**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)**

Колледж инновационных технологий и предпринимательства

**кафедра физики и прикладной математики**

**ОТЧЕТ**

**ПО производственной ПРАКТИКЕ**

**ПО МОДУЛЮ ПМ.03 «Участие в интеграции программных модулей»**

Студента Кузьминовой Виктории Викторовны

Специальность 09.02.03 Программирование в компьютерных системах

Группа ПКсп-120

Время прохождения практики с «21» марта 2024 г. по «17» апреля 2024 г.

Руководитель от ВлГУ

преподаватель КИТП \_\_\_\_\_\_\_\_\_\_\_\_\_М.Ю. Кабанова

Руководитель от предприятия:

начальник участка по автоматизации

\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Е. Филатов

(Андрей Евгеньевич тел.: 89103827553)

Владимир, 2024

|  |  |
| --- | --- |
|  | Утверждаю  Зав. кафедрой ФиПМ\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
|
|

**ЗАДАНИЕ**

на производственную практику по модулю ПМ.03 «Участие в интеграции программных модулей»

студента \_\_*Кузьминовой Виктории Викторовны*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

**4** курса, направления 09.02.03 Программирование в компьютерных системах группы **ПКсп-120**

Предприятие АО «Выксунский металлургический завод», г. Выкса, Нижегородская область

Последовательность прохождение практики получить задание, пройти инструктаж по технике безопасности и охране труда, выполнить все задания, оформить отчет согласно требованиям.

За время прохождения практики необходимо:

1. Изучить вопросы, предусмотренные программой по всем разделам.

2. Изучить и исследовать: **Автоматическое формирование шихтовки плавки для дуговой сталеплавильной печи**

*2.1. На языке UML построить необходимые диаграммы для выбранного приложения.*

*2.2. Развернуть клиентскую часть системы контроля версий SmartGit.*

*2.3. Поместить в репозиторий исходные файлы разработанного приложения.*

*2.4. Изучить принципы использования компилятора вне интегрированной среды разработки.*

*2.5. Разработать тест-кейсы, на их основании протестировать приложение.*

3. Получить навыки

*3.1. оформления отчета по результатам практики согласно требованиям к оформлению документации.*

4. Задание по охране труда, технике безопасности и охране окружающей среды:

*4.1. изучить правила техники безопасности при работе за компьютером, вопросы охраны труда на месте прохождения практики.*

Отчет по практике составить к \_\_17.04.2024\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание выдал:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Ю. Кабанова

(Ф.И.О, руководителя практики от университета)

Задание получил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Кузьминова

(подпись студента, дата)

**Примечание:** задание должно быть приложено к отчету по практике (вторым листом после титульного листа)

СОДЕРЖАНИЕ

[Введение 4](#_Toc164009337)

[1 Разработка приложения 5](#_Toc164009338)

[1.1 Основные определения 5](#_Toc164009339)

[1.2 Предприятие 6](#_Toc164009340)

[1.3 Описание предметной области 6](#_Toc164009341)

[2 Проектирование приложения 13](#_Toc164009342)

[2.1 Проектирование процессов функционирования 13](#_Toc164009343)

[2.1.1 Диаграмма вариантов использования 13](#_Toc164009344)

[2.1.2 Диаграмма активностей приложения 15](#_Toc164009345)

[3 Разработка приложения 22](#_Toc164009346)

[3.1 Инструментальные средства разработки приложения 22](#_Toc164009347)

[3.2 Разработка интерфейса 26](#_Toc164009348)

[4 Система контроля версий 34](#_Toc164009349)

[4.1 Описание системы контроля версий 34](#_Toc164009350)

[4.2 Работа с репозиторием 34](#_Toc164009351)

[5 Тестирование 37](#_Toc164009352)

[5.1 Тест-требования 37](#_Toc164009353)

[5.2 Тест-план 38](#_Toc164009354)

[5.3 Результаты тестирования 43](#_Toc164009355)

[Заключение 52](#_Toc164009356)

[Список использованных источников 53](#_Toc164009357)

[Приложение А 56](#_Toc164009358)

# ВВЕДЕНИЕ

Целью прохождения производственной практики, которая проходит с 21 марта 2024 года по 17 апреля 2024 года, является получение теоретических и практических знаний, умений и навыков в рамках профессионального модуля ПМ.03 «Участие в интеграции программных модулей».

Для реализации задания выданного на данную практику необходимо выполнить все задачи, охватывающие все разделы программы практики, а именно:

* изучить правила техники безопасности и охраны окружающей среды;
* качественно выполнить задания, выданные руководителем производственной практики от предприятия;
* изучить предметную область программы, производственный процесс и построить UML-диаграммы для приложения;
* выбрать программные средства, с помощью которых будет разрабатываться программный продукт;
* произвести разработку проекта;
* написать тес-план и тест-требования;
* выполнить тестирование программного продукта;
* подготовить отчет по производственной практике и защитить его на кафедре.

# 1 РАЗРАБОТКА ПРИЛОЖЕНИЯ

## 1.1 Основные определения

Бадья ⎯ сосуд, в которой загружается металл для плавления.

Дуговая сталеплавильная печь (ДСП) ⎯ это печь, предназначенная для плавки металлов с использованием теплоты электрической дуги.

Карбюризаторы ⎯ предназначен для науглероживания металла при недостаточном количестве чугуна [1].

Лигатура в металлургии ⎯ специальный сплав, предназначенный для легирования жидкого металла (сплава) летучими, сильно окисляющимися или тугоплавкими компонентами (легирующими добавками) [2].

Насыпная масса материала ⎯ соотношение массы материалов ко всему занимаемому им объемом бадьи.

Окислитель ⎯ предназначены для окисления примесей чугуна и должны содержать большое количество оксидов железа и меньшее количество серы, фосфора, кремния [3].

Технолог ⎯ это специалист, который обеспечивает эффективность работы производства, качества продукции и безопасности производственных процессов.

Флюс ⎯ материал, применяемый в металлургических процессах с целью образования или регулирования состава шлака, предохранения расплавленных металлов от взаимодействия с внешней газовой средой, а также служащий для связывания оксидов при пайке и сварке металлов [4].

Футеровка печи ⎯ процесс укрепления внутренних стен печи или камина с помощью специальных огнеупорных материалов [5].

Шлак ⎯ расплав, покрывающий поверхность жидкого металла или очаговые остатки, образующиеся при сжигании твердого топлива [6].

Шихта ⎯ это комплекс минералов, загружаемых внутрь доменной печи или другого высокотемпературного оборудования, для получения конечных продуктов заданного химического состава и свойств [7].

## 1.2 Предприятие

АО «Выксунский металлургический завод» ⎯ крупнейший отечественный производитель стальных сварных труб и железнодорожных колес. Основан в 1757 году, с 1999 года входит в состав Объединенной металлургической компании (ОМК) [8].

ОМК ввела в эксплуатацию новый современный литейно-прокатный комплекс (ЛПК). ЛПК ⎯ первый полномасштабный металлургический завод, основной задачей которого является импортозамещение [9].

Данное программное средство разрабатывалось для предприятия АО «ВМЗ» ЛПК.

## 1.3 Описание предметной области

Руководство предприятия, в котором проходила производственная практика, предоставило тему разработки веб-приложения, предназначенного для автоматического формирования шихтовки плавки для дуговой сталеплавильной печи.

Перед началом разработки необходимо произвести анализ предметной области, исходя из нее создать базу данных, определиться с входными и выходными параметрами, а также с участниками данного процесса.

Плавка стали осуществляется при высоких температурах и сопровождается сложным физико-химическими процессами взаимодействия расплава, флюсов, шлаков, печных и атмосферных газов, футеровки печи. Поэтому в литейных цехах обычно сталь выплавляют в дуговых сталеплавильных печах.

В ДСП можно выполнять литейные стали различного состава. Преимуществом является применение кислой футеровки, более стойкой, чем основная, особенно при работе с перерывами [10]. Емкость данной печи на предприятии достигает 180 тонн.

Технологический процесс выплавки стали происходит следующим образом. Начинается все с заправки печи и загрузки шихты, затем происходит плавление шихтовки, ее окисление (выброс шлака) и восстановление (доводка до химического состава и выпуск металла) [11].

Весь процесс плавки в ДСП условно делится на два этапа − период плавления, когда в печи есть твердая не расплавившаяся шихта, и окислительный период, во время которого жидкий металл интенсивно продувается газообразным кислородом.

Сырьем для ДСП выступает стальной лом (старая техника, крепежные элементы, оконные решетки и т.д.). Весь материал, который будет задействован в процессе выплавки делится на три специальные корзины(бадьи), с открывающем днищем и ограниченным объемом. В одной заходке могут быть задействованы от двух до трех бадей. Загрузка метала осуществляется в определенной последовательности: сначала легкий, затем средний, потом тяжелый и завещающий слой − легкий. В какой слой попадет той или иной лом зависит его насыпной массы.

В первую корзину загружают максимальный допустимый объем материала, который может поместить в себе данная емкость. Вторая же корзина заполняется остававшимся ломом, либо оставшимся процентом от всей общей вместительности печи. Третья корзина заполняется в том случае, если в двух других свободного места нет, а некоторый объем лома остался, необходимый для выплавки. Именно данный остаток и загружается в третью свободную емкость.

В качестве шихтовых материалов используют различные металлические (лом, чугун, ферросплавы, металлизованное сырье) и неметаллические (флюсы − известь, боксит; окислители − железная руда, кислород; карбюризаторы − электронный бой, отсев кокса) материалы.

Задача расчета шихты ⎯ определение необходимой комбинации исходных материалов и соотношения между ними, при котором обеспечивается получение стали, по химическому составу и механическим свойствам соответствующей требованиям ГОСТ 977-8 «Отливки стальные. Общие технические требования» или специальным техническим условиям.

Шихту составляют таким образом, чтобы содержание углерода в исходной шихте было на 0,3% выше нижнего предела ГОСТа на данную марку стали. Для его повышения вводят чугун, а также электродный бой или кокс.

Каждый материал, который входит в состав любой шихтовки, имеет ряд собственных свойств:

* химический состав (элементы, которые входят в состав и окисление этих элементов);
* насыпная масса (отношение тонны материала на занимаемый объем);
* угар (материал без засора);
* цена;
* засор (мусор, который уходит в шлак).

Химический состав содержит в себе определение количества углерода, кремния, марганца, серы, фосфора, меди, никеля, вольфрама, молибдена, хрома и железа, который относится к определенному виду металла, то есть каждый вид имеет свои коэффициенты содержания данных элементов. Также химический состав содержит в себе значения элементов после окисления, которые уходят в шлак.

Насыпная масса определяется по формуле. Полученное значение сортируется по соотношению слоев, загруженные в бадью, то есть если насыпная масса будет большой, данный металл будет располагаться в тяжелом слое, а если наименьшая, то в легком.

Угар получается в результате выплавки загруженного материала в печь.

Засор ⎯ это мусор, который состоит из окисей химических элементов и другого мусора, например, пластика.

Цена металла определяется из поставки. В ней также указана дата поставки, масса прихода, масса остатка, наименование металла и соответствующий код.

Обычно расчет шихты ведется по одному из двух методов:

* По среднему химическому составу сплава.
* По оптимальному химическому составу, при котором сплав имеет наилучшие технологические и эксплуатационные свойства.

При расчете определяется необходимое количество шихтовых материалов, которое нужно загрузить в печь, чтобы получить заданный вес готового сплава. Также при расчете увеличиваются потери, получающиеся в процессе приготовления сплава, называемые угаром.

Угар складывается из потерь вследствие испарения, в результате взаимодействия жидкого металла с атмосферой печи и образования не растворимых в металле соединений, а также вследствие взаимодействия сплава и футеровки печи. В случае необходимости иметь более точный расчет шихты нужно учитывать угар отдельно по каждому компоненту.

Этапы расчета шихтовки:

* Определение расчетного состава шихты.
* Определение количества лигатура.
* Определения количество чистых металлов.
* Шихтовая карта.

После расчета шихты, которая будет использоваться в процессе расплава, необходимо рассчитать минимизацию времени на выплавку, т.е. сколько киловатт в чат необходимо на одну тонну лома.

После того, как весь лом полностью расплавлен, проводится проверка и корректировка химического состава стали.

Для удобной работы с данной программной на огромном предприятии наиболее подходящий вид разработки является веб-приложение, т.к. его можно открыть в любой части завода, на любом персональном компьютере, перейдя по ссылке в любом доступном для работы браузере.

Известно, что для корректной работы программы необходимы материалы. Для их добавления в базу данных программы выделена отдельная роль «Администратор базы данных», которому поступает документация с поставками лома на предприятие, информацию о котором необходимо включить в базу.

Остальным функционалом программы будет пользоваться «Технолог». Его задачей является грамотно подобрать нужный лом по всем ограничениям, определить соотношение масс слоев каждой корзины, указать, какой материал будет загружен в ту или иную бадью.

Для упрощения данной работы, разрабатывается программное средство, которое будет содержать в себе весь перечень его задач, для более эффективной работы.

Входными параметрами программы являются такие данные, как:

* масса выплавки;
* количество бадей загрузки;
* ограничения по химическому составу;
* соотношение слоев.

В поле масса выплавки необходимо указать общий вес загрузки металла в бадью. Данные указываются без учета угара и засора, то есть общий вес, не превышающий 180 тонн.

Количество бадей определяется из поля масса выплавки. В зависимости от выбранного металла, его насыпной плотности и веса определяется сколько необходимо бадей для загрузки материала. Максимальное количество что доступно это три бадьи, минимумом является две.

В ограничении по химическому составу подразумевается, что у какого-либо метала есть значения химических элементов. Пользователь вводит значение и все материалы фильтруются так, чтобы значение химического элемента в составе не превышало заданного ограничения.

В соотношении слоев задается процентное отношение тяжелого, легкого и среднего на общий объем бадьи.

Количество значений шихтовки зависит от количества бадей, используемых в процессе загрузки материала.

Критерии, необходимые для расчета шихты:

* масса материала;
* доступный материал для работы;
* химический ограничения;
* цена либо минимизация времени на выплавку;
* соотношение слоев;

Минимизация времени на выплавку ⎯ это отношение затраченной электрической и химической энергии на тонну металла.

При соблюдении всех критериев на выходе технолог получает шихтовку на все бадьи. Каждая из них делится по условиям, которые прописываются в критериях, а также происходит вывод химического состава конечного продукта.

Выходными данными является рассчитанная шихтовка, содержащая такие данные, как:

* коллекцию материалов, которые загружаются в бадью;
* итоговый объем загруженной бадьи;
* масса;
* затраченная электрическая энергия.

Кратко описанная предметная область дает возможность понимания того, для кого будет разработана программа, каком образом она будет разработана (веб-приложение или мобильное приложение), а также получили входные и выходные данные, которые необходимо учесть.

# 2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

## 2.1 Проектирование процессов функционирования

### 2.1.1 Диаграмма вариантов использования

Диаграмма вариантов использования ⎯ диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей [12].

Данная диаграмма отображает множество актеров, взаимодействующих с проектируемой системой с помощью вариантов использования. Основными элементами любой диаграммы вариантов использования являются актер и вариант использования. Диаграмма вариантов использования изображена на рисунке 1.

Актер ⎯ это внешняя по отношению к модулируемой системе сущность, взаимодействующая с системой для решения некоторых задач.

В данном случае, актером будет являться «Технолог», который использует данное программное средство для более эффективной работы, а также в качестве актера будет выступать «Администратор базы данных».

Вариант использования определяется некоторый набор действий, которые должны быть выполнены моделируемой системой или программным средством при взаимодействии с актером.

Набор вариантов использования актера «Администратор БД»:

− Авторизация/регистрация пользователя в системе.

− Добавлять/удалять/обновлять информацию о каждом материале, имеющимся в наличие.

− Работа с пользователями системы (удаление и редактирование прав доступа).

Набор вариантов использования для актера «Технолог»:

1. Определение задания (ввода ограничений):
   1. по химическому составу;
   2. по обязательным к загрузке материалов;
   3. по цене;
   4. по электрической энергии;
   5. по массе выплавки.
2. Определение соотношения массы каждого слоя корзины.
3. По ограничениям получить шихтовку для процесса выплавки.

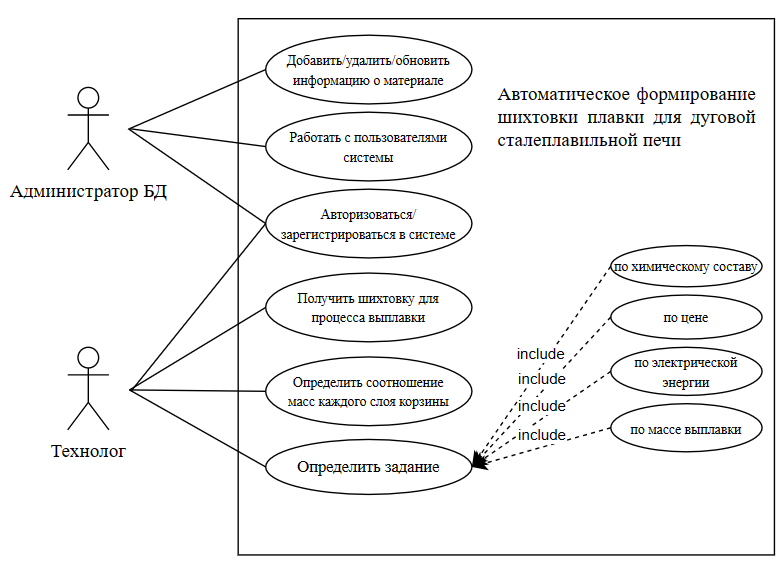


Рисунок 1 − Диаграмма вариантов использования

После разработки диаграммы вариантов использования можно наглядно понять каким основным функционалом должна быть оснащена разрабатываемая программа, а также определить количество ролей, которые будут с ней взаимодействовать.

### 2.1.2 Диаграмма активностей приложения

Диаграмма активности отображает динамические аспекты поведения системы, позволяет более детально визуализировать конкретный случай использования. Это диаграмма, которая иллюстрирует поток деятельности через систему [13].

Для каждой задачи на диаграмме деятельности указывается, кто выполняет задачу, какие ресурсы необходимы для ее выполнения, какие результаты ожидаются на каждом этапе процесса, а также какая последовательность задач и операций необходима для успешного завершения проекта.

Все разработанные диаграммы деятельности представлены на рисунках 2-13. При рассмотрении данных иллюстраций, можно более глубоко изучить тот или иной функционал программы и понять какими входными и выходными параметрами должен обладать пользователь.

Диаграмма деятельности «Регистрация пользователя в системе» изображена на рисунке 2.

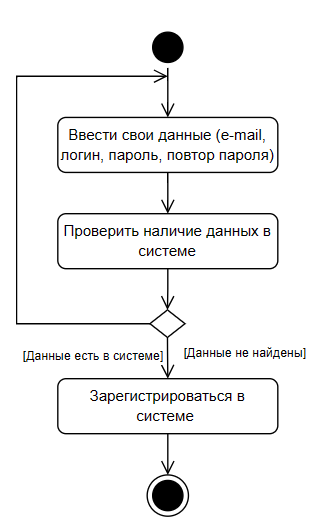


Рисунок 2 − Диаграмма деятельности «Регистрация пользователя в системе»

Диаграмма деятельности «Авторизация пользователя в системе» представлена на рисунке 3.

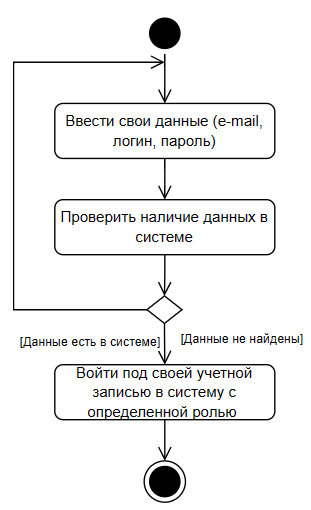


Рисунок 3 − Диаграмма деятельности «Авторизация пользователя в системе»

Диаграмма деятельности «Добавить/изменить информацию о материале» изображена на рисунке 4.

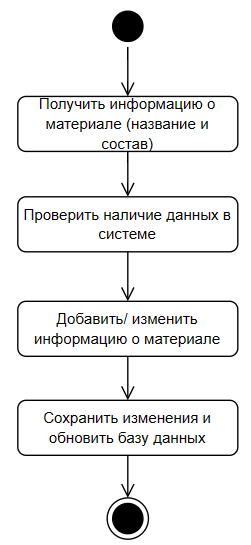


Рисунок 4 − Диаграмма деятельности «Добавить/изменить информацию о материале»

Диаграмма деятельности «Удаление материала» представлена на рисунке 5.

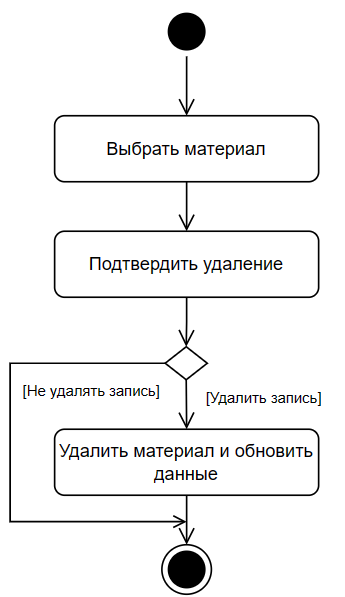


Рисунок 5 − Диаграмма деятельности «Удаление материала»

Диаграмма деятельности «Удаление пользователя системы» изображена на рисунке 6.

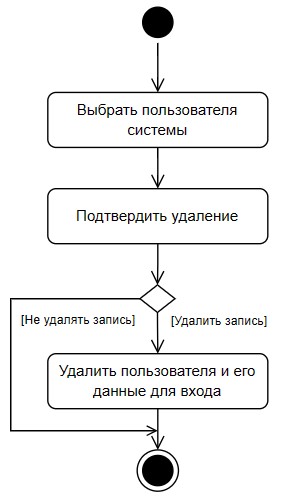


Рисунок 6 − Диаграмма деятельности «Удаление пользователя системы»

Диаграмма деятельности «Изменить роль пользователя системы» представлена на рисунке 7.

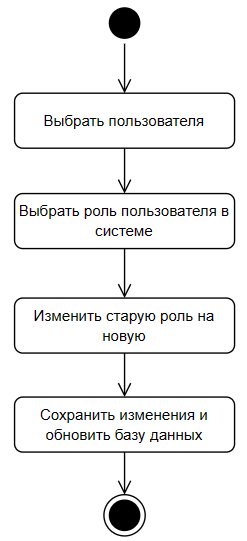


Рисунок 7 − Диаграмма деятельности «Изменить роль пользователя системы»

Диаграмма деятельности «Ограничение химического состава» изображена на рисунке 8.



Рисунок 8 − Диаграмма деятельности «Ограничение химического состава»

Диаграмма деятельности «Ограничение стоимости выплавки» представлена на рисунке 9.

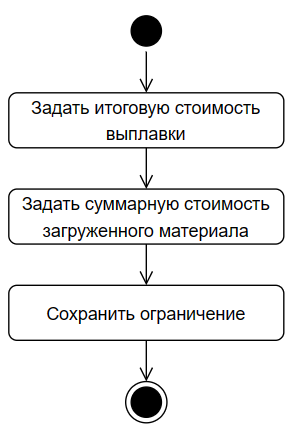


Рисунок 9 − Диаграмма деятельности «Ограничение стоимости выплавки»

Диаграмма деятельности «Ограничение электрической энергии» изображена на рисунке 10.

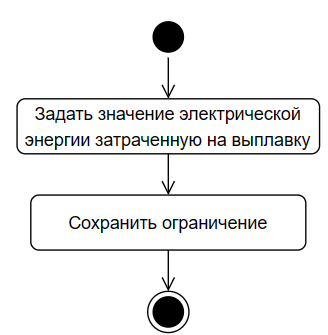


Рисунок 10 − Диаграмма деятельности «Ограничение электрической энергии»

Диаграмма деятельности «Ограничение общей массы выплавки» представлена на рисунке 11.

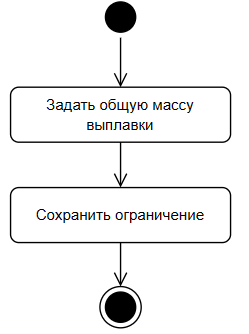


Рисунок 11 − Диаграмма деятельности «Ограничение общей массы выплавки»

Диаграмма деятельности «Соотношение масс слоев бадьи» изображена на рисунке 12.

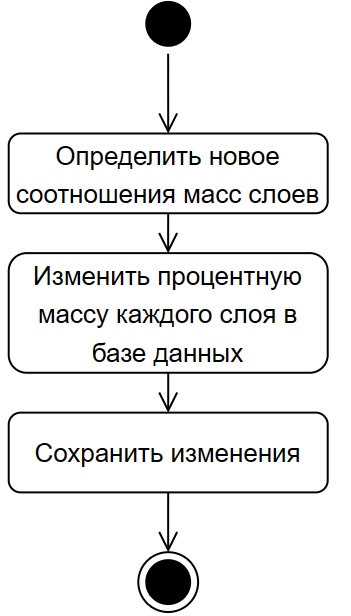


Рисунок 12 − Диаграмма деятельности «Соотношение масс слоев бадьи»

Диаграмма деятельности «Получение шихтовки» представлена на рисунке 13.

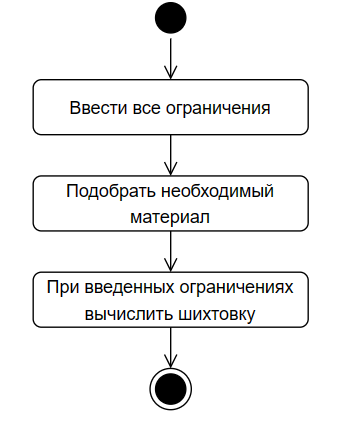


Рисунок 13 − Диаграмма деятельности «Получение шихтовки»

После представления всех вышеописанных диаграмм разработка приложения будет наиболее комфортна, так как весь функционал программы в них описан максимально подробно и изображен графически, с учетом всех действий со стороны пользователя.

# 3 РАЗРАБОТКА ПРИЛОЖЕНИЯ

## 3.1 Инструментальные средства разработки приложения

Для разработки программного средства необходимые определенные инструментальные средства такие, как текстовый редактор, для написания технической записки, площадка для написания кода и сам язык программирования, на котором будет написана программы.

В ходе выполнения данной работы были использованы:

* Microsoft Word [14].
* Draw.io [15].
* SQL Server Management Studio [16].
* Microsoft Visual Studio [17].

Языки программирования:

* C# [18].
* JavaScript [19].
* HTML [20].
* CSS [21].

Microsoft Word (MS Word, WinWord или просто Word) ⎯ текстовый процессор, предназначенный для создания, просмотра, редактирования и форматирования текстов статей, деловых бумаг, а также иных документов, с локальным применением простейших форм таблично-матричных алгоритмов.

Данное программное средство использовалось для написания технической записки, его альтернативой могли бы быть Яндекс.Документы, LibreOffice или SoftMaker FreeOffice.

Выбор был осуществлен в пользу данной программы из-за следующих преимуществ:

* удобная функциональная панель инструментов;
* имеются средства для работы с таблицами и диаграммами;
* высокая степень защиты данных и самого документа;
* широкий спектр распространения.

Draw.io ⎯ это бесплатный онлайн-сервис, который помогает создавать блок- схемы, прототипы, инфографику и диаграммы любого вида. Чаще всего его используют именно для построения диаграмм.

Данное программное средство использовалось для создания блок-схем и различных диаграмма, которые необходимы для графического представления функционала программы, его альтернативой могли бы быть Miro, Visio или Lucidchart.

Выбор был осуществлен в пользу данной программы из-за следующих преимуществ:

* понятный и привычный интерфейс для пользователя;
* не нужна регистрация в системе;
* множественный выбор сохранения файлов, например, на GitHub или на устройстве или Google Диске;
* разнообразные форматы сохранения файлов, например, svg, png, html или xml.

SQL Server Management Studio ⎯ это интегрированная среда для управления инфраструктурой SQL Server и базы данных SQL Azure. Среда Management Studio предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server.

Данное инструментальное средство использовалось для разработки базы данных и ее администрирования, также, через программу можно было заполнять таблицы данными, а также создавать представления, триггеры и так далее. Также были рассмотрены программы-аналоги данного средства такие, как Navicat, Database .NET или DataGrip.

Выбор был осуществлен в пользу SQL Server Management Studio из-за следующих преимуществ:

* удобный и привычный интерфейс программы;
* автоматизация задач по управлению базой данных;
* разнообразный функционал;
* высокая надежность и стабильная работа.

Microsoft Visual Studio ⎯ то стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

Основной функционал программы и внешний вид разрабатывался в рассматриваемом инструментальном средстве. Также были аналоги, среди которых происходил выбор, это такие программы, как Eclipse, NetBeans или Lazarus.

Основными достоинствами Microsoft Visual Studio перед другими являются:

* поддержка языка программирования C#;
* разработка веб-сайтов;
* привычный и интеллектуально понятный интерфейс;
* доступна интеграция с различными технологиями, которые необходимы в разработки программы.

C# ⎯ современный объектно-ориентированный и безопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET. C# относится к широко известному семейству языков C.

Данный язык программирования выбран в качестве языка для написания основного функционала программы. Основными аналогами были такие языки программирования, как Kotlin, Object Pascal или Objective-C.

Основным преимуществом выбранного языка заключается в том, что он легко функционирует с языком HTML, также данный язык является компактным и легко читаемым, имеет огромный функционал, который необходим для разрабатываемого приложения.

JavaScript ⎯ это интерпретируемый язык программирования, который используют для написания frontend- и backend-частей сайтов, а также мобильных приложений.

Данный язык программирования выбран для написания функционала со стороны веб-приложения и объединения функционала, написанного на языке программирования C#. Имеются аналоги данного языка такие, как TypeScript, CoffeeScript или ReasonML.

Основными достоинствами данного языка являются:

* легкого взаимодействует с серверной частью и самой страницей;
* быстра реализация различных задач;
* привычный интерфейс и хорошая производительность.

HTML ⎯ это язык разметки гипертекстовых документов. Он нужен, чтобы отображать в браузере специальным образом отформатированный документ с множеством вложенных элементов: заголовками, абзацами, списками, гиперссылками, медиа-источниками, расположением изображений, видео и аудио.

Этот язык разметки необходим для создания основных компонентов веб-страницы, которые будут на ней отображены. Было произведено сравнение между различными языками разметки, такими, как Markdown или ReAdABLE Human Format.

Основное преимущество данного языка состоит в том, что HTML является стандартом веб-разработки и поддерживается всеми браузерами, обладает широкими возможностями для создания страниц, то есть поддерживает текст, изображения, таблицы, формы, аудио и видео, а также работу с ними.

CSS ⎯ язык описания внешней страницы. Он отвечает за визуальную красоту и оформление сайта.

Данный язык использовался для стилевого оформления сайта, то есть создание фона и таблиц.

Основным преимуществом является то, что данный язык существенно упрощает верстку и снижает временные затраты. Созданный файл разработчиком может использоваться на все страницы разрабатываемого приложения сразу. Для внесения изменений не нужно искать данную строчку во всех файлах, а достаточно просто в одном файле найти нужный блок и внести изменения.

После определения с инструментальными средствами разработки можно с легкостью написать веб-приложение исходя из выбранных языков программирования и площадки для написания кода, чтобы в итоге выпустить готовый продукт.

Перед тем как приступить к разработке веб-приложения, необходимо создать базу данных, которая будет хранить в себе всю информацию, нужную для взаимодействия пользователя с приложением.

## 3.2 Разработка интерфейса

Перед началом разработки программного продукта необходимо продумать интерфейс данного приложения.

Проектирование макетов ⎯ это важный момент разработки ПО, он позволяет заранее определить местоположение кнопок управления, отображения данных из базы данных, а также продумать степень удобства использования программы и так далее.

При первом запуске программы пользователю будет отображена страница его авторизации в систему, представленная на рисунке 14. Листинг данной страницы представлен в приложении А.

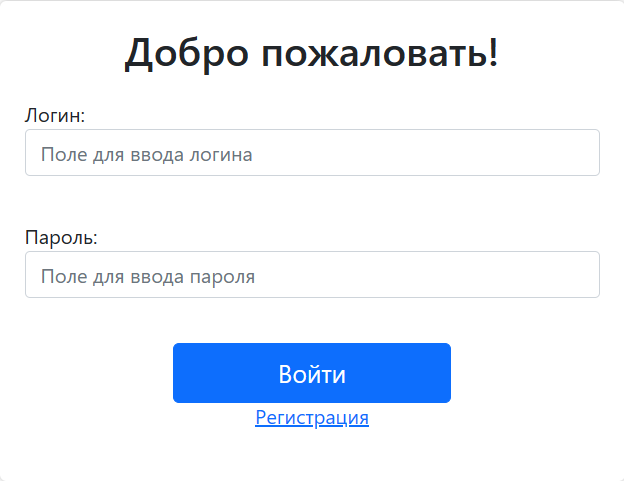


Рисунок 14 − Страница «Авторизация пользователя»

Если пользователь не зарегистрирован в системе, то он может это сделать с помощью перехода на страницу «Регистрация» и ввести свои данные и выполнить регистрацию. Макет данной страницы расположен на рисунке 15.

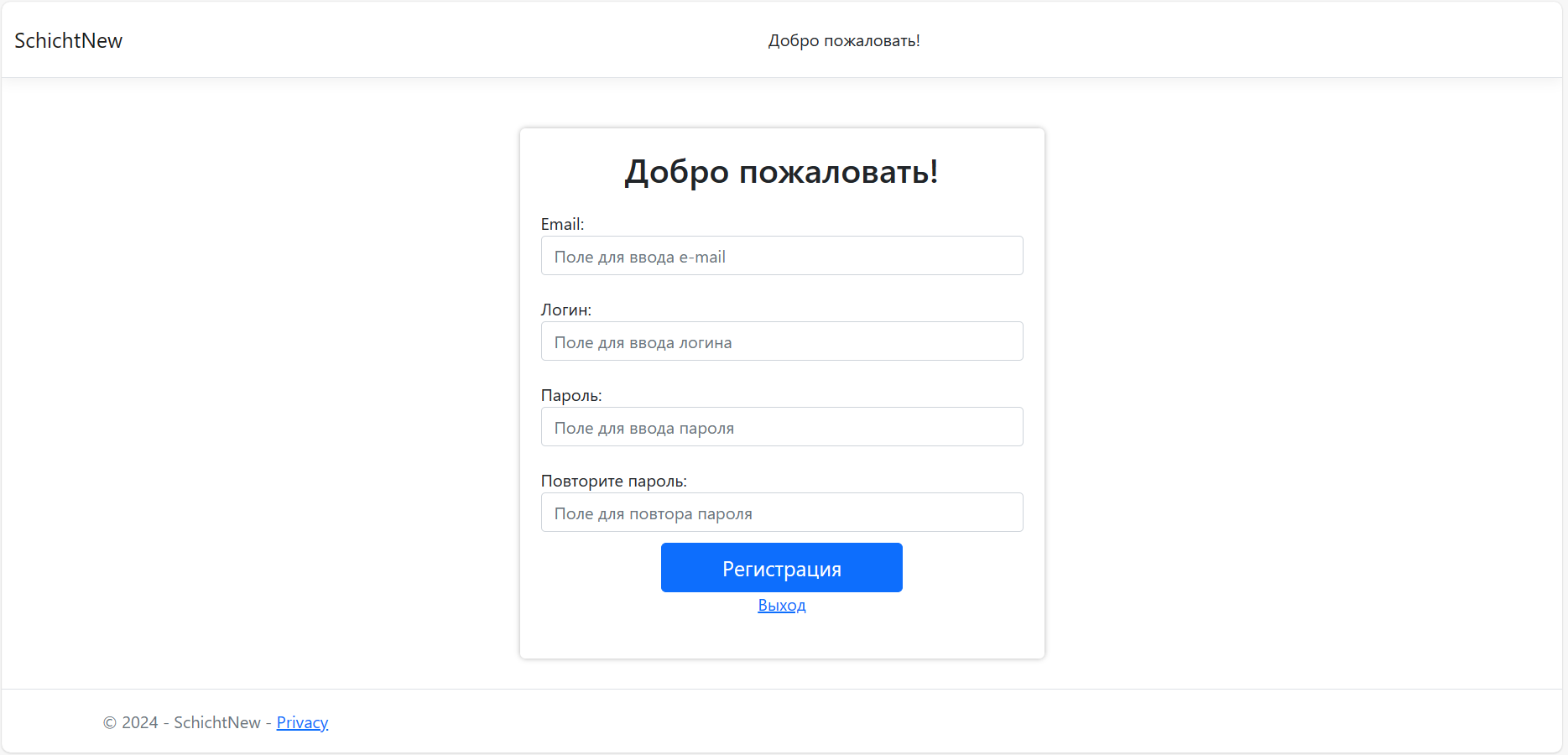


Рисунок 15 − Страница «Регистрация пользователя»

После авторизации пользователь может перейти на одну из двух страниц, в зависимости от роли авторизуемого. Первый вариант, представленный на рисунке 16, авторизация пользователя с ролью «Администратор БД», а на рисунке 17, авторизация пользователя с ролью «Технолог», листинги данных страниц описаны в приложении А.

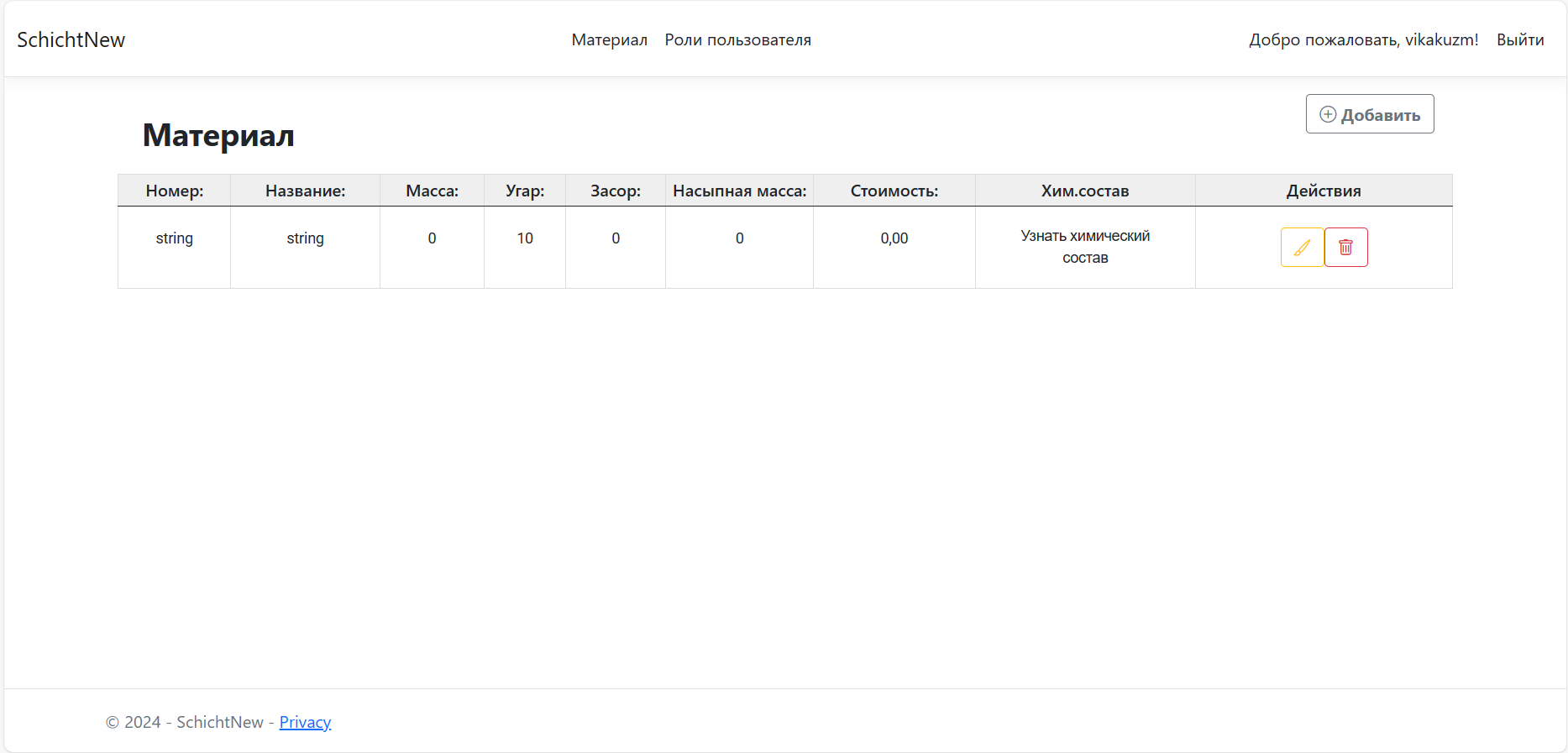


Рисунок 16 − Главная страница роли «Администратор БД»

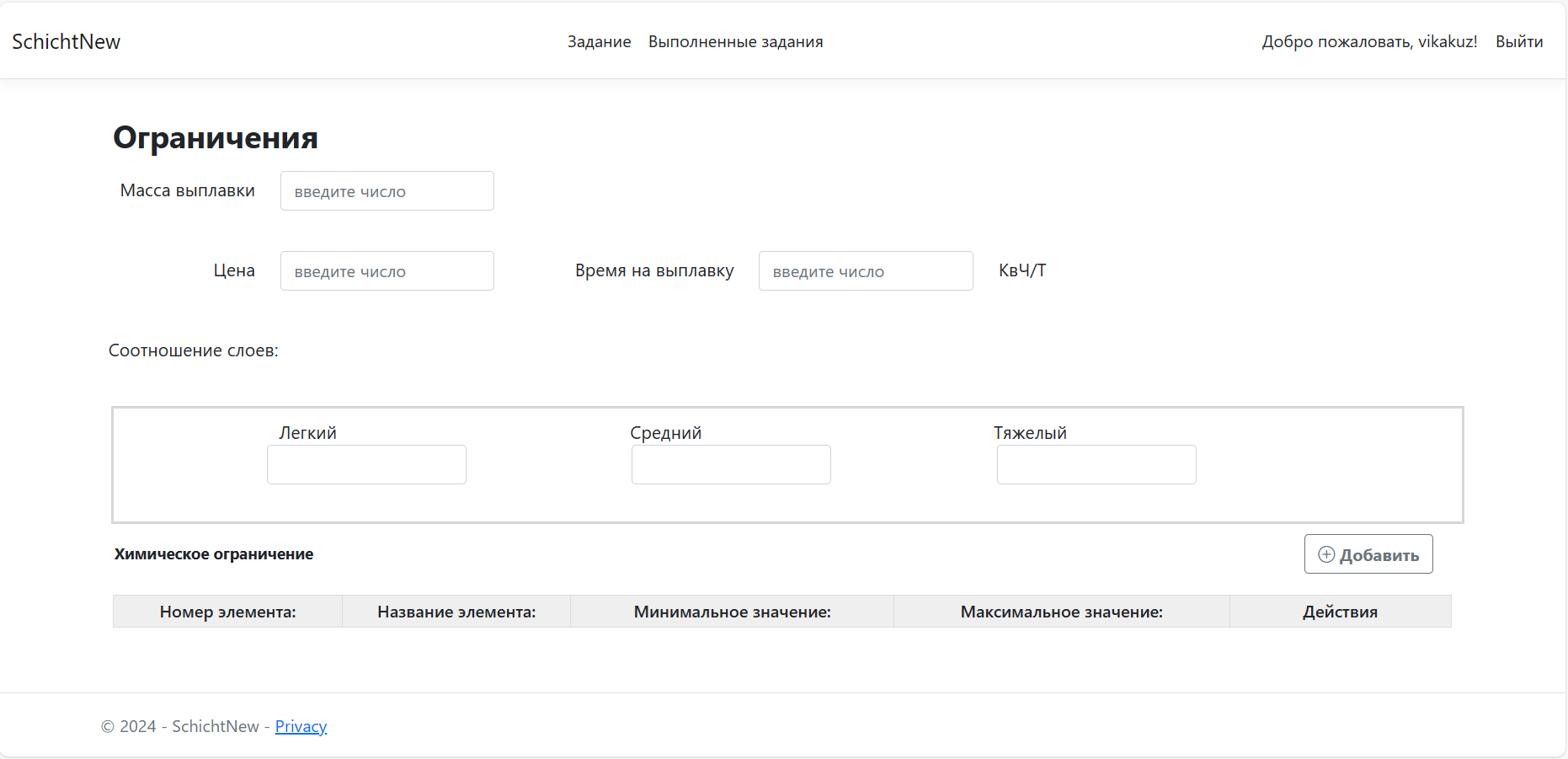


Рисунок 17 − Главная страница роли «Технолог»

Для того, чтобы узнать химический состав каждого материала, необходимо перейти на другую страницу, с помощью ссылки в таблице. Данная ссылка переводит пользователя на страницу с таблицами, в которых указаны шлаковые и металлические фазы каждого материала. Описанная страница представлена на рисунке 18.

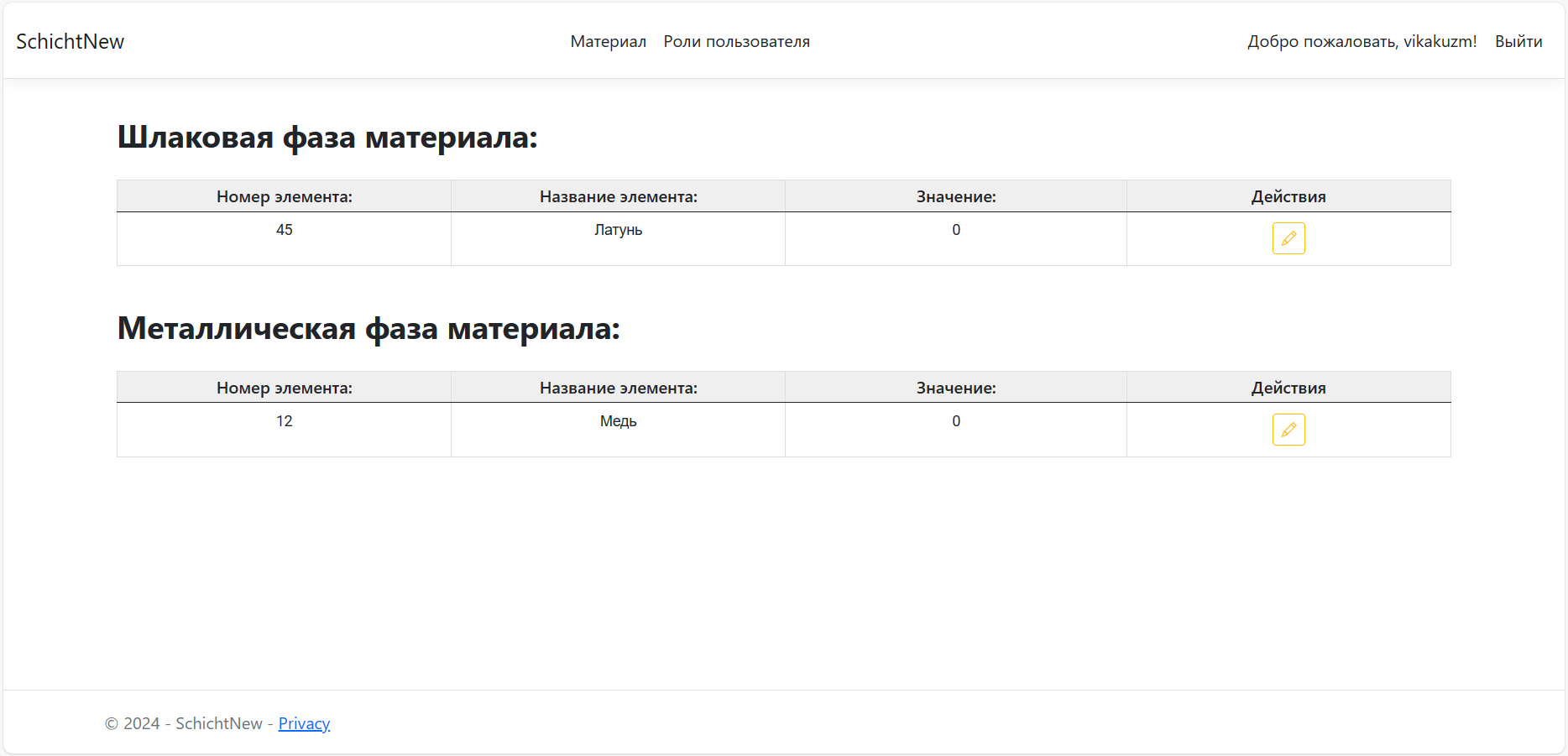


Рисунок 18 − Страница «Химический состав материала»

Для изменения значения химического элемента необходимо перейти на форму, представленную на рисунке 19.

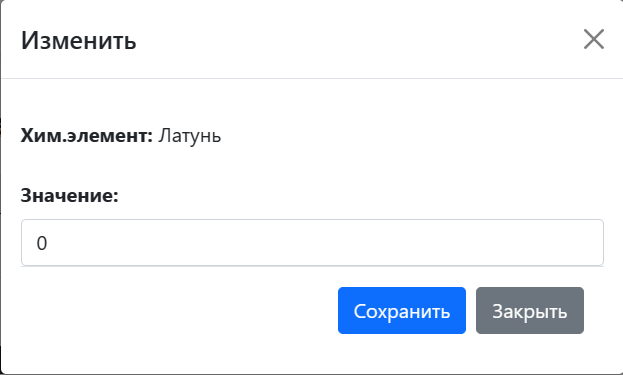


Рисунок 19 − Форма «Изменение химического состава»

В таблицу материалов можно добавить новую запись, перейдя на дополнительную форму, предназначенную для ввода нового материала. Данная форма представлена на рисунке 20.

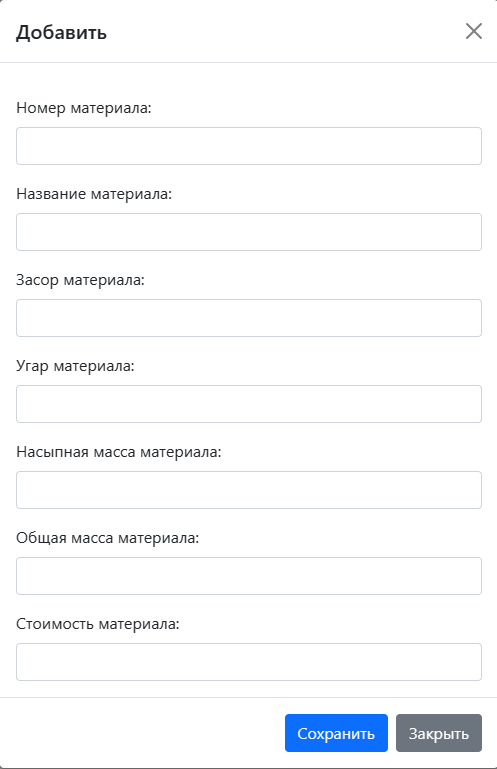


Рисунок 20 − Форма «Добавить новый материал»

Изменение свойств материала происходит в форме, представленной на рисунке 20, только пустые поля заполняются значениями.

Так же администратор может взаимодействовать с пользователями системы, перейдя на страницу, изображенная на рисунке 21.

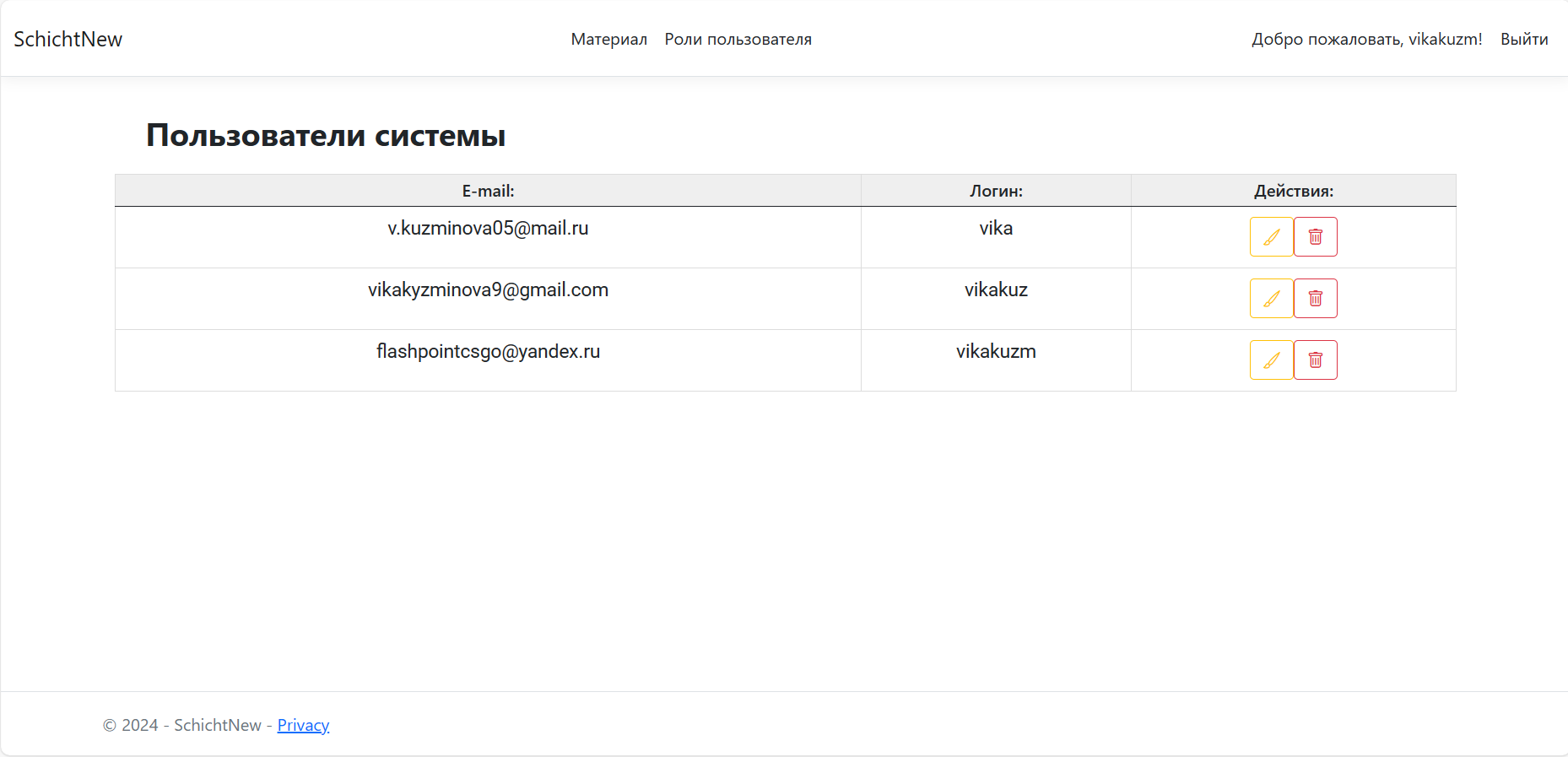


Рисунок 21 − Страница «Перечень пользователей системы»

Основным взаимодействием администратора с пользователями системы является изменение их роли. Чтобы изменить роль пользователя необходимо перейти на форму, представленная на рисунке 22.

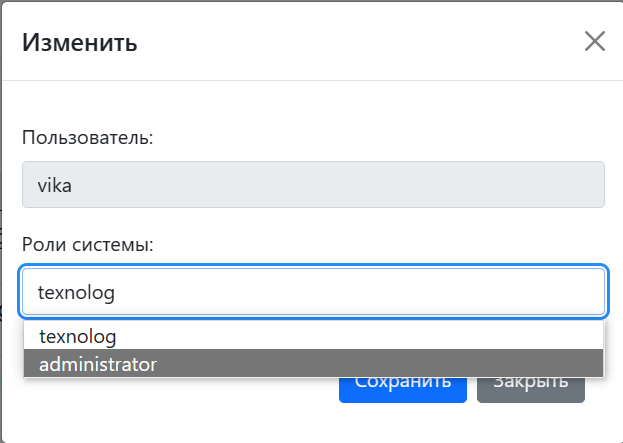


Рисунок 22 − Форма «Изменить роль пользователя в системе»

После занесения всех доступных материалов в базу данных можно задавать ограничительные значения, благодаря котором произойдет расчет шихтовки. Интерфейс данной странице изображен на рисунке 17.

При переходе на форму задание ограничения химического состава открывается форма, в которой для каждого химического элемента можно ввести свое значение. Данная форма представлена на рисунке 23

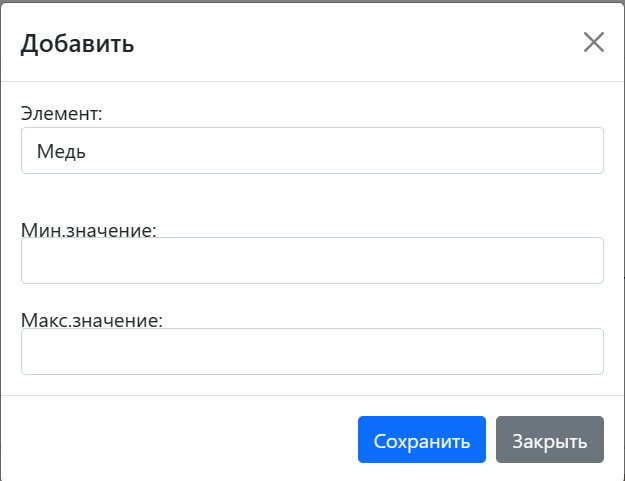


Рисунок 23 − Форма «Добавление химического ограничения»

После ввода всех необходимых ограничений происходит автоматический расчет шихтовки, результат которой будет отображен в форме, представленная на рисунке 24.

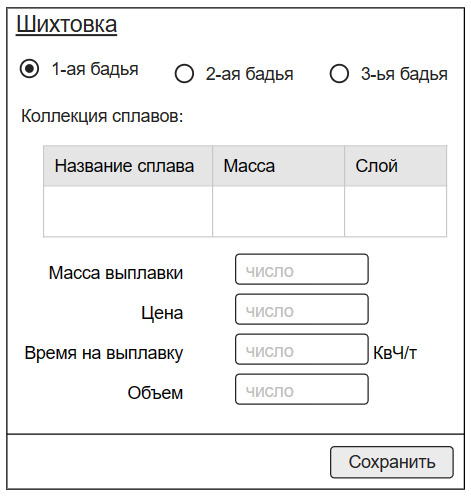


Рисунок 24 − Форма «Результат расчет шихтовки»

После нажатия кнопки «Сохранить» весь результат расчета будет сохранен в базу данных и все раннее рассчитанные шихтовки можно посмотреть на странице «Выполненные задания», изображенная на рисунке 25.

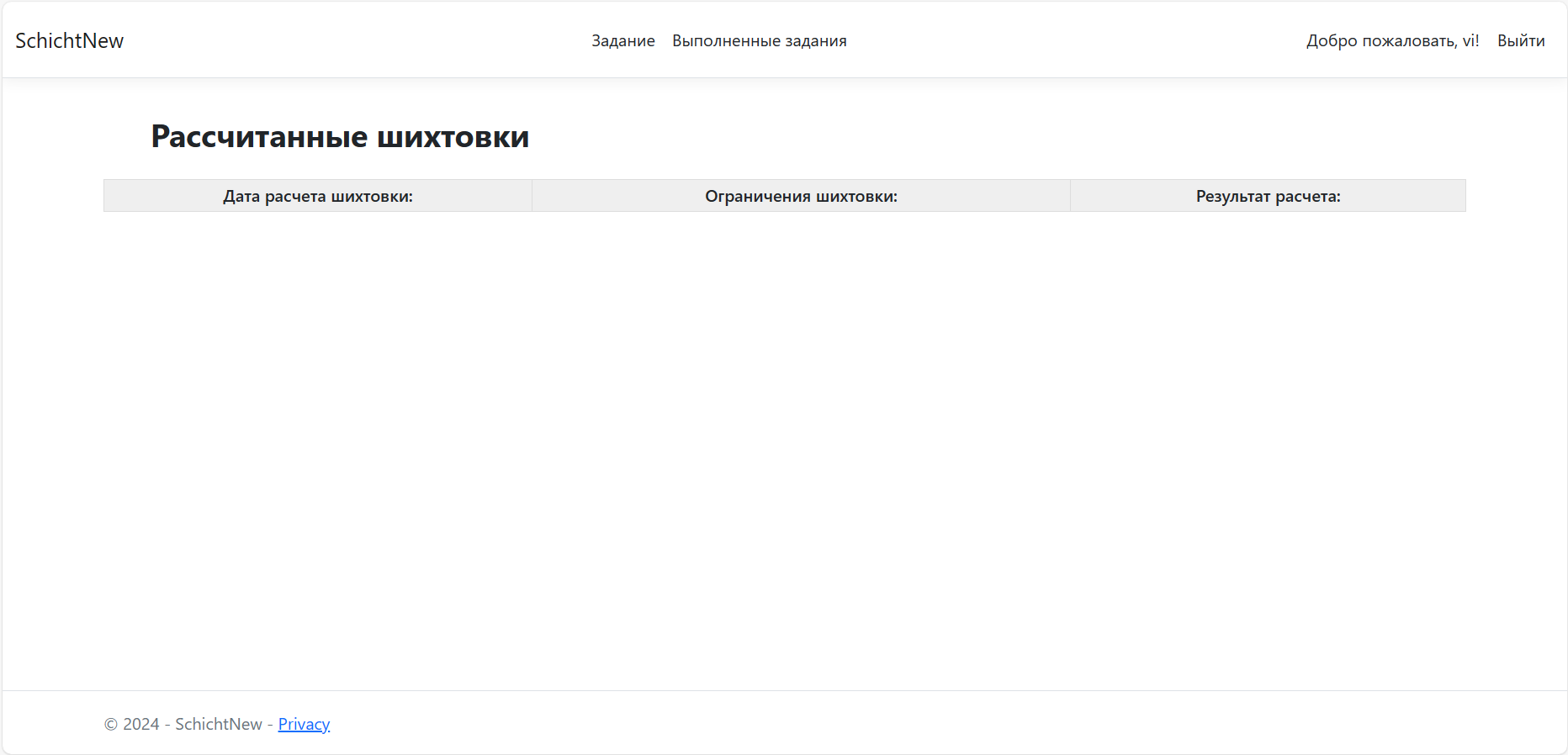


Рисунок 25 − Страница «Выполненные задания»

После разработки интерфейса всех страниц можно приступить к разработки самого приложения, так как сразу видно, как должно выглядеть итоговое приложение и можно учесть все нюансы удобного пользования веб-сервисом.

# 4 СИСТЕМА КОНТРОЛЯ ВЕРСИЙ

## 4.1 Описание системы контроля версий

GitHub ⎯ облачная платформа для хостинга IT-проектов и совместной разработки, которая поддерживается популярной системой контроля версий Git, а также полноценная сеть для разработчиков [22].

Функционал сервиса обеспечивает удобный и быстрый контроль над текущим состоянием программного продукта, а также сохраняет его последние версии и авторство внесенных изменений.

В настоящее время на данной платформе размещают проекты, имеющие открытый код, в разработке которых может принять участие любой желающий программист. Также есть возможность размещения персонального портфолио с примерами работ, результат ⎯ интерес со стороны потенциальных работодателей [23].

Все проекты хранятся в репозитории ⎯ коллекции всех изменений создаваемого кода. Внутри репозитория изменения кода хранятся в виде веток и коммитов. Основной веткой проекта, как правило, считается ветка main или master [24].

## 4.2 Работа с репозиторием

Перед началом работы на платформу GitHub необходимо зарегистрировать аккаунт на данном сервере. Результат регистрации представлен на рисунке 26.

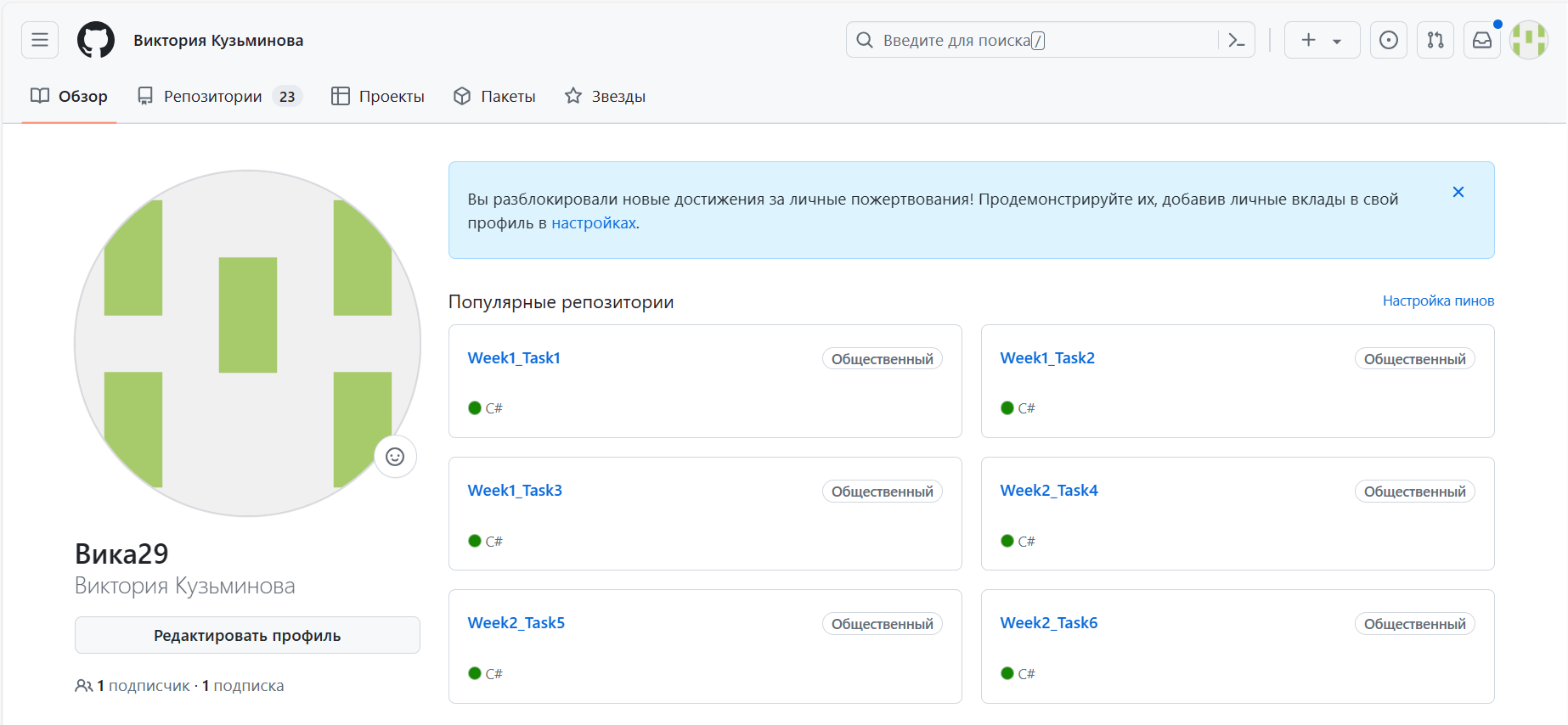


Рисунок 26 − Регистрация аккаунта на сервере GitHub

После регистрации аккаунта необходимо создать репозиторий, в котором, в дальнейшем, размещается проект, редактирующий на протяжении всей работы создаваемой программы. Созданный репозиторий представлен на рисунке 27.

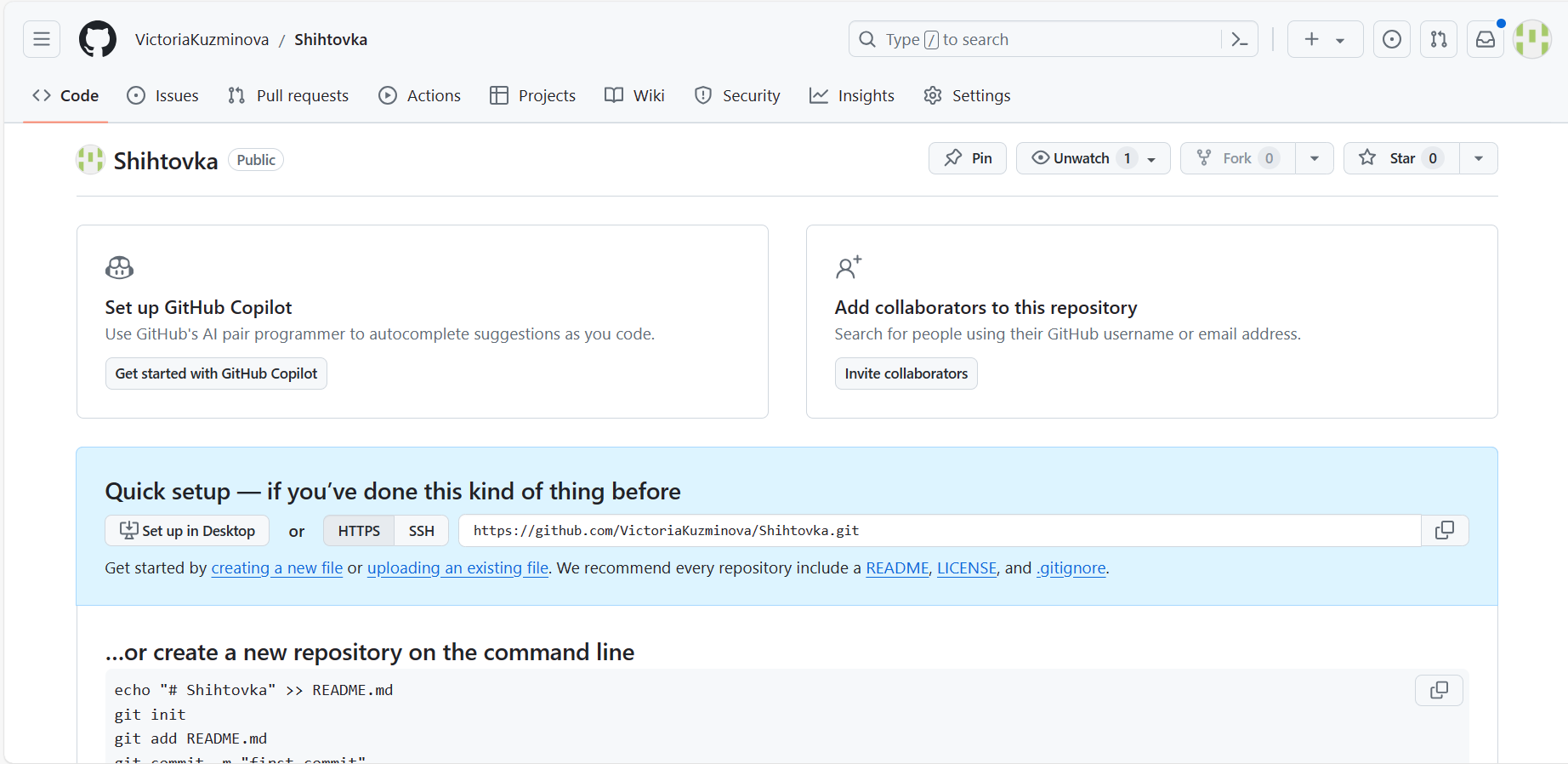


Рисунок 27 − Создание репозитория на сервере GitHub

После создания репозитория, ссылка на репозиторий указана в приложении А, можно начинать работу с программой, но для отслеживания изменений, происходящих с приложением, на сервере, его необходимо выгрузить. Результат выгрузки проекта на сервер, представлено на рисунке 28.

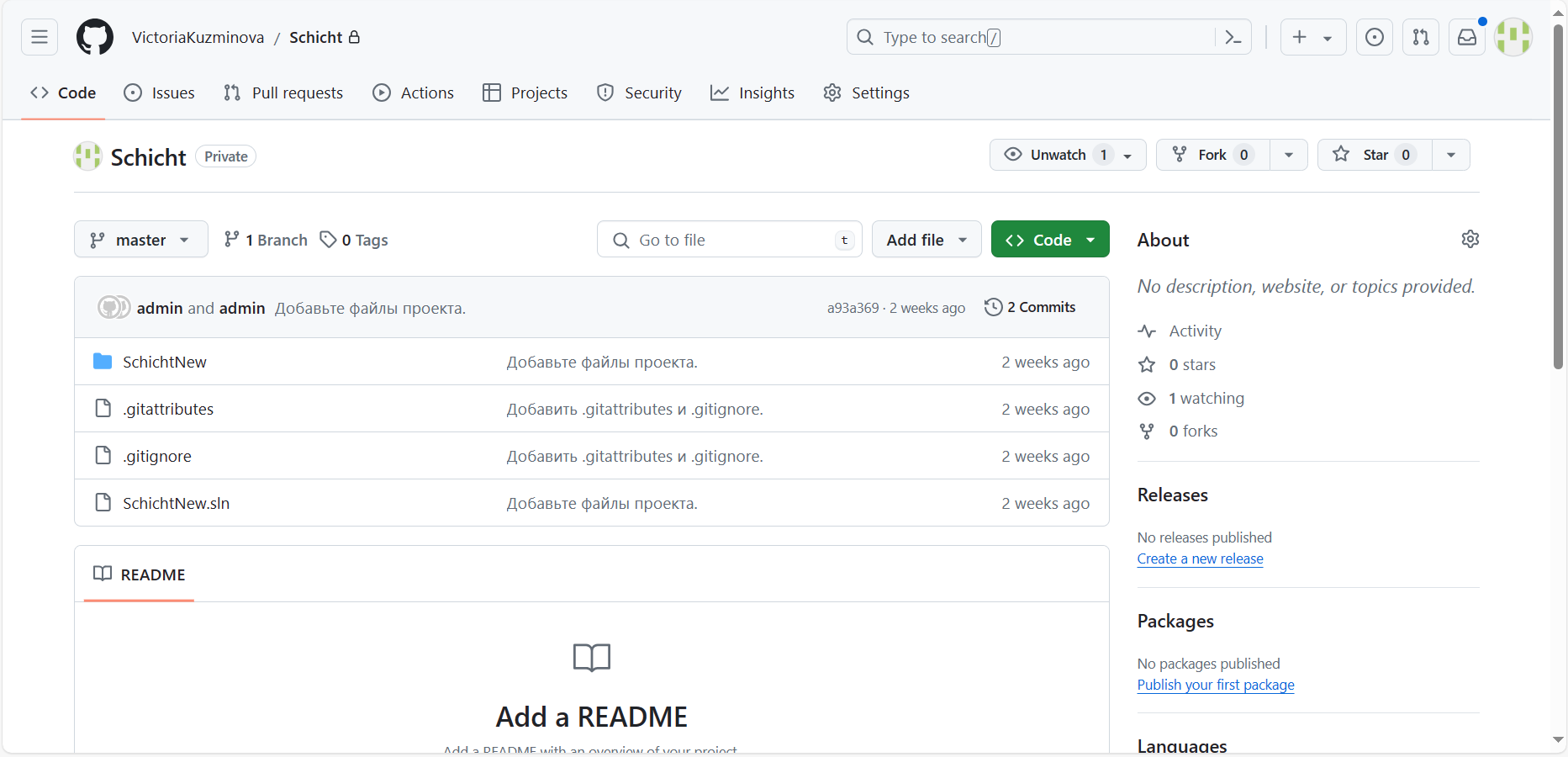


Рисунок 28 − Добавление проекта в репозиторий на сервере GitHub

Выгрузка проекта позволяет другим пользователям, которые заинтересованы в изменениях данного проекта могут спокойно контролировать это с помощью представленного сервиса, а разработчик может спокойно обновлять и вносить изменения в свой проект, удаленно отправляя их в репозиторий.

# 5 ТЕСТИРОВАНИЕ

## 5.1 Тест-требования

После изучения предметной области и составления диаграммы действия можно определить некоторые тест-требования, которые необходимо учесть при разработки приложения и исходя из них, провести тестирование, с предусмотренными требованиями к системе, а если что-то неучтённо, провести доработку программы.

1. Регистрация пользователя в системе
   1. Проверить, что пользователь ввел все данные для регистрации, если нет, то вывести сообщение о их отсутствии.
   2. Проверить, что в поле «E-mail» имеется символ «@».
   3. Проверить, что данные в полях «Пароль» и «Повторите пароль» совпадают.
2. Авторизация пользователя в системе

2.1) Проверить, что пользователь зарегистрирован в системе.

2.2) Проверить, что всех поля заполнены данными.

3) Добавление материала

3.1) Проверить, что при вводе вещественного числа, после запятой всего два знака.

3.2) Проверить, что все поля заполнены данными.

4) Работа с пользователями системы

4.1) Проверить, что логин нельзя менять, а возможна смена только роли пользователя.

После написанных тест-требований легко составить тест-план, с помощью которого можно проверить корректную работу разрабатываемого приложения.

## 5.2 Тест-план

После составленных тест-требований, необходимо описать тест-план для того, чтобы при проведении тестирования разработанного приложения знать, что и куда нужно вводить и определиться с тем, какой результат должен быть получен.

Тестовый пример 1.

Номер тест-требования: 1.1

Описание теста: в данном тесте проверятся отсутствие пустых полей данных на странице регистрация.

Входные данные:

* приложение «Расчет шихтовки ДСП»;
* страница «Регистрация пользователя»;
* поле «Пароль».

Ожидаемые выходные данные: при отсутствии данных в поле «Пароль» система должна подсветить данное поле и вывести сообщение с просьбой ввести данные в это поле.

Сценарий:

* запустить приложение;
* перейти на страницу «Регистрация»;
* в поле «e-mail» − [vika@gmail.com](mailto:vika@gmail.com);
* в поле «Логин» − vikakuzminova;
* поле «Пароль» оставить пустым;
* в поле «Повторите пароль» − 123456.

Тестовый пример 2.

Номер тест-требования: 1.2

Описание теста: в данном тесте проверятся отсутствие в поле «E-mail» символа «@».

Входные данные:

* приложение «Расчет шихтовки ДСП»;
* страница «Регистрация пользователя»;
* поле «E-mail».

Ожидаемые выходные данные: при отсутствии символа «@» в поле «E-mail» система должна подсветить данное поле и вывести сообщение с просьбой ввести данные в это поле.

Сценарий:

* запустить приложение;
* перейти на страницу «Регистрация»;
* в поле «e-mail» − [vikagmail.com](mailto:vika@gmail.com);
* в поле «Логин» − vikakuzminova;
* в поле «Пароль» − 123456;
* в поле «Повторите пароль» − 123456.

Тестовый пример 3.

Номер тест-требования: 1.3

Описание теста: в данном тесте проверятся совпадение данных в полях «Пароль» и «Повторите пароль».

Входные данные:

* приложение «Расчет шихтовки ДСП»;
* страница «Регистрация пользователя»;
* поле «Пароль»;
* поле «Повторите пароль».

Ожидаемые выходные данные: при вводе различных данных в поля «Пароль» и «Повторите пароль» система должна вывести сообщение о том, пароли различны.

Сценарий:

* запустить приложение;
* перейти на страницу «Регистрация»;
* в поле «e-mail» − [vika@gmail.com](mailto:vika@gmail.com);
* в поле «Логин» − vikakuzminova;
* в поле «Пароль» − 1234567;
* в поле «Повторите пароль» − 123456.

Тестовый пример 4.

Номер тест-требования: 2.1

Описание теста: в данном тесте проверятся наличие в системе пользователя с веденными данными.

Входные данные:

* приложение «Расчет шихтовки ДСП»;
* страница «Авторизация пользователя»;
* поле «Логин»;
* поле «Пароль».

Ожидаемые выходные данные: при вводе данных, которые не зарегистрированы в системе обновлять страницу и очищать поля «Логин» и «Пароль».

Сценарий:

* запустить приложение;
* перейти на страницу «Авторизация»;
* в поле «Логин» − vikakuzminova;
* в поле «Пароль» − 1234567.

Тестовый пример 5.

Номер тест-требования: 2.2

Описание теста: в данном тесте происходит проверка на пустые поля при авторизации.

Входные данные:

* приложение «Расчет шихтовки ДСП»;
* страница «Авторизация пользователя»;
* поле «Логин»;
* поле «Пароль».

Ожидаемые выходные данные: при оставлении полей «Логин» и «Пароль» пустыми система должна подсветить данные поля и вывести сообщение с просьбой о вводе данных.

Сценарий:

* запустить приложение;
* перейти на страницу «Авторизация»;
* поле «Логин» оставить пустым;
* поле «Пароль» оставить пустым.

Тестовый пример 6.

Номер тест-требования: 3.1

Описание теста: в данном тесте проверяется ограничение символов после запятой.

Входные данные:

* приложение «Расчет шихтовки ДСП»;
* страница «Материал»;
* поле «Угар».

Ожидаемые выходные данные: при вводе вещественного числа с тремя символами после запятой система должна вывести сообщение о данной ошибке.

Сценарий:

* запустить приложение;
* перейти на страницу «Материал»;
* в поле «Номер материала» − 123456;
* в поле «Название материала» − Латунь;
* в поле «Засор» − 1,2;
* в поле «Насыпная масса» − 10;
* в поле «Общая масса» − 1238,52;
* в поле «Стоимость» − 1254;
* в поле «Угар» − 12,145.

Тестовый пример 7.

Номер тест-требования: 3.2

Описание теста: в данном тесте проверяется полнота заполненности данными.

Входные данные:

* приложение «Расчет шихтовки ДСП»;
* страница «Материал»;
* поле «Название материала».

Ожидаемые выходные данные: при не заполнении поля данными система должна выделить его цветом и вывести сообщение с ошибкой.

Сценарий:

* запустить приложение;
* перейти на страницу «Материал»;
* в поле «Номер материала» − 123456;
* поле «Название материала» оставить пустым;
* в поле «Засор» − 1234;
* в поле «Насыпная масса» − 10;
* в поле «Общая масса» − 1238,52;
* в поле «Стоимость» − 1254;
* в поле «Угар» − 12,14.

Тестовый пример 8.

Номер тест-требования: 4.1

Описание теста: в данном тесте проверяется запрет на ввод значений в поле «Логин».

Входные данные:

* приложение «Расчет шихтовки ДСП»;
* страница «Роли пользователей»;
* поле «Логин».

Ожидаемые выходные данные: при нажатии на поле нельзя изменить данные, хранящиеся в нем.

Сценарий:

* запустить приложение;
* перейти на страницу «Роли пользователей»;
* в поле «Логин» − Виктория;

Подробное описание тест-плана с помощью тест-требований дает возможность человеку, который производит тестирование системы, правильно ввести все ограничения, благодаря которым, можно легко оценить качество работы программы.

## 5.3 Результаты тестирования

Результат тестирования 1.1.

Входные данные изображены на рисунке 29.

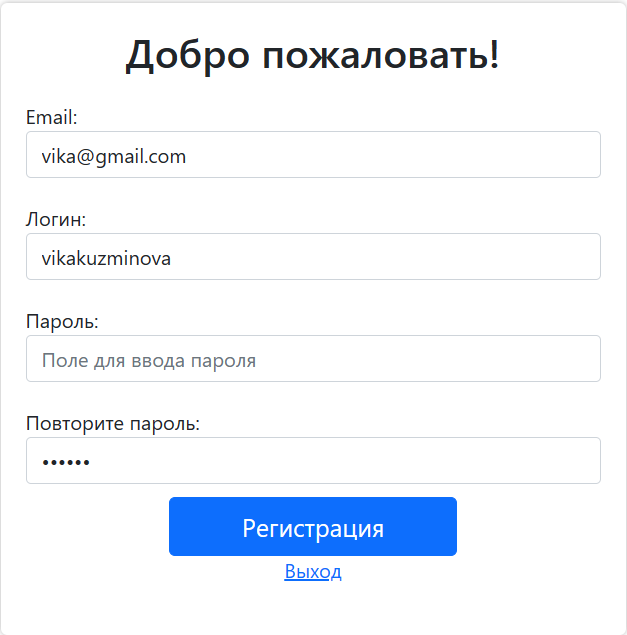


Рисунок 29 − Входные данные тестового примера 1

Выходные данные представлены на рисунке 30.

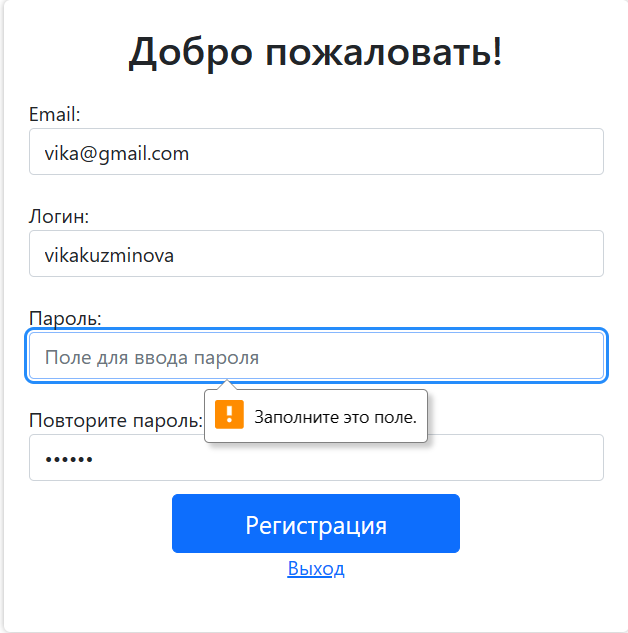


Рисунок 30 − Результат тестового примера 1

Результат тестирования 1.2.

Входные данные изображены на рисунке 31.

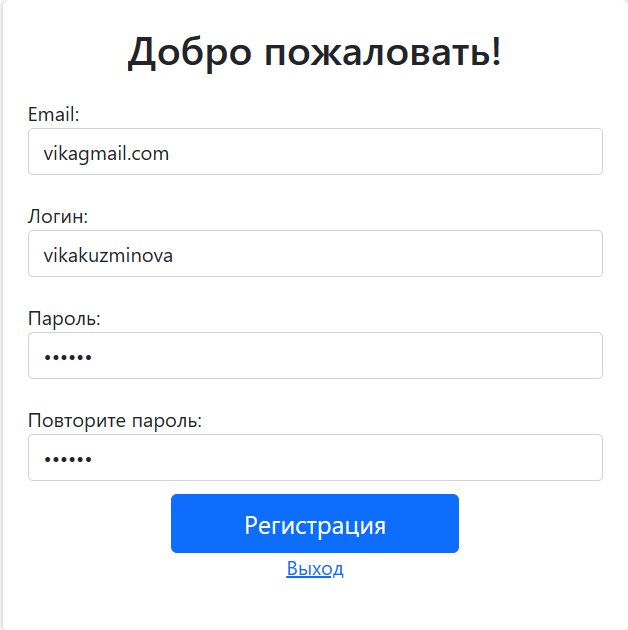


Рисунок 31 − Входные данные тестового примера 2

Выходные данные представлены на рисунке 32.

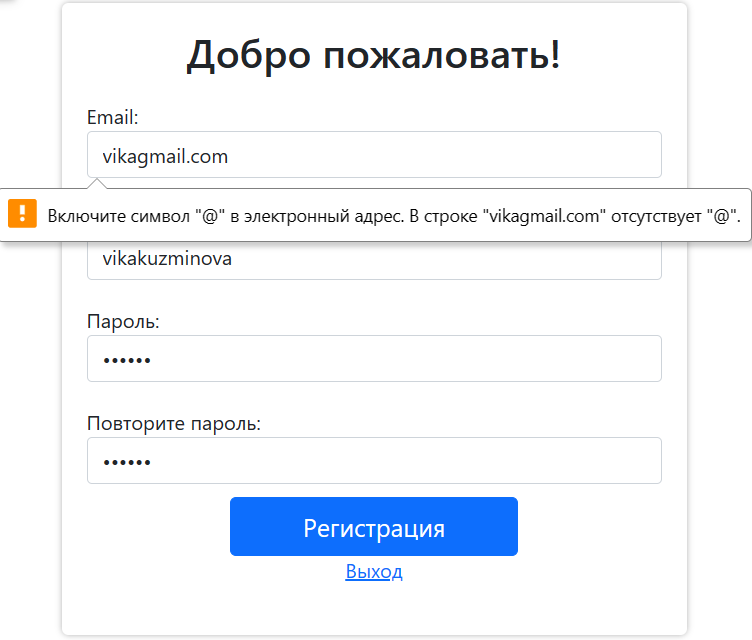


Рисунок 32 − Результат тестового примера 2

Результат тестирования 1.3.

Входные данные изображены на рисунке 33.

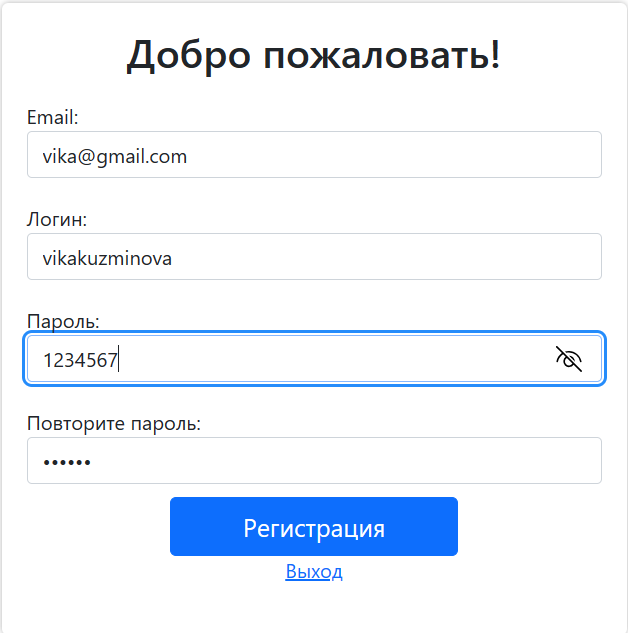


Рисунок 33 − Входные данные тестового примера 3

Выходные данные представлены на рисунке 34.

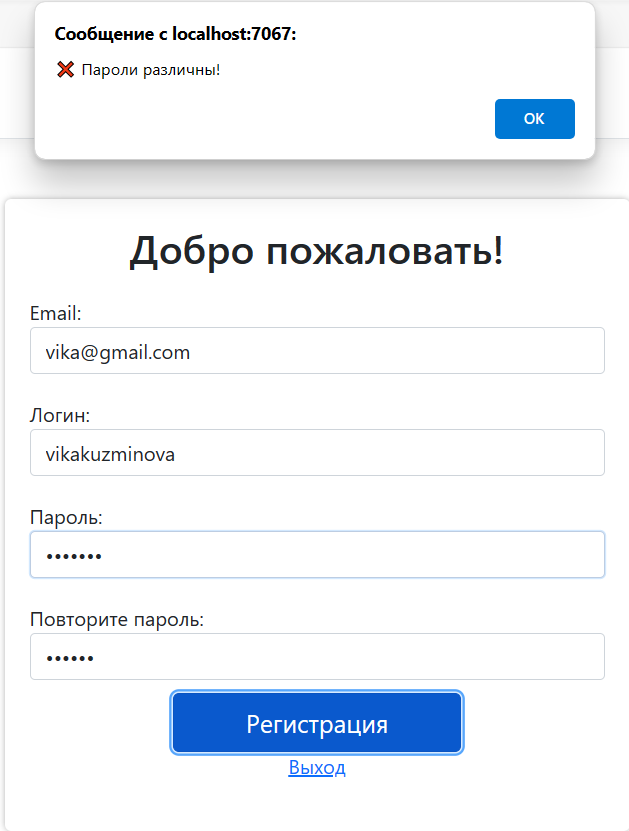


Рисунок 34 − Результат тестового примера 3

Результат тестирования 2.1.

Входные данные изображены на рисунке 35

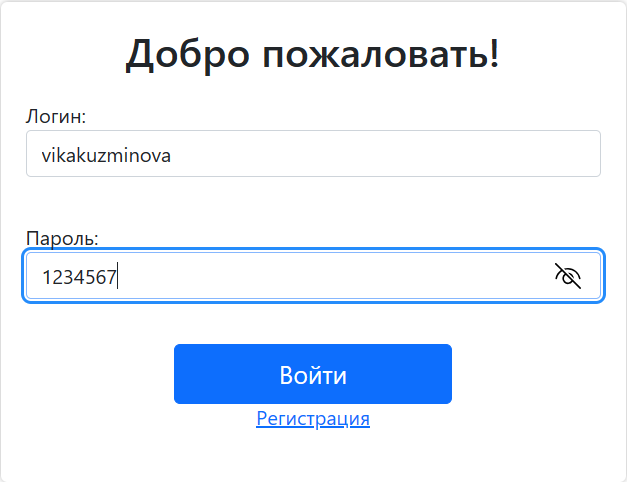


Рисунок 35 − Входные данные тестового примера 4

Выходные данные представлены на рисунке 36.

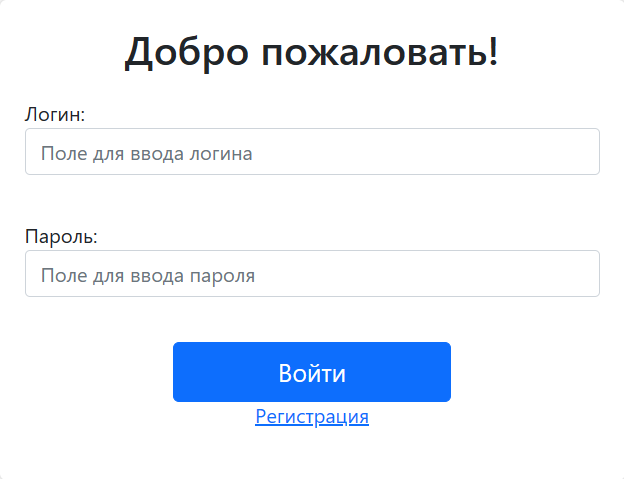


Рисунок 36 − Результат тестового примера 4

Результат тестирования 2.2.

Входные данные изображены на рисунке 37.

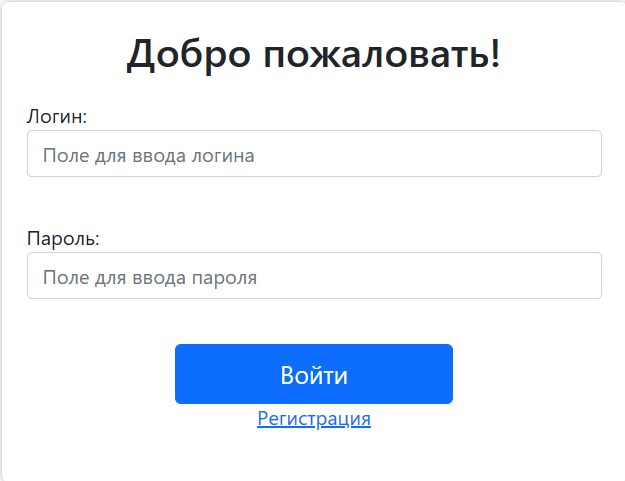


Рисунок 37 − Входные данные тестового примера 5

Выходные данные представлены на рисунке 38.

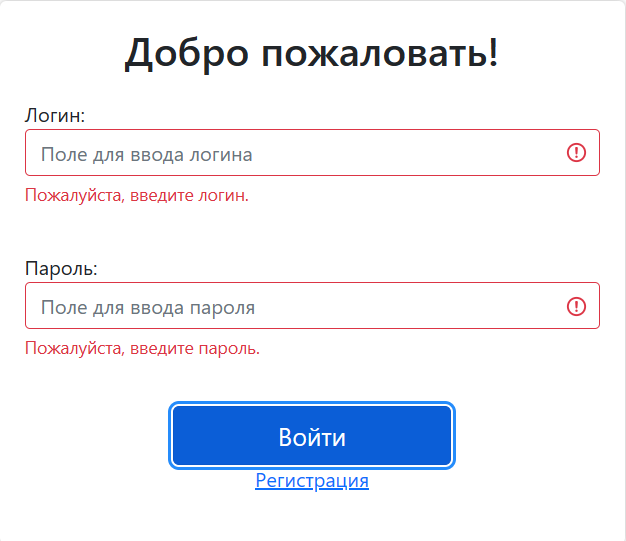


Рисунок 38 − Результат тестового примера 5

Результат тестирования 3.1.

Входные данные изображены на рисунке 39.

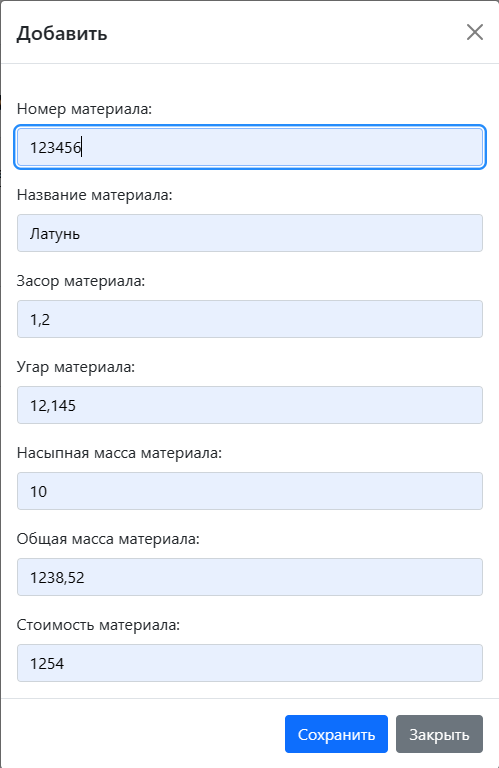


Рисунок 39 − Входные данные тестового примера 6

Выходные данные представлены на рисунке 40.

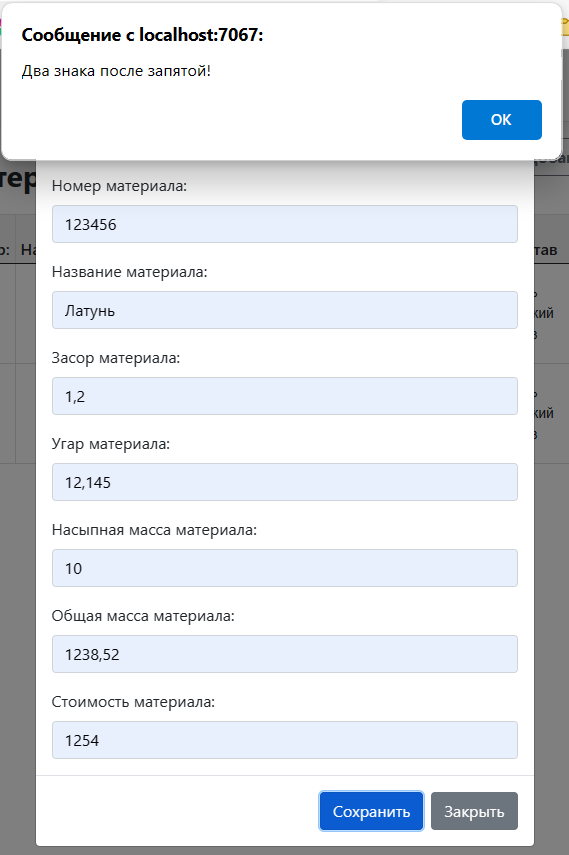


Рисунок 40 − Результат тестового примера 6

Результат тестирования 3.2.

Входные данные изображены на рисунке 41.

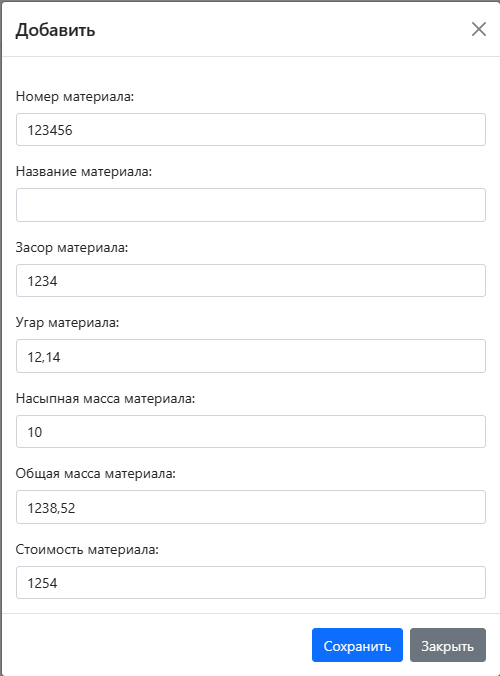


Рисунок 41 − Входные данные тестового примера 7

Выходные данные представлены на рисунке 42.

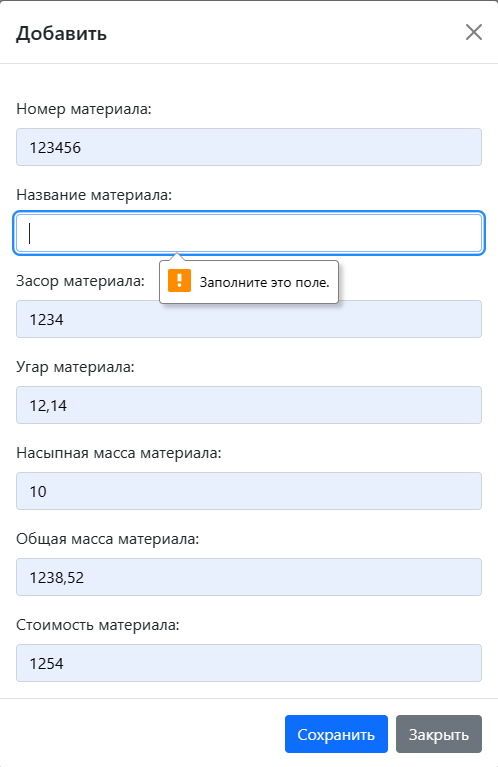


Рисунок 42 − Результат тестового примера 7

Результат тестирования 4.1.

Входные данные изображены на рисунке 43.

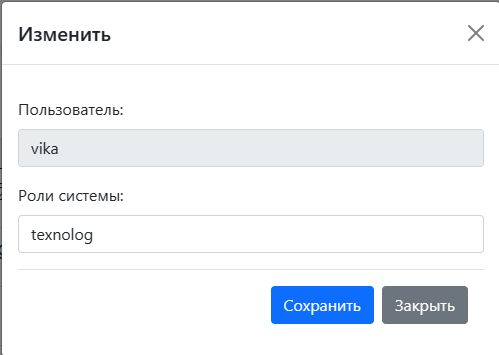


Рисунок 43 − Входные данные тестового примера 8

Выходные данные: поле выделено серым цветом, что означает то, что поле заблокировано для ввода других данных и изменить данные нельзя.

После проведения тестирования можно сказать, что приложение работает корректно, и все ограничения были учтены на данном этапе разработки.

# ЗАКЛЮЧЕНИЕ

В процессе прохождения производственной практики по модулю ПМ.03 «Участие в интеграции программных модулей» были рассмотрены и реализованы всевозможные способы достижения поставленной цели, анализ которых позволил выбрать наиболее оптимальный набор универсальных функций.

В ходе работы были выполнены все поставленные задачи:

− был выполнен анализ предметной области и произведен поиск программ-аналогов;

− успешное проектирование базы данный, состоящая из восьми взаимосвязанных таблиц;

− проектирование программного продукта, конечным результатом которого стал набор взаимосвязанных веб-страниц;

− успешное тестирование конечного продукта.

Программа имеет понятный интерфейс, является законченной и удобной для выполнения поставленных задач.

Для продолжения разработки продукта предполагаются следующие шаги:

− выполнить расчет шихтовки сплавов по введенным ограничениям;

− осуществить поиск по всем таблицам.

Дальнейшая разработка позволит улучшить взаимодействие пользователя с программой с помощью системы поиска, что позволит быстрее найти необходимую ему информацию.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Bstudy.ru [Электронный ресурс] − Расчеты материальных и энергетических балансов при выплавке стали в дуговых сталеплавильных печах − Режим доступа: [https://bstudy.net/876362/tehnika/](https://bstudy.net/876362/tehnika/teoreticheskie_osnovy_staleplavilnyh_protsessov?ysclid=luzldtcoxy599800575) (дата обращения 25.03.2024).
2. Большая российская энциклопедия [Электронный ресурс] − Лигатура в металлургии − Режим доступа: [https://bigenc.ru/c/ligatura](https://bigenc.ru/c/ligatura-v-metallurgii-66058f?ysclid=luzllck83w68614675) (дата обращения 25.03.2024).
3. StudFile.net [Электронный ресурс] − Общие основы сталеплавильного производства − Режим доступа: [https://studfile.net/preview](https://studfile.net/preview/1865385/) (дата обращения 25.03.2024).
4. Большая российская энциклопедия [Электронный ресурс] − Флюс в технике − Режим доступа: [https://bigenc.ru/c/flius](https://bigenc.ru/c/flius-v-tekhnike-1f8631?ysclid=luzlt5aub7557694296) (дата обращения 25.03.2024).
5. Отопление дома [Электронный ресурс] − Футеровка печи: что это такое? − Режим доступа: [https://pechi-ufa.ru/article/futerovka](https://pechi-ufa.ru/article/futerovka-pechi-chto-ehto-takoe?ysclid=luzlujgm1l563040710) (дата обращения 25.03.2024).
6. Большая российская энциклопедия [Электронный ресурс] − Шлак − Режим доступа: [https://bigenc.ru/c/shlak](https://bigenc.ru/c/shlak-5609a1?ysclid=luzlxtacdq399047812) (дата обращения 25.03.2024).
7. Академик [Электронный ресурс] − Шихта − Режим доступа: [https://dic.academic.ru/dic.nsf](https://dic.academic.ru/dic.nsf/dic_fwords/30244/%D0%A8%D0%98%D0%A5%D0%A2%D0%90?ysclid=luzlz79gjo659145265) (дата обращения 25.03.2024).
8. ОМК [Электронный ресурс] − Выксунский завод − Режим доступа: [https://omk.ru/factory/vyksa](https://omk.ru/factory/vyksa/?ysclid=luzm1pe8va884899428) (дата обращения 25.03.2024).
9. ОМК [Электронный ресурс] − Успех государственного масштаба− Режим доступа: [https://omk.ru/press/media](https://omk.ru/press/media/6337/?ysclid=luzm35rezz9730832) (дата обращения 25.03.2024).
10. Довнар Г.В., Неменёнок Б.М., Румянцева Г.А. Расчет шихты для плавки стали. − Минск: БНТУ, 2022. − 44с.
11. StudFile.net [Электронный ресурс] − Выплавка стали методом окисления − Режим доступа: [https://studfile.net/preview](https://studfile.net/preview/1731863/page:16/) (дата обращения 25.03.2024).
12. Хабр [Электронный ресурс] − Использование диаграммы вариантов использования UML при проектировании программного обеспечения − Режим доступа: [https://habr.com/ru/articles](https://habr.com/ru/articles/566218/) (дата обращения 27.03.2024).
13. Ensi [Электронный ресурс] − Диаграмма активностей (Activity diagram) − Режим доступа: [https://docs.ensi.tech/analyst-guides](https://docs.ensi.tech/analyst-guides/tools/diagrams/uml/activity-diagram) (дата обращения 27.03.2024).
14. Wikipedia. Microsoft Word [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Word (дата обращения 04.04.2024).
15. Образовательные технологии Яндекса [Электронный ресурс] – Сервис Draw.io: за что его любят и как им пользоваться – Режим доступа: https://practicum.yandex.ru/blog/drawio (дата обращения 04.04.2024).
16. SSK [Электронный ресурс] – SQL Server Management Studio Режим доступа: https://sqlserver-kit.org/ru/ssms (дата обращения 04.04.2024).
17. Microsoft [Электронный ресурс] – Добро пож4аловать в интегрированную среду разработки Visual Studio – Режим доступа: https://web.archive.org/web/20210110233636/visual-studio (дата обращения 04.04.2024).
18. Microsoft [Электронный ресурс] – Краткий обзор языка C# – Режим доступа: https://docs.microsoft.com/tour-of-csharp/ (дата обращения 04.04.2024).
19. Skillfactory media [Электронный ресурс] – JavaScript – Режим доступа: https://blog.skillfactory.ru/glossary/javascript (дата обращения 04.04.2024).
20. Skillfactory media [Электронный ресурс] – HTML – Режим доступа: https://blog.skillfactory.ru/glossary/html/ (дата обращения 04.04.2024).
21. Образовательные технологии Яндекса [Электронный ресурс] – Что такое CSS и почему без него не стать веб‑разработчиком – Режим доступа: https://practicum.yandex.ru/blog/chto-takoe-css (дата обращения 04.04.2024).
22. Skillbox Media [Электронный ресурс] − Что такое GitHub и как им пользоваться − Режим доступа: [https://skillbox.ru/media/code/](https://skillbox.ru/media/code/chto-takoe-github-i-kak-im-polzovatsya/?ysclid=luzmnft7qn95959058) (дата обращения 10.04.2024).
23. Сравни [Электронный ресурс] − GitHub – обзор − Режим доступа: <https://www.sravni.ru/kursy/info/github/> (дата обращения 10.04.2024).
24. Skill Factory Media [Электронный ресурс] − GitHub − Режим доступа: [https://blog.skillfactory.ru/glossary/github](https://blog.skillfactory.ru/glossary/github/?ysclid=luzmqmmqgi434728871) (дата обращения 10.04.2024).

# ПРИЛОЖЕНИЕ А

Ссылка на репозиторий

<https://github.com/VictoriaKuzminova/Schicht.git>

Листинг страницы «Авторизация»

@{

ViewData["Title"] = "Авторизация";

}

@model SchichtNew.Models.LoginUser

<style>

.form-container {

max-width: 500px;

margin: auto;

padding: 20px;

background-color: #fff;

border-radius: 5px;

box-shadow: 0 0 5px rgba(0, 0, 0, 0.3);

}

h2 {

text-align: center;

margin-bottom: 20px;

}

.form-group {

margin-bottom: 20px;

}

.error {

color: red;

margin-top: 5px;

}

.success {

color: green;

margin-top: 5px;

}

</style>

<div class="container mt-5">

<div class="form-container">

<h2>Добро пожаловать!</h2>

<**form** class="row g-3 needs-validation" novalidate method="post" **asp-controller**="Home" **asp-action**="Login" **asp-route-returnUrl**="@Model.ReturnUrl">

<div class="form-group">

<**label** **asp-for**="UserName">Логин:</**label**>

<**input** **asp-for**="UserName" **type**="text" class="form-control" id="UserName" **name**="UserName" placeholder="Поле для ввода логина" required>

<div class="invalid-feedback">

Пожалуйста, введите логин.

</div>

</div>

<div class="form-group">

<**label** **asp-for**="Password">Пароль:</**label**>

<**input** **asp-for**="Password" **type**="password" class="form-control" id="password" **name**="password" placeholder="Поле для ввода пароля" required>

<div class="invalid-feedback">

Пожалуйста, введите пароль.

</div>

</div>

<div class="form-group">

<**input** **asp-for**="RememberMe" **type**="hidden" class="form-control" id="rememberMe" **name**="rememberMe" **value**="false">

<!--<span asp-validation-for="RememberMe"></span>-->

<div class="error"></div>

</div>

<div class="d-grid gap-2 col-6 mx-auto" style="margin-top:-25px">

<button type="submit" id="submit" name="submit" class="btn btn-primary btn-lg">Войти</button>

</div>

<div class="form-group" style="margin-top:-1px; text-align: center">

<**a** **asp-controller**="Home" **asp-action**="Register">Регистрация</**a**>

</div>

</**form**>

</div>

</div>

<script>

// Пример стартового JavaScript для отключения отправки форм при наличии недопустимых полей

(function () {

'use strict'

// Получите все формы, к которым мы хотим применить пользовательские стили проверки Bootstrap

var forms = document.querySelectorAll('.needs-validation')

// Зацикливайтесь на них и предотвращайте отправку

Array.prototype.slice.call(forms)

.forEach(function (form) {

form.addEventListener('submit', function (event) {

if (!form.checkValidity()) {

event.preventDefault()

event.stopPropagation()

}

form.classList.add('was-validated')

}, false)

})

})()

</script>

[HttpPost]

[AllowAnonymous]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Login(LoginUser model)

{

var models = new LoginUser { Password = null!, UserName = null! };

if (ModelState.IsValid)

{

var result =

await \_signInManager.PasswordSignInAsync(model.UserName, model.Password, false, false);

var user=\_userManager.FindByNameAsync(model.UserName);

var roles = \_userManager.GetRolesAsync(user.Result);

if (result.Succeeded)

{

// проверяем, принадлежит ли URL приложению

if (!string.IsNullOrEmpty(model.ReturnUrl) && Url.IsLocalUrl(model.ReturnUrl))

{

return Redirect(model.ReturnUrl);

}

else

{

foreach (var item in roles.Result)

{

if (item == "administrator")

return RedirectToAction("Material", "Material");

else

return RedirectToAction("Mission", "Mission");

}

return RedirectToAction("Login", "Home");

}

}

else

{

return RedirectToAction("Login", "Home");

}

}

else

{

return RedirectToAction("Login", "Home");

}

}

Листинг страницы «Материал»

@model IEnumerable<SchichtNew.Models.Material>

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="container container-fluid">

<div class="container-ovrLayout">

<div class="PageHeader">

<div class="row" style="text-align:center; text-shadow: 1px; margin-bottom: 1em">

<div class="col-sm-2">

<label class="align-middle" style="font-style: normal; font-weight: bold; padding-top: 0.5em; font-size:30px">@ViewBag.TagName</label>

</div>

<div class="col-sm" style="padding-right:2vW">

<a id="bucketsView" class="btn btn-outline-secondary float-end " data-bs-toggle="modal" data-bs-target="#editMaterial" onclick="AddMaterial()"><i style="font-style: normal; font-weight: bold; " class="bi bi-plus-circle"> Добавить</i></a>

</div>

</div>

</div>

<div class="MemoList" id="materialContainer">

<table class="table" id="myTable">

<thead>

<tr>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Номер:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Название:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Масса:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Угар:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Засор:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Насыпная масса:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Стоимость:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Хим.состав</h6>

</th>

<th id="tableLabel1">

<h6 style="margin-bottom : 0px">Действия</h6>

</th>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr style="cursor: pointer">

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.Number)</p>

</td>

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.Name)</p>

</td>

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.Mass)</p>

</td>

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.Ugar)</p>

</td>

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.Blockage)</p>

</td>

<td style="text-align:center;font-weight: bold">

<p id="label\_s">@Html.DisplayFor(modelItem => item.BulkMass)</p>

</td>

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.Price)</p>

</td>

<td style="text-align:center; padding:10px; font-weight: bold">

@if (Model.Count() >= 1)

{

<a class="nav-link text-dark" id="label\_s" onclick=" SlagAndMetalStades(@item.Id)">Узнать химический состав</a>

}

</td>

<td style="text-align:center; padding:10px; font-weight: bold">

@if (Model.Count() >= 1)

{

<div class="btn-group btn-group-sm" role="group" aria-label="Edit">

<div>

<a class="btn btn-outline-warning" style="margin:0px;" data-bs-toggle="modal" data-bs-target="#editMaterial" onclick="EditMaterial(@item.Id,'@item.Number','@item.Name',@item.Blockage, @item.Ugar,

@item.BulkMass,@item.Mass,@item.Price)"> <i class="bi bi-brush"></i></a>

</div>

<div>

<a class="btn btn-outline-danger" style="margin:0px;" onclick="DeleteMaterial(@item.Id)"><i class="bi bi-trash"></i></a>

</div>

</div>

}

</td>

</tr>

}

</tbody>

</table>

</div>

</div>

</div>

<form>

<div class="modal" id="editMaterial" data-bs-backdrop="true" tabindex="-1">

<!--modal form-->

<div class="modal-dialog modal-md">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="wnd-descr">Изменить</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<input type="hidden" id="materialId" name="materialId" value="-1" />

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label" for="inpNumber">Номер материала:</label>

<input class="form-control" type="text" id="inpNumber" name="inpNumber" required />

</div>

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label" for="inpName">Название материала:</label>

<input class="form-control" type="text" id="inpName" name="inpName" required />

</div>

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label" for="inpBlockage">Засор материала:</label>

<input class="form-control" type="number" step="0.01" min="0" id="inpBlockage" name="inpBlockage" required />

</div>

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label" for="inpUgar">Угар материала:</label>

<input class="form-control" type="number" step="0.01" min="0" id="inpUgar" name="inpUgar" required />

</div>

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label" for="inpBulkMass">Насыпная масса материала:</label>

<input class="form-control" type="number" step="0.01" min="0" id="inpBulkMass" name="inpBulkMass" required />

</div>

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label" for="inpMass">Общая масса материала:</label>

<input class="form-control" type="number" step="0.01" min="0" id="inpMass" name="inpMass" required />

</div>

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label" for="inpPrice">Стоимость материала:</label>

<input class="form-control" type="number" step="0.01" min="0" id="inpPrice" name="inpPrice" required />

</div>

</div>

<div class="modal-footer">

<button id="btn-confirm1" type="submit" class="btn btn-primary">Сохранить</button>

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Закрыть</button>

</div>

</div>

</div>

</div>

</form>

<script>

var isMatrial = 1;

var materialContainer = document.getElementById("materialContainer");

var slagAndMetal = document.getElementById("slagAndMetalContainer");

function GetMaterialById(idMaterial) {

var idMaterial = idMaterial;

fetch('/Material/GetMaterialById/' + idMaterial)

.then(function (response) {

return response.text();

})

.then(function (html) {

location.reload();

});

}

function AddMaterial() {

isMatrial = 0;

document.getElementById("wnd-descr").innerHTML = "Добавить";

document.getElementById("inpNumber").value = null;

document.getElementById("inpName").value = null;

document.getElementById("inpBlockage").value = null;

document.getElementById("inpUgar").value = null;

document.getElementById("inpBulkMass").value = null;

document.getElementById("inpMass").value = null;

document.getElementById("inpPrice").value = null;

}

function EditMaterial(Id, Number, Name, Blocksge, Ugar, BulkMass, Mass, Price) {

isMatrial = 1;

document.getElementById("wnd-descr").innerHTML = "Изменить";

document.getElementById("materialId").value = Id;

document.getElementById("inpNumber").value = Number;

document.getElementById("inpName").value = Name;

document.getElementById("inpBlockage").value = Blocksge;

document.getElementById("inpUgar").value = Ugar;

document.getElementById("inpBulkMass").value = BulkMass;

document.getElementById("inpMass").value = Mass;

document.getElementById("inpPrice").value = Price;

}

document.addEventListener('DOMContentLoaded', function () {

var btnEditMaterial = document.getElementById("btn-confirm1");

btnEditMaterial.addEventListener("click", (e) => {

var idMaterial = document.getElementById("materialId").value;

var number = document.getElementById("inpNumber").value;

var name = document.getElementById("inpName").value;

var blockage = document.getElementById("inpBlockage").value;

var ugar = document.getElementById("inpUgar").value;

var bulkMass = document.getElementById("inpBulkMass").value;

var mass = document.getElementById("inpMass").value;

var price = document.getElementById("inpPrice").value;

if ((name !== "") && (number !== "")) {

if (isMatrial === 0) {

fetch('/Material/AddMaterial/', {

method: 'POST',

headers: {

'Content-Type': 'application/json;charset=utf-8'

},

body: JSON.stringify({

Number: number,

Name: name,

Blockage: blockage,

Ugar: ugar,

BulkMass: bulkMass,

Mass: mass,

Price: price

})

})

.then(() => { GetMaterialById(idMaterial); });

}

if (isMatrial === 1) {

fetch('/Material/EditMaterial/', {

method: 'PUT',

headers: {

'Content-Type': 'application/json;charset=utf-8'

},

body: JSON.stringify({

Id: idMaterial,

Number: number,

Name: name,

Blockage: blockage,

Ugar: ugar,

BulkMass: bulkMass,

Mass: mass,

Price: price

})

})

.then(() => {

location.reload();

});

}

}

})

})

function DeleteMaterial(Id) {

if (confirm("Удалить запись?")) {

fetch('/Material/DeleteMaterial/' + Id)

.then(function (response) {

return response.text();

})

.then(() => {

location.reload();

});

}

}

function SlagAndMetalStades(idMaterial) {

if (confirm("Посмотреть химию?")) {

fetch('/SlagAndMetalStades/SlagAndMetalStades/' + idMaterial)

.then(() => {

window.location.replace("https://localhost:7067/SlagAndMetalStades/SlagAndMetalStades/" + idMaterial);

});

}

}

function f(n) {

let i = 0;

if (n % 1 != 0) {

do {

n = n \* 10;

i++;

} while (n % 1 != 0);

}

return i;

}

</script>

public async Task<ActionResult> AddMaterial([FromBody] MaterialCreateDto material)

{

if (material != null)

{

if ((SearchMaterial(material.Number) == null) && (SearchMaterial(material.Name) == null))

{

Material newMaterial = new Material();

if (material.Number != null) newMaterial.Number = material.Number; else return BadRequest("Error");

if (material.Name != null) newMaterial.Name = material.Name; else return BadRequest("Error");

if ((material.Blockage != null) && (GetNumberOfDecimalPlaces((double)material.Blockage) <= 2))

newMaterial.Blockage = (double)material.Blockage;

else return BadRequest("Error");

if ((material.Ugar != null) && (GetNumberOfDecimalPlaces((double)material.Ugar) <= 2))

newMaterial.Ugar = (double)material.Ugar;

else return BadRequest("Error");

if ((material.BulkMass != null) && (GetNumberOfDecimalPlaces((double)material.BulkMass) <= 2))

newMaterial.BulkMass = (double)material.BulkMass;

else return BadRequest("Error");

if ((material.Mass != null) && (GetNumberOfDecimalPlaces((double)material.Mass) <= 2))

newMaterial.Mass = (double)material.Mass;

else return BadRequest("Error");

if ((material.Price != null) && (GetNumberOfDecimalPlaces((double)material.Price) <= 2))

newMaterial.Price = (decimal)material.Price;

else return BadRequest("Error");

await \_materialService.AddMaterial(newMaterial);

await \_slagStageService.AddSlagStage(newMaterial.Id);

await \_metalStageService.AddMetalStage(newMaterial.Id);

var model = GetAllMaterial();

return View(new { model });

}

}

return BadRequest("Error");

}

Листинг страницы «Ограничения»

@model IEnumerable<SchichtNew.Contracts.ChemicalLimits.ChemicalLimitsCteateDto>

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<div class="container container-fluid">

<div class="container-ovrLayout">

<div class="PageHeader">

<div class="row" style="text-align:center; margin-bottom: 1em">

<div class="col-sm-2">

<label class="align-middle" style="font-style: normal; font-weight: bold; padding-top: 0.5em; font-size:30px ">@ViewBag.TagName</label>

</div>

</div>

</div>

</div>

<form>

<div class="container-ovrLayout">

<div class="PageHeader">

<div class="row" style="text-shadow: 1px; margin-left:-5em; margin-bottom: 1em">

<div class="col-sm-2" align="right">

<label class="form-label"> Масса выплавки</label>

</div>

<div class="col-sm-2">

<input class="form-control" type="number" step="0.01" min="0" name="inpMass" id="inpMass" placeholder="введите число">

</div>

</div>

<div class="row" style="text-shadow: 1px; margin-left:-5em; margin-bottom: 1em">

<div class="col-sm-2" align="right">

<label class="form-label"> Цена</label>

</div>

<div class="col-sm-2">

<input class="form-control" type="number" step="0.01" min="0" name="inpPrice" id="inpPrice" placeholder="введите число">

</div>

<div class="col-sm-2" align="right">

<label class="form-label"> Время на выплавку</label>

</div>

<div class="col-sm-2">

<input class="form-control" type="number" step="0.01" min="0" name="inpTime" id="inpTime" placeholder="введите число">

</div>

<div class="col-sm-2">

<label class="form-label" style="text-align:left;"> КвЧ/Т</label>

</div>

</div>

<div class="row" style="text-shadow: 1px; margin-left:-5em; margin-bottom: 1em">

<div class="col-sm-2" align="right">

<label class="form-label" style="margin-left:25px"> Соотношение слоев: </label>

</div>

</div>

<div class="row" style="height:7em; border:3px #d7d7d7 solid; text-align:center; text-shadow: 1px;margin-left:-0.1em;margin-top:-10px">

<div class="col-sm-2">

<label class="form-label" style="margin-top:10px;">Легкий</label>

</div>

<div class="col-sm-2" style=" padding-top:2.5em;margin-left:-5em">

<input id="num1" class="form-control" type="number" step="0.01" min="0" max="100">

</div>

<div class="col-sm-2">

<label class="form-label" style="margin-top:10px;">Средний</label>

</div>

<div class="col-sm-2" style=" padding-top:2.5em;margin-left:-5em">

<input id="num2" class="form-control" type="number" step="0.01" min="0" max="100">

</div>

<div class="col-sm-2">

<label class="form-label" style="margin-top:10px;">Тяжелый</label>

</div>

<div class="col-sm-2" style=" padding-top:2.5em;margin-left:-5em">

<input id="num3" class="form-control" type="number" step="0.01" min="0" max="100">

</div>

</div>

<div class="row" style="text-align:center; text-shadow: 1px; margin-bottom: 1em">

<div class="col-sm-2">

<label class="align-middle" style="font-style: normal; font-weight: bold; padding-top: 0.5em; margin-top:10px;font-size:15px">Химическое ограничение</label>

</div>

<div class="col-sm" style="padding-right:2vW">

<a id="bucketsView" class="btn btn-outline-secondary float-end " style="margin-top:10px" data-bs-toggle="modal" data-bs-target="#addChemLimits" onclick="AddChemicalLimits()"><i style="font-style: normal; font-weight: bold;" class="bi bi-plus-circle"> Добавить</i></a>

</div>

</div>

<table class="table" id="myTable">

<thead>

<tr>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Название элемента:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Минимальное значение:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Максимальное значение:</h6>

</th>

<th id="tableLabel">

<h6 style="margin-bottom : 0px">Действия</h6>

</th>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr style="cursor: pointer">

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.ChemicalName)</p>

</td>

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.minValue)</p>

</td>

<td style="text-align:center;font-weight: bold ">

<p id="label\_s">@Html.DisplayFor(modelItem => item.maxValue)</p>

</td>

<td style="text-align:center; padding:10px; font-weight: bold">

@if (Model.Count() >= 1)

{

<div class="btn-group btn-group-sm" role="group" aria-label="Edit">

<div>

<a class="btn btn-outline-warning" style="margin:0px;" data-bs-toggle="modal" data-bs-target="#editMaterial" onclick="EditMaterial()"> <i class="bi bi-brush"></i></a>

</div>

<div>

<a class="btn btn-outline-danger" style="margin:0px;" onclick="DeleteMaterial()"><i class="bi bi-trash"></i></a>

</div>

</div>

}

</td>

</tr>

}

</tbody>

</table>

</div>

</div>

</form>

</div>

<form>

<div class="modal" id="addChemLimits" data-bs-backdrop="true" tabindex="-1">

<!--modal form-->

<div class="modal-dialog modal-md">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="wnd-descr">Добавить</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label1" style="margin-top:-20px;" for="inpNumber"> Элемент: </label>

@Html.DropDownList("chemicalElements", ViewBag.ChemicalElements as SelectList)

</div>

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label1" for="minValue">Мин.значение:</label>

<input class="form-control" type="number" step="0.01" min="0" id="minValue" name="minValue" required />

</div>

<div class="form-group d-md-grid" style="padding-top: 1em">

<label class="form-label1" for="maxValue">Макс.значение:</label>

<input class="form-control" type="number" step="0.01" min="0" id="maxValue" name="maxValue" required />

</div>

</div>

<div class="modal-footer">

<button id="confirm2" type="submit" class="btn btn-primary">Сохранить</button>

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Закрыть</button>

</div>

</div>

</div>

</div>

</form>

<script>

var list = document.getElementById("chemicalElements");

function AddChemicalLimits() {

list.setAttribute("class", "form-control");

list.setAttribute("style", "margin-bottom: 1em");

}

document.addEventListener('DOMContentLoaded', function () {

var btnAddLimit = document.getElementById("confirm2");

btnAddLimit.addEventListener("click", (e) => {

var minValue = document.getElementById("minValue").value;

var maxValue = document.getElementById("maxValue").value;

fetch('/Mission/AddChemicalLimits/', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

ChemicalName: list.value,

minValue: minValue,

maxValue: maxValue

})

}).then(() => {

console.log(1);

});

})

})

</script>