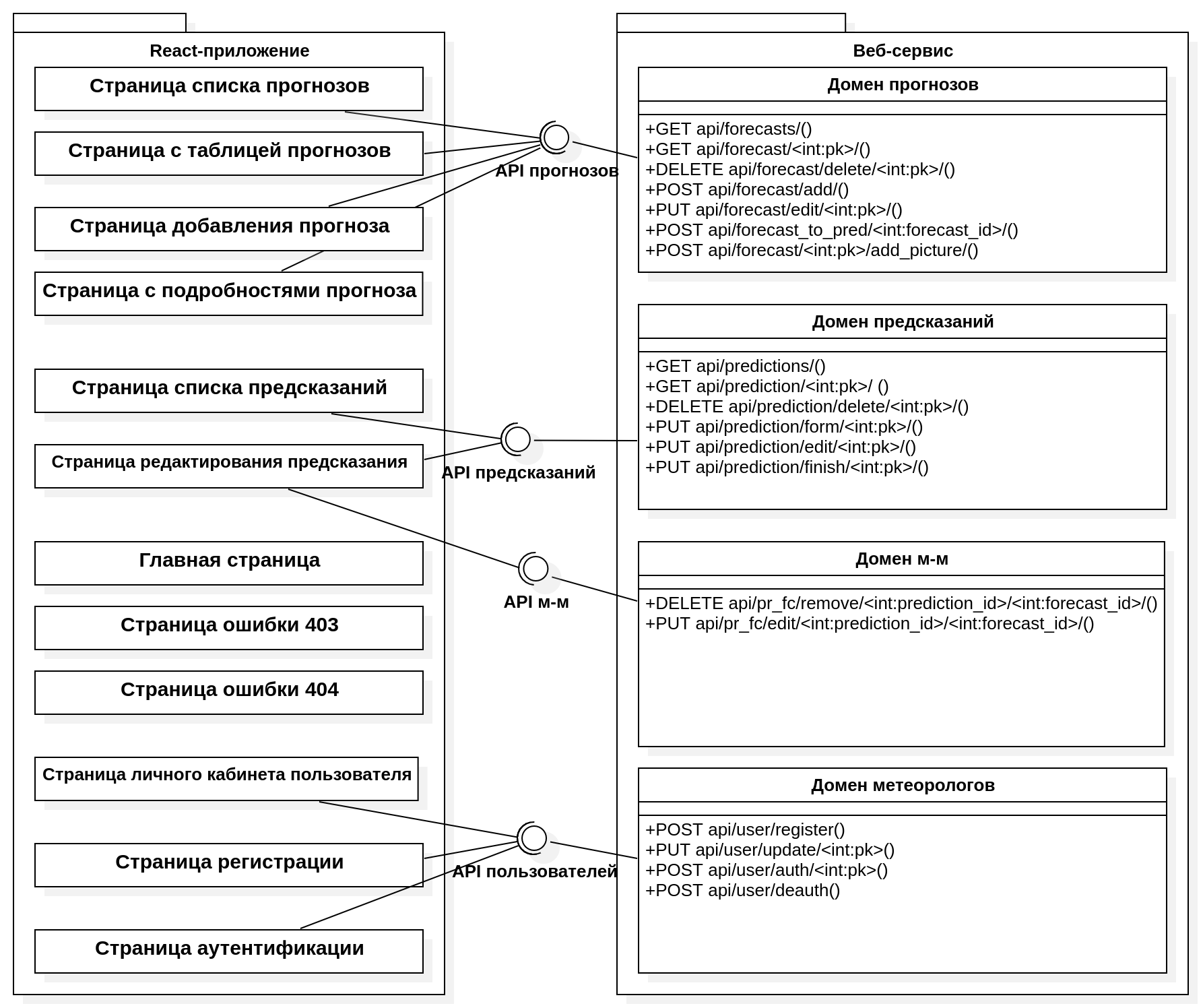


Рисунок 7 — Диаграмма классов фронтенда

# 3 АЛГОРИТМЫ

Алгоритм работы разработанной системы отображен на диаграмме последовательности, приведенной на рисунке 8. В основе системы лежит веб-сервис, внутри которого реализована вся бизнес-логика. Он предоставляет доступ к методам следующих доменов: прогнозы, предсказания, выбранные прогнозы, пользователи (метеорологи и синоптики). Методы следуют правилам REST API.

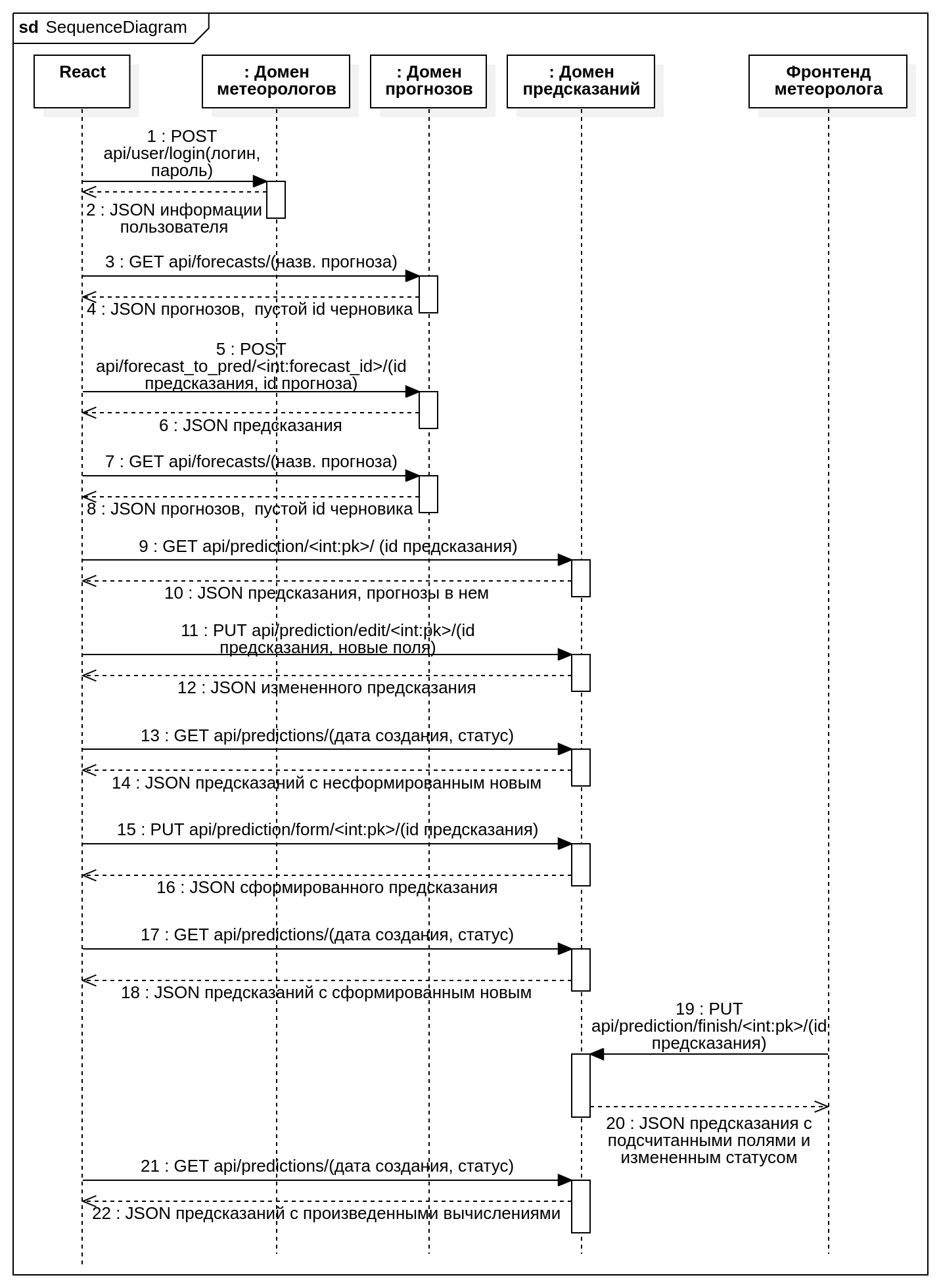


Рисунок 8 — Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. При помощи графического интерфейса пользователь отправляет запрос, передавая в нем логин и пароль. Если аккаунт с указанными данными существует в базе, на клиент возвращается информация о пользователе и устанавливаются JWT в local storage, внутри которых закодированы user\_id, role в ответном запросе. Если же аккаунта с введённым логином не существует, или пароль введен неверно, на клиент возвращается ошибка. В таком случае пользователю надо пройти регистрацию, либо ввести корректные логин или пароль. На этом же этапе происходит проверка: является пользователь метеорологом или синоптиком. Графический интерфейс пользователя запрашивает у веб-сервиса список прогнозов системы, которые возвращаются в JSON формате. Метеоролог выбирает прогноз, которую хочет добавить в предсказание, и, нажимая на кнопку «Добавить» в графическом интерфейсе, отправляет запрос на добавление прогноза в свой черновую заявку. Это действие может быть выполнено несколько раз.

Когда пользователь определится со списком прогнозов, а также указал необходимые значения полей, он нажимает кнопку «Сохранить», либо сразу «Сформировать» в графическом интерфейсе. После чего приложение отправляет на веб-сервис запрос на формирование предсказания. Пользователь может отслеживать статус сформированных предсказаний на соответствующей странице в графическом интерфейсе.

Процесс отслеживания предсказаний так же происходит при помощи графического интерфейса. Синоптики могут просматривать список всех оформленных метеорологом операций; завершать или отклонять их при помощи соответствующих кнопок. Синоптики имеют возможность редактировать список предлагаемых службой прогнозов, изменять данные о каждом из них. Для каждой из этих возможностей существует соответствующий метод, отправляемый на веб-сервис.

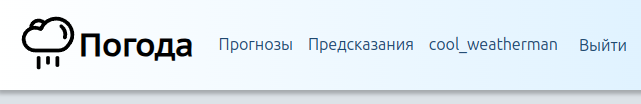
# 4 ОПИСАНИЕ ИНТЕРФЕЙСА

Главное меню приложения включает разделы, доступность которых зависит от роли пользователя. Рисунок 9 иллюстрирует разделы доступные гостю.

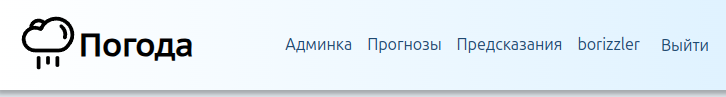


Рисунок 9 — Меню приложения для гостя

На рисунке 10 изображено меню, которое видит метеоролог.

Рисунок 10 — Меню приложения для метеоролога

На рисунке 11 изображено меню, которое видит синоптик.

Рисунок 11 — Меню приложения для синоптика

На странице с формой регистрации (рисунок 12) отображается форма, при помощи которой гость может создать новый аккаунт. После успешной регистрации открывается главная страница.

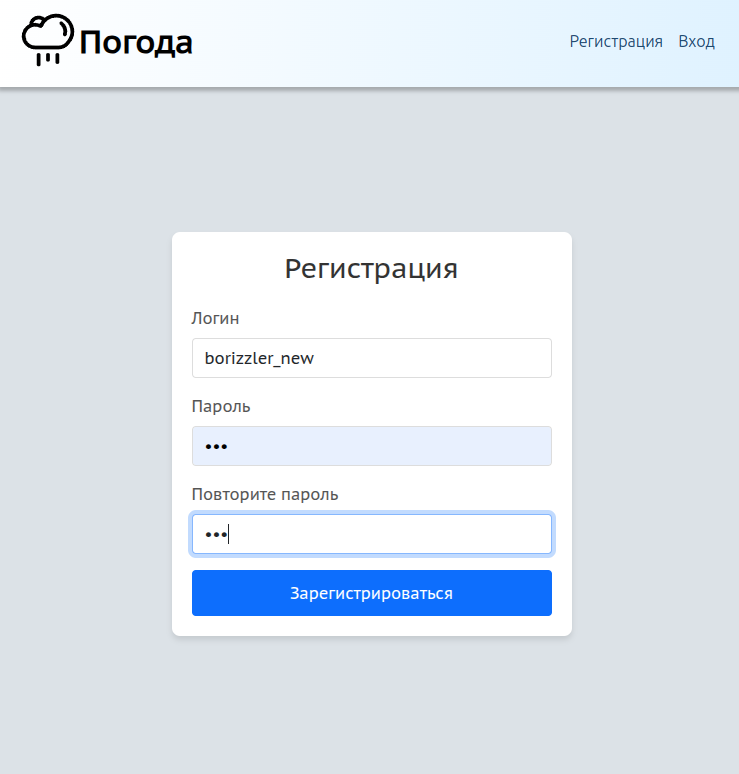


Рисунок 12 — Форма регистрации

На странице с формой входа (рисунок 13) отображается форма, через которую пользователь может войти в свой аккаунт. При успешном вводе данных аккаунта на клиент приходит JWT с идентификатором пользователя и дополнительной информацией.

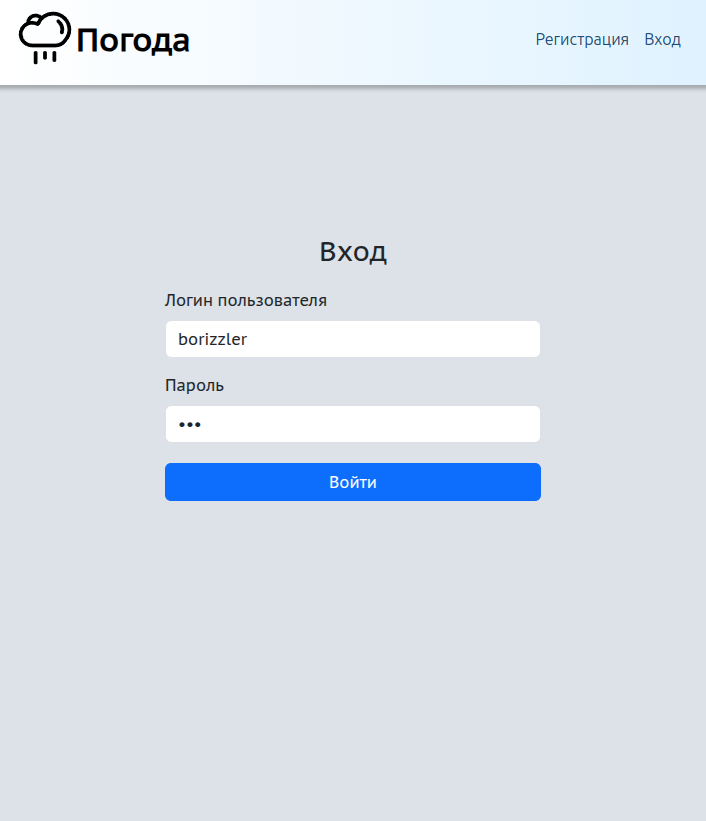


Рисунок 13 — Форма входа

После входа в аккаунт пользователь попадает на страницу списка прогнозов (рисунок 14), на которой располагается информация о доступных прогнозах. При нажатии кнопки подробнее, открывается страница с более подробной информацией о выбранном прогнозе.

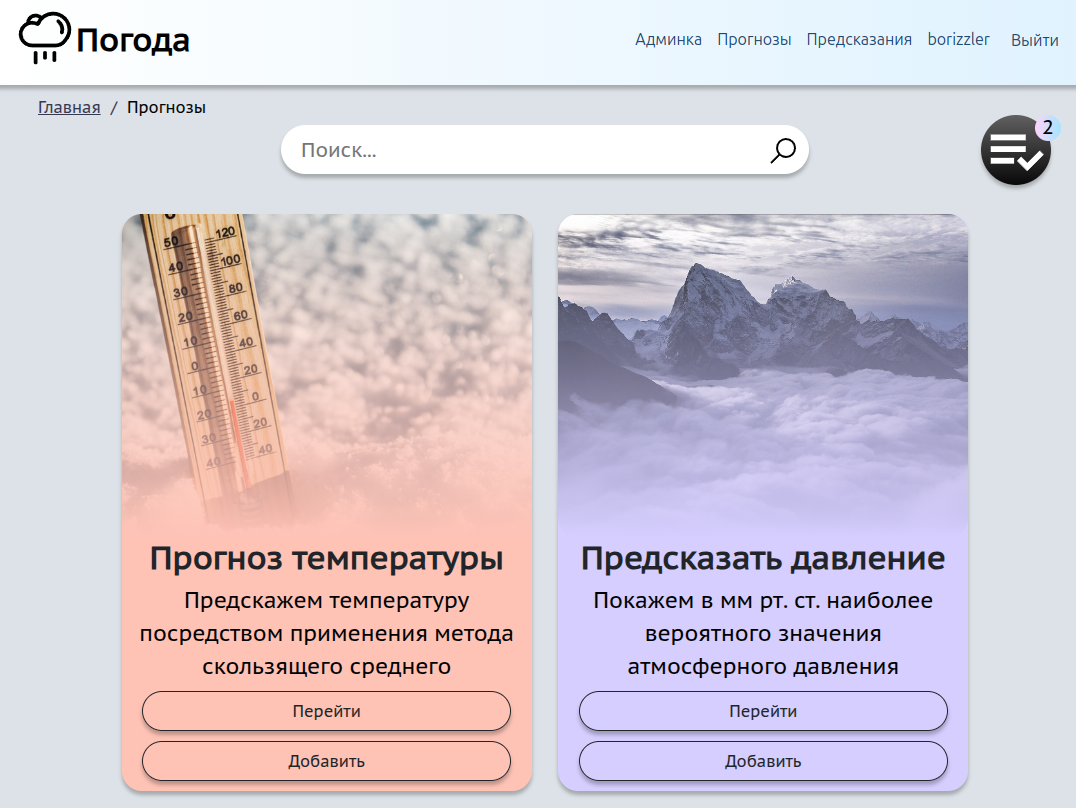


Рисунок 14 — страница прогнозов

Главная страница с информацией о системе (рисунок 15).

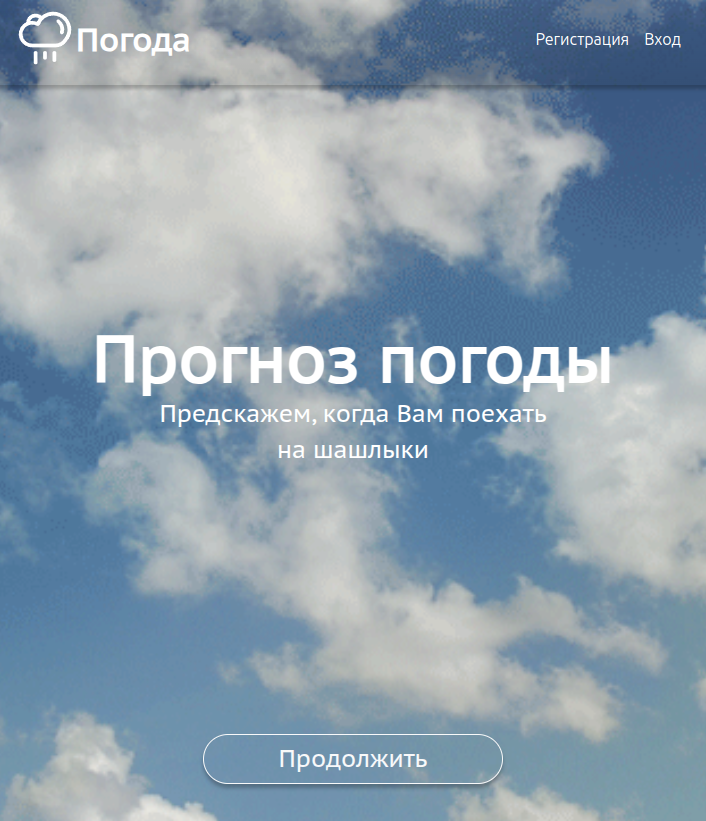


Рисунок 15 — Страница с информацией об сбер-банке

На странице прогноза представлена информация о названии, описании, единицах измерения (рисунок 16).

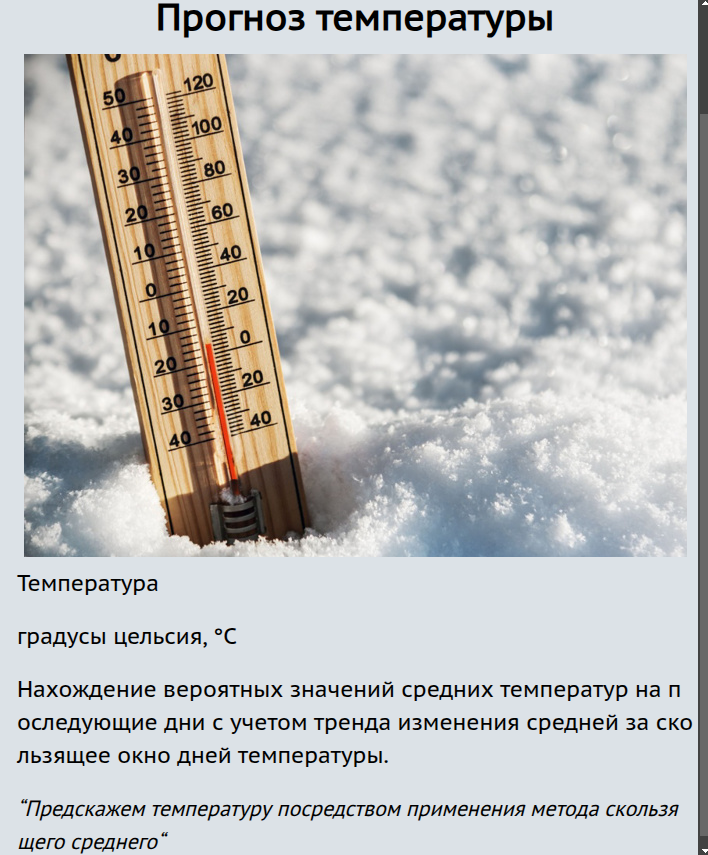


Рисунок *16* — Страница с описанием прогноза

В отличие от гостя метеорологу на карточке каждого прогноза доступна кнопка «Добавить» (рисунок 17), с помощью которой он может добавить прогноз в предсказание. Просмотреть черновик предсказания, при его наличии, метеоролог может, нажав на соответствующую кнопку в правом верхнем углу страницы. Вверху страницы также располагается поисковая строка, позволяющая найти интересующий прогноз.

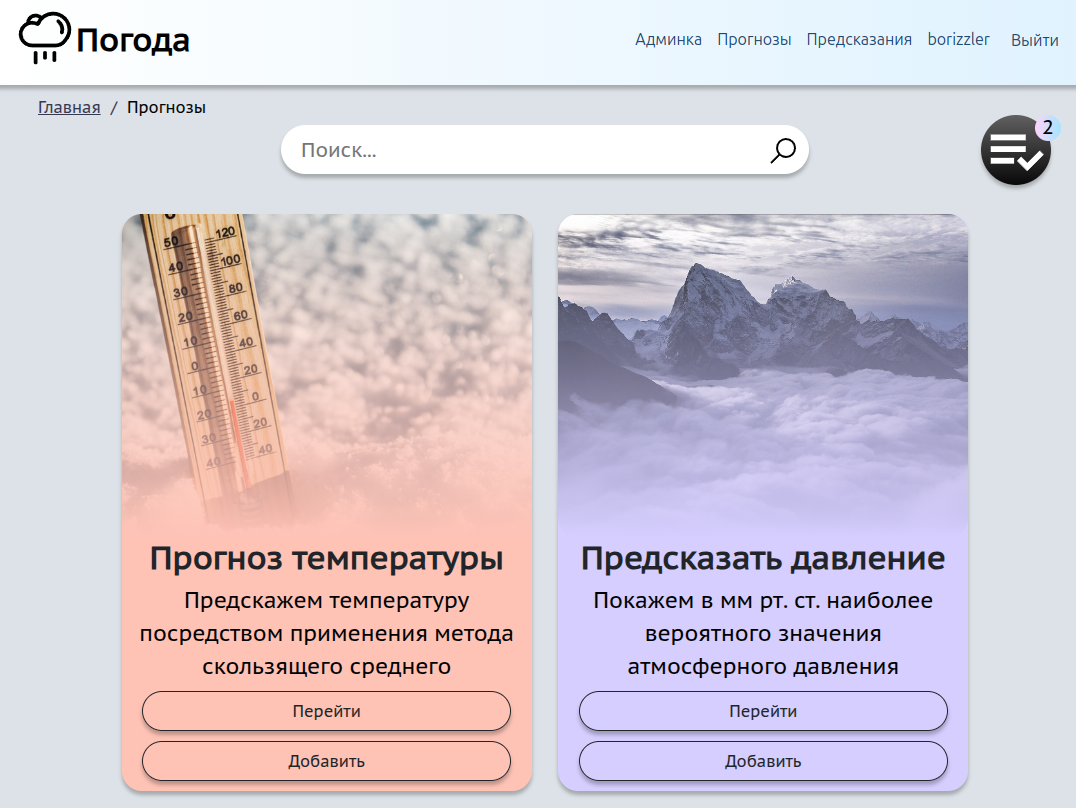
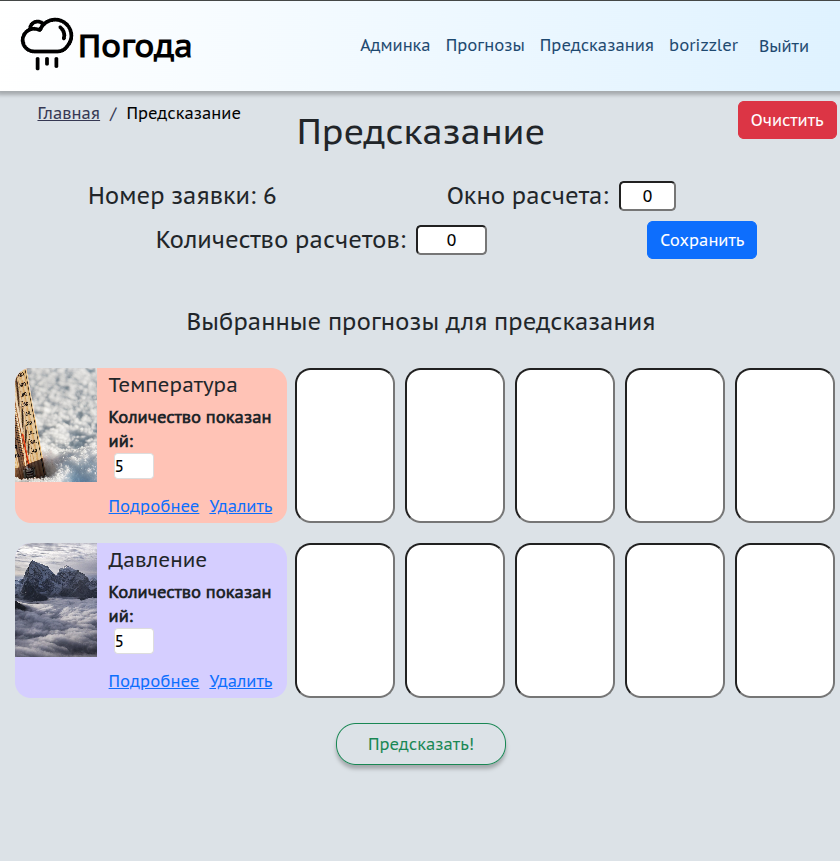
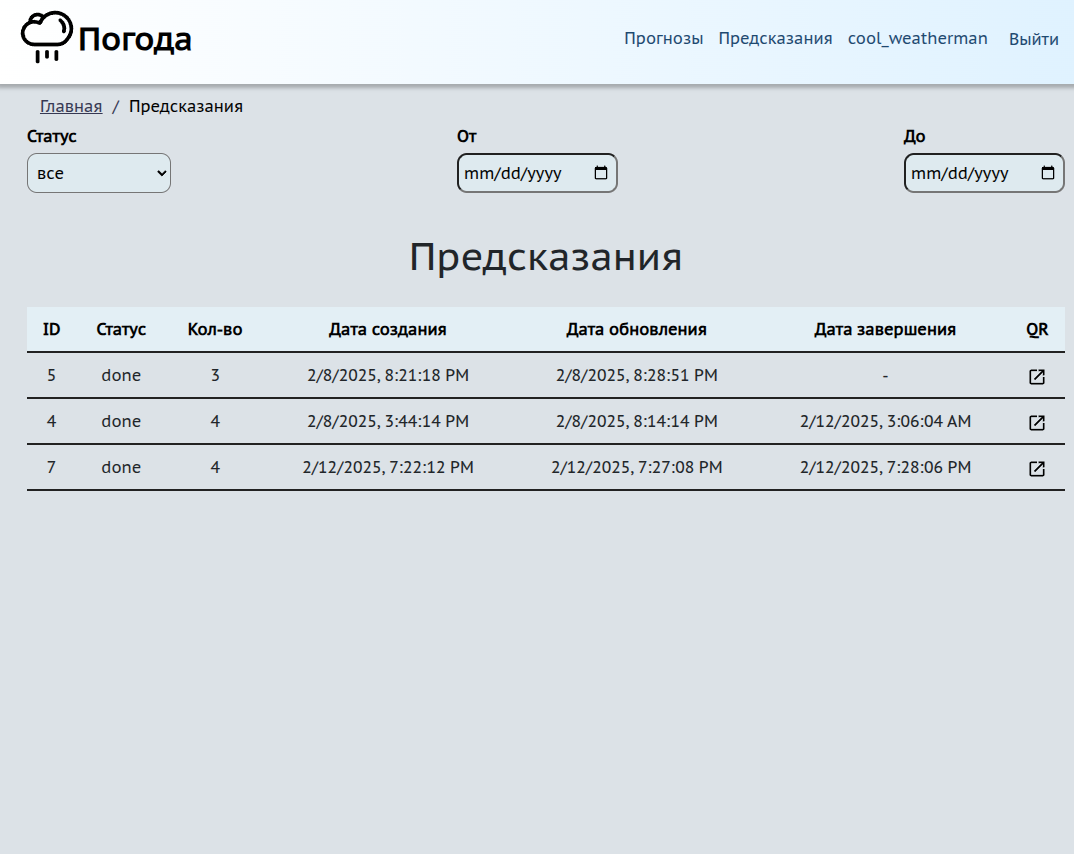


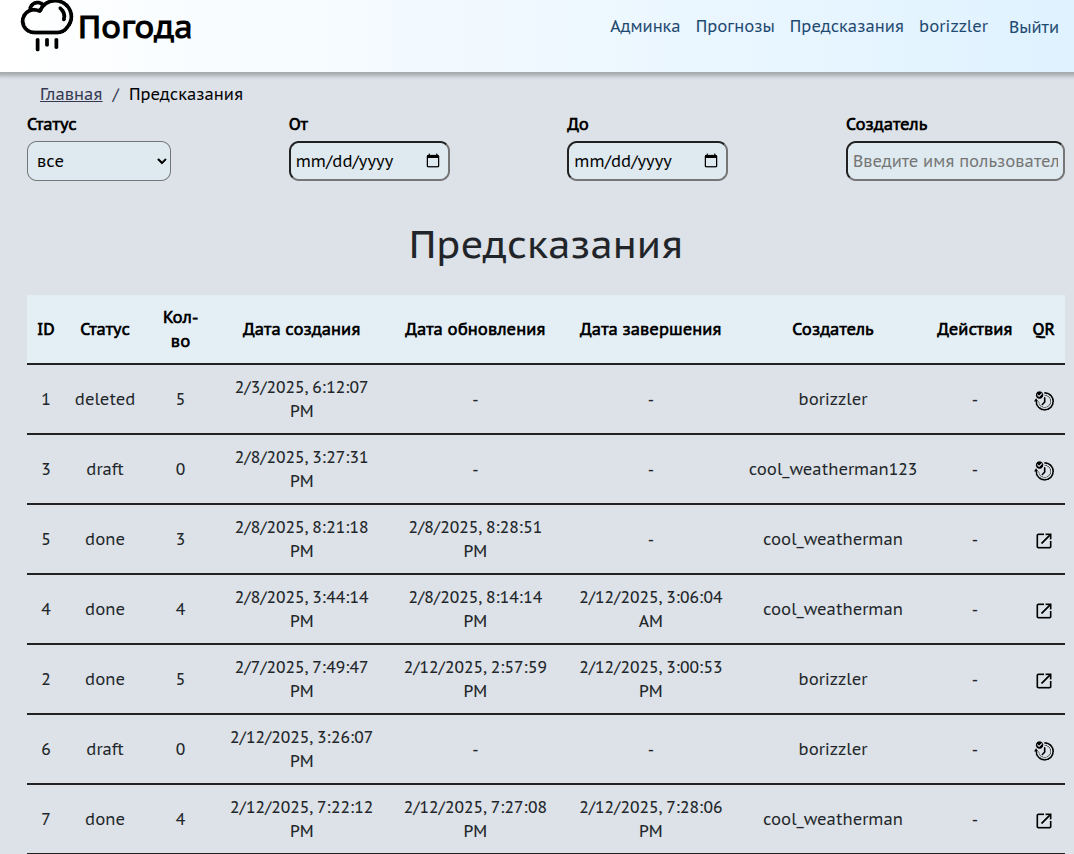
Рисунок 17 — Страница списка прогнозов для метеоролога

На странице корзины (рисунок 18) отображается текущее черновое предсказание. Метеоролог может удалить прогноз из предсказания нажатием «Удалить». Кнопки «Сохранить», «Сформировать» и «Удалить» позволяют выполнить одноименные операции над предсказанием. Редактирование оформленных предсказаний недоступно.

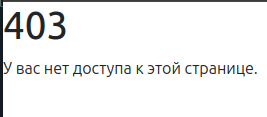
 Рисунок 17 — Страница чернового предсказания

На странице списка предсказаний (рисунок 19.1) метеорологи могут просматривать созданные ими операции в виде длинных карточек. А синоптики могут принимать или откланять заявки пользователей (рисунок 19.2)

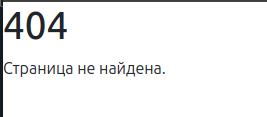
Рисунок 18.1 — Страница списка сформированных операций

Рисунок 19.2 — Страница списка сформированных операций для операциониста

В случае, если пользователь пытается получить доступ к страницам, для просмотра которых ему не хватает прав, он перенаправляется на страницу ошибки 403 (рисунок 20).

Рисунок 20 — Страница ошибки 403

В случае, если пользователь запрашивает несуществующую станицу, он перенаправляется на страницу ошибки 404 (рисунок 26).

Рисунок 21 — Страница ошибки 404

Пользователь может изменить свой пароль на странице редактирования личного кабинета (рисунок 22). Она открывается при нажатии на логин пользователя в меню.

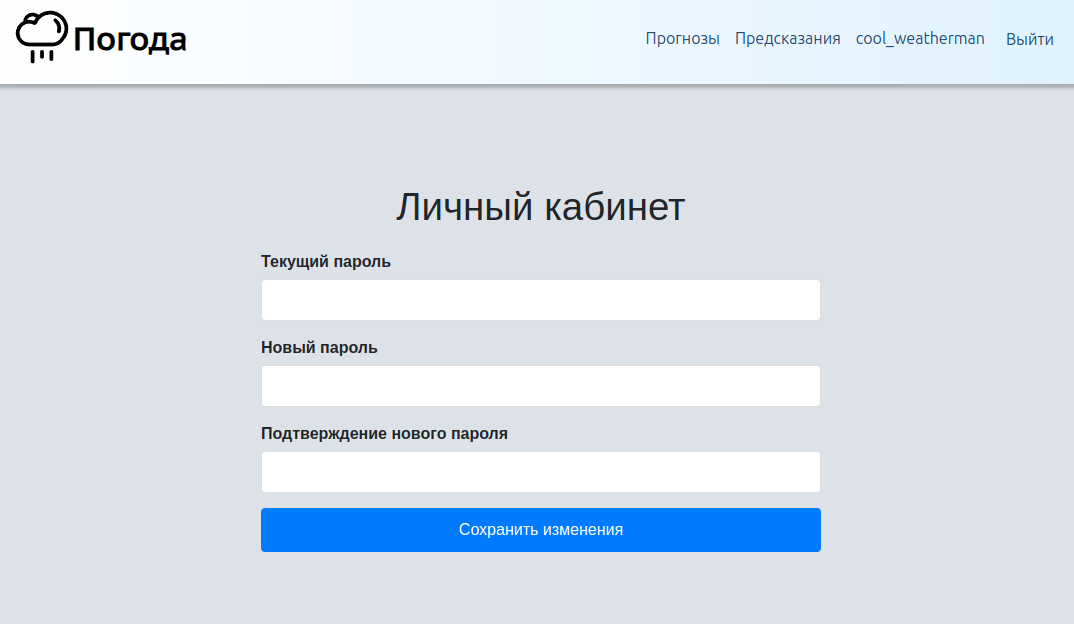
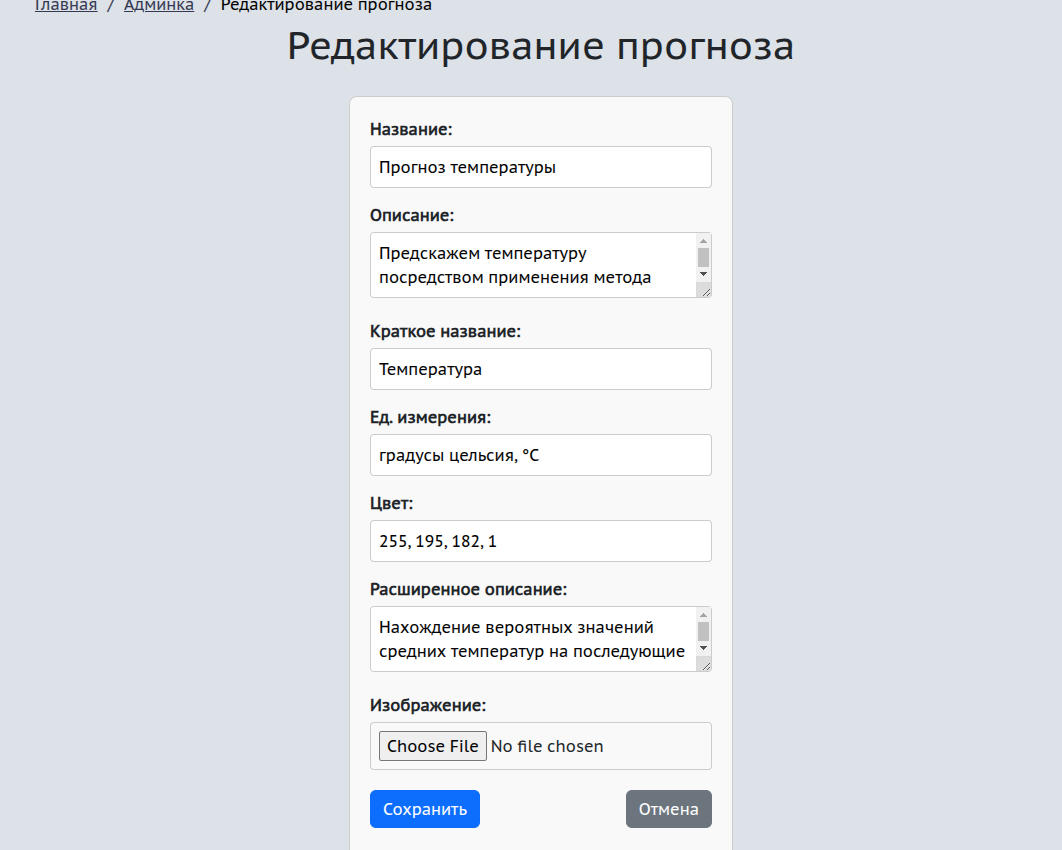


Рисунок 22 – Страница личного кабинета

Страница редактирования прогнозов(рисунок 23). Синоптик может редактировать уже существующие прогнозы (рисунок 24), добавлять новые прогнозы(рисунок 25).



Рисунок 23 - страница редактирования прогнозов

Рисунок 24 - страница редактирования прогноза

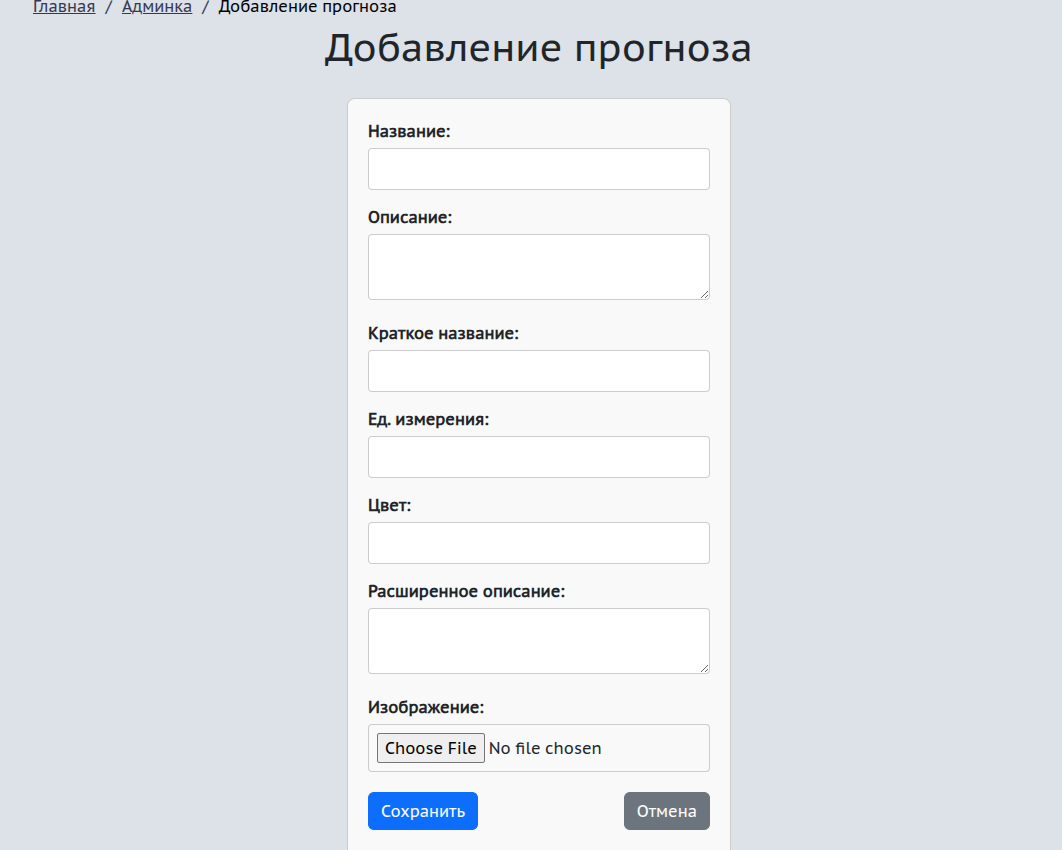


Рисунок 25 - страница создания прогноза

# ЗАКЛЮЧЕНИЕ

В ходе работы были достигнуты следующие результаты:

1. Разработан дизайн приложения в Figma на основе yandex.weather. После ознакомления с разработкой бэкенда с использованием фреймворка Gin создан MVP.
2. Спроектирована и создана база данных PostgreSQL, а затем подключена к бэкенду.
3. Создан веб-сервис со всей итоговой бизнес-логикой, кроме авторизации, в бэкенде системы для использования его в SPA.
4. Добавлена авторизация с использованием Redis для хранения токенов, в веб-сервис внедрен Swagger.
5. Разработан базовый интерфейс приложения для гостя на React.
6. Внедрен менеджер состояний Redux Toolkit для хранения значений фильтров, добавлена адаптивность и PWA.
7. Завершена разработка интерфейса пользователя в React, для обращений к методам веб-сервиса использован Axios.
8. Реализован React интерфейс издателя, внедрен Real-time web.
9. Разработано десктопное приложение Tauri.
10. Приложение развернуто при помощи сервиса GitHub Pages и доступно по ссылке: https://borislarkin.github.io/Forecast-System-Frontend/
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформлен git-репозиторий на сервисе GitHub, содержащий исходный код проекта: https://github.com/BorisLarkin/Forecast-System-Frontend

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Система прогнозирования [Электронный ресурс]. URL: https://yandex.com/weather/ (дата обращения: 13.12.2024).
2. Документация Golang [Электронный ресурс] // Go. URL: https://go.dev/doc/ (дата обращения: 10.10.2024).
3. Документация Minio [Электронный ресурс] // Min. URL: <https://min.io/docs/minio/kubernetes/upstream/index.html> (дата обращения: 09.09.2024).
4. Документация по Redis [Электронный ресурс] // Netlify. URL: <https://master--redis-doc.netlify.app/docs/> (дата обращения: 10.10.2024).
5. Документация PostgreSQL [Электронный ресурс] // Postgresql. URL: <https://www.postgresql.org/docs/> (дата обращения: 15.09.2024).
6. Документация по GORM [Электронный ресурс] // GORM. URL: https://gorm.io/ (дата обращения: 20.10.2024).
7. Документация по Tauri [Электронный ресурс] // Tauri. URL: <https://v2.tauri.app/develop/> (дата обращения: 20.11.2024).
8. Документация по React [Электронный ресурс] // React. URL: <https://react.dev/learn> (дата обращения: 01.11.2024).

**ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
| \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_А.И. Канев  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Прогноз погоды.**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

8

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Ларкин Борис Владимирович |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г. |  |

Москва - 2024

1. Введение

Цель системы, включающая в себя веб-сервис, веб-приложение, нативное приложение, которая дает возможность метеорологам и пользователям сервиса прогнозировать различные виды будущих погодных условий на основе математических методов и предварительно задаваемых уже известных измерений.

1. Назначение разработки

Данная система предназначена для взаимодействия с пользователями для прогнозирования погоды, хранения данных о предсказаниях для каждого пользователя. Клиент сервиса, заготовивший информацию о наблюдениях погодных измерений за некоторый период времени, передает информацию сервису, выбирая окно измерений и количество предсказаний (будущих дней). Также пользователь может потом просматривать свои созданные предсказания и результаты их вычислений. Модератор сервиса проверяет запросы на предсказания пользователей и может их отклонять или принимать на расчет. Также модераторы могут редактировать набор доступных прогнозов, просматривать предсказания всех пользователей.

1. Стадии и этапы разработки
   1. Создание MVP и базового дизайна на основе yandex.ru/pogoda
   2. Реализовать хранение данных в БД (Postgres)
   3. Создать веб-сервис на бэкенде (Golang)
   4. Реализовать авторизацию и хранение сессий в Redis
   5. Разработать SPA на React для гостя
   6. Внедрить адаптивность, разработка Tauri приложения
   7. Интерфейс метеоролога на React с менеджером состояний Redux Toolkit
   8. Добавить в приложение React интерфейс модератора
   9. Реализовать нативное приложение
   10. Развернуть веб-приложение в GitHub Pages
   11. Подготовка всей документации (РПЗ, ТЗ и набор диаграмм)
   12. Подготовка репозитория на GitHub
2. Требования к функциональным характеристикам
   1. Методы HTTP
      1. GET Получение всех прогнозов
      2. POST Добавление прогноза
      3. GET Получить один прогноз
      4. PUT Изменить прогноз
      5. DELETE Удалить прогноз
      6. POST Добавить прогноз в предсказание
      7. POST Изменение/добавление картинки прогноза
      8. GET Получить все предсказания
      9. GET Получить одно предсказание
      10. PUT Изменить поля предсказания
      11. DELETE Удалить предсказание
      12. PUT Сформировать предсказание
      13. PUT Завершить предсказание
      14. DELETE Удалить прогноз из предсказания
      15. PUT Изменить входные данные для прогноза
      16. POST Регистрировать пользователя
      17. PUT Изменить данные пользователя
      18. POST Аутентификация
      19. POST Деавторизация
   2. Меню
      1. Главная - перенаправляет на страницу 4.6
      2. Список прогнозов – перенаправляет на страницу 4.7 (вызывается метод 4.1.1)
      3. Список предсказаний – перенаправляет на страницу 4.9
      4. Редактирование прогнозов – перенаправляет на страницу 4.12, доступно только модераторам
      5. Зарегистрироваться – перенаправляет на страницу 4.3 появляется только для гостей
      6. Редактирование пользователя – перенаправляет на страницу 4.5, только аутентифицированным пользователям
      7. Войти – перенаправляет на страницу 4.4, только для гостей, появляется только для гостей
      8. Выйти – перенаправляет на страницу 4.6 (вызывается метод 4.1.19), доступно только аутентифицированным пользователям
   3. Регистрация
      1. Доступно только гостям
      2. Отображает форму регистрации
         1. Поле логина
         2. Поле пароля
      3. Действия
         1. Регистрация пользователя – (вызывается метод 4.1.16)
         2. Вернуться к аутентификации – перенаправляет на страницу 4.4
   4. Аутентификация
      1. Доступно только гостям
      2. Отображает форму аутентификации
         1. Поле логина
         2. Поле пароля
      3. Действия
         1. Войти – (вызывается метод 4.1.18)
         2. Регистрация – перенаправляет на страницу 4.3
   5. Личный кабинет
      1. Доступно аутентифицированному пользователю
      2. Действия
         1. Изменить данные пользователя – (вызывается метод 4.1.17)
   6. Главная
      1. Доступна всем
      2. Отображается статическая информация
         1. Назначение сервиса
         2. Контакты для связи
   7. Список прогнозов
      1. Доступен всем
      2. Отображаются элементы карточек с прогнозами, вызывается метод 4.1.1
         1. Название прогноза
         2. Краткое описание
      3. Действия
         1. Поиск – перенаправляет на страницу 4.6, (используется метод 4.1.1), с фильтрующем параметром
         2. Подробнее – перенаправляет на страницу 4.8 (используется метод 4.1.3)
         3. Добавить в предсказание – добавляет прогноз в предсказание-черновик, (вызывается метод 4.1.2), только аутентифицированные пользователи.
         4. Кнопка корзины – перенаправляет на страницу 4.10, только аутентифицированные пользователи.
   8. Один прогноз
      1. Доступна всем
      2. Отображается подробная информация выбранного прогноза, (вызывается метод 4.1.3)
   9. Одно предсказание
      1. Доступно только аутентифицированным пользователям
      2. Отображает текущее предсказание-черновик пользователя, (метод 4.1.9)
         1. Список выбранных прогнозов
         2. Поле количества предсказаний
         3. Поля вводов данных для каждого прогноза
         4. Поле величины скользящего окна для предсказаний
      3. Действия, доступны только в случае, если статус «черновик»
         1. Убрать прогноз – удалить прогноз из предсказания, (вызывается метод 4.1.14)
         2. Сохранить – сохраняет текущее предсказание-черновик, (вызывается метод 4.1.12)
         3. Очистить – удаляет предсказание-черновик, (вызывается метод 4.1.1)
         4. Заполнить – вносит данные предсказания (окно, количество, вызывается метод 4.1.10)
   10. Список предсказаний
       1. Доступно аутентифицированному пользователю
       2. Отображается список предсказаний (метод 4.1.8)
          1. Только предсказания, созданные данным пользователем, если он не модератор
          2. Все сохраненные предсказания в противном случае
       3. Действия
          1. Фильтрация – фильтрует предсказания по дате создания или статусу, вызывается (метод 4.1.8)
          2. Сформировать – формирует предсказание, выполняется метод 4.1.13, доступно только пользователю
          3. Завершить – заврешает предсказание со статусом «отклонено» или «завершено», вызывается метод 4.1.13, доступно только модератору
          4. Посмотреть подробную информацию о предсказании – перенаправляет на страницу 4.9 (вызывается метод 4.1.9)
   11. Список прогнозов таблицей
       1. Доступно доверенному модератору
       2. Отображаются все существующие в системе прогнозы (вызывается метод 4.1.1)
       3. Действия
          1. Удалить – удаляет прогноз (вызывается метод 4.1.5)
          2. Редактирование/создание – переход на страницу 4.12
   12. Редактирование/создание прогноза
       1. Доступно модератору
       2. Отображается информация об изменяемом/добавляемом прогнозе (вызывается метод 4.1.3)
          1. Название
          2. Детальное описание
          3. Подробное описание
          4. Единица измерения
          5. Цвет в интерфейсе
          6. Картинка
       3. Действия
          1. Сохранить – добавляет новый прогноз, вызывается метод, 4.1.2, если изменяется существующий, то вызывается метод 4.1.4
          2. Картинка – добавляет/изменяет картинку прогноза (вызывается метод 4.1.7)
   13. 404
       1. Доступно всем
       2. Отображается в случае отсутствия ресурса
   14. 403
       1. Доступно всем
       2. Отображается в случае запрета на использование ресурса
3. Требования к составу и параметрам технических средств
   1. Сервер
      1. Процессор AMD Ryzen™ 5 5600H with Radeon™ Graphics × 12
      2. Оперативная память 16 Гб
      3. Свободное пространство на диске 10 Гб
   2. Клиент
      1. Процессор AMD Ryzen™ 5 5600H with Radeon™ Graphics × 12
      2. Оперативная память 16 Гб
      3. Свободное пространство на диске 10 Гб
4. Требования к информационной и программной совместимости
   1. Сервер
      1. ОС Linux (6.4.12)
      2. Redis (7.4.1)
      3. Minio (RELEASE 2022-10-15T19-57-03Z)
      4. PostgreSQL (15.7)
      5. Docker
      6. Go 1.23.1
   2. Клиент
      1. **Браузер (Safari 16.5.2, Firefox 121.0, Chrome 119.0.6045, Yandex 24.6.3.729, Opera 105.0.4970.16)**

# ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ

Таблица 1 – HTTP методы разрабатываемого веб-сервиса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Тип** | **Описание** | **URL** | **Входные данные** | **Выходные данные** |
| 4.1.1 | GET | /forecasts | Возвращает все прогнозы  Доступно всем | forecast\_name=string | {  "forecasts": [  {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "forecast\_id": int,  "image": "string",  "measure\_type": "string",  "short\_title": "string",  "title": "string"  }  ],  "forecasts\_empty": true,  "predicction\_id": "string",  "prediction\_size": int  } |
| 4.1.2 | POST | /forecast/add | Добавляет прогноз,  доступно модератору | {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "image": "string",  "measure\_type": "string",  "short\_title": "string",  "title": "string"  } | {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "image": "string",  "measure\_type": "string",  "short\_title": "string",  "title": "string"  } |
| 4.1.3 | GET | /forecast/{pk} | Получает данные прогноза,  Доступно всем | pk: int | {  "color": "string",  "descr": "string",  "extended\_desc": "string",  "id": int,  "img\_url": "string",  "measure\_type": "string",  "short": "string",  "title": "string"  } |
| 4.1.4 | PUT | /forecast/edit/{pk} | Изменяет данные прогноза (кроме картинки), доступно модератору | pk: int  {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "image": "string",  "measure\_type": "string",  "short\_title": "string",  "title": "string"  } | {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "image": "string",  "measure\_type": "string",  "short\_title": "string",  "title": "string"  } |
| 4.1.5 | DELETE | /forecast/delete/{pk} | Удаляет прогноз, доступно модератору | pk: int |  |
| 4.1.6 | POST | /forecast\_to\_pred/{pk} | Добавляет прогноз в предсказание, доступно аутентифицированному пользователю | pk: int |  |
| 4.1.7 | POST | /forecast/{pk}/add\_picture | Добавляет/изменяет картинку прогноза, доступно модератору | pk: int  {  pic: formdata/Image } |  |
| 4.1.8 | GET | /predictions | Получает предсказания, доступно любому аутентифицированному пользователю | dateCreated-min=date  dateCreated-max=date  status=string | [ {  "date\_completed": "string",  "date\_created": "string",  "date\_formed": "string",  "prediction\_amount": int,  "prediction\_id": int,  "prediction\_window": int,  "status": "string"  }  ] |
| 4.1.9 | GET | /prediction/{pk} | Получает предсказание, любому аутентифицированному пользователю | pk: int | {  "forecasts": [  {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "id": int,  "image": "string",  "input": "string",  "measure\_type": "string",  "result": "string",  "short\_title": "string",  "title": "string"  }  ],  "prediction": {  "date\_completed": "string",  "date\_created": "string",  "date\_formed": "string",  "prediction\_amount": int,  "prediction\_id": int,  "prediction\_window": int,  "status": "string"  }  } |
| 4.1.10 | PUT | /prediction/edit/{pk} | Изменяет поля предсказания, доступно аутентифицированному пользователю | pk: int  {  "prediction\_amount": int,  "prediction\_window": int  } | {  "forecasts": [  {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "id": int,  "image": "string",  "input": "string",  "measure\_type": "string",  "result": "string",  "short\_title": "string",  "title": "string"  }  ],  "prediction": {  "date\_completed": "string",  "date\_created": "string",  "date\_formed": "string",  "prediction\_amount": int,  "prediction\_id": int,  "prediction\_window": int,  "status": "string"  }  } |
| 4.1.11 | DELETE | /prediction/delete/{pk} | Удаляет предсказание, доступно владельцу предсказания | pk: int |  |
| 4.1.12 | PUT | /prediction/form/{pk} | Формирует предсказание, доступно владельцу предсказания | pk: int | {  "forecasts": [  {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "id": int,  "image": "string",  "input": "string",  "measure\_type": "string",  "result": "string",  "short\_title": "string",  "title": "string"  }  ],  "prediction": {  "date\_completed": "string",  "date\_created": "string",  "date\_formed": "string",  "prediction\_amount": int, "prediction\_id":int,  "prediction\_window": int,  "status": "string"  }  } |
| 4.1.13 | PUT | /prediction/finish/{pk} | Завершает предсказание, доступно модератору | pk: int  status:string | {  "forecasts": [  {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "id": int,  "image": "string",  "input": "string",  "measure\_type": "string",  "result": "string",  "short\_title": "string",  "title": "string"  }  ],  "prediction": {  "date\_completed": "string",  "date\_created": "string",  "date\_formed": "string",  "prediction\_amount":int,  "prediction\_id":int, "prediction\_window": int,  "status": "string"  }  } |
| 4.1.14 | DELETE | /pr\_fc/remove/{prediction\_id}/{forecast\_id} | Удаляет прогноз из предсказания, доступно аутентифицированному пользователю | prediction\_id: int,  forecast\_id: int |  |
| 4.1.15 | PUT | /pr\_fc/edit/{prediction\_id}/{forecast\_id} | Меняет поле input для прогноза в предсказании, доступно авторизованным пользователям | prediction\_id: int,  forecast\_id: int  {  "input": "string"  } | {  "forecast": {  "color": "string",  "descr": "string",  "ext\_desc": "string",  "forecast\_id":int,  "image": "string",  "measure\_type": "string",  "short\_title": "string",  "title": "string"  },  "forecast\_id": int,  "input": "string",  "prediction": {  "date\_completed": "string",  "date\_created": "string",  "date\_formed": "string",  "prediction\_amount": int, "prediction\_id":int,  "prediction\_window": int,  "status": "string"  },  "prediction\_id":int,  "preds\_forecs\_id": int,  "result": "string"  } |
| 4.1.16 | POST | /user/register | Регистрирует пользователя, доступно всем | {  "login": "string",  "password": "string"  } | {  "ok": true  } |
| 4.1.17 | PUT | /user/update/{pk} | Личный кабинет, доступно любому аутентифицированному пользователю | pk: int {  "login": "string",  "password": "string",  "role": int  } | {  "login": "string",  "password": "string",  "role": int  } |
| 4.1.18 | POST | /user/login | Аутентификация, доступна всем | {  "guest": true,  "login": "string",  "password": "string"  } | {  "access\_token": "string",  "expires\_in": "string",  "token\_type": "string"  } |
| 4.1.19 | POST | /user/logout | Деавторизация, доступна дюбому аутентифицированному пользователю |  |  |