

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

«___» _____ 2024 г.

**Разработка веб-сервиса для организации процесса ремонта
секций 300x360 МНЛЗ-5 ККЦ для ПАО «ЧМК»**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.04.2024.308-358.ВКР

Научный руководитель,
ст. преподаватель кафедры СП
_____ Н.С. Силкина

Автор работы,
студент группы КЭ-404
_____ А.О. Горбачев

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
«___» _____ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой СП

_____ Л.Б. Соколинский

29.01.2024 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра

студенту группы КЭ-404

Горбачеву Александру Олеговичу,

обучающемуся по направлению

09.03.04 «Программная инженерия»

1. Тема работы (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)

Разработка веб-сервиса для организации процесса ремонта секций 300x360

МНЛЗ-5 ККЦ для ПАО «ЧМК».

2. Срок сдачи студентом законченной работы: 03.06.2024 г.

3. Исходные данные к работе

3.1. Документация по C#. [Электронный ресурс] URL:

<https://learn.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 02.02.2024 г.).

3.2. Документация по TypeScript. [Электронный ресурс] URL:

<https://www.typescriptlang.org/docs/> (дата обращения: 02.02.2024 г.).

3.3. Документация по Microsoft SQL. [Электронный ресурс] URL:

<https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver16/> (дата обращения: 02.02.2024 г.).

3.4. Документация по React. [Электронный ресурс] URL:

<https://legacy.reactjs.org/docs/getting-started.html?> (дата обращения: 02.02.2024 г.).

4. Перечень подлежащих разработке вопросов

4.1. Провести анализ предметной области.

4.2. Спроектировать веб-сервис.

4.3. Реализовать веб-сервис.

4.4. Протестировать разработанный веб-сервис.

5. Дата выдачи задания: 29.01.2024 г.

Научный руководитель,
ст. преподаватель кафедры СП

Н.С. Силкина

Задание принял к исполнению

А.О. Горбачев

ГЛОССАРИЙ

1. *Сортамент* – это полный перечень различных видов сортов металлической продукции, который применяется в строительной и других отраслях [1].

2. *Чушковый чугун* – чугун, поставляемый в виде слитков в литейные цехи с металлургических заводов для переплавки при производстве [1].

3. *МНЛЗ* – это машина непрерывного литья заготовок. Используется на металлургических предприятиях для изготовления длинных заготовок [1].

4. *Кристаллизатор* – узел в МНЛЗ, предназначенный для приема жидкого металла, а также перевода части жидкой стали в твердое состояние посредством интенсивного отвода тепла охлаждающей водой [1].

5. *Затравка* – устройство в виде металлической штанги или цепи со съемной головкой, вводимое в кристаллизатор МНЛЗ для сцепления формирующего непрерывного слитка с головкой и его движения до входа в устройство для вытягивания [1].

6. *REST* – это программная архитектура, которая определяет условия работы API [2].

7. *API* – это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными [2].

ОГЛАВЛЕНИЕ

ГЛОССАРИЙ.....	4
ВВЕДЕНИЕ.....	6
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1. Описание предметной области	8
1.2. Обзор инструментов для реализации.....	10
2. ПРОЕКТИРОВАНИЕ	12
2.1. Анализ требований	12
2.2. Диаграмма вариантов использования.....	13
2.3. Хранилище данных	15
2.4. Архитектура веб-сервиса	17
2.5. Диаграмма деятельности.....	18
3. РЕАЛИЗАЦИЯ	20
3.1. Серверная часть веб-сервиса	20
3.2. Клиентская часть веб-сервиса	22
4. ТЕСТИРОВАНИЕ	29
ЗАКЛЮЧЕНИЕ	32
ЛИТЕРАТУРА.....	33
ПРИЛОЖЕНИЯ.....	35
Приложение А. Техническое задание.....	35
Приложение Б. Диаграмма деятельности.....	41

ВВЕДЕНИЕ

Актуальность

В современном промышленном производстве внедрение цифровых технологий и автоматизированных систем играет ключевую роль в оптимизации производственных процессов и повышении эффективности деятельности предприятий. В этом контексте особое внимание уделяется разработке веб-сервисов, специализированных программных решений, направленных на автоматизацию управления и организацию технологических процессов.

Тема разработки веб-сервиса для организации процесса ремонта секций 300x360 МНЛЗ-5 ККЦ для ПАО «ЧМК» является актуальной в свете необходимости оптимизации и улучшения производственных процессов на металлургических предприятиях. ПАО Челябинский металлургический комбинат (ЧМК) является крупным производителем стальной продукции, и эффективность его деятельности напрямую зависит от надежности и эффективности процессов технического обслуживания и ремонта оборудования, включая секции 300x360 МНЛЗ-5.

Введение веб-сервиса для организации процесса ремонта указанных секций позволит значительно сократить время на планирование и координацию работ, оптимизировать расходы на запасные части и материалы, а также повысить прозрачность и контролируемость процесса ремонта. Кроме того, такой сервис может предусматривать функционал для мониторинга состояния оборудования, прогностического анализа возможных отказов и оптимизации запасных частей на основе данных о работе оборудования.

В контексте стремления предприятий к повышению эффективности и снижению затрат, разработка веб-сервиса для организации процесса ремонта секций 300x360 МНЛЗ-5 ККЦ для ПАО «ЧМК» является важным шагом в направлении цифровой трансформации и современного управления производственными процессами.

Постановка задачи

Целью выпускной квалификационной работы является разработка веб-сервиса для организации процесса ремонта секций 300x360 МНЛЗ-5 ККЦ для ПАО «ЧМК». Для достижения поставленной цели необходимо решить следующие задачи:

- 1) провести анализ предметной области;
- 2) спроектировать веб-сервис;
- 3) реализовать веб-сервис;
- 4) протестировать разработанный веб-сервис.

Структура и содержание работы

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 41 страницу, объем списка литературы – 15 источников.

В первой главе описывается анализ предметной области, обзор инструментов для реализации.

Вторая глава посвящена проектированию разрабатываемой системы. Определяются основные функциональные и нефункциональные требования к системе. Также в главе описывается архитектура системы, схема базы данных и диаграмма деятельности.

В третьей главе приведены детали реализации и листинги, также отображены основные элементы интерфейса веб-сервиса.

Четвертая глава посвящена тестированию веб-сервиса. В ней были описаны методы тестирования и приведены результаты его проведения.

В приложении А содержится техническое задание на разработку системы.

В приложении Б приведена диаграмма деятельности, описывающая вариант использования «Создать новый отчет».

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание предметной области

Публичное акционерное общество «Челябинский металлургический комбинат» (ЧМК) [3] – одно из крупнейших в России предприятий полного металлургического цикла по выпуску качественных и высококачественных сталей.

ЧМК выпускает широкий сортамент продукции металлургического производства: чушковый чугун, полуфабрикаты стальные для дальнейшего передела, сортовой и листовой металлопрокат из углеродистых, конструкционных, инструментальных и коррозионностойких марок стали, фасонный прокат и рельсовую продукцию.

Для изготовления заготовок, которые впоследствии будут использоваться при производстве рельсовой продукции используется МНЛЗ. МНЛЗ (машина непрерывного литья заготовок) – металлургический агрегат для разливки стали. На рисунке 1 представлена схема МНЛЗ.

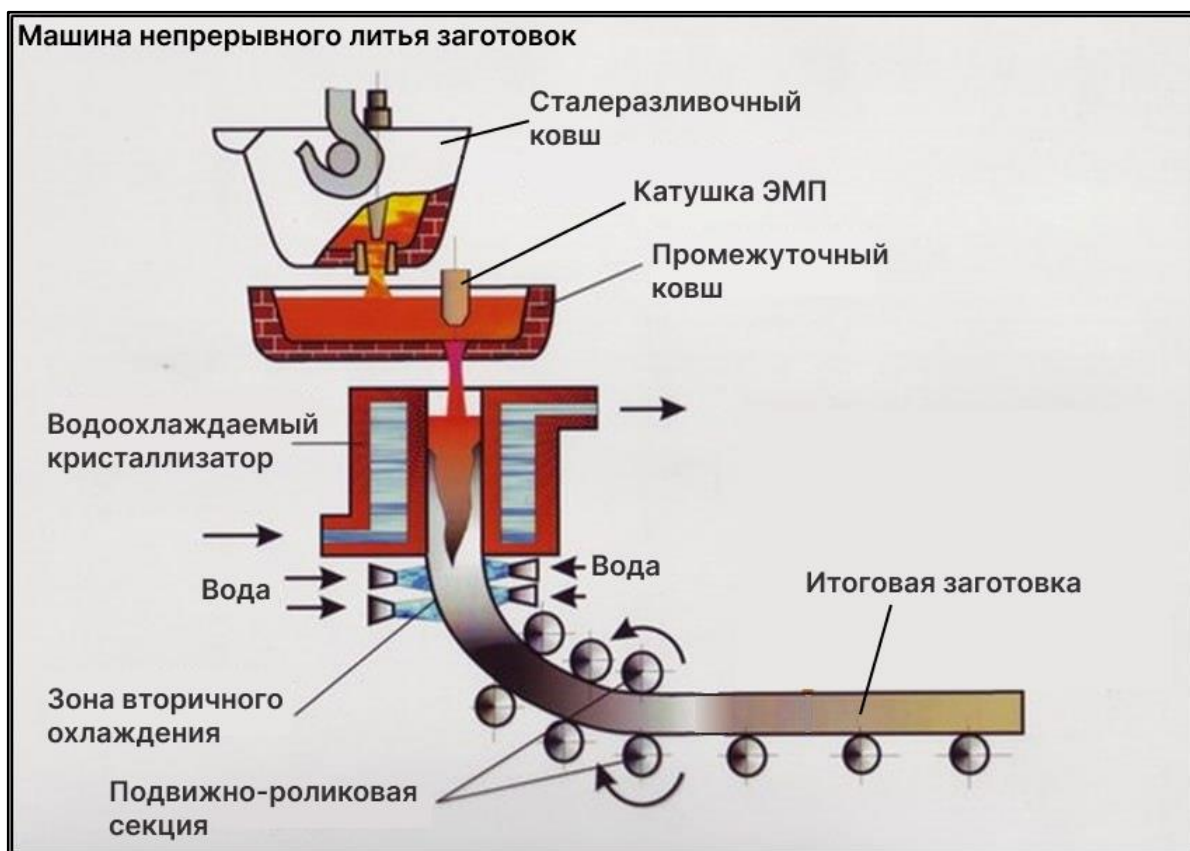


Рисунок 1 – Схема МНЛЗ

Принцип действия агрегата заключается в том, что жидкая сталь непрерывно заливается в водоохлаждаемую форму, называемую кристаллизатором. Перед началом заливки в кристаллизатор вводится специальное устройство с замковым захватом («затравка»), как дно для первой порции металла. После затвердевания металла затравка вытягивается из кристаллизатора, увлекая за собой формирующийся слиток. Поступление жидкого металла продолжается и слиток непрерывно наращивается. В кристаллизаторе затвердевают лишь поверхностные слои металла, образуя твердую оболочку слитка, сохраняющего жидкую фазу по центральной оси. Поэтому за кристаллизатором располагают зону вторичного охлаждения, называемая также второй зоной кристаллизации. В этой зоне в результате форсированного поверхностного охлаждения заготовка затвердевает по всему сечению. После чего происходит вытягивание заготовки с помощью подвижно-роликовой секции. Этот процесс слиткообразования является способом получения слитков неограниченной длины.

Поскольку МНЛЗ работает в непрерывном режиме, многие узлы агрегата в условиях высокой температуры часто изнашиваются и требуют постоянного наблюдения и ремонта. Чаще всего изнашиваются кристаллизаторы, роликовые секции и катушки ЭМП. Чтобы избежать полного выхода из строя агрегата, необходимо постоянно проводить отчетности о состоянии каждого узла.

В агрегате установлены пять комплексов, которые находятся в эксплуатации. Чтобы уменьшить время простоя агрегата в случае прорыва кристаллизатора на смену работающих комплексов готовится горячий резерв, это отремонтированные комплексы, готовые к быстрой замене на агрегате. Также выделяют холодный резерв, внедрение которого займет гораздо больше времени для установки. Секции, снятые с агрегата, разбираются и отправляются на ремонт. Кристаллизаторам заменяются прокладки, а подвижно-роликовые секции разукomплектовываются и обслуживаются. Катушки ЭМП заменяются на другие.

1.2. Обзор инструментов для реализации

Для реализации веб-сервиса используются языки программирования TypeScript и C#.

TypeScript [4] – язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript.

C# [5] – это объектно-ориентированный язык программирования, разработанный в Microsoft в конце 1990-х годов. Он является частью платформы .NET и широко используется для создания различных типов приложений, включая десктопные приложения, веб-приложения, игры, мобильные приложения и многое другое.

Для реализации веб-сервиса используются инструменты ASP.NET Core, React и Node.js, SCSS, HTML.

ASP.NET Core [6] – свободно-распространяемый кроссплатформенный фреймворк для создания веб-приложений на платформе .NET с открытым исходным кодом.

React [7] – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram (социальные сети, запрещенные в РФ) и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений.

Node.js [8] – это среда выполнения JavaScript, построенная на базе движка JavaScript V8 от Google Chrome. Она позволяет выполнять JavaScript на сервере, а не только в браузере. Node.js предоставляет средства для создания масштабируемых и быстрых сетевых приложений. Он обладает асинхронной и событийно-ориентированной архитектурой, что делает его особенно подходящим для работы с вводом-выводом и сетевыми операциями. Node.js широко используется для создания веб-серверов, API, микросервисов, приложений реального времени и многих других типов программного обеспечения.

SCSS [9] – это надмножество CSS, которое добавляет множество дополнительных функций и возможностей для улучшения процесса написания и организации кода стилей веб-приложений. SCSS предоставляет мощные инструменты, такие как переменные, вложенные правила, наследование и операторы, что значительно упрощает написание и поддержку стилей.

HTML [10] – это стандартизированный язык разметки, который используется для создания веб-страниц и их структурирования. HTML состоит из различных элементов, называемых тегами, которые определяют структуру содержимого страницы, такие как заголовки, абзацы, списки, ссылки и изображения. Эти теги обрамляют контент и сообщают браузеру, как должна отображаться информация на веб-странице.

Для реализации серверной части была выбрана среда разработки Visual Studio 2022, поскольку она предоставляет большое количество инструментов для реализации и развертывания веб-сервисов. Для реализации клиентского приложения было решено использовать редактора кода Visual Studio Code, так как он имеет большое количество возможностей, таких как подсветка синтаксиса, автозаполнение кода, применение различных расширений для ускорения разработки и улучшения качества кода.

Вывод по первой главе

В данной главе был проведен анализ функционирования агрегата, что позволяет определить его основные элементы, необходимые для ведения отчетности. Также был определен набор инструментов, необходимых для реализации проекта.

2. ПРОЕКТИРОВАНИЕ

2.1. Анализ требований

Функциональные требования к системе

Исходя из технического задания, указанного в приложении А, были выдвинуты следующие функциональные требования.

1. Пользователь должен получать информацию о том, какие секции находятся в эксплуатации.

2. Пользователь должен получать информацию о готовности горячего резерва.

3. Пользователь должен получать информацию о состоянии холодного резерва.

4. Пользователь должен получать информацию о том, какие секции разукomплектованы.

5. Пользователь должен иметь возможность получения информации о зафиксированных прорывах.

6. Пользователь должен иметь возможность изменения деталей в комплексах.

7. Пользователь должен иметь возможность получения информации о наличии на складе необходимых для ремонта материалов.

8. Пользователь должен иметь возможность оставить примечания о прорывах, состоянии горячего резерва и запаса материалов.

9. Пользователь должен иметь возможность просмотра предыдущих отчетов о комплексном состоянии МНЛЗ-5.

10. Пользователь должен иметь возможность создания новых отчетов о комплексном состоянии МНЛЗ-5.

11. Пользователь должен иметь возможность редактировать наличие составных частей комплексов.

12. Пользователь должен иметь возможность просмотра чертежей материалов для ремонта оборудования.

Нефункциональные требования к системе

Нефункциональные требования описывают свойства и ограничения, налагаемые на систему, которые определяют ее атрибуты качества. Исходя из технического задания, указанного в приложении А, были выдвинуты следующие нефункциональные требования.

1. Веб-сервис должен быть реализован на языках программирования TypeScript и C#.
2. Веб-сервис должен использовать для реализации реляционную базу под управлением Microsoft SQL Server [11].

2.2. Диаграмма вариантов использования

На рисунке 2 представлена диаграмма вариантов использования разрабатываемого веб-сервиса.

Мастер является авторизованным пользователем веб-сервиса. Пользователь может выполнять следующие функции:

- 1) посмотреть информацию о МНЛЗ-5 – просмотр информации о состоянии всех комплексов;
- 2) посмотреть информацию о подвижно-роликовой секции – просмотр информации о подвижно-роликовых секциях, внедренных в комплексы;
- 3) посмотреть информацию о кристаллизаторе – просмотр информации о кристаллизаторах, внедренных в комплексы;
- 4) посмотреть информацию о катушке – просмотр информации о катушках, внедренных в комплексы;
- 5) поменять или убрать деталь – мастер изменяет деталь в комплексе;
- 6) внести отметки – мастер вносит отметки о состоянии горячего резерва и о прорывах;
- 7) посмотреть наличие составных частей комплексов – просмотр составных частей комплексов, имеющихся на текущий момент;
- 8) добавить деталь – мастер добавляет новую деталь;

- 9) удалить деталь – мастер удаляет выбранную деталь;
- 10) поменять номер детали – мастер меняет номер выбранной детали;
- 11) удалить отчет – мастер удаляет выбранный отчет;

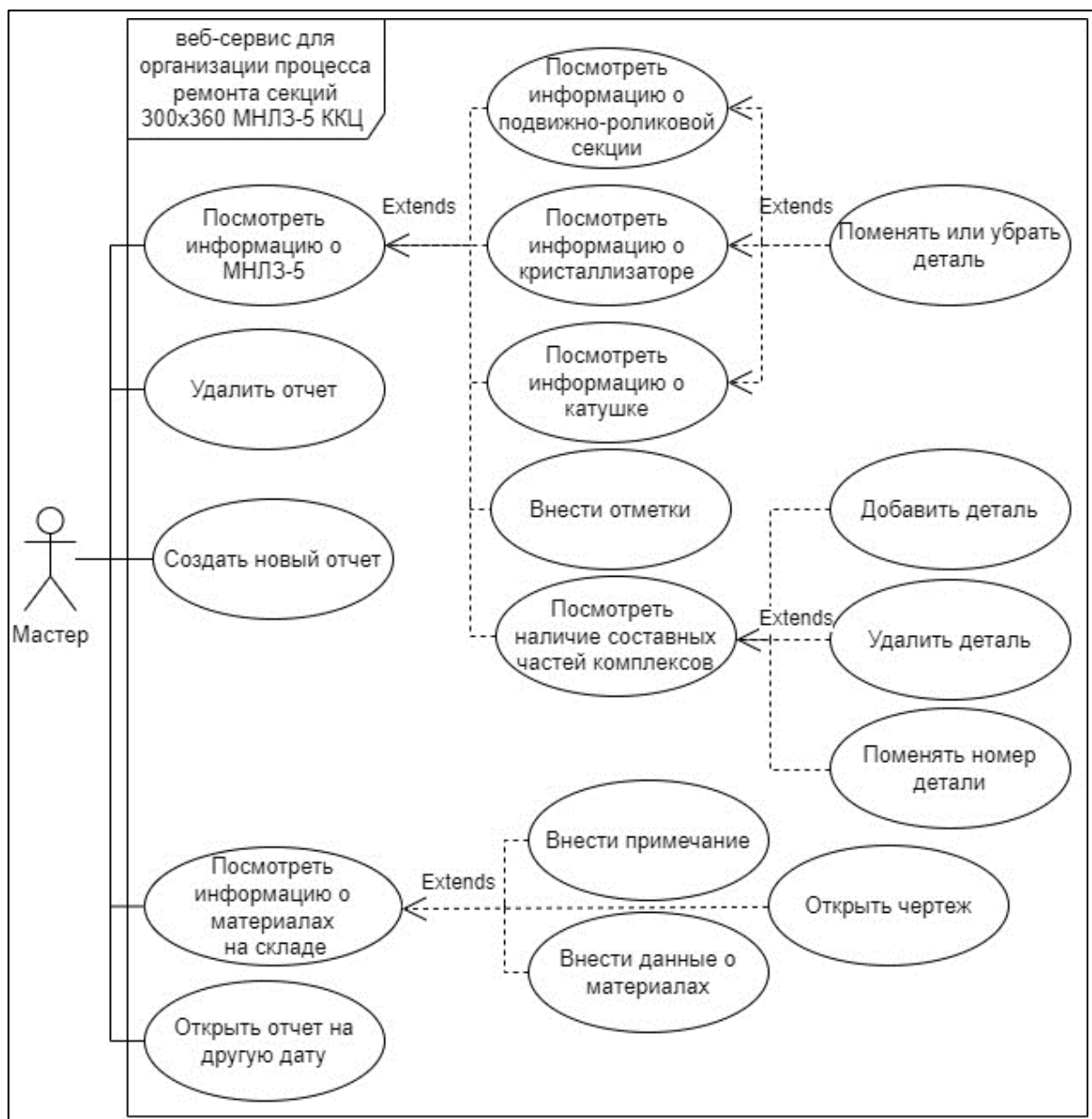


Рисунок 2 – Диаграмма вариантов использования

- 12) создать новый отчет – мастер создает новый отчет;
- 13) посмотреть информацию о материалах на складе – просмотр информации о необходимых для ремонта комплектующих;
- 14) внести примечание – мастер вводит примечания о материалах для ремонта;

15) открыть чертеж – мастер открывает чертеж, выбранного им материала;

16) внести данные о материалах – мастер вносит данные о актуальном наличии детали и номера заявки;

17) открыть отчет на другую дату – мастер открывает выбранный отчет о состоянии комплексов МНЛЗ-5 и о наличии материалов на складе.

2.3. Хранилище данных

База данных (БД) – это основная составляющая структуры веб-сервиса, предназначенная для хранения, изменения и обработки взаимосвязанной информации, которая необходима для функционирования системы. В качестве хранилища данных исходя из нефункциональных требований используется СУБД Microsoft SQL Server, которая обеспечивает высокую производительность, надежность и масштабируемость. СУБД Microsoft SQL Server предоставляет широкий набор функциональных возможностей, таких как поддержка транзакций, механизмы резервного копирования и восстановления данных, а также возможности для оптимизации запросов и управления доступом к данным.

Для хранения информации была спроектирована схема базы данных. База данных состоит из следующих десяти таблиц.

1. ReportForCCM5 – таблица хранит информацию о отметках, мастера, составившего отчет, дату формирования и статус отчета.

2. Materials – хранит информацию о всех материалах и инструментах, необходимых для ремонта агрегата.

3. ReportMaterials – таблица хранит связь материалов и отчетов, также хранит информацию о текущем количестве материалов на складе и номер заявки заказа.

4. Section – хранит информацию о секции.

5. Coil – хранит информацию о катушке.

6. Crystallizer – хранит информацию о кристаллизаторе.

7. `Complex` – хранит информацию о дате установки и снятия с эксплуатации комплекса, также хранит информацию о статусе комплекса.

8. `CoilComplex` – хранит связь катушки и комплекса, также хранит информацию о статусе, дате установки и снятия с эксплуатации катушки.

9. `CrystallizerComplex` – хранит связь кристаллизатора и комплекса, также хранит информацию о статусе, дате установки и снятия с эксплуатации кристаллизатора.

10. `ComplexSection` – хранит связь секции и комплекса, также хранит информацию о статусе, дате установки и снятия с эксплуатации секции.

Схема базы данных представлена на рисунке 3.

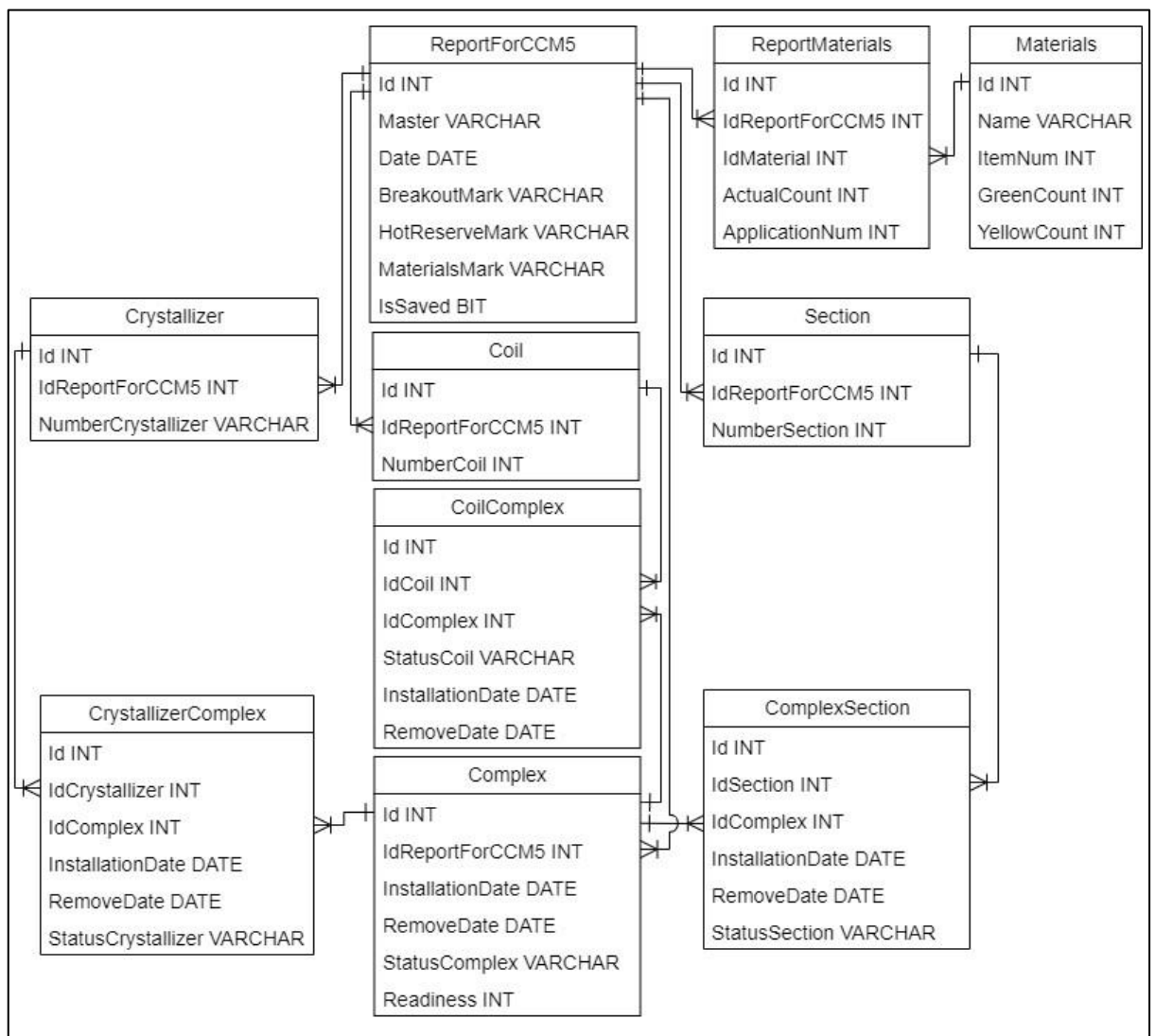


Рисунок 3 – Схема базы данных

Таблица `ReportForCCM5` связана связью один ко многим со следующими таблицами:

- `Coil`;
- `Section`;
- `Crystallizer`;
- `Complex`.

Таблица `Complex` связана с таблицами: `Crystallizer`, `Coil`, `Section` связью многие ко многим.

Таблица `Materials` связана с таблицей `ReportForCCM5` связью многие ко многим.

2.4. Архитектура веб-сервиса

Спроектированная система реализует трехслойную клиент-серверную архитектуру [12]. Архитектура представлена в виде диаграммы размещения на рисунке 4.

К плюсам трехслойной архитектуры можно отнести возможность более четкого разделения ответственностей между различными компонентами приложения, что способствует повышению модульности, гибкости и масштабируемости системы. Отделение представления данных от бизнес-логики и доступа к данным также облегчает процессы тестирования, отладки и сопровождения кода, поскольку изменения в одном слое могут быть реализованы без необходимости модификации других слоев. Ниже описаны уровни трехслойной архитектуры.

1. Уровень данных. Отвечает за хранение и поддержания целостности данных. В разрабатываемой системе представлен в виде реляционной базы данных.

2. Уровень обработки. Отвечает за обработку данных. Представляет собой серверную часть системы.

3. Уровень пользовательского интерфейса. Представлен в виде клиентского приложения. Отвечает за передачу задач и результатов, понятных пользователю.

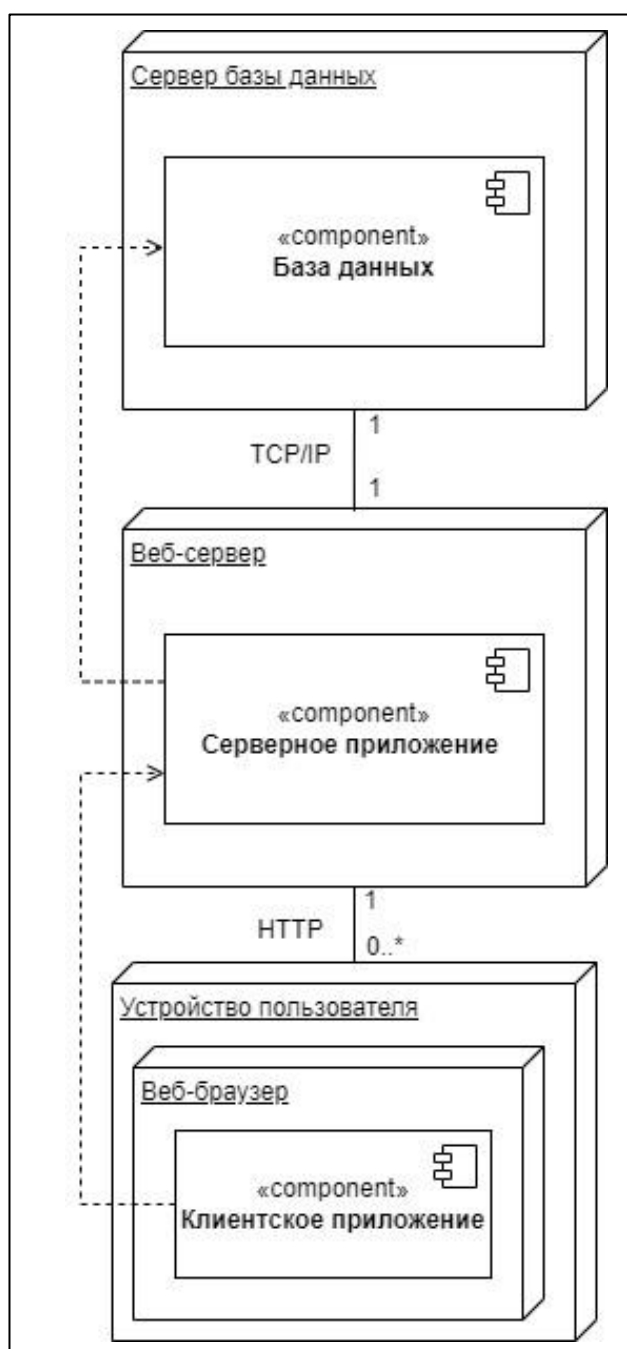


Рисунок 4 – Архитектура системы

2.5. Диаграмма деятельности

В ходе проектирования веб-сервиса была разработана диаграмма деятельности, которая описывает вариант использования «Создать новый отчет». Диаграмма деятельности представлена в приложении Б.

В предоставленной диаграмме подразумевается, что пользователь находится в режиме сохраненного отчета, в котором появляется возможность добавления нового отчета.

При нажатии пользователем на кнопку «Создать новый отчет», в клиентском приложении происходит запрос к серверу для получения данных всех существующих дат отчетов. После возвращения данных с сервера в клиентское приложение, происходит проверка, есть ли в полученном списке дат сегодняшняя, если в базе данных существует запись отчета с сегодняшней датой, то на веб-странице появляется модальное окно, в котором пользователю предлагается ввести другую дату для создания отчета. Получив дату для нового отчета, происходит запрос на сервер для его создания. На сервере происходит запрос к базе данных для получения дат всех отчетов, из которых выбирается последняя. По полученной дате определяется отчет, из которого будут копироваться данные. После этого происходит запрос на получение данных о кристаллизаторах, катушках ЭМП, секциях, комплексах и связующих их таблицах. Далее формируется запрос на создание нового отчета, заполняются таблицы компонентов и комплексов, также данными заполняются связующие таблицы компонентов комплексов и материалов. После создания нового отчета клиентскому приложению возвращается дата созданного отчета, для которой происходит запрос на серверную часть веб-сервиса для получения данных о новом отчете. Получив данные с серверной части веб-сервиса, в клиентском приложении происходит обновление соответствующих полей, отвечающих за отображение данных об отчете.

Вывод по второй главе

В данной главе были составлены функциональные и нефункциональные требования. Разработана диаграмма вариантов использования с описанием всех функций. Дано описание архитектуры системы, также была спроектирована база данных с описанием, содержащихся в ней таблиц. Также была описана диаграмма деятельности варианта использования «Создать новый отчет».

3. РЕАЛИЗАЦИЯ

3.1. Серверная часть веб-сервиса

При реализации серверной части веб-сервиса было решено использовать REST API. Основной идеей данного подхода является разделение операций на отдельные ресурсы, которые могут быть адресованы и изменены через стандартные HTTP методы:

- 1) GET – используется для получения данных;
- 2) POST – используется для создания новых данных;
- 3) PUT – используется для полного обновления ресурса;
- 4) DELETE – используется для удаления данных.

Это позволяет создавать гибкую и масштабируемую архитектуру, где каждый ресурс представляет собой некоторую сущность или коллекцию данных.

Рассмотрим пример реализации метода получения данных об отчете. При отправке HTTP запроса на сервере активируется метод контроллера, который был указан в URL ссылке запроса. В случае получения данных об отчете, используется метод `GetReportForCCM5`. Реализация метода получения данных об отчете представлена в листинге 1.

Листинг 1 – Метод `GetReportForCCM5`

```
[HttpGet("[action]")]
public async Task<IActionResult> GetReportForCCM5(string date)
{
    try
    {
        IEnumerable<ReportForCCM5> content = await repo.GetReport-
ForCCM5Async(date);
        return Ok(content);
    }
    catch (Exception ex) { return BadRequest(ex.Message); }
}
```

Чтобы получить данные, контроллер вызывает метод `GetReportForCCM5Async` для взаимодействия с базой данных. В нем происходит обращение к базе данных с входными данными, полученными из контроллера. Реализация метода представлена в листинге 2.

Листинг 2 – Метод GetReportForCCM5Async

```
public async Task<IEnumerable<ReportForCCM5>> GetReportForCCM5Async(string
date)
{
    string stmt = @"SELECT *
                    FROM ReportForCCM5
                    WHERE Date = @date";

    using (IDbConnection db = new SqlConnection(connectionString))
    {
        IEnumerable<ReportForCCM5> reportForCCM5 = await db.QueryAsync<Re-
portForCCM5>(stmt, new { date });
        if (reportForCCM5.Count() == 0 && date == DateTime.To-
day.ToString("yyyy-MM-dd"))
        {
            reportForCCM5 = await db.QueryAsync<ReportForCCM5>(stmt, new {
date });
        }
        return reportForCCM5;
    }
}
```

Таким образом было реализовано тридцать пять методов для взаимо-
действия с серверной частью веб-сервиса (таблица 1).

Таблица 1 – API маршруты

№	Метод	URL	Описание
1.	GET	/GetAllDatesFromReport- ForCCM5	Получение всех дат отчетов
2.	GET	/GetComplexReport	Получение всех комплексов отчета
3.	GET	/GetAllMaterialsForReport	Получение данных о материалах для ремонта
4.	GET	/GetCoils	Получение всех катушек
5.	GET	/GetSections	Получение всех секций
6.	GET	/GetCrystallizers	Получение всех кристаллизаторов
7.	GET	/GetRepSections	Получение всех секций, не находя- щихся в комплексах
8.	GET	/GetRepCrystallizers	Получение всех кристаллизаторов, не находящихся в комплексах
9.	GET	/GetRepCoils	Получение всех катушек, не находя- щихся в комплексах
10.	GET	/GetReportForCCM5	Получение данных об отчете
11.	POST	/CreateNewReportForCCM5	Создание нового отчета
12.	POST	/CreateFirstReport	Создание нового, пустого отчета
13.	POST	/AddNewCoil	Создание новой катушки
14.	POST	/AddNewSection	Создание новой секции
15.	POST	/AddNewCrystallizer	Создание нового кристаллизатора
16.	POST	/CreateComplex	Создание комплекса
17.	PUT	/UpdateReportForCCM5	Обновление данных об отчете
18.	PUT	/EditReport	Изменения состояния отчета для ре- дактирования
19.	PUT	/UpdateCoil	Обновление данных о катушке

№	Метод	URL	Описание
20.	PUT	/UpdateSection	Обновление данных о секции
21.	PUT	/UpdateCrystallizer	Обновление данных о кристаллизаторе
22.	PUT	/UpdateReportMaterial	Обновление данных о материалах
23.	PUT	/UpdateComplexReadiness	Обновление готовности комплекса
24.	PUT	/NullifyCoilsInComplex	Удаление катушки из комплекса
25.	PUT	/NullifyCrystallizersInComplex	Удаление кристаллизатора из комплекса
26.	PUT	/NullifySectionInComplex	Удаление секции из комплекса
27.	PUT	/UpdateCoilInComplex	Добавление катушки в комплекс
28.	PUT	/UpdateCrystallizerInComplex	Добавление кристаллизатора в комплекс
29.	PUT	/UpdateSectionInComplex	Добавление секции в комплекс
30.	PUT	/ChangeBreakoutDate	Изменение даты прорыва комплекса
31.	DELETE	/DeleteComplex	Удалить комплекс
32.	DELETE	/DeleteCoil	Удалить катушку
33.	DELETE	/DeleteSection	Удалить секцию
34.	DELETE	/DeleteCrystallizer	Удалить кристаллизатор
35.	DELETE	/DeleteReport	Удалить отчет

3.2. Клиентская часть веб-сервиса

Клиентская часть веб-сервиса представляет собой веб-приложение, созданное с использованием шаблона «create-react-app». Данное приложение разработано на языке программирования TypeScript.

Веб-сервис реализован в виде одностраничного приложения, что означает, что весь код загружается один раз при начальной загрузке страницы, а затем динамически обновляется при взаимодействии пользователя с интерфейсом, без необходимости перезагрузки страницы.

При разработке клиентской части был реализован основной компонент веб-сервиса, в котором пользователю можно создавать новые отчеты, редактировать данные, такие как:

- 1) мастер, сформировавший отчет;
- 2) комментарий о состоянии горячего резерва;
- 3) комментарий о произошедших прорывах комплексов;
- 4) комментарий о наличии материалов для ремонта.

Интерфейс отчета представлен на рисунке 5.

ВИЗУАЛИЗАЦИЯ РАБОЧИХ И РЕЗЕРВНЫХ СЕКЦИЙ 300X360 МНЛЗ-5

Дата: Мастер:



КОМПЛЕКСЫ В ЭКСПЛУАТАЦИИ

№ п/п	Номер кристаллизатора	Номер катушки ЭМП	Номер секции	Действия
1	<input type="text" value="5-03"/>	<input type="text" value="4"/>	<input type="text" value="13"/>	✘
2	<input type="text" value="5-08"/>	<input type="text" value="3"/>	<input type="text" value="15"/>	✘
3	<input type="text" value="5-04"/>	<input type="text" value="2"/>	<input type="text" value="14"/>	✘
4	<input type="text" value="5-05"/>	<input type="text" value="7"/>	<input type="text" value="16"/>	✘
5	<input type="text" value="5-02"/>	<input type="text" value="1"/>	<input type="text" value="17"/>	✘

Рисунок 5 – Интерфейс отчета

При открытии компонента с сервера запрашиваются данные отчета, комплексов и материалов. Пример данных о комплексах представлен в листинге 3.

Листинг 3 – Данные комплексов в формате JSON

```
{
  "complexId": 2116,
  "coilId": 1611,
  "coilNumber": 4,
  "coilStatus": "Эксплуатация",
  "sectionId": 1578,
  "complexStatus": "Эксплуатация",
  "readiness": 0,
  "removeDate": "0001-01-01T00:00:00",
  "sectionNumber": 13,
  "sectionStatus": "Эксплуатация",
  "crystallizerId": 1878,
  "crystallizerNumber": "5-03"
},
{
  "complexId": 2132,
  "coilId": 1614,
  "coilNumber": 3,
  "coilStatus": "Эксплуатация",
  "sectionId": 1580,
  "complexStatus": "Холодный",
  "readiness": 0,
```

```

        "removeDate": "2024-05-01T00:00:00",
        "sectionNumber": 15,
        "sectionStatus": "Эксплуатация",
        "crystallizerId": 1880,
        "crystallizerNumber": "5-08"
    },
    {
        "complexId": 2133,
        "coilId": 1613,
        "coilNumber": 2,
        "coilStatus": "Эксплуатация",
        "sectionId": 1579,
        "complexStatus": "Горячий",
        "readiness": 0,
        "removeDate": "2024-05-02T00:00:00",
        "sectionNumber": 14,
        "sectionStatus": "Эксплуатация",
        "crystallizerId": 1879,
        "crystallizerNumber": "5-04"
    }
}

```

После получения данных о комплексах, они передаются в дочерние компоненты таблиц, в которых объекты сортируются по ключу `complexStatus`. После сортировки массива объектов, данные заполняются в таблицу. Реализация сортировки и заполнения данных в таблицу в компоненте `BreakoutCCM5Table` представлена в листингах 4 и 5 соответственно.

Листинг 4 – Функция сортировки данных

```

const filteredData = data ? data.filter(item => {
  if (item.complexStatus !== "Авария") return false;
  if (saved) {
    return item.sectionNumber;
  }
  return true;
}) : [];

```

Листинг 5 – Реализация заполнения данных в таблицу

```

<td>
  {!saved ? (<select style={{minHeight: '30px'}} id = "coilNumSelect"
    value={item.coilNumber}
    onChange={e => handleCoilChange(item.complexId, parseInt(e.target.options[e.target.selectedIndex].id), item.coilId, date)}
    onClick={() =>{fetchCoils(reportId) setSelectedItem(`coil_${index}`)}}>
    <option value={item.coilNumber}>{item.coilNumber}</option>
    <option id="remove" value="remove">Удалить</option>
    {availableCoils.map((coil) => (
      <option id={coil.id} value={coil.numberCoil}>
        {coil.numberCoil}
      </option>
    ))}
  </select>) : (<>{item.coilNumber}</>)}</td>

```


В зависимости от содержимого параметра `saved` меняется отображение таблицы. Если в `saved` хранится `true`, то ячейки таблицы заполняются данными в виде текста. В обратном случае ячейки заполняются выпадающими списками, взаимодействие с которыми приводит к изменению содержания таблицы.

Для работы с материалами отчета также был создан отдельный компонент, представляющий из себя таблицу. Таблица фактического наличия комплектующих для ремонта представлен на рисунке 6.

№ п/п	Наименование материала	Ном. №	Кол-во	№ заявки
1	РВД 10 20 350 09/09 R/R 0 9205	49109361	124	55321
2	РВД 10 20 500 09/09 R/R 9205	49109288	247	31242
3	ФУТОРКА Ф69, 2x35	4999905	123	14212
4	СГОН 1/2 230 СЕКТОРА 300x360	1614992	213	41233
5	СГОН 1/2 257 СЕКТОРА 300x360	1614992	432	74904
6	СГОН 1/2 286 СЕКТОРА 300x360	1614991	19	
7	ТРУБКА МЕДНАЯ М1 О6 (кг)	1700103	234	23455
8	ШТУЦЕР УГЛОВ. G1/8-G1/4 90 DROPSA 911012	49100603	68	
9	ШТУЦЕР G1/4-G1/4 DROPSA 910007	4912989	65	32693
10	ШТУЦЕР G14 О6 DROPSA 92014	4913000	34	
11	НИППЕЛЬ О6 DROPSA 93006	4913005	203	23145
12	НИППЕЛЬ О10 DROPSA 93010	4913007	432	17355
13	ШТУЦЕР G3/8- G38 DROPSA 92141	4913020	56	29043
14	ШТУЦЕР G3/8 О10 DROPSA 92010	4913006	174	

Рисунок 6 – Таблица наличия комплектующих для ремонта

Для просмотра чертежей материалов был создан файл, в котором они хранятся. При нажатии на номенклатурный номер в строке таблицы, происходит поиск чертежа с таким названием материала. В случае нахождения

файла открывается модальное окно с его содержимым. Модальное окно чертежа представлено на рисунке 7.

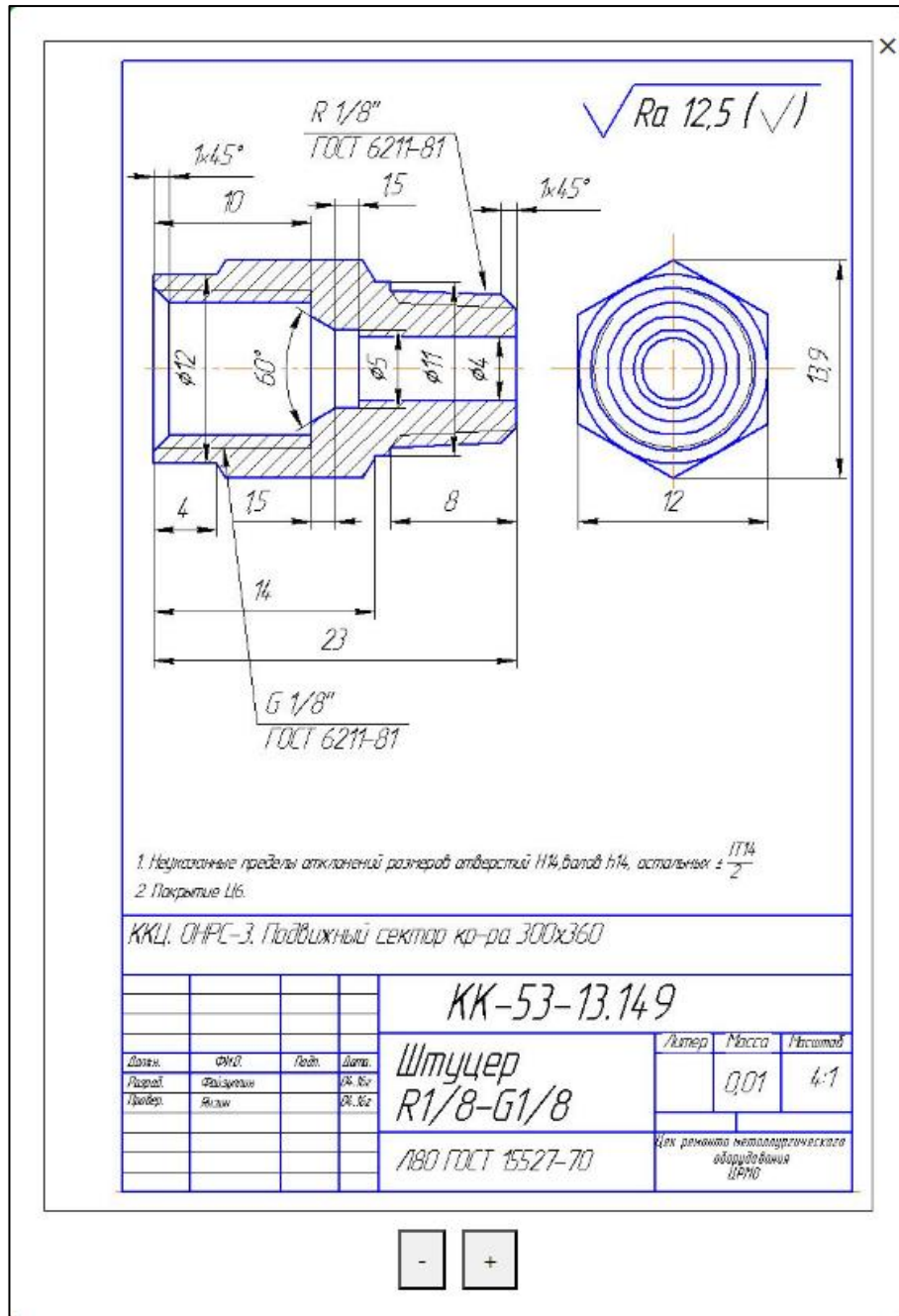


Рисунок 7 – Модальное окно чертежа

Функция открытия модального окна представлена в листинге 6.

Листинг 6 – Функция открытия модального окна

```
const openPopup = (itemName: string, itemNum: string) => {
  const imagePath = path.join('Materials', `${itemName.replaceAll("/", "_")}`);
  const imageExtensions = ['.jpg'];
  const findImage = (ext: string) => {
    const filePath = `${imagePath}.${ext}`;
```

```

const fileReader = new FileReader();

fileReader.onload = () => {
  if (fileReader.result) {
    setPopupImage(filePath);
    setHighlightedItemNums(prevState => [...prevState,
itemNum]);
  }
};

fileReader.onerror = () => {
  console.error(`Ошибка при чтении файла ${filePath}`);
};

fetch(filePath)
  .then(response => response.blob())
  .then(blob => fileReader.readAsBinaryString(blob))
  .catch(error => console.error(`Ошибка при загрузке файла
${filePath}:`, error));
};

for (const ext of imageExtensions) {
  findImage(ext);
}
};

```

Для актуализации данных о компонентах комплексов, также был создан отдельный компонент, представляющий из себя модальное окно, в котором посредством выбора в выпадающем списке кристаллизаторов, секций и катушек меняется содержимое окна. Модальное окно компонентов комплексов представлено на рисунке 8.

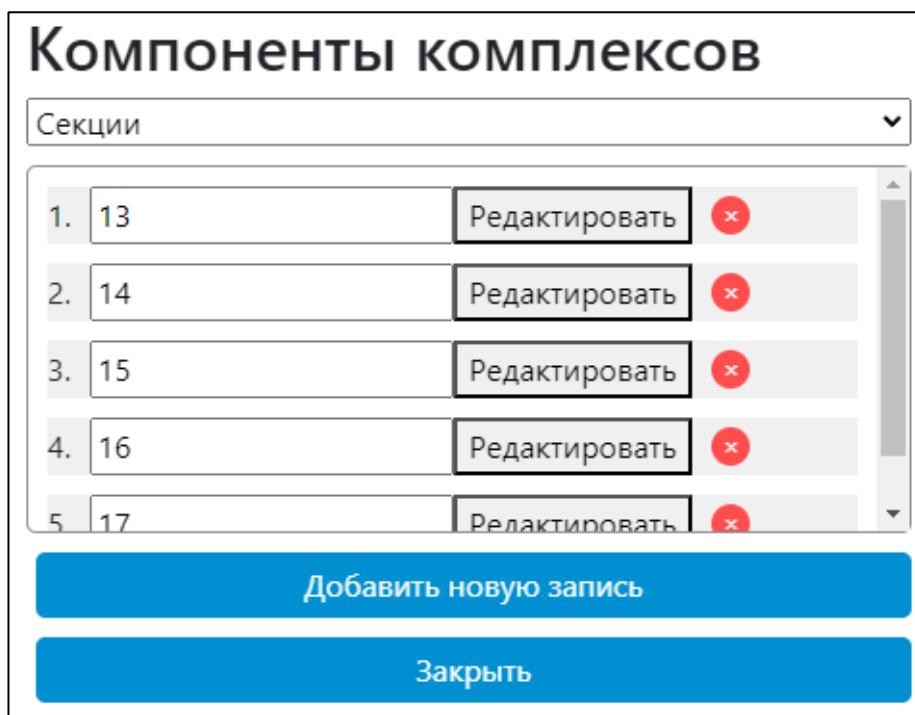


Рисунок 8 – Модальное окно компонентов комплексов

Реализация сохранения отредактированных компонентов комплексов

представлена в листинге 7.

Листинг 7 – Реализация сохранения отредактированных компонентов

```
const existingComponent = (components as Array<Crystallizer | Section |
Coil>).find((component): component is Crystallizer | Section | Coil => {
    if (componentType === 'Crystallizers') {
        return (component as Crystallizer).numberCrystallizer === re-
namedNumber;
    } else if (componentType === 'Sections') {
        return (component as Section).numberSection === renamedNumber;
    } else if (componentType === 'Coils') {
        return (component as Coil).numberCoil === renamedNumber;
    }
    return false;
});
if (existingComponent && existingComponent !== components[edit-
ingIndex]) {
    alert(`Компонент с номером ${renamedNumber} уже существует. Пожа-
луйста, введите другой номер.`);
    return;
}
switch (componentType) {
    case 'Crystallizers':
        renamedNumber = `${newCrystallizerNumber}`;
        await cHandler.RenameCrystallizer(components[editingIndex].id,
renamedNumber);
        break;
    case 'Sections':
        renamedNumber = parseInt(newSectionNumber as string, 10);
        await cHandler.RenameSection(components[editingIndex].id, re-
namedNumber);
        break;
    case 'Coils':
        renamedNumber = parseInt(newCoilNumber as string, 10);
        await cHandler.RenameCoil(components[editingIndex].id, re-
namedNumber);
        break;
    default:
        throw new Error('Invalid component type');
}
```

Для обеспечения адаптивности страниц при реализации клиентской части приложения использовались компоненты Bootstrap [13].

Вывод по третьей главе

В данной главе был описан способ реализации серверной и клиентской части веб-сервиса. Также были описаны маршруты взаимодействия с серверной частью.

4. ТЕСТИРОВАНИЕ

Функциональное тестирование

Функциональное тестирование представляет собой проверку программного обеспечения на его способность эффективно выполнять функциональные требования, т.е. на его способность решать задачи, которые необходимы пользователям в конкретных сценариях использования [14]. Было проведено функциональное тестирование веб-сервиса (таблица 2).

Таблица 2 – Функциональное тестирование

№	Название теста	Действие	Ожидаемый результат	Результат теста
1	Создание нового отчета	Пользователь, находясь на главной странице, нажимает кнопку «Создать новый отчет»	Происходит формирование и отображение нового отчета	Пройден
2	Редактирование информации о комплексах	Пользователь нажимает на компонент, который хочет заменить, и выбирает новый	Строка таблицы, соответствующего комплекса, заменяется на актуальные данные	Пройден
3	Просмотр чертежа детали	Пользователь нажимает на номенклатурный номер материала для ремонта	Отображается модальное окно с чертежом, выбранной детали	Пройден
4	Просмотр отчетов на другую дату	Пользователь нажимает на выпадающий список дат отчетов и выбирает необходимую дату	Происходит отображение отчета, на выбранную дату	Пройден
5	Сохранение отчета	Пользователь, после работы с отчетом, нажимает на кнопку «Сохранить»	Поля ввода и выбора деталей становятся недоступными для редактирования	Пройден
6	Редактирование предыдущих отчетов	Пользователь, перейдя на предыдущий отчет, нажимает на кнопку «Редактировать»	Поля ввода и выбора деталей становятся доступными для редактирования	Пройден
7	Создание новых компонентов комплексов	Пользователь нажимает на кнопку «Компоненты комплексов» и, находясь в модальном окне, нажимает кнопку «Добавить запись»	В список компонентов добавляется новая запись	Пройден

№	Название теста	Действие	Ожидаемый результат	Результат теста
8	Редактирование компонентов комплексов	Пользователь, находясь в модальном окне компонентов комплексов, нажимает на кнопку «Редактировать», вводит новый номер компонента и нажимает на кнопку «Сохранить»	Список компонентов обновляется с новыми данными	Пройден
9	Удалить компонент комплекса	Пользователь, находясь в модальном окне компонентов комплексов, нажимает на кнопку в виде крестика	Компонент удаляется и в список компонентов выводятся обновленные данные	Пройден
10	Удалить отчет	Пользователь нажимает на кнопку «Удалить отчет» и в модальном окне подтверждает удаление отчета	Происходит удаление отчета и открытие самого нового отчета, существующего в системе	Пройден

Юзабилити тестирование

Юзабилити тестирование – это исследование, выполняемое в целях определения, удобен ли некоторый искусственный объект для его предполагаемого применения [15]. Целью являлась проверка удобства разработанного интерфейса. Для этого были сформированы следующие задачи:

- 1) создать новый отчет;
- 2) создать новые записи во всех таблицах;
- 3) открыть модальное окно компонентов комплексов и создать новые кристаллизаторы, секции, катушки;
- 4) добавить новые компоненты во все таблицы;
- 5) внести отметки и примечания в соответствующие поля;
- 6) внести данные о актуальном количестве материалов и номере заявки в таблицу о наличии материалов;
- 7) открыть чертеж;
- 8) удалить отчет.

В проведенном тестировании приняли участие 4 сотрудника компании-заказчика. В ходе тестирования поставленные задачи были выполнены успешно, затруднений при использовании веб-сервиса не выявлено.

Тестирование пользовательского интерфейса

Было проведено тестирование разработанного интерфейса для экранов с разным разрешением. В результате тестирования ошибок не было обнаружено.

Также была осуществлена проверка работоспособности приложения в следующих браузерах:

- 1) Google Chrome;
- 2) Opera GX;
- 3) Яндекс Браузер;
- 4) Microsoft Edge;
- 5) Mozilla Firefox.

Во всех браузерах сайт отображается корректно.

Вывод по четвертой главе

В данной главе было описано проведение функционального и юзабилити тестирования системы, в результате которого все тесты были выполнены успешно, продемонстрировав, что разработанный веб-сервис выполняет все поставленные функции и является удобным для пользователя. Также описано тестирование пользовательского интерфейса, которое продемонстрировало корректное отображение интерфейса на устройствах с разным разрешением и при открытии веб-сервиса в разных браузерах.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы был разработан веб-сервис для организации процесса ремонта секций 300x360 МНЛЗ-5 ККЦ для ПАО «ЧМК». При выполнении работы были выполнены задачи, представленные ниже.

1. Проведен анализ предметной области, в результате которого была определена специфика работы предприятия. Также были выявлены необходимые средства для реализации системы.

2. Был спроектирован веб-сервис. Определены основные функциональные требования, варианты использования системы. Спроектирована схема хранилища данных и была определена архитектура системы.

3. Реализован веб-сервис с использованием технологий ASP.NET, Node.js, React.

4. Произведено тестирование веб-сервиса, в результате которого была успешно проведена проверка работоспособности системы.

Произведено опытное внедрение разработанного веб-сервиса на предприятии, результатом которого стал акт, подписанный 22.05.2024 г. исполняющим обязанности директора департамента АСУТП Резепиным И.Н.

Разработанный веб-сервис имеет базовый функционал. В качестве развития можно выделить функции, необходимые для улучшения функциональности системы: добавить синхронизацию базы данных с базой данных материалов на складе, добавить функцию автоматического определения необходимых к закупке материалов и формирования заявки на их покупку, для поддержания непрерывного процесса изготовления заготовок.

ЛИТЕРАТУРА

1. Бахтинов В.Б. Технология прокатного производства. // Москва: Металлургия, 1983. – 232 с.
2. Ивонин О.А. Создание клиент-серверного приложения на основе restful api архитектуры / О.А. Ивонин, А.Г. Гречихин, И.Р. Гильматдинов.// Москва: Молодой ученый, 2021. – С. 18–20.
3. Мечел. [Электронный ресурс] URL: <https://mechel.ru/sector/steel/chelyabinskiy-metallurgicheskiy-kombinat/> (дата обращения: 04.02.2024 г.).
4. Документация по TypeScript. [Электронный ресурс] URL: <https://www.typescriptlang.org/docs/> (дата обращения: 02.02.2024 г.).
5. Документация по C#. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 02.02.2024 г.).
6. ASP.NET Core | Microsoft Learn. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio> (дата обращения: 04.02.2024 г.).
7. Документация по React. [Электронный ресурс] URL: <https://legacy.reactjs.org/docs/getting-started.html?> (дата обращения: 02.02.2024 г.).
8. Node.js. [Электронный ресурс] URL: <https://nodejs.org/en> (дата обращения: 06.02.2024 г.).
9. Фримен Э., Фримен Э. Изучаем HTML, XHTML и CSS. 2-е изд. – СПб.: Питер, 2014 г. – С. 36–40.
10. Препроцессор SCSS. [Электронный ресурс] URL: <https://sass-lang.com/guide/#preprocessing> (дата обращения: 15.03.2024 г.).
11. Документация по Microsoft SQL. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver16> (дата обращения: 02.02.2024 г.).
12. Стин ван М., Таненбаум Э.С. Распределенные системы. / пер. с англ. В.А. Яроцкого. – М.: ДМК Пресс, 2021. – 584 с.

13. Документация Bootstrap для React. [Электронный ресурс] URL: <https://react-bootstrap.netlify.app/> (дата обращения: 24.04.2024 г.).

14. Старолетов С.М. Основы тестирования и верификации программного обеспечения: учебное пособие – 2-е изд. // Санкт–Петербург: Лань, 2020. – 344 с.

15. Карпович Е.Е. Методы тестирования и отладки программного обеспечения: учебник. // Москва: МИСИС, 2020. – 136 с.

ПРИЛОЖЕНИЯ

Приложение А. Техническое задание

Техническое задание на разработку веб-сервиса для организации процесса ремонта секций 300X360 МНЛЗ-5 ККЦ для ПАО «ЧМК»

1. Общие сведения

1.1. Полное наименование системы

Веб-сервис для организации процесса ремонта секций 300x360 МНЛЗ-5 ККЦ.

1.2. Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы

Исполнитель: Горбачев Александр Олегович, студент группы КЭ-404, ВШЭКН, ЮУрГУ.

Заказчик: ПАО «ЧМК», Департамент АСУТП.

Адрес: 2-я Павелецкая ул., 14, Челябинск, Челябинская обл., 45404.

1.3. Плановые сроки начала и окончания работы по созданию системы

Дата начала работы: 1 февраля 2024.

Дата окончания работы: 22 мая 2024.

2. Назначение и цели создания системы

2.1. Назначение системы

Веб-сервис для организации процесса ремонта секций 300x360 МНЛЗ-5 ККЦ предназначен для организации процесса ремонта комплексов МНЛЗ-5. Система предназначена для мастеров, исполняющих ремонтные работы над агрегатом. Мастер вносит данные о аварийных ситуациях, произошедших в процессе работы агрегата, а также заполняет данные о состоянии разукomплектованных подвижно-роликовых секций и о состоянии

горячего и холодного резерва. Также одной из основных задач системы является контроль наличия на складе необходимых материалов для ремонта оборудования.

2.2. Цели создания системы

Создание системы должно обеспечить достижение следующих целей:

- оптимизация процесса ремонта;
- повышение эффективности работы;
- улучшение управления ресурсами;
- повышение прозрачности процесса;
- повышение безопасности на предприятии;
- оптимизация проведения аналитики и отчетности.

3. Требования к системе

3.1. Функциональные требования

1. Пользователь должен получать информацию о том, какие секции находятся в эксплуатации.

2. Пользователь должен получать информацию о готовности горячего резерва.

3. Пользователь должен получать информацию о состоянии холодного резерва.

4. Пользователь должен получать информацию о том, какие секции разукомплектованы.

5. Пользователь должен иметь возможность получения информации о зафиксированных прорывах.

6. Пользователь должен иметь возможность изменения деталей в комплексах.

7. Пользователь должен иметь возможность получения информации о наличии на складе необходимых для ремонта материалов.
8. Пользователь должен иметь возможность оставить примечания о прорывах, состоянии горячего резерва и запаса материалов.
9. Пользователь должен иметь возможность просмотра предыдущих отчетов о комплексном состоянии МНЛЗ-5.
10. Пользователь должен иметь возможность создания новых отчетов о комплексном состоянии МНЛЗ-5.
11. Пользователь должен иметь возможность редактировать наличие составных частей комплексов.
12. Пользователь должен иметь возможность просмотра чертежей материалов для ремонта оборудования.

3.2. Нефункциональные требования

1. Веб-сервис должен быть реализован на языках программирования TypeScript и C#.
2. Веб-сервис должен использовать для реализации реляционную базу под управлением Microsoft SQL Server.

3.3. Исходные данные

3.3.1. Таблица «Комплексы в эксплуатации»

Таблица содержит следующий набор полей:

- 1) номер кристаллизатора – порядковый номер кристаллизатора, в эксплуатируемом комплексе;
- 2) номер катушки ЭМП – порядковый номер катушки ЭМП, в эксплуатируемом комплексе;
- 3) номер секции – порядковый номер секции, в эксплуатируемом комплексе.

Пример таблицы изображен на рисунке 1.

№ п/п	Номер кристаллизатора	Номер катушки ЭМП	Номер секции
1	5-05.	5	6
2	5-07.	4	3
3	5-06.	1	10
4	5-03.	3	1
5	5-10.	3	9

Рисунок 1 – Таблица «Комплексы в эксплуатации»

3.3.2. Таблица «Зафиксированные прорывы в период разливки»

Таблица содержит следующие поля:

- 1) номер кристаллизатора – порядковый номер кристаллизатора комплекса, на котором произошел прорыв;
- 2) номер секции – порядковый номер секции комплекса, на котором произошел прорыв;
- 3) дата прорыва НЛЗ – дата, в которую произошел прорыв комплекса.

Пример таблицы изображен на рисунке 2.

Номер кристаллизатора	Номер секции	Дата прорыва НЛЗ
5-01.	7	16.01.24г.

Рисунок 2 – Таблица «Зафиксированные прорывы в период разливки»

3.3.3. Таблица «Состояние горячего резерва»

Таблица содержит следующие поля:

- 1) номер кристаллизатора – порядковый номер кристаллизатора, находящегося в горячем резерве;
- 2) номер катушки ЭМП – порядковый номер катушки ЭМП, находящейся в горячем резерве;

- 3) номер секции – порядковый номер секции, находящейся в горячем резерве;
- 4) процент готовности – готовность комплекса на момент формирования отчета.

Пример таблицы изображен на рисунке 3.

Номер кристаллизатора	Номер катушки ЭМП	Номер секции	% Готовности
5-08	5	4	100 %

Рисунок 3 – Таблица «Состояние горячего резерва»

3.3.4. Таблица «Состояние холодного резерва»

Таблица содержит следующие поля:

- 1) номер секции – порядковый номер секции, находящейся в холодном резерве;
- 2) процент готовности – готовность комплекса на момент формирования отчета.

Пример таблицы изображен на рисунке 4.

Номер секции	% Готовности
5	50 %
7	60 %
1	70 %

Рисунок 4 – Таблица «Состояние холодного резерва»

3.3.5. Таблица «Разукомплектованные секции»

Таблица содержит поле номер секции. Номер секции – порядковый номер секции, находящейся в разукомплектованном состоянии.

Пример таблицы изображен на рисунке 5.

Номер секции
2
8
11
12

Рисунок 5 – Таблица «Разукомплектованные секции»

3.3.6. Таблица «Фактическое наличие комплектующих»

Таблица содержит следующие поля:

- 1) наименование материала – наименование материала, необходимого для ремонта узлов агрегата;
- 2) номенклатурный номер – номенклатурный номер материала;
- 3) количество – количество материала, которое находится на складе;
- 4) номер заявки – номер заявки на приобретения материала.

Пример таблицы изображен на рисунке 6.

№ п/п	Наименование материала	Ном. №	Кол-во	№ заявки
1	РВД 10 20 350 09/09 R/R 0 9205	4912432	350	5477
2	РВД 10 20 500 09/09 R/R 0 9205	4325546	33	4576
3	ФУТОРКА Ф69, 2х35	3465767	0	6876

Рисунок 6 – Таблица «Фактическое наличие комплектующих»

Приложение Б. Диаграмма деятельности

На рисунке 7 представлена диаграмма деятельности варианта использования «Создать новый отчет».

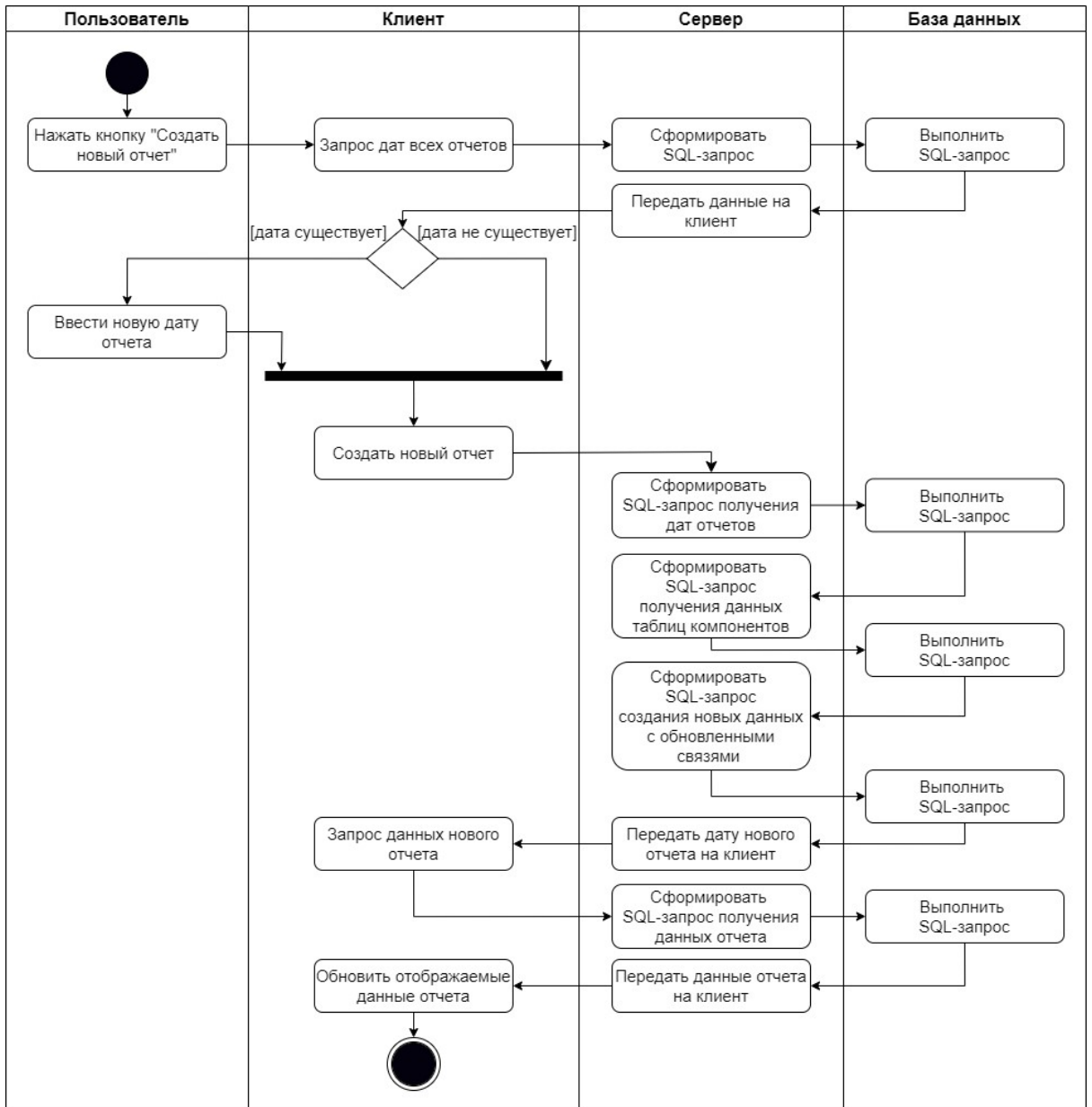


Рисунок 7 – Диаграмма деятельности создания нового отчета