

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 описания состояния  
кристаллизатора и подвижно-роликовой секции 300x360  
МНЛЗ-5 ККЦ для ПАО «ЧМК»**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 09.03.04.2024.308-357.ВКР

Научный руководитель,  
ст. преподаватель кафедры СП  
\_\_\_\_\_ А.А. Шаблей

Автор работы,  
студент группы КЭ-404  
\_\_\_\_\_ О.В. Горб

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

Горб Олегу Владимировичу,

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 описания состояния кристаллизатора и подвижно-роликовой секции 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: 10.02.2024 г.).

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

<https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver16/> (дата обращения: 10.02.2024 г.).

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

<https://www.typescriptlang.org/docs/> (дата обращения: 10.02.2024 г.).

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

<https://ru.react.js.org/docs/getting-started.html/> (дата обращения: 10.02.2024 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 веб-сервис.

**5. Дата выдачи задания: 29.01.2024 г.**

**Научный руководитель,**  
ст. преподаватель кафедры СП

А.А. Шаблей

**Задание принял к исполнению**

О.В. Горб

## ГЛОССАРИЙ

1. *JavaScript* – это язык программирования, который используют разработчики для создания интерактивных веб-страниц. Функции JavaScript могут улучшить удобство взаимодействия пользователя с веб-сайтом: от обновления ленты новостей в социальных сетях и до отображения анимации и интерактивных карт [1].

2. *TypeScript* – это расширенная версия языка JavaScript. TypeScript помогает избавиться от типичных проблем JavaScript: ошибок типов в рантайме и неконтролируемо разрастающегося кода [2].

3. *.NET Framework* – это технология, которая поддерживает создание и выполнение веб-служб и приложений Windows [3].

4. *Node* или *Node.js* – программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API [4].

5. *ASP.NET Core* – свободно распространяемый кроссплатформенный фреймворк для создания веб-приложений на платформе .NET с открытым исходным кодом [5].

6. *Microsoft SQL Server* – система управления базой данных от Microsoft, которая обеспечивает масштабируемость и широкий набор функциональности для предприятий и крупных организаций [6].

7. *React* – это библиотека JavaScript с открытым кодом для создания внешних пользовательских интерфейсов [7].

8. *МНЛЗ* – это машина непрерывного литья заготовок. Используется на металлургических предприятиях, для изготовления заготовок [8].

9. *API* – это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными [9].

10. *SCSS* – это надмножество CSS, которое добавляет множество дополнительных функций и возможностей для организации кода стилей [10].

## **ОГЛАВЛЕНИЕ**

ГЛОССАРИЙ.....	4
ВВЕДЕНИЕ.....	6
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	8
1.1. Описание предметной области .....	8
1.2. Обзор инструментов для реализации.....	9
2. ПРОЕКТИРОВАНИЕ .....	10
2.1. Анализ требований.....	10
2.2. Архитектура приложения.....	11
2.3. Диаграмма вариантов использования .....	13
3. РЕАЛИЗАЦИЯ .....	15
3.1. Хранилище данных .....	15
3.2. Реализация серверной части .....	18
3.3. Реализация клиентской части .....	20
3.4. Интеграция клиентской и серверной частей.....	24
4. ТЕСТИРОВАНИЕ .....	27
ЗАКЛЮЧЕНИЕ .....	29
ЛИТЕРАТУРА .....	30
ПРИЛОЖЕНИЕ. Техническое задание.....	32

## **ВВЕДЕНИЕ**

### **Актуальность**

Актуальность этой темы в сфере промышленности заключается в повышении эффективности и автоматизации производственных процессов. Разработкой такого веб-сервиса можно значительно упростить и ускорить мониторинг и анализ состояния оборудования. Документированные состояния кристаллизаторов и подвижно-роликовых секций 300x360 МНЛЗ-5 ККЦ позволят значительно облегчить и автоматизировать управление ремонтом этого оборудования. Это также позволит оперативно реагировать на возможные проблемы и предотвращать аварии, что в свою очередь сэкономит время и ресурсы компании. ПАО Челябинский металлургический комбинат (ЧМК) [11] является крупным производителем стальной продукции, и эффективность его деятельности напрямую зависит от надежности и эффективности процессов технического обслуживания и ремонта оборудования, включая кристаллизаторы и подвижно-роликовые секции 300x360 МНЛЗ-5 ККЦ.

В рамках стремления предприятий к повышению эффективности и снижению затрат, разработка веб-сервиса для описания состояния кристаллизаторов и подвижно-роликовых секций 300x360 МНЛЗ-5 ККЦ для ПАО «ЧМК» является важным шагом в направлении цифровой трансформации и современного управления производственными процессами.

Веб-сервис с использованием React позволяет создать удобный и интуитивно понятный интерфейс для пользователей, платформа ASP.Net позволяет создать надежное и производительное серверное приложение для веб-сервиса, а база данных является основой для хранения и организации информации о кристаллизаторах и подвижно-роликовых секциях.

### **Постановка задачи**

Целью выпускной квалификационной работы является Разработка веб-сервиса для описания состояния кристаллизатора и подвижно-роликовой секции 300x360 МНЛЗ-5 ККЦ для ПАО «ЧМК».

Для достижения поставленной цели необходимо решить следующие задачи:

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

### **Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 43 страницы, объем списка литературы – 15 источников.

В первой главе описывается предметная область проекта, приводится обзор на средства разработки.

Вторая глава посвящена проектированию разрабатываемой системы. Определяются основные функциональные и нефункциональные требования к системе. Также в ней описывается архитектура системы, приведена диаграмма размещения основных компонентов системы, приведена диаграмма вариантов использования системы.

Третья глава посвящена реализации системы. В ней приведены схема базы данных, описаны выбранные средства разработки. Также в ней приведены детали реализации и листинги серверной и клиентской части приложения, описаны методы интеграции этих частей, элементы интерфейса разработанного веб-сервиса.

В четвертой главе приведено тестирование системы. В ней были описаны методы тестирования и приведены результаты его проведения.

В приложении приведено техническое задание на разработку системы.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Описание предметной области

ПАО «Челябинский металлургический комбинат» является одним из крупнейших предприятий полного металлургического цикла в России. Комбинат специализируется на производстве высококачественных сталей и предлагает широкий выбор металлургической продукции, включая чушковый чугун, полуфабрикаты стальные, различные категории листового и сортового металлопроката, а также фасонный прокат и рельсовую продукцию.

В процессе производства рельсовой продукции на ЧМК используется Машина Непрерывного Литья Заготовок (МНЛЗ). МНЛЗ используется для разливки стали. Работа агрегата основана на непрерывной заливке жидкой стали в охлаждаемую водой форму, называемую кристаллизатором. Процесс начинается с внесения специального устройства, называемого «затравкой», в кристаллизатор. Затравка служит основой для первой плавки стали. После затвердевания, затравка вытягивается, захватывая формирующийся слиток. Затем продолжается поступление жидкой стали, и слиток непрерывно наращивается. Во время процесса затвердевания в кристаллизаторе образуется твердая оболочка, сохраняющая жидкую фазу по центральной оси слитка. Для полного затвердевания, заготовка проходит через вторую зону кристаллизации, где происходит форсированное поверхностное охлаждение, чтобы обеспечить затвердевание по всему сечению. Этот процесс создания слитков может быть осуществлен без ограничения по длине слитка.

Поскольку МНЛЗ работает в непрерывном режиме, многие его компоненты подвержены износу и требуют постоянного мониторинга и ремонта из-за высоких температур. Чаще всего износу подвержены кристаллизаторы и роликовые секции. Для предотвращения поломок агрегата необходимо постоянно отслеживать состояние каждого компонента и проводить регулярное обслуживание.



## **1.2. Обзор инструментов для реализации**

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

Для реализации серверной части используется платформа ASP.NET и язык программирования C#. ASP.NET обладает мощным функционалом и хорошей масштабируемостью. C# – современный и удобный язык программирования, который обеспечивает высокую производительность и безопасность кода. Благодаря эффективной интеграции между ASP.NET и C# можно создать надежное и производительное серверное приложение для веб-сервиса.

Для реализации клиентской части используется язык программирования TypeScript и технология React. Использование TypeScript с React обеспечивает множество преимуществ. TypeScript добавляет статическую типизацию к JavaScript, что способствует удобству разработки и обнаружению ошибок на ранних этапах. React, в свою очередь, позволяет создавать динамичные и масштабируемые пользовательские интерфейсы, делая взаимодействие с веб-сервисом более удобным и приятным для пользователей.

В качестве СУБД используется Microsoft SQL Server. Microsoft SQL Server – надежная и производительная реляционная СУБД. Ее широкие возможности в области безопасности, масштабируемости и управления данными делают ее отличным выбором для хранения и обработки информации, необходимой для функционирования веб-сервиса. Благодаря интеграции с платформой ASP.NET, Microsoft SQL обеспечит эффективное взаимодействие между данными и серверным приложением.

### **Вывод по первой главе**

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

## **2. ПРОЕКТИРОВАНИЕ**

### **2.1. Анализ требований**

#### **Функциональные требования к системе**

Функциональные требования описывают требуемое поведение системы. Для реализации приложения были определены следующие функциональные требования:

- 1) пользователь должен иметь возможность выбора даты, выбора секции и кристаллизатора;
- 2) пользователь должен иметь возможность просматривать информацию о состоянии выбранного кристаллизатора в определенный день;
- 3) пользователь должен иметь возможность просматривать информацию о состоянии выбранной секции в определенный день;
- 4) пользователь должен иметь возможность редактировать информацию о состоянии выбранного кристаллизатора в определенный день;
- 5) пользователь должен иметь возможность редактировать информацию о состоянии выбранной секции в определенный день;
- 6) система должна оснащаться функцией валидации, чтобы обеспечить проверку введенных пользователем данных на соответствие заданным критериям.

#### **Нефункциональные требования к системе**

Нефункциональные требования определяют свойства и ограничения которых должна придерживаться реализуемая система. Для реализации будущего приложения были выдвинуты следующие нефункциональные требования:

- 1) серверная часть должна быть реализована на платформе ASP.NET;
- 2) клиентская часть должна быть реализована с использованием языка TypeScript и технологии React;
- 3) в качестве СУБД для реляционной базы данных необходимо использовать Microsoft SQL.

## 2.2. Архитектура приложения

Архитектура приложения построена на основе клиент-серверного взаимодействия. Можно выделить 3 основных части приложения, обеспечивающие это взаимодействие, а именно: клиентскую часть, серверную часть и базу данных. Архитектура представлена в виде диаграммы размещения на рисунке 1.

Клиентская часть приложения представляет собой интерфейс, через который пользователь взаимодействует с данными. В данном случае, клиентская часть разработана с использованием фреймворка React.js, обеспечивающего быструю и отзывчивую работу интерфейса. Также использование этого фреймворка позволяет масштабировать и поддерживать проект, если в будущем будут вноситься изменения и улучшения.

Серверная часть играет ключевую роль в обработке запросов от клиентской части, содержащей важную бизнес-логику для удовлетворения функциональных требований. Кроме того, серверная часть активно взаимодействует с базой данных Microsoft SQL, где хранятся и запрашиваются данные, обеспечивая устойчивость и эффективное управление необходимой информацией.

База данных (Microsoft SQL). База данных Microsoft SQL играет ключевую роль в хранении и обработке данных приложения. Здесь данные структурируются и хранятся в таблицах, обеспечивая целостность данных и эффективную обработку запросов. Также имеется возможность оптимизировать работу с базой данных, создавая связи между таблицами и индексируя необходимые поля для ускорения операций чтения и записи данных. Грамотное проектирование базы данных позволяет обеспечить быстрый доступ ко всей необходимой информации.

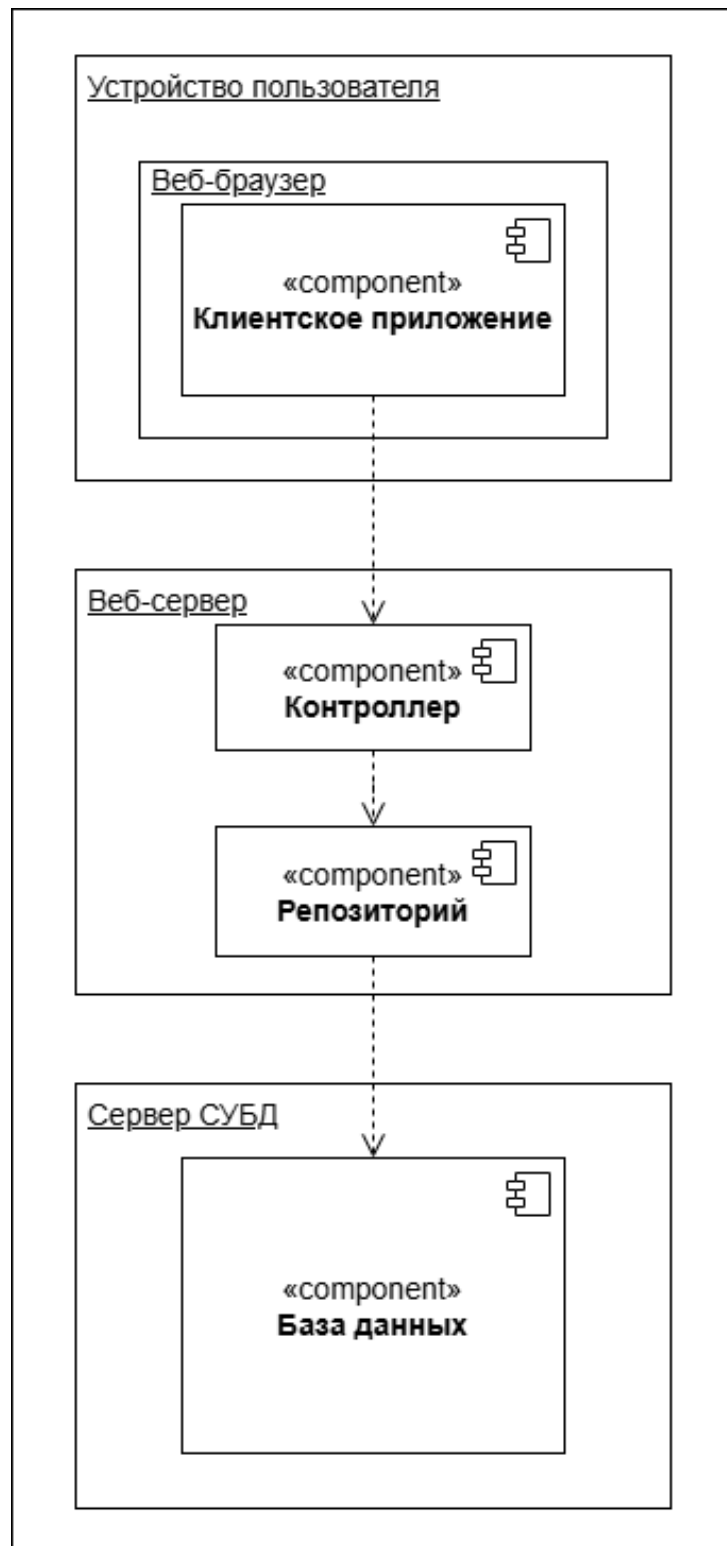


Рисунок 1 – Архитектура системы

Ниже описаны уровни трехслойной архитектуры.

1. Уровень данных – это место, где хранятся и поддерживаются данные для обеспечения их целостности. В разрабатываемой системе используется реляционная база данных.

2. Уровень обработки – место, где данные обрабатываются для выполнения необходимых операций. Этот уровень представлен серверной частью системы. На диаграмме размещения показано взаимодействие двух компонентов, а именно репозитория и контроллера.

3. Уровень пользовательского интерфейса – это клиентское приложение, которое отвечает за передачу пользовательских задач и результатов в понятном для пользователя виде, а также отправляет запросы на сервер с целью получения и обновления необходимой информации, на клиентскую часть отправляется набор данных, где он обрабатывается и выводится в удобном для пользователя виде.

### 2.3. Диаграмма вариантов использования

На рисунке 2 представлена диаграмма вариантов использования разрабатываемого приложения.



Рисунок 2 – Диаграмма вариантов использования

Пользователь является авторизованным пользователем веб-сервиса для описания состояния кристаллизатора и секции МНЛЗ-5 ККЦ. Пользователь может выполнять следующие функции:

1) просмотреть информацию о кристаллизаторах – пользователь может посмотреть список кристаллизаторов, использующихся на производстве в составе комплексов МНЛЗ-5 ККЦ;

2) просмотреть информацию о секциях – пользователь может посмотреть список секций, использующихся на производстве в составе комплексов МНЛЗ-5 ККЦ;

3) посмотреть данные о состоянии кристаллизатора – пользователь может узнать данные о состоянии выбранного кристаллизатора МНЛЗ-5 ККЦ в выбранную дату;

4) посмотреть данные о состоянии секции – пользователь может узнать данные о состоянии выбранной секции МНЛЗ-5 ККЦ в выбранную дату;

5) изменить данные о состоянии кристаллизатора – пользователь может изменить данные о состоянии выбранного кристаллизатора МНЛЗ-5 ККЦ в выбранную дату;

6) изменить данные о состоянии секции – пользователь может изменить данные о состоянии выбранной секции МНЛЗ-5 ККЦ в выбранную дату.

### **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. Была приведена диаграмма размещения.

.

### 3. РЕАЛИЗАЦИЯ

#### 3.1. Хранилище данных

Для хранения информации была реализована база данных. База данных состоит из шестнадцати таблиц, представленных ниже.

1. `Crystallizer` – хранит информацию о кристаллизаторе, его номере, дату начала ремонта, мастере, при котором начался этот ремонт, мастере, при котором закончился, дате завершения ремонта.

2. `CheckingAfterPouring` – хранит информацию о герметичности кристаллизатора после разливки.

3. `DefectiveRollers` – хранит информацию о состоянии нижнего фланца.

4. `SleeveDimensions` – хранит информацию снятых размерах гильзы во время проверки.

5. `CheckingAfterRepair` – хранит информацию о герметичности кристаллизатора после ремонта.

6. `LowerFlangeRevision` – хранит информацию нижнем фланце, о его состоянии после ревизии.

7. `LowerFlangeSizes` – хранит информацию о размерах, нижнего фланца.

8. `InstallingCollectors` – хранит информацию о готовности коллекторов.

9. `Section` – таблица хранит информацию о секции, ее номере, дату начала ремонта, мастере, при котором начался этот ремонт, мастере, при котором закончился, дате завершения ремонта.

10. `CheckingAfterPouring` – хранит информацию о состоянии роликов секции после работы.

11. `RepairRollersSection` – хранит информацию о состоянии роликов секции после ремонта/ревизии и бригадах, проводивших ремонт их.

12. `RepairMasters` – хранит информацию о мастерах бригад, проводивших ремонт/ревизию роликов секции.

13. RollerCoolingSystem – хранит информацию о системе охлаждения после ремонта.

14. RollerLubricationSystem – хранит информацию о системе смазки после ремонта.

15. GaspValues – хранит информацию о величинах зазоров между роликами секции и шаблоном.

16. Notes – хранит информацию об отметках к таблицам дефектовки, ревизии/ремонта секций.

На рисунках 3 и 4 изображена схема базы данных для описания состояния секции и кристаллизатора.

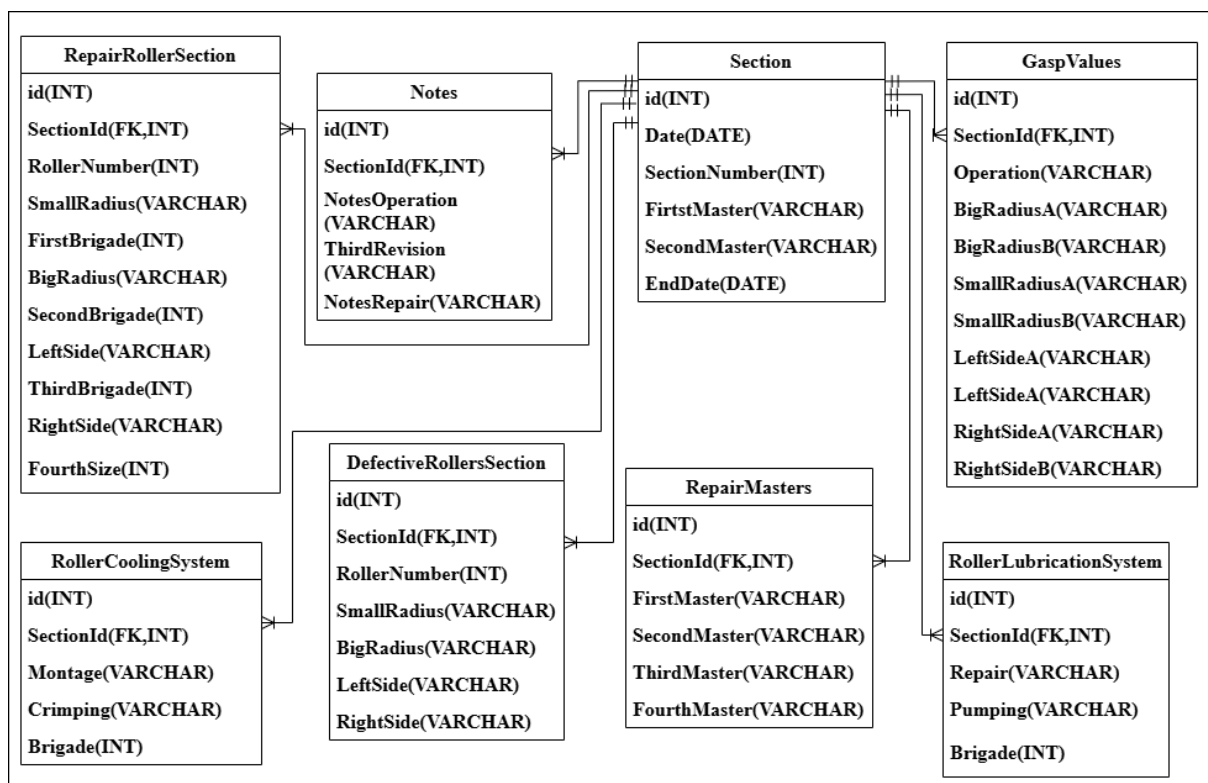


Рисунок 3 – Схема базы данных для описания состояния секции

Связь между таблицей Section и следующими таблицами один ко многим:

- CheckingAfterPouring;
- RepairRollersSection;
- RepairMasters;



- RollerCoolingSystem;
- RollerLubricationSystem;
- GaspValues;
- Notes.

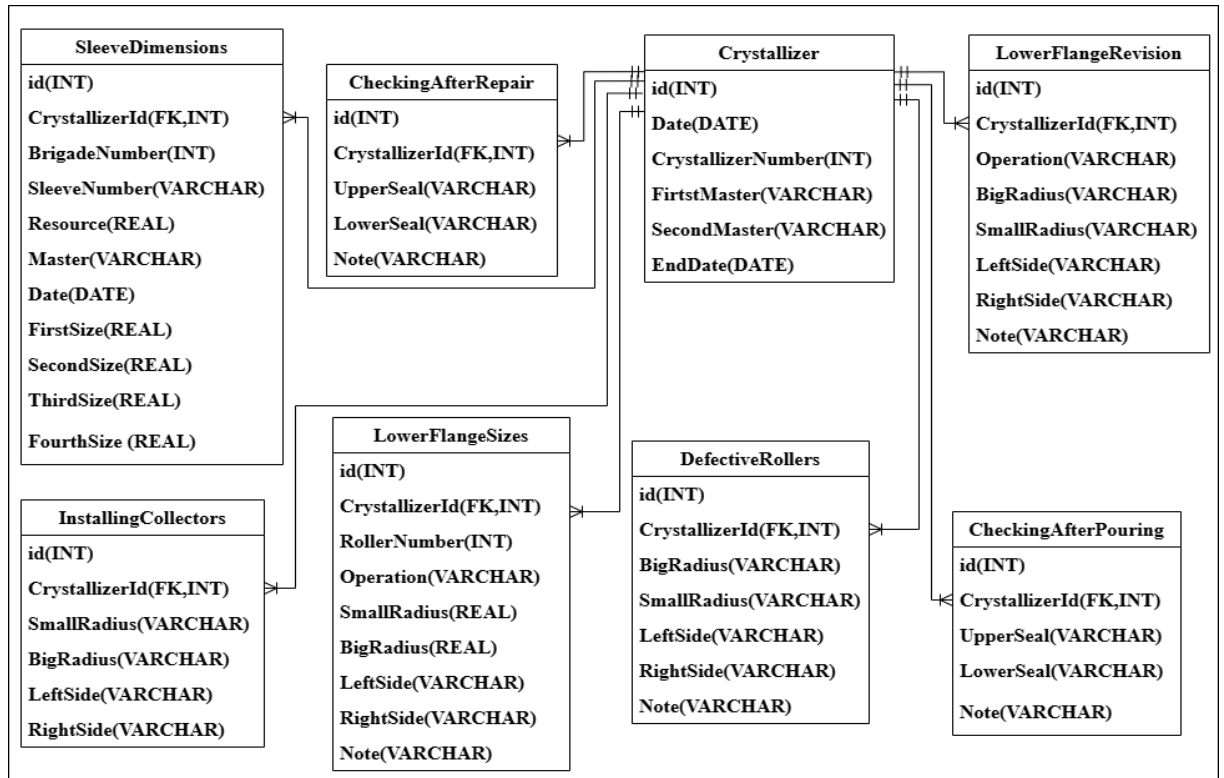


Рисунок 4 – Схема базы данных для описания состояния кристаллизатора

Связь между таблицей Crystallizer и следующими таблицами один ко  
МНОГИМ:

- CheckingAfterPouring;
- DefectiveRollers;
- SleeveDimensions;
- CheckingAfterRepair;
- LowerFlangeRevision;
- LowerFlangeSizes;
- InstallingCollectors.

## 3.2. Реализация серверной части

Серверная часть реализована на фреймворке ASP.NET Core. В программе имеется 3 компонента: Модели (Models), Репозитории (DdRepositories) и Контроллеры (Controllers).

DbRepositories существуют для получения, обновления и обработки данных в БД. Получение и изменение данных происходит при помощи SQL-запросов. В листинге 1 показана функция получения данных из базы данных на примере основной информации о секции.

### Листинг 1 – Получение данных из базы данных

```
public async Task<IEnumerable<SectionInfo>> GetSectionInfoAsync(string
date)
{
    string stmt = @"SELECT *
                    FROM SectionInfo
                    WHERE Date= @date";
    using (IDbConnection db = new SqlConnection(conString))
    {
        IEnumerable<SectionInfo> sectionInfo = await db.QueryAsync<Section-
Info>(stmt, new { date });
        if (sectionInfo.Count() == 0 && date == DateTime.To-
day.ToString("yyyy-MM-dd"))
        {
            int rows = await CreateEmptySectionInfoAsync();
            if (rows > 0)
            {
                sectionInfo = await db.QueryAsync<SectionInfo>(stmt, new {
date });
            }
        }
        return sectionInfo;
    }
}

async Task<int> CreateEmptySectionInfoAsync()
{
    string date = DateTime.Today.ToString("yyyy-MM-dd").Substring(0, 10);
    int rows = 0;
    string stmt = @"INSERT INTO SectionInfo (
                    Date,
                    SectionNumber,
                    FirstMaster,
                    SecondMaster,
                    EndDate)
                    Values (@date, @sectionNumber, '', '', @date)";
    using (IDbConnection db = new SqlConnection(conString))
    {
        db.Open();
        using (IDbTransaction tran = db.BeginTransaction())
        {
            try
            {
                {
                    int sectionNumber = 0;
                    for (int i = 1; i <= 12; i++)
                    {
                        sectionNumber = i;
                    }
                }
            }
        }
    }
}
```

```

        Console.WriteLine("Value of sectionNumber: " + section-
Number);
        rows += await db.ExecuteAsync(stmt, new { date, sec-
tionNumber }, tran);
    }
    tran.Commit();
}
catch
{
    tran.Rollback();
    db.Close();
}
}
return rows;
}

```

Models существуют для представления данных в серверной части проекта. В листинге 2 показано, что для представления данных используется интерфейс `IEnumerable`, модели – компоненты, которые используются в качестве элементов этого интерфейса. Пример описания модели приведен на листинге 2.

### Листинг 2 – Описание модели

```

public class SectionInfo
{
    public int Id { get; set; }
    public string Date { get; set; }
    public int SectionNumber { get; set; }
    public string FirstMaster { get; set; }
    public string SecondMaster { get; set; }
    public string EndDate { get; set; }
}

```

Controllers существуют для связи клиентской и серверной частей проекта. При помощи HTTP-запросов эти функции вызываются клиентской частью, обратно в клиентскую часть отправляется набор данных. Пример такой функции приведен в листинге 3. Здесь на примере функций взаимодействия с основной информацией о секции показано использование стандартных HTTP методов. Метод GET используется для получения данных, метод PUT для их обновления.

### Листинг 3 – Описание контроллера

```

[HttpGet("[action]")]
public async Task<IActionResult> GetSectionInfo(string date)
{
    try
    {

```

```

        IEnumerable<SectionInfo> content = await repo.GetSectionInfoAsync(date);
        return Ok(content);
    }
    catch (Exception ex) { return BadRequest(ex.Message); }
}
[HttpPUT("[action]")]
public async Task<IActionResult> UpdateSectionInfo(SectionInfo[] sectionInfo)
{
    try
    {
        return Ok(await repo.UpdateSectionInfoAsync(sectionInfo));
    }
    catch (Exception ex) { return BadRequest(ex.Message); }
}

```

### 3.3. Реализация клиентской части

При разработке клиентской части был использован компонентный подход. Для страницы существует компонент, который содержит HTML-шаблон страницы [12], файл стилей, а также файл typescript, отвечающий за логику компонента. Данные в клиентской части хранятся в файлах «types», пример кода такого файла показан в листинге 4.

#### Листинг 4 – Описание данных в клиентской части

```

export type SectionInfo = {
    id: number
    date: string
    sectionNumber: number
    firstMaster: string
    secondMaster: string
    endDate: string
}

```

Для выбора даты отчета о состоянии секции или кристаллизатора на странице веб-сервиса реализован интерактивный календарь, при открытии веб-сервиса, автоматически открывается отчет о состоянии секции или кристаллизатора на текущую дату, кнопками «влево» и «вправо» можно переместиться на предыдущую и следующую дату, также можно развернуть календарь и выбрать любую дату отчета. Как выглядит интерактивный календарь для выбора даты отчета показано на рисунке 5.

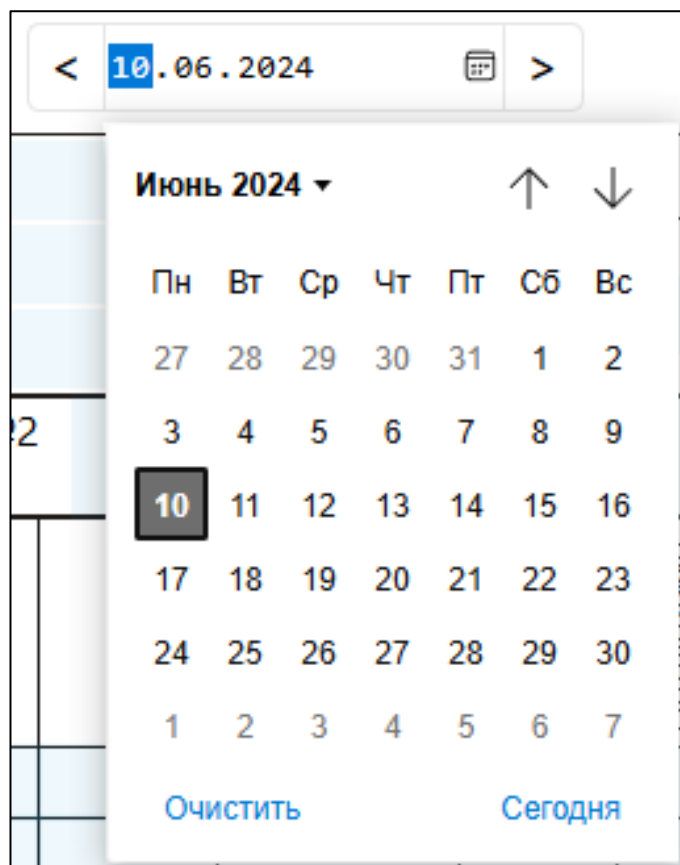


Рисунок 5 – Календарь для выбора даты отчета

Когда пользователь открывает отчет о состоянии секции или кристаллизатора в определенную дату, клиентское приложение запрашивает с сервера всю информацию о состоянии секции или кристаллизатора в данную дату. Пример данных о дефектовке роликов для первой секции 10 июня показан в листинге 5.

Листинг 5 – Данные о состоянии секций в формате JSON

```
{
  "id": 49,
  "date": "2024-06-10",
  "sectionNumber": 1,
  "number": 1,
  "smallRadius": "подклинивает",
  "bigRadius": "подклинивает",
  "rightSight": "вращается",
  "leftSight": "вращается"
},
{
  "id": 50,
  "date": "2024-06-10",
  "sectionNumber": 1,
  "number": 2,
  "smallRadius": "вращается",
  "bigRadius": "вращается",
```

```

    "rightSight": "вращается",
    "leftSight": "вращается"
  },
  {
    "id": 51,
    "date": "2024-06-10",
    "sectionNumber": 1,
    "number": 3,
    "smallRadius": "зациклен",
    "bigRadius": "подклинивает",
    "rightSight": "вращается",
    "leftSight": "подклинивает"
  },
  {
    "id": 52,
    "date": "2024-06-10",
    "sectionNumber": 1,
    "number": 4,
    "smallRadius": "вращается",
    "bigRadius": "вращается",
    "rightSight": "зациклен",
    "leftSight": "вращается"
  }
}

```

Для пользователя данные выводятся в формате таблиц, пример одной из таких таблиц, а именно таблицы «Дефектовка роликов подвижной секции», изображен на рисунке 6.

№ Ролика	Малый радиус	Большой радиус	Правая сторона	Левая сторона
1	вращается	вращается	вращается	вращается
2	вращается	вращается	вращается	вращается
3	вращается	вращается	вращается	вращается
4	вращается	вращается	вращается	вращается
5	вращается	вращается	вращается	вращается
6	вращается	вращается	вращается	вращается
7	вращается	вращается	вращается	вращается
8	вращается	вращается	вращается	вращается

Рисунок 6 – Дефектовка роликов подвижной секции

После загрузки таблиц у пользователя появляется возможность перейти в режим редактирования. Кнопки «Отменить» и «Сохранить» стано-

вятся доступными для нажатия. После нажатия кнопки «Отменить» все изменения отменяются. После нажатия кнопки «Сохранить», новые данные о состоянии секции или кристаллизатора отправляются на сервер и сохраняются в базе данных. Элементы управления отчетом изображены на рисунке 7.



Рисунок 7 – Элементы управления отчетом

После перехода в режим редактирования таблицы преобразуются. Некоторые ячейки таблиц представляются в виде выпадающих списков, иные ячейки, где находятся поля ввода, подсвечиваются специальным цветом. Для удобства пользователя, при наведении курсором, ячейка, доступная для редактирования получает дополнительное обрамление. Примеры таблиц в режиме редактирования показаны на рисунках 8 и 9.

№ Ролика	Малый радиус	Большой радиус	Правая сторона	Левая сторона
1	вращается ▾	вращается ▾	вращается ▾	вращается ▾
2	вращается ▾	вращается ▾	вращается ▾	вращается ▾
3	вращается ▾	вращается ▾	вращается ▾	вращается ▾
4	вращается ▾	вращается ▾	вращается ▾	вращается ▾
5	вращается ▾	вращается ▾	вращается ▾	вращается ▾
6	вращается ▾	вращается ▾	вращается ▾	вращается ▾
7	вращается ▾	вращается ▾	вращается ▾	вращается ▾
8	вращается ▾	вращается ▾	вращается ▾	вращается ▾

Рисунок 8 – Таблица «Дефектовка роликов секции»

в режиме редактирования данных

№ Ролика	Малый радиус, А	Малый радиус, Б	Большой радиус, А	Большой радиус, Б	Правая сторона, А	Правая сторона, Б	Левая сторона, А	Левая сторона, Б
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Рисунок 9 – Таблица «Величины зазоров между роликами секции и шаблоном» в режиме редактирования данных

### 3.4. Интеграция клиентской и серверной частей

Взаимодействие клиентской и серверной части осуществляется путем отправления HTTP-запросов с клиента на сервер. Пример функции получения данных показан в листинге 6.

#### Листинг 6 – Функция получения данных

```

async GetSectionInfoAsync(date: string): Promise<SectionInfo[]> {
    const resp = await fetch(`api/Convshop/GetSectionInfo?date=${date}`)
    if (resp.status >= 400)
        throw new Error(`[ConvShopDbHandler (GetSectionInfoAsync)]: ${await
resp.text()}`)
    const sectionInfoList = await resp.json() as SectionInfo[]
    return sectionInfoList
}

```

Пример функции обновления данных показан в листинге 7.

#### Листинг 7 – Функция обновления данных

```

async UpdateSectionInfoAsync(sectionInfo: SectionInfo[]): Promise<number> {
    var stringifiedList = JSON.stringify(sectionInfo)
    console.log(`[Stringified SectionInfo]: ${stringifiedList}`)
    var resp = await fetch(`api/Convshop/UpdateSectionInfo`, {
        method: "PUT",
        headers: {
            "Content-Type": "application/json"
        },
        body: stringifiedList
    })
    if (resp.status >= 400)
        throw new Error(`[ConvShopDbHandler (UpdateSectionInfoAsync)]:
${await resp.text()}`)
    const data = await resp.json() as number
    return data
}

```



Таким образом было реализовано тридцать два метода для взаимодействия с серверной частью веб-сервиса [13]. В таблице 1 представлены API маршруты для описания состояния секции.

Таблица 1 – API маршруты для описания состояния секции

№	Метод	URL	Описание
1.	GET	/GetSectionInfo	Получение основной информации о секции за выбранную дату
2.	GET	/GetSectionDefect	Получение информации о дефектовке роликов секции
3.	GET	/GetSectionRepair	Получение информации о ремонте и ревизии роликов секции
4.	GET	/GetSectionCooling	Получение информации о готовности системы охлаждения роликов
5.	GET	/GetSectionLubricant	Получение информации о готовности системы смазки роликов
6.	GET	/GetSectionMasters	Получение информации о мастерах бригад, проводивших ремонт и ревизию
7.	GET	/GetSectionGaps	Получение информации о величине зазоров между роликами секции и шаблоном
8.	GET	/GetSectionNotes	Получение информации о дополнительных комментариях по ревизии, ремонту и дефектовке роликов
9.	PUT	/UpdateSectionInfo	Обновление основной информации о секции за выбранную дату
10.	PUT	/UpdateSectionDefect	Обновление информации о дефектовке роликов секции
11.	PUT	/UpdateSectionRepair	Обновление информации о ремонте и ревизии роликов секции
12.	PUT	/UpdateSectionCooling	Обновление информации о готовности системы охлаждения роликов
13.	PUT	/UpdateSectionLubricant	Обновление информации о готовности системы смазки роликов
14.	PUT	/UpdateSectionMasters	Обновление информации о мастерах бригад, проводивших ремонт и ревизию
15.	PUT	/UpdateSectionGaps	Обновление информации о величине зазоров между роликами секции и шаблоном
16.	PUT	/UpdateSectionNotes	Обновление информации о дополнительных комментариях по ревизии, ремонту и дефектовке роликов

В таблице 2 представлены API маршруты для описания состояния кристаллизатора.

Таблица 2 – API маршруты для описания состояния кристаллизатора

№	Метод	URL	Описание
1.	GET	/GetCrystallizerInfo	Получение основной информации о кристаллизаторе за выбранную дату
2.	GET	/GetCrystallizerTightness	Получение информации о герметичности кристаллизатора после разливки
3.	GET	/GetCrystallizerDefect	Получение информации о дефектовке роликов нижнего фланца кристаллизатора
4.	GET	/GetCrystallizerSleeveDimensions	Получение информации о снятых размерах гильзы кристаллизатора
5.	GET	/GetCrystallizerTightnessAfterRepair	Получение информации о герметичности кристаллизатора после ремонта и ревизии
6.	GET	/GetCrystallizerFlangeRevision	Получение информации о ревизии нижнего фланца кристаллизатора
7.	GET	/GetCrystallizerFlangeAdjustment	Получение информации о настройке нижнего фланца кристаллизатора
8.	GET	/GetCrystallizerCollector	Получение информации об установке коллекторов кристаллизатора
9.	PUT	/UpdateCrystallizerInfo	Обновление основной информации о кристаллизаторе за выбранную дату
10.	PUT	/UpdateCrystallizerTightness	Обновление информации о герметичности кристаллизатора после разливки
11.	PUT	/UpdateCrystallizerDefect	Обновление информации о дефектовке роликов нижнего фланца кристаллизатора
12.	PUT	/UpdateCrystallizerSleeveDimensions	Обновление информации о снятых размерах гильзы кристаллизатора
13.	PUT	/UpdateCrystallizerTightnessAfterRepair	Обновление информации о герметичности кристаллизатора после ремонта и ревизии
14.	PUT	/UpdateCrystallizerFlangeRevision	Обновление информации о ревизии нижнего фланца кристаллизатора
15.	PUT	/UpdateCrystallizerFlangeAdjustment	Обновление информации о настройке нижнего фланца кристаллизатора
16.	PUT	/UpdateCrystallizerCollector	Обновление информации об установке коллекторов кристаллизатора

### Вывод по третьей главе

В данной главе был описан способ реализации серверной и клиентской части веб-сервиса, описана база данных, а также интеграция этих частей.

## 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 3).

Таблица 3 – Функциональное тестирование

№	Название теста	Действие	Ожидаемый результат	Результат теста
1	Создание нового отчета о состоянии кристаллизатора или секции	Пользователь переходит на текущую дату, отчет создается автоматически	Происходит формирование и отображение нового отчета	Пройден
2	Просмотр отчета о состоянии кристаллизатора или секции	Пользователь переходит на необходимую ему дату	Отображается отчет о состоянии кристаллизатора на момент выбранной даты	Пройден
3	Редактирование отчета	Пользователь нажимает на кнопку «Редактировать»	Поля ввода становятся доступными для редактирования	Пройден
4	Сохранение отчета	Пользователь после работы с отчетом, нажимает на кнопку «Сохранить»	Поля ввода становятся недоступными для редактирования	Пройден
5	Введение данных неподходящего типа	Пользователь, пытается ввести данные неподходящего типа в выбранную ячейку	Система ограничивает пользователя, не дает ему возможность ввести данные неподходящего типа	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 для кристаллизатора или секции;
- 2) редактировать записи во всех таблицах;

- 3) выбрать отчет за другую дату, отредактировать его;
- 4) внести отметки и примечания в соответствующие поля.

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 функциональное и юзабилити тестирование системы, в результате которого все тесты были выполнены успешно, продемонстрировав, что разработанное приложение выполняет все поставленные функции и является удобным для пользователя. Также описание тестирования 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 описания состояния кристаллизатора и подвижно-роликовой секции 300x360 МНЛЗ-5 ККЦ для ПАО «ЧМК». При выполнении работы были выполнены задачи, представленные ниже.

1. Проведен анализ предметной области.
2. Спроектирован веб-сервис.
3. Реализован веб-сервис.
4. Протестирован веб-сервис.

В ходе анализа предметной области была определена специфика работы предприятия. Также были выявлены необходимые средства для реализации системы.

На этапе проектирования были определены основные функциональные требования, варианты использования системы. Спроектирована схема базы данных и была определена архитектура системы.

Для реализации веб-сервиса были использованы технологии ASP.NET, Node.js, React.

Произведено опытное внедрение разработанного веб-сервиса на предприятии, результатом которого стал акт, подписанный 14.06.2024 г. директором департамента АСУТП Шевченко А.А.

Разработанный веб-сервис имеет базовый функционал. В качестве развития можно добавить визуализацию кристаллизаторов и секций для наглядности.

## ЛИТЕРАТУРА

1. Документация по Java. [Электронный ресурс] URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript/> (дата обращения: 26.02.2024 г.).
2. Документация по TypeScript. [Электронный ресурс] URL: <https://www.typescriptlang.org/docs/> (дата обращения: 26.02.2024 г.).
3. Документация по ASP.NET. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-6.0> (дата обращения: 26.02.2024 г.).
4. Node.js. [Электронный ресурс] URL: <https://nodejs.org/en/> (дата обращения: 26.02.2024 г.).
5. Документация по ASP.NET. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-6.0> (дата обращения: 26.02.2024 г.).
6. Документация ASP.NET Core MVC. [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/aspnet/core/mvc/overview?view=aspnetcore-7.0> (дата обращения: 26.02.2024 г.).
7. Документация по React. [Электронный ресурс] URL: <https://ru.react.js.org/docs/getting-started.html/> (дата обращения: 26.02.2024 г.).
8. Бахтинов В. Б. Технология прокатного производства. // Москва: Металлургия, 1983. – 232 с.
9. 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.
10. Фримен Эрик, Фримен Элизабет. Изучаем HTML, XHTML и CSS. 2-е изд. – СПб.: Питер, 2014 г. – С. 36–40.
11. Мечел. [Электронный ресурс] URL: <https://mchel.ru/sector/steel/chelyabinskiy-metallurgicheskiy-kombinat/> (дата обращения: 04.02.2024 г.).

12. Препроцессор SCSS. [Электронный ресурс] URL: <https://sasscss.ru> (дата обращения: 15.03.2024 г.).
13. Документация по Postman. [Электронный ресурс] URL: <https://learning.postman.com/docs/introduction/overview/> (дата обращения: 25.03.2024 г.).
14. Старолетов С.М. Основы тестирования и верификации программного обеспечения: учебное пособие – 2-е изд., стер. // Санкт-Петербург: Лань, 2020. – 344 с.
15. Карпович Е.Е. Методы тестирования и отладки программного обеспечения: учебник. // Москва: МИСИС, 2020. – 136 с.

## **ПРИЛОЖЕНИЕ. Техническое задание**

### 1. Общие сведения

#### 1.1. Полное наименование системы

Веб-сервис для описания состояния кристаллизатора и подвижно-роликовой секции 300x360 МНЛЗ-5 ККЦ.

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

Разработчик: Горб Олег Владимирович, студент группы КЭ-404, факультет 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: 31 мая 2024.

### 2. Назначение и цели создания системы

#### 2.1. Назначение системы

Веб-сервис для описания состояния кристаллизатора и подвижно-роликовой секции 300x360 МНЛЗ-5 ККЦ предназначен для организации процесса ремонта комплексов МНЛЗ-5. Для достижения этой цели, мастер фиксирует состояния различных частей кристаллизатора или секции после эксплуатации, во время дефектовки, после ревизии или ремонта, а также сравнивает размеры определенных компонентов с шаблоном.

#### 2.2. Цели создания системы

Создание системы должно обеспечить достижение следующих целей:

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

### 3. Требования к системе



### 3.1. Функциональные требования

- 1) мастер должен иметь возможность выбора даты, выбора секции и кристаллизатора;
- 2) мастер должен иметь возможность просматривать информацию о состоянии выбранного кристаллизатора в определенный день;
- 3) мастер должен иметь возможность просматривать информацию о состоянии выбранной секции в определенный день;
- 4) мастер должен иметь возможность редактировать информацию о состоянии выбранного кристаллизатора в определенный день;
- 5) мастер должен иметь возможность редактировать информацию о состоянии выбранной секции в определенный день;
- 6) мастер должен иметь возможность оставлять примечания, которые несут в себе дополнительную важную информацию о состоянии отдельных компонентов кристаллизатора или секции, не отраженную в таблицах;
- 7) система должна оснащаться функцией валидации, чтобы обеспечить проверку введенных пользователем данных на соответствие заданным критериям;

### 3.2. Нефункциональные требования

- 1) серверная часть должна быть реализована на платформе ASP.NET;
- 2) клиентская часть должна быть реализована с использованием языка TypeScript и технологии React;
- 3) в качестве СУБД для реляционной базы данных необходимо использовать Microsoft SQL.

### 3.3. Требования к таблицам, схемам и примечаниям

#### 3.3.1. Таблица «Проверка на герметичность кристаллизатора после разливки»

Таблица содержит следующий набор полей:

- 1) верхнее уплотнение – информация о герметичности верхнего уплотнения кристаллизатора (герметично/течь);

2) нижнее уплотнение – информация о герметичности нижнего уплотнения кристаллизатора (герметично/течь).

Пример таблицы приведен на рисунке 1.

<b>Верхнее уплотнение</b>	<b>Нижнее уплотнение</b>
<b>герметично</b>	<b>герметично</b>

Рисунок 1 – Пример таблицы «Проверка на герметичность кристаллизатора после разливки»

### 3.3.2. Таблица «Дефектовка роликов нижнего фланца»

Таблица содержит следующие поля:

1) большой радиус – информация о состоянии большого радиуса (ролика нижнего фланца, находящегося сверху, вращается/подклинивает/зациклен);

2) малый радиус – информация о состоянии малого радиуса (ролика нижнего фланца, находящегося снизу, вращается/подклинивает/зациклен);

3) левая сторона – состояние левой стороны (ролика нижнего фланца, находящегося слева, вращается/подклинивает/зациклен);

4) правая сторона – состояние правой стороны (ролика нижнего фланца, находящегося справа, вращается/подклинивает/зациклен).

Пример таблицы приведен на рисунке 2.

<b>операция</b>	<b>малый радиус</b>	<b>большой радиус</b>	<b>правая сторона</b>	<b>левая сторона</b>
<b>ревизия</b>	<b>готов</b>	<b>готов</b>	<b>готов</b>	<b>готов</b>

Рисунок 2 – Пример таблицы «Дефектовка роликов нижнего фланца»

### 3.2.3. Таблица «Снятие размеров гильзы»

Таблица содержит следующие поля:

1) номер бригады – порядковый номер бригады, проводившей замеры;

2) номер гильзы – порядковый номер гильзы;

3) мастер – мастер, проводивший замеры;

4) дата – дата проведения замеров.

Пример таблицы приведен на рисунке 3.

<b>Бригада №</b>	1
<b>Гильза №</b>	68001
<b>Выработанный ресурс, м</b>	7454
<b>Замеры провел</b>	Чариков В.Е.
<b>Дата</b>	26.01.2024

Рисунок 3 – Пример таблицы «Снятие размеров гильзы»

3.2.4. Таблица «Проверка на герметичность кристаллизатора после ремонта»

Таблица содержит следующий набор полей:

- 1) верхнее уплотнение – информация о герметичности верхнего уплотнения (герметично/течь);
- 2) нижнее уплотнение – информация о герметичности нижнего уплотнения (герметично/течь).

Пример таблицы приведен на рисунке 4.

<b>Верхнее уплотнение</b>	<b>Нижнее уплотнение</b>
герметично	герметично

Рисунок 4 – Пример таблицы «Проверка на герметичность кристаллизатора после ремонта»

3.2.5. Таблица «Ревизия роликов нижнего фланца»

Таблица содержит следующие поля:

- 1) операция – операция, совершаемая над роликами нижнего фланца (ревизия/ремонт);
- 2) большой радиус – информация о состоянии большого радиуса (ролика нижнего фланца, находящегося сверху, готов/не готов);

3) малый радиус – информация о состоянии малого радиуса (ролика нижнего фланца, находящегося снизу, готов/не готов);

4) левая сторона – состояние левой стороны (ролика нижнего фланца, находящегося слева, готов/не готов);

5) правая сторона – состояние правой стороны (ролика нижнего фланца, находящегося справа, готов/не готов).

Пример таблицы приведен на рисунке 5.

<b>операция</b>	<b>малый радиус</b>	<b>большой радиус</b>	<b>правая сторона</b>	<b>левая сторона</b>
<b>ревизия</b>	<b>готов</b>	<b>готов</b>	<b>готов</b>	<b>готов</b>

Рисунок 5 – Пример таблицы «Ревизия роликов нижнего фланца»

### 3.2.6. Таблица «Зазор роликов нижнего фланца»

Таблица содержит следующие поля:

1) операция – операция, совершаемая над роликами нижнего фланца (настройка);

2) большой радиус – информация о размере зазора большого радиуса (наибольший зазор между роликом и шаблоном, 0,1 – допустимый зазор, 0,2 – предельный зазор, >0,2 – недопустимый зазор);

3) малый радиус – информация о состоянии малого радиуса (наибольший зазор между роликом и шаблоном, 0,1 – допустимый зазор, 0,2 – предельный зазор, >0,2 – недопустимый зазор);

4) левая сторона – состояние левой стороны (наибольший зазор между роликом и шаблоном, 0,1 – допустимый зазор, 0,2 – предельный зазор, >0,2 – недопустимый зазор);

5) правая сторона – состояние правой стороны (наибольший зазор между роликом и шаблоном, 0,1 – допустимый зазор, 0,2 – предельный зазор, >0,2 – недопустимый зазор).

Пример таблицы приведен на рисунке 6.

<b>операция</b>	<b>малый радиус</b>	<b>большой радиус</b>	<b>правая сторона</b>	<b>левая сторона</b>
<b>настройка</b>	<b>0,1</b>	<b>0,2</b>	<b>0,1</b>	<b>0,1</b>

Рисунок 6 – Пример таблицы «Зазор роликов нижнего фланца»

### 3.2.7. Таблица «Установка коллекторов»

Таблица содержит следующие поля:

- 1) большой радиус – информация о коллекторе большого радиуса (установлен/не установлен);
- 2) малый радиус – информация о коллекторе малого радиуса (установлен/не установлен);
- 3) левая сторона – информация о коллекторе левой стороны (установлен/не установлен);
- 4) правая сторона – информация о коллекторе правой стороны (установлен/не установлен).

Пример таблицы приведен на рисунке 7.

<b>малый радиус</b>	<b>большой радиус</b>	<b>правая сторона</b>	<b>левая сторона</b>
<b>установлен</b>	<b>установлен</b>	<b>установлен</b>	<b>установлен</b>

Рисунок 7 – Пример таблицы «Установка коллекторов»

### 3.2.8. Таблица «Дефектовка Подвижно-Роликовой секции»

Таблица содержит следующие поля:

- 1) номер ролика – порядковый номер ролика (1–8);
- 2) большой радиус – состояние большого радиуса (вращается/подклинивает/зациклен);
- 3) малый радиус – состояние малого радиуса (вращается/подклинивает/зациклен);
- 4) правая сторона – состояние правой стороны (вращается/подклинивает/зациклен);

5) левая сторона – состояние левой стороны (вращается/подклинивает/зациклен);

Пример таблицы приведен на рисунке 8.

№ ролика	Малый радиус	Большой радиус	Правая сторона	Левая сторона
1	вращается	вращается	вращается	вращается
2	вращается	вращается	вращается	вращается
3	вращается	вращается	вращается	вращается
4	вращается	вращается	вращается	вращается
5	вращается	вращается	вращается	вращается
6	вращается	вращается	вращается	вращается
7	вращается	вращается	заклинен	вращается
8	вращается	вращается	вращается	вращается

Рисунок 8 – Пример таблицы «Дефектовка Подвижно-Роликовой секции»

### 3.2.9. Таблица «Ремонт и ревизия Подвижно-Роликовой секции»

Таблица содержит следующие поля:

- 1) номер ролика – порядковый номер ролика (1–8);
- 2) большой радиус состояние – состояние большого радиуса (готов/не готов);
- 3) большой радиус номер бригады – номер бригады, проводившей ревизию/ремонт большого радиуса (1–4);
- 4) малый радиус состояние – состояние малого радиуса (готов/не готов);
- 5) малый радиус номер бригады – номер бригады, проводившей ревизию/ремонт малого радиуса (1–4);
- 6) правая сторона состояние – состояние правой стороны (готов/не готов);
- 7) правая сторона номер бригады – номер бригады, проводившей ревизию/ремонт правой стороны (1–4);
- 8) левая сторона состояние – состояние левой стороны (готов/не готов);
- 9) левая сторона номер бригады – номер бригады, проводившей ревизию/ремонт левой стороны (1–4).

Пример таблицы приведен на рисунке 9.

№ ролика	Малый радиус		Большой радиус		Правая сторона		Левая сторона	
	состояние	№ бр.	состояние	№ бр.	состояние	№ бр.	состояние	№ бр.
1	готов	4	готов	4	готов	4	готов	4
2	готов	4	готов	4	готов	4	готов	4
3	готов	4	готов	4	готов	4	готов	4
4	готов	4	готов	4	готов	4	готов	4
5	готов	4	готов	4	готов	4	готов	4
6	готов	4	готов	4	готов	4	готов	4
7	готов	4	готов	4	готов	4	готов	4
8	готов	4	готов	4	готов	4	готов	4

Рисунок 9 – Пример таблицы «Ремонт и ревизия Подвижно-Роликовой секции»

### 3.2.10. Таблица «Мастера, проводившие ревизию/ремонт»

Таблица содержит следующие поля:

- 1) сменный мастер бригада №1 – Ф.И.О. сменного мастера первой бригады;
- 2) сменный мастер бригада №2 – Ф.И.О. сменного мастера второй бригады;
- 3) сменный мастер бригада №3 – Ф.И.О. сменного мастера третьей бригады;
- 4) сменный мастер бригада №4 – Ф.И.О. сменного мастера четвертой бригады.

Пример таблицы приведен на рисунке 10.

Отв. исполнитель:	Ф.И.О.
Сменный мастер бригада №1	
Сменный мастер бригада №2	
Сменный мастер бригада №3	
Сменный мастер бригада №4	<b>Дьяченко М.В.</b>

Рисунок 10 – Пример таблицы «Мастера, проводившие ревизию/ремонт»

### 3.2.11. Таблица «Готовность системы охлаждения роликов»

Таблица содержит следующие поля:

- 1) монтаж РВД – информация о монтаже РВД (ремонт/ревизия);

- 2) опрессовка – информация об опрессовке (испытано/ревизия);
- 3) номер бригады – номер бригады, проводившей проверку готовности системы охлаждения.

Пример таблицы приведен на рисунке 11.

Монтаж РВД	Опрессовка	№ бригады
ремонт	испытано	4

Рисунок 11 – Пример таблицы «Готовность системы охлаждения роликов»

### 3.2.12. Таблица «Готовность системы смазки роликов»

Таблица содержит следующие поля:

- 1) ремонт – информация о ремонте системы смазки;
- 2) прокачка – информация о прокачке (испытано/ревизия);
- 3) номер бригады – номер бригады, проводившей проверку готовности системы смазки.

Пример таблицы приведен на рисунке 12.

Ремонт	Прокачка	№ бригады
ремонт	испытано	1

Рисунок 12 – Пример таблицы «Готовность системы смазки роликов»

### 3.2.13. Таблица «Зазоры между роликами секции и шаблоном»

Таблица содержит следующие поля:

- 1) номер ролика – порядковый номер ролика (1-8);
- 2) большой радиус А – состояние большого радиуса (наибольший зазор между роликом и шаблоном с левой стороны, 0,1 – допустимый зазор, 0,15 – предельный зазор, >0,15 – недопустимый зазор);
- 3) большой радиус В – состояние большого радиуса (наибольший зазор между роликом и шаблоном с правой стороны, 0,1 – допустимый зазор, 0,15 – предельный зазор, >0,15 – недопустимый зазор);



4) малый радиус А – состояние малого радиуса (наибольший зазор между роликом и шаблоном с левой стороны, 0,1 – допустимый зазор, 0,15 – предельный зазор, >0,15 – недопустимый зазор);

5) малый радиус В – состояние малого радиуса (наибольший зазор между роликом и шаблоном с правой стороны, 0,1 – допустимый зазор, 0,15 – предельный зазор, >0,15 – недопустимый зазор);

6) правая сторона А – состояние правой стороны (наибольший зазор между роликом и шаблоном с левой стороны правого ряда роликов, 0,1 – допустимый зазор, 0,15 – предельный зазор, >0,15 – недопустимый зазор);

7) правая сторона В – состояние правой стороны (наибольший зазор между роликом и шаблоном с правой стороны правого ряда роликов, 0,1 – допустимый зазор, 0,15 – предельный зазор, >0,15 – недопустимый зазор);

8) левая сторона А – состояние левой стороны (наибольший зазор между роликом и шаблоном с левой стороны левого ряда роликов, 0,1 – допустимый зазор, 0,15 – предельный зазор, >0,15 – недопустимый зазор);

9) левая сторона В – состояние левой стороны (наибольший зазор между роликом и шаблоном с правой стороны левого ряда роликов, 0,1 – допустимый зазор, 0,15 – предельный зазор, >0,15 – недопустимый зазор).

Пример таблицы приведен на рисунке 13.

№ ролика	Малый радиус		Большой радиус		Правая сторона		Левая сторона	
	А	В	А	В	А	В	А	В
1	0,1	0,1	0,2	0,1	0,1	0,1	0,1	0,1
2	0,2	0,2	0,1	0,1	0,1	0,1	0,1	0,1
3	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
4	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
5	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
6	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
7	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
8	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1

Рисунок 13 – Пример таблицы «Зазоры между роликами секции и шаблоном»

### 3.2.14. Примечания к таблицам

Страницы содержат следующие примечания к таблицам.

1. Примечание к таблице «Проверка на герметичность кристаллизатора после разливки» – дополнительная информация, почему кристаллизатор не герметичен. Пример примечания приведен на рисунке 14.

<b>Примечание:</b> <b>По какой причине не герметично</b>

Рисунок 14 – Пример примечания к таблице «Проверка на герметичность кристаллизатора после разливки»

2. Примечание к таблице «Дефектовка роликов нижнего фланца» – дополнительная информация о состоянии роликов нижнего фланца. Пример примечания приведен на рисунке 15.

<b>Примечание:</b> <b>Особые отметки о состоянии нижнего фланца</b>

Рисунок 15 – Пример примечания к таблице «Дефектовка роликов нижнего фланца»

3. Примечание к таблице «Проверка на герметичность кристаллизатора после ремонта» – дополнительная информация, почему кристаллизатор не герметичен. Пример примечания приведен на рисунке 16.

<b>Примечание:</b> <b>По какой причине не герметично</b>

Рисунок 16 – Пример примечания к таблице «Проверка на герметичность кристаллизатора после ремонта»

4. Примечание к таблице «Ревизия роликов нижнего фланца» – дополнительная информация о состоянии роликов нижнего фланца. Пример примечания приведен на рисунке 17.

<b>Примечание:</b>
<b>Особые отметки по ревизии нижнего фланца</b>

Рисунок 17 – Пример примечания к таблице «Ревизия роликов нижнего фланца»

5. Примечание к таблице «Дефектовка роликов подвижной секции» – дополнительная информация о состоянии подвижной секции после эксплуатации. Пример примечания приведен на рисунке 18.

<b>Особые отметки о состоянии подвижной секции после эксплуатации:</b>

Рисунок 18 – Пример примечания к таблице «Дефектовка роликов подвижной секции»

6. Примечание к таблице «Ремонт и ревизия Подвижно-Роликовой секции» – дополнительная информация о состоянии подвижной секции после ревизии и ремонта. Пример примечания приведен на рисунке 19.

<b>Особые отметки по ревизии и ремонту роликов секции:</b>

Рисунок 19 – Пример примечания к таблице «Ремонт и ревизия Подвижно-Роликовой секции»