

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 клиентской части системы отслеживания
срока годности продукции**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2024.308-310.ВКР**

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.
_____ Е.В. Иванова

Автор работы,
студент группы КЭ-401
_____ А.А. Сараева

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

студентке группы КЭ-401

Сараевой Анастасии Андреевны,

обучающейся по направлению

02.03.02 «Фундаментальная информатика и информационные технологии»

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

Разработка клиентской части системы отслеживания срока годности продукции.

2. Срок сдачи студентом законченной работы: 03.06.2024 г.

3. Исходные данные к работе

3.1. Мардан А. React быстро. Веб-приложения на React, JSX, Redux и GraphQL. – СПб.: Питер. 2019. – 560 с.

3.2. React: A JavaScript library for building user interfaces. [Электронный ресурс] URL: <https://ru.legacy.reactjs.org> (дата обращения: 29.01.2024 г.).

3.3. Современный учебник JavaScript. [Электронный ресурс] URL: <https://learn.javascript.ru> (дата обращения: 29.01.2024 г.).

4. Перечень подлежащих разработке вопросов

4.1. Выполнить анализ предметной области, описать требования к системе отслеживания срока годности продукции.

4.2. Спроектировать клиентскую часть системы.

4.3. Выполнить реализацию клиентской части системы.

4.4. Выполнить тестирование клиентской части системы.

5. Дата выдачи задания: 29.01.2024 г.

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.

Е.В. Иванова

Задание принял к исполнению

А.А. Сараева

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	7
1.1. Анализ аналогичных проектов.....	7
1.2. Анализ существующих решений для реализации проекта.....	10
2. ПРОЕКТИРОВАНИЕ.....	12
2.1. Требования к системе.....	12
2.2. Варианты использования системы.....	14
2.3. Проектирование интерфейса.....	17
2.4. Архитектура системы.....	19
2.5. Структура JSON данных.....	21
3. РЕАЛИЗАЦИЯ.....	23
3.1. Средства реализации.....	23
3.2. Реализация аутентификации.....	24
3.3. Взаимодействие с серверной частью.....	25
3.4. Информационная архитектура.....	28
3.5. React разработка.....	30
3.6. Адаптивный интерфейс.....	31
4. ТЕСТИРОВАНИЕ.....	33
ЗАКЛЮЧЕНИЕ.....	37
ЛИТЕРАТУРА.....	38
ПРИЛОЖЕНИЯ.....	40
Приложение А. Диаграмма вариантов использования.....	40
Приложение Б. Спецификация вариантов использования.....	41
Приложение В. Листинг запроса.....	58
Приложение Г. Скриншоты разработанной системы.....	59

ВВЕДЕНИЕ

Актуальность

Сегодня в розничной торговле и производстве продуктов питания обеспечение качества и безопасности продукции играет ключевую роль. Отслеживание сроков годности становится все более важным аспектом этого процесса, при этом традиционные методы контроля уступают место автоматизированным системам. Эти системы не только оптимизируют процессы контроля, но и сокращают риски продажи просроченной продукции, повышая операционную эффективность и обеспечивая высокий стандарт качества и безопасности.

Разработка и внедрение систем отслеживания сроков годности становятся неотъемлемой частью современного бизнеса. Такие системы не только повышают эффективность работы предприятий, но и предотвращают продажу просроченной продукции, что важно для поддержания репутации компании и обеспечения ее конкурентоспособности на рынке.

Таким образом, разработка системы для отслеживания сроков годности продукции становится актуальной задачей.

Постановка задачи

Командой разработчиков реализуется система отслеживания сроков годности продукции в виде веб-приложения. Система разделена на две части: клиентскую и серверную. В рамках выпускной квалификационной работы выполняется разработка клиентской части системы.

Целью выпускной квалификационной работы является разработка клиентской части системы отслеживания срока годности продукции. Для достижения поставленной цели необходимо решить следующие задачи:

- 1) выполнить анализ предметной области, описать требования к системе отслеживания срока годности продукции;
- 2) спроектировать клиентскую часть системы;
- 3) выполнить реализацию клиентской части системы;
- 4) выполнить тестирование клиентской части системы.

Структура и содержание работы

Работа состоит из введения, четырех глав, заключения, списка литературы и приложений. Объем работы составляет 67 страниц, объем списка литературы – 26 источников.

В первой главе описывается анализ предметной области. В ней проводится обзор аналогов и существующих решений для реализации веб-системы.

Вторая глава посвящена проектированию. В ней описываются требования к системе, варианты использования, архитектура системы и проектирование интерфейса.

В третьей главе приводится реализация системы. В ней рассказывается о реализации интерфейса веб-системы, его работе и взаимодействии с API.

В четвертой главе описывается тестирование системы. В ней приводится протокол функционального тестирования интерфейса приложения.

В приложении А представлена диаграмма вариантов использования веб-системы.

В приложении Б представлена спецификация вариантов использования веб-системы.

В приложении В представлен листинг запроса на получение информации о продукции.

В приложении Г представлены скриншоты веб-системы.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Анализ аналогичных проектов

Проекты, включающие в себя функционал отслеживания сроков годности продукции: Контур.Маркет, S-Mobile, CloudShop.

Контур.Маркет [1] – это веб-сервис и набор программ для учета товаров и организации работы мест общественного питания. С помощью этой системы можно вести учет товаров, контролировать продажи, а также обеспечивать соблюдение законодательства в сфере розничной торговли. Контур.Маркет обладает различными функциями для руководителей, кассиров и персонала по работе с товарами (рисунок 1).

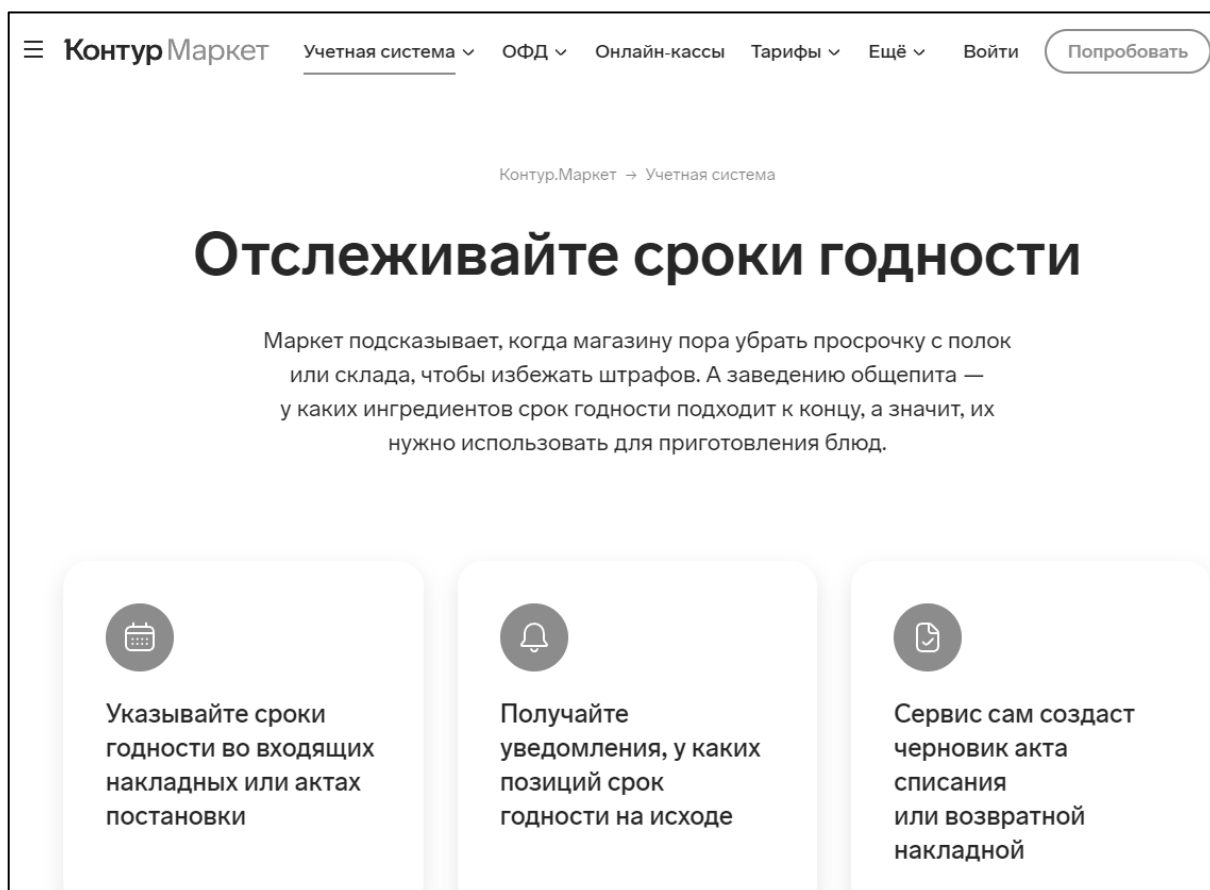


Рисунок 1 – Сайт сервиса отслеживания сроков годности продукции в Контур.Маркет

S-Mobile [2] – это мобильное приложение, которое значительно упрощает процессы приемки товаров и инвентаризации в магазинах. Оно позволяет сократить время выполнения этих задач и снизить риск возникновения

ошибок в учете. Приложение автоматически формирует необходимые документы для корректного ведения учета, что помогает улучшить эффективность работы бизнеса и избежать расхождений при переносе данных из первичных документов в систему товаручета (рисунок 2).

[Автоматизация торговли](#) [Сервис](#) [Блог](#) [Оборудование](#)

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ УЧЕТА ТОВАРОВ В МАГАЗИНЕ "S-MOBILE" ИСКЛЮЧИТ ОШИБКИ УЧЕТА, ПРОСТОЙ МАГАЗИНА И ПОТЕРИ ТОВАРА

Позвольте вашему товароведу видеть реальные остатки в реальном времени для абсолютно точных закупок

- Максимизируем доход
- Снижаем риски неэффективных закупок
- Гарантируем сохранность товаров
- Улучшаем большинство торговых процессов
- Снимаем нагрузку на торговый персонал
- Освобождаем время и силы руководства для занятия выгодной коммерцией

☰ +7 (999) 999-99-99

Попробуйте демо бесплатно

Нажимая на кнопку, вы даете согласие на обработку своих персональных данных. [Политика обработки персональных данных](#)




Рисунок 2 – Сайт мобильного приложения S-Mobile

CloudShop [3] – это веб-приложение для учета продаж в розничной торговле с возможностью онлайн-доступа. Оно предоставляет основные операции, включая закупку, продажу и возврат товаров, а также обеспечивает удобную загрузку номенклатуры из таблиц и импорт/экспорт данных о поставщиках и покупателях, предоставляет возможность управления складскими запасами и автоматического формирования отчетов о продажах (рисунок 3).

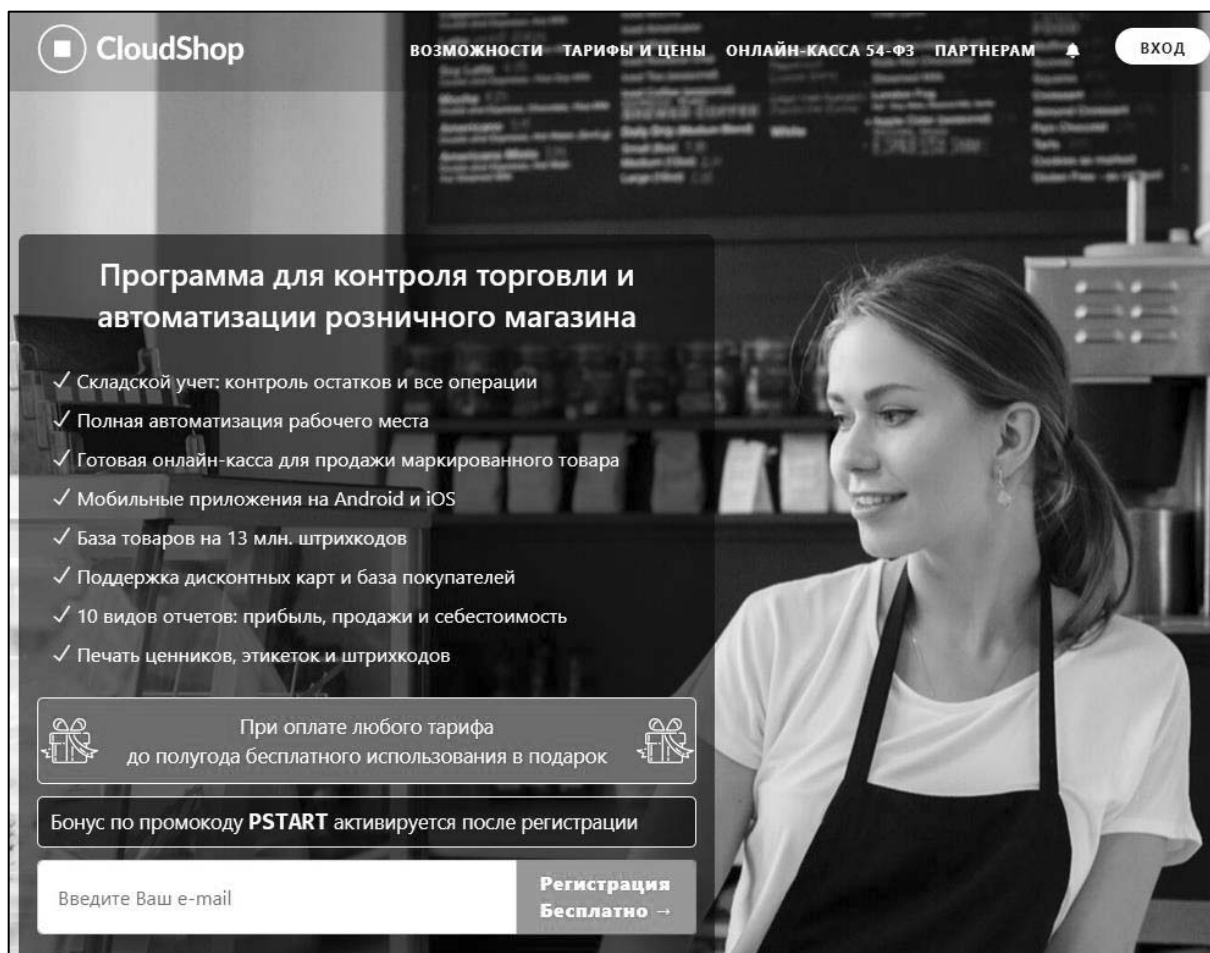


Рисунок 3 – Сайт веб-приложения CloudShop

В таблице 1 представлены характеристики аналогов системы.

Таблица 1 – Структура JSON данных

Функция	Контур.Маркет	S-Mobile	CloudShop
Сканирование кода товара сканом	Нет	Да	Да
Сканирование кода товара камерой	Нет	Нет	Нет
Возможность отдельного использования функционального модуля по отслеживанию срока годности продукции	Нет	Нет	Нет
Сканирование штрих-кода	Нет	Да	Да
Сканирование QR-кода	Нет	Нет	Нет

Все перечисленные сервисы являются частями обширных систем обслуживания предприятий. Отдельно приобрести нужные сервисы от CloudShop и Контур.Маркет невозможно; оплачивать полные системы с функциями, не нужными бизнесу, невыгодно. Многие компании в России

уже установили рабочие процессы и используют соответствующее программное обеспечение, но хотят решить вопрос автоматизации отслеживания срока годности продукции, приобретая соответствующее ПО, направленное исключительно на эту задачу. Кроме того, эти сервисы не обладают универсальностью сканирования QR-кодов и штрихкодов. Некоторые предприятия не пользуются сканерами, а предпочитают использовать камеры телефонов для сканирования QR-кодов, такой функционал в данных системах отсутствует.

1.2. Анализ существующих решений для реализации проекта

Для разработки современных веб-систем зачастую используются следующие технологии.

React [4] – это популярный JavaScript-фреймворк, который используется для разработки пользовательских интерфейсов веб-приложений. Он обладает модульной структурой и позволяет создавать компоненты, которые могут эффективно взаимодействовать между собой. React используется для создания интерактивных пользовательских интерфейсов с использованием компонентного подхода, что обеспечивает высокую скорость работы приложения и удобство его разработки и поддержки.

JavaScript [5] – это широко используемый язык программирования, который применяется для создания интерактивных элементов на веб-страницах. Он поддерживается всеми современными браузерами и позволяет добавлять динамическое поведение к веб-приложениям. JavaScript используется в различных областях веб-разработки, включая создание клиентских и серверных приложений, а также для разработки игр и мобильных приложений.

HTML [6] и CSS [7] – это основные технологии, используемые для создания структуры и внешнего вида веб-страниц. HTML используется для

определения структуры содержимого веб-страницы, в то время как CSS применяется для задания стилей и оформления элементов на странице. Вместе они обеспечивают создание красивых и функциональных веб-интерфейсов.

JSON (JavaScript Object Notation) [8] – это легкий формат обмена данными, часто используемый в веб-разработке, основанный на языке JavaScript. Он прост в чтении и записи как людьми, так и компьютерами, что делает его удобным для передачи данных между клиентом и сервером в веб-приложениях.

Вывод по первой главе

При анализе современных технологий веб-разработки были выделены ключевые инструменты: React, JavaScript, HTML, CSS и JSON. React, созданный Facebook, широко используется для создания динамичных пользовательских интерфейсов. JavaScript обеспечивает интерактивность веб-страниц, тогда как HTML и CSS формируют их структуру и внешний вид. JSON служит для обмена данными между клиентом и сервером.

Исследование аналогичных технологий показало, что многофункциональные платформы часто предлагают обширный набор возможностей. Однако они нередко не позволяют пользователям выбирать и оплачивать только нужные функции, что приводит к избыточности и усложнению использования. Пользователи вынуждены изучать и оплачивать ненужные им функции, что увеличивает сложность и стоимость разработки.

Дополнительные сложности могут возникнуть из-за уникальных методов и подходов, используемых этими платформами, что требует времени для освоения. Эти платформы могут иметь свои системы управления зависимостями и методы развертывания, отличающиеся от стандартов индустрии. Это создает трудности для разработчиков, привыкших к более гибким и модульным инструментам.

2. ПРОЕКТИРОВАНИЕ

2.1. Требования к системе

Анализ предметной области позволил выделить следующие функциональные и нефункциональные требования.

Функциональные требования

На основе проведенного анализа были выделены функциональные требования, определяющие ключевые функции и сервисы, которые должна обеспечивать система для удовлетворения потребностей пользователей и достижения целей проекта.

Были выделены функциональные требования.

1. Система не должна предоставлять доступ неавторизованным пользователям. Авторизация должна производиться по логину и паролю.
2. Система должна позволять пользователям сканировать штрих-коды и QR-коды сканером.
3. Система должна позволять пользователям сканировать штрих-коды и QR-коды камерой.
4. Система должна позволять пользователям вносить данные о продукции и ее сроке годности.
5. Система должна позволять пользователям создавать цеха, на которых производится собственная продукция предприятия.
6. Система должна позволять пользователям создавать тип продукции, который производится на предприятии или поступает на него.
7. Система должна обеспечивать возможность пользователям просматривать список всей продукции, которая используется на предприятии.
8. Система должна обеспечивать возможность пользователям просматривать данные о всей зарегистрированной продукции, которая поступила.
9. Система должна позволять пользователям просматривать отслеживаемую продукцию с информацией о ее сроке годности.

10. Система должна позволять пользователям удалять продукцию из списка отслеживаемой продукции после того, как ее убрали с прилавка.

11. Система должна позволять пользователям фильтровать список отслеживаемой продукции по цехам.

12. Система должна позволять пользователям фильтровать список зарегистрированной продукции по цехам.

13. Система должна позволять пользователям экспортировать список зарегистрированной продукции по цехам для печати.

14. Система должна позволять пользователям создавать аккаунты в веб-приложении.

15. Система должна позволять пользователям регистрироваться в веб-приложении.

16. Система должна позволять пользователям авторизоваться в веб-приложении.

17. Система должна позволять пользователям выходить из учетной записи веб-приложения.

18. Система должна позволять пользователям управлять ролями аккаунтов веб-приложения.

19. Система должна позволять пользователям создавать организации в веб-приложении.

20. Система должна позволять пользователям управлять организациями в веб-приложении.

21. Система должна позволять пользователям блокировать аккаунты в веб-приложении.

22. Система должна позволять пользователям разблокировать аккаунты в веб-приложении.

Нефункциональные требования

На основе проведенного анализа были выделены нефункциональные требования, которые определяют качественные характеристики и ограничения системы.

Были выделены нефункциональные требования.

1. Система должна обеспечивать безопасность пользовательских данных, а также использовать HTTPS [9] для защищенной передачи данных.
2. Система должна быть доступной постоянно, чтобы пользователи могли иметь доступ к ней в любое время.
3. Система должна быть полностью протестирована, чтобы обеспечить стабильную работу при ее обновлениях.

2.2. Варианты использования системы

Для проектирования приложения был использован язык графического описания для объектного моделирования UML [10]. По составленному списку требований к проектируемой системе была построена модель взаимодействия актеров с программной системой в виде диаграммы вариантов использования, представленной в приложении А.

Были выделены 4 актера, возможные варианты использования для них представлены ниже.

1. Неавторизованный пользователь – это человек или программа, которые не прошли процесс аутентификации и не имеют доступа к системе.
2. Работник – это пользователь, выполняющий конкретные обязанности на предприятии, такие как кладовщик, повар или продавец.
3. Менеджер – это пользователь, ответственный за управление операциями, управление персоналом и обеспечение эффективной работы предприятия в целом.
4. Администратор – это пользователь, обладающий привилегиями для управления настройками, доступом, безопасностью и другими аспектами функционирования приложения.

Неавторизованный пользователь может выполнить авторизацию в приложении (войти в аккаунт для получения прав пользователя).

Работник имеет ограниченный доступ к функционалу системы и может выполнять в системе следующие действия:

- 1) сканирование сканом – сканирование штрихкодов и QR кодов с помощью устройств со сканером;
- 2) сканирование камерой – сканирование штрихкодов и QR кодов с помощью камеры смартфона или планшета;
- 3) регистрация поступившей продукции – оценка органолептических характеристик продукта (по шкале от 1 до 5), ввод веса или количества продукции, срока годности и температуры;
- 4) просмотр списка отслеживаемой продукции – просмотр списка зарегистрированной продукции, находящейся на прилавках магазина, для получения информации об истечении срока ее годности;
- 5) удаление отслеживаемой продукции – процесс исключения из списка отслеживаемой продукции той продукции, которая либо просрочена и не пригодна для продажи, либо уже была продана;
- 6) фильтрация по цехам – функция, позволяющая фильтровать продукцию по цехам, в которых она была произведена или к которым она относится.

Менеджер, помимо функционала работника, имеет доступ к следующим функциям:

- 1) просмотр списка цехов – просмотр списка названий цехов, на которых предприятие изготавливает продукцию или из которых ее получает;
- 2) создание цехов – возможность добавления нового цеха, из которого поступает продукция;
- 3) просмотр списка зарегистрированной продукции – просмотр списка с данными о всей продукции, которая поступала в продажу;
- 4) экспорт списка зарегистрированной продукции – возможность экспортировать данные о зарегистрированной продукции в файл для печати;
- 5) просмотр списка типов продукции предприятия – просмотр списка всех типов продаваемой предприятием продукции;

6) создание типа продукции предприятия – добавление нового типа продаваемой продукции с указанием его названия, цеха, на котором он произведен и кода для сканирования;

7) просмотр списка настроек – просмотр списка управляемых параметров;

8) изменение времени срабатывания звуковых оповещений – изменение настройки времени срабатывания звуковых уведомлений до момента истечения срока годности продукции;

9) включение и отключение звуковых уведомлений – изменение параметра настройки включения или отключения звуковых уведомлений;

10) фильтрация по дате – функция, позволяющая фильтровать продукцию по дате регистрации продукции;

11) изменение пароля – функция изменения пароля текущего аккаунта в настройках с вводом текущего и нового пароля.

Администратор может выполнять в системе следующие действия:

1) просмотр списка аккаунтов – просмотр списка аккаунтов, зарегистрированных в системе;

2) фильтрация списка аккаунтов – возможность фильтровать список аккаунтов по заблокированным и незаблокированным;

3) создание аккаунтов – возможность зарегистрировать новый аккаунт с указанием ФИО работника, логина, пароля, организации и роли;

4) редактирование аккаунтов – возможность изменить ФИО работника и роль;

5) блокировка аккаунтов – возможность заблокировать аккаунт, тем самым ограничив ему доступ к системе;

6) разблокировка аккаунтов – возможность разблокировать аккаунт, тем самым вернув ему доступ к системе;

7) просмотр списка ролей – просмотр списка ролей в системе;

8) создание ролей – возможность добавить новую роль в системе, указав наименование и доступы, которые будет иметь аккаунт с этой ролью;

9) редактирование ролей – возможность изменить наименование и доступы, которые будет иметь аккаунт с этой ролью;

10) просмотр списка организаций – просмотр списка организаций в системе;

11) создание организаций – возможность добавить новую организацию в системе, указав наименование.

Спецификация вариантов использования представлена в приложении Б.

2.3. Проектирование интерфейса

Разработан дизайн страниц системы. Макет страницы отслеживаемой продукции представлен на рисунке 4.

Выберите цех								
Все цеха	Горячий	Кондитерский	Мясной	Продукция поставщиков	Рыбный	Хлебобулочный	Холодный	С...
11:00								
<input checked="" type="checkbox"/> Просрочен (2)	<input type="checkbox"/> Менее 15 минут до истечения срока годности	<input type="checkbox"/> Свежий	Удалить все просроченные					
СОУС Количество: 1		Создано: 28.08.2023 17:44 Истекает: 30.08.2023 17:44	Убрать с прилавка					
САЛАТ ОЛИВЬЕ Вес: 0.31 кг		Создано: 28.08.2023 17:44 Истекает: 30.08.2023 17:44	Убрать с прилавка					
МАНТЫ Вес: 0.31 кг		Создано: 28.08.2023 17:44 Истекает: 30.08.2023 17:44	Убрать с прилавка					
САЛАТ ОЛИВЬЕ Вес: 0.31 кг		Создано: 28.08.2023 17:44 Истекает: 30.08.2023 17:44	Убрать с прилавка					

Рисунок 4 – Макет страницы отслеживаемой продукции

Сверху страницы представлена карусель фильтров по цехам. Ниже показывается текущее время и напоминания о цветовых обозначениях продукции в списке, а также кнопка для удаления из списка всех просроченных

товаров. Всю оставшуюся часть страницы занимает список отслеживаемой продукции, показывающий состояния товаров на витринах и информацию о них, а также содержащий кнопку для удаления определенных товаров из списка.

Часть дизайн-системы страницы заполнения органолептических характеристик представлен на рисунке 5.

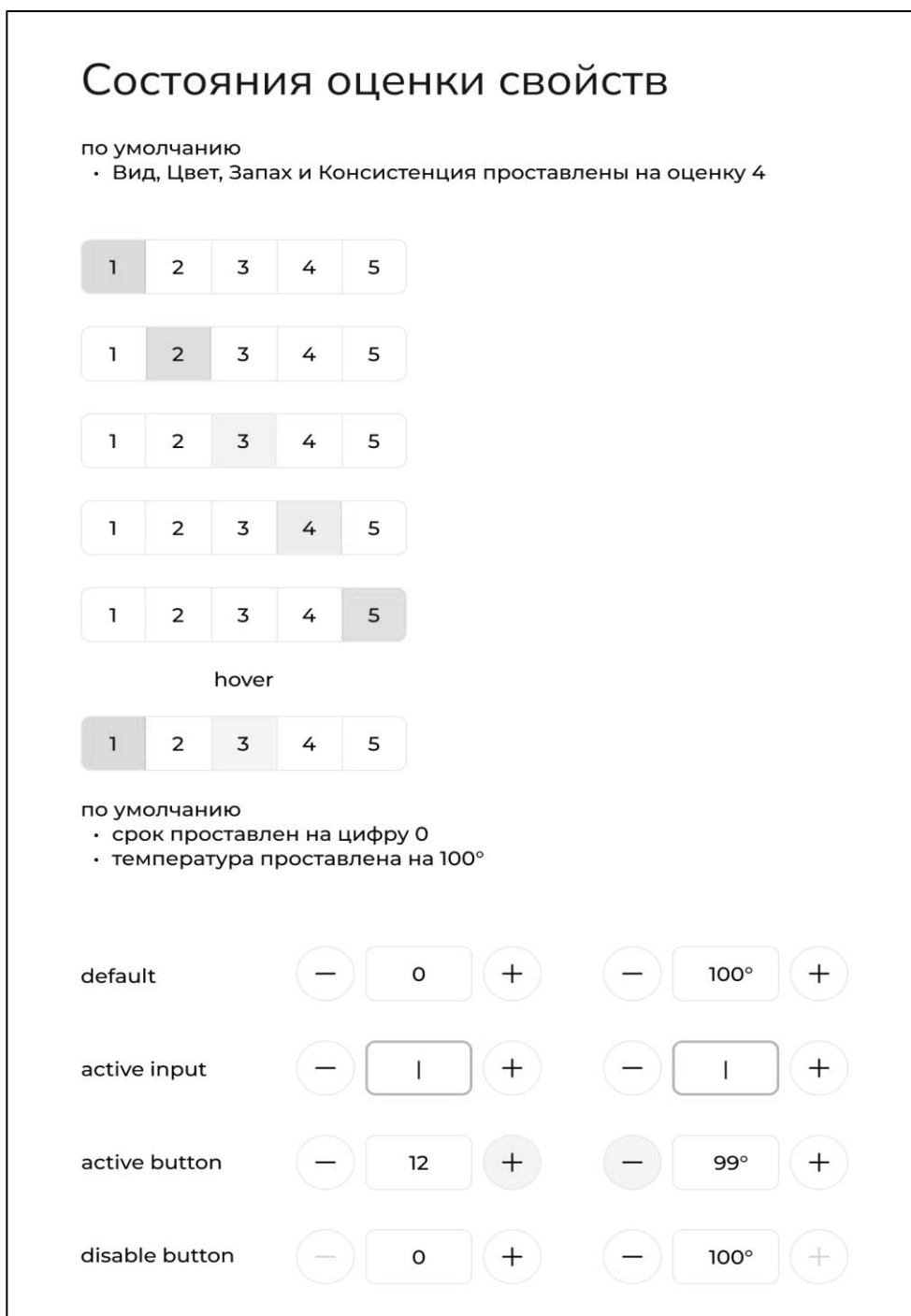


Рисунок 5 – Часть дизайн-системы страницы заполнения органолептических характеристик

В данной дизайн-системе представлена информация о состояниях оценки свойств товаров и полей для ввода их температуры и срока годности. Дизайн боковой навигационной панели представлен на рисунке 6.

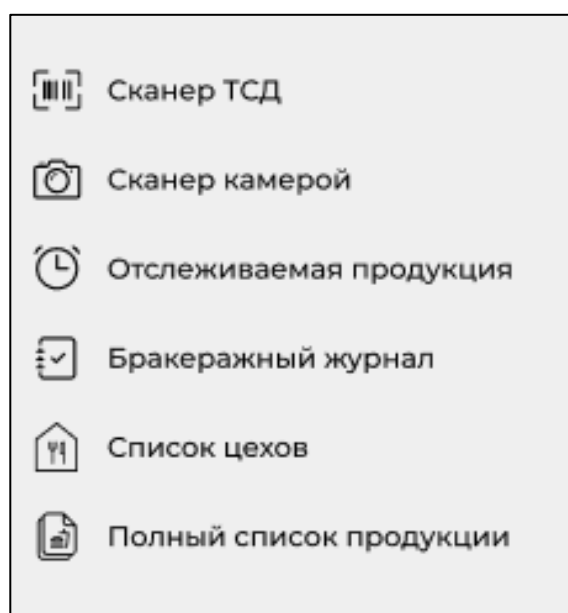


Рисунок 6 – Дизайн боковой навигационной панели

С помощью данной боковой навигационной панели пользователь может перемещаться по таким страницам как:

- 1) страница для сканирования через ТСД;
- 2) страница для сканирования камерой;
- 3) страница отслеживаемой продукции;
- 4) страница зарегистрированной продукции (бракеражный журнал);
- 5) страница списка цехов;
- 6) страница полного списка продукции.

2.4. Архитектура системы

Система представляет собой клиент-серверное приложение, состоящее из следующих частей: веб-интерфейса, базы данных и веб приложения, поделенных на микро-сервисы, такие как основная веб-система, сервис аккаунт-менеджмента и сервис аутентификации. Для представления архитектуры системы была построена диаграмма компонентов (рисунок 7).

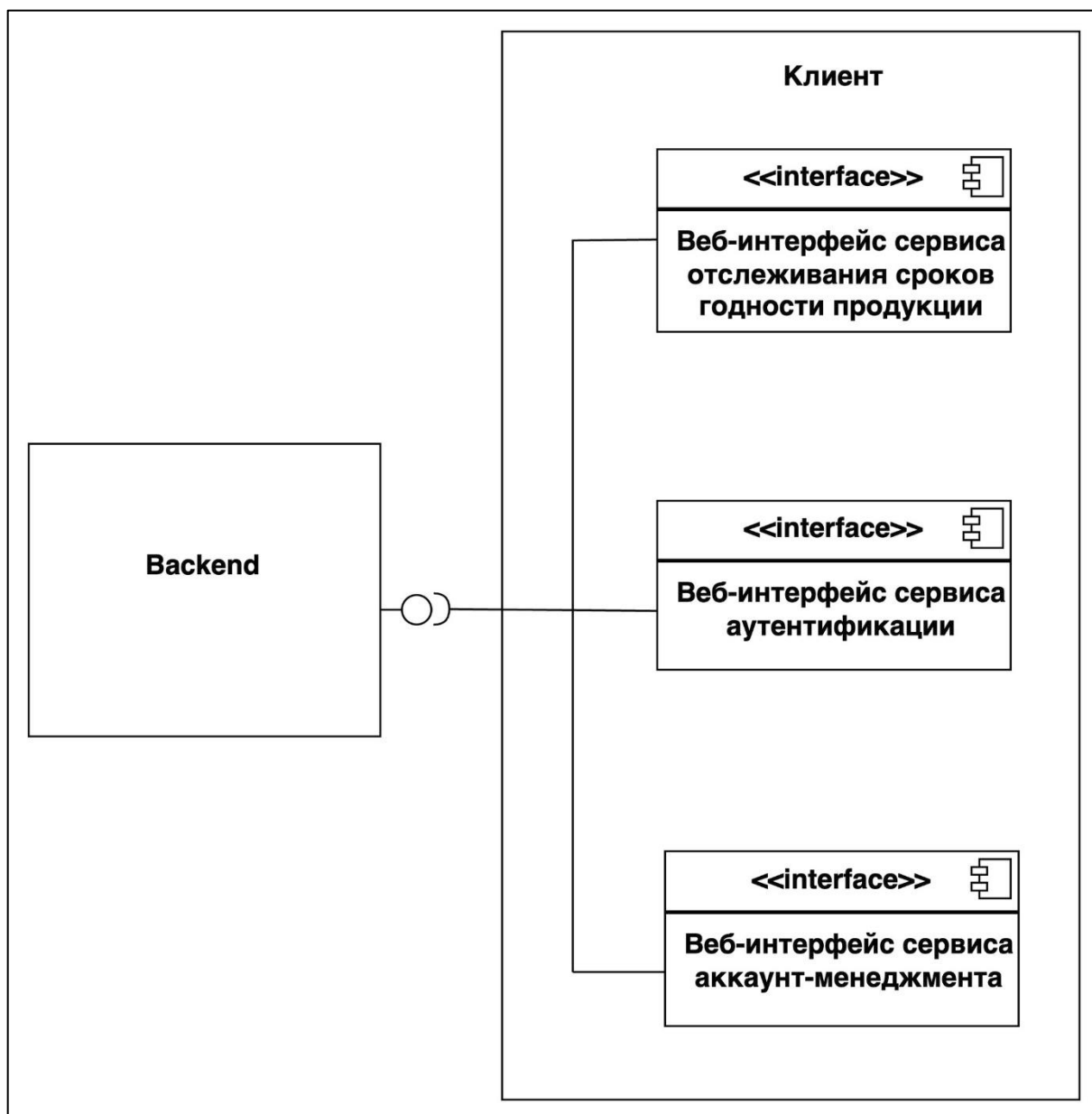


Рисунок 7 – Диаграмма компонентов

На диаграмме представлены компоненты системы.

1. Веб-интерфейс сервиса отслеживания сроков годности продукции – интерфейс системы, позволяющий внешним клиентам взаимодействовать с веб-системой, поддерживает возможность просмотра данных, их фильтрацию, изменение, добавление, вход и выход из аккаунта, загрузку и скачивание файлов.

2. Веб-интерфейс сервиса аутентификации – интерфейс системы, позволяющий внешним клиентам взаимодействовать с веб-системой через

интернет-браузер. Дает возможность пользователям, имеющим аккаунт в системе получить к ней доступ путем аутентификации.

3. Веб-интерфейс сервиса аккаунт-менеджмента – интерфейс системы, позволяющий внешним клиентам взаимодействовать с веб-системой через интернет-браузер.

4. Backend– API-сервис веб-системы отслеживания сроков годности продукции, отвечающий за обработку данных, поступающих с клиентской части, хранение данных и реализацию функционала.

2.5. Структура JSON данных

Разрабатываемая клиентская часть системы отслеживания срока годности продукции получает данные от серверной части в виде JSON-структур и не имеет прямого доступа к базам данных системы.

В таблице 2 приведены основные JSON данные сервиса аутентификации, которые будут использоваться для реализации веб-интерфейса.

Таблица 2 – Структура JSON данных сервиса аутентификации

Наименование	Тип	Описание
login	string	логин
tenantid	number	идентификатор организации
accountid	number	идентификатор аккаунта
currentpassword	string	текущий пароль
newpassword	string	новый пароль

В таблице 3 приведены основные JSON данные сервиса управления аккаунтами, которые будут использоваться для реализации веб-интерфейса.

Таблица 3 – Структура JSON данных сервиса управления аккаунтами

Наименование	Тип	Описание
firstname	string	имя
lastname	string	фамилия
middlename	string	отчество
isBlocked	boolean	флаг блокировки аккаунта
tenantName	string	Название организации
roles	array	роли
permissions	array of string	разрешения

В таблице 4 приведена основная часть JSON данных сервиса отслеживания сроков годности продукции, которые будут использоваться для реализации веб-интерфейса.

Таблица 4 – Структура JSON данных сервиса отслеживания сроков годности продукции

Наименование	Тип	Описание
id	number	идентификатор
name	string	наименование
shopName	string	наименование цеха
isPiece	boolean	штучность продукции
barcodes	array	штрих-коды продукции
productId	number	идентификатор продукции
lifetimeInMinutes	number	срок годности в минутах
barcode	string	штрих-код продукции
trackedProductCount	number	вес продукции
temperature	object	Органолептические свойства продукции
organolepticProps	string	штрих-код продукции
organolepticProps: color	number	цвет
organolepticProps: appearance	number	вид
organolepticProps: smell	number	запах
organolepticProps: consistency	number	консистенция

Вывод по второй главе

На основе рассмотрения аналогичных систем с функционалом отслеживания сроков годности продукции и технологий разработки, были проработаны функциональные и нефункциональные требования к клиентской части системы, подготовлен дизайн интерфейса системы отслеживания срока годности продукции, рассмотрены варианты использования системы. Разработана архитектура веб-системы, определены компоненты системы, их взаимодействие и функционал. Спроектирована структура JSON данных, которая обеспечивает гибкий обмен информацией и адаптацию к новым требованиям.

3. РЕАЛИЗАЦИЯ

3.1. Средства реализации

Разработка системы была реализована с помощью следующих технологий.

1. Visual Studio Code 1.84.2 [11] – редактор кода, предоставляющий широкие возможности для разработки программного обеспечения.
2. React 18.2.0 [12] – библиотека JavaScript для разработки пользовательских интерфейсов, использующая компонентный подход.
3. Node.js 18.17.1 [13] – среда выполнения JavaScript, используемая для запуска серверного кода.
4. MobX 6.9.0 [14] – библиотека для управления состоянием приложения в React.
5. Axios 1.4.0 [15] – библиотека для выполнения HTTP-запросов из JavaScript.
6. Clsx 1.2.1 [16] – утилита для генерации классов для React-компонентов на основе условий.
7. Lodash 4.17.20 [17] – утилиты JavaScript для работы с массивами, объектами и функциями.
8. Moment 2.29.4 [18] – библиотека для удобной работы с датами и временем в JavaScript.
9. React-table 7.8.0 [19] – библиотека для создания гибких таблиц в React.
10. Vite 3.2.0 [20] – плагин для преобразования SVG в React-компоненты.
11. ZXing 0.20.0 [21] – библиотека для работы с штрихкодами и QR-кодами в JavaScript, основана на классических алгоритмах компьютерного зрения и обработки изображений.
12. HTML [22] – стандартный язык разметки, используемый для создания и структурирования веб-страниц.

13. Sass [23] – метаязык CSS с расширенным функционалом, включающим переменные, вложенные правила и другие улучшения.

14. TypeScript [24] – язык программирования, расширяющий возможности JavaScript путем добавления статической типизации.

3.2. Реализация аутентификации

JWT (JSON Web Token) – это компактный и самодостаточный способ представления информации между двумя сторонами в виде объекта JSON. Он используется для передачи утверждений (claims) между клиентом и сервером. JWT состоит из трех частей: заголовка (header), полезной нагрузки (payload) и подписи (signature).

JWT используется в механизмах аутентификации и авторизации веб-приложений. Работа с JWT состоит из нескольких этапов.

1. Получение JWT. Когда пользователь успешно аутентифицируется на сервере (например, введя правильные учетные данные), сервер создает JWT для этого пользователя. Этот токен возвращается в ответе сервера после успешной аутентификации.

2. Хранение JWT. Полученный JWT сохраняется на клиентской стороне для последующего использования. Хранится в локальном хранилище (localStorage) или в куки (cookies).

3. Включение JWT в каждый запрос к серверу. После того, как JWT получен и сохранен, он должен быть включен в каждый запрос к серверу. Это делается путем добавления заголовка Authorization с значением Bearer <JWT> в каждый HTTP запрос.

4. Проверка JWT на сервере. При получении запроса на сервере, сервер должен проверить переданный JWT. Это включает в себя проверку подписи токена, а также проверку его срока действия.

5. Обработка и обновление JWT. JWT имеет срок действия, после истечения которого он становится недействительным. При необходимости

сервер может вернуть новый JWT в ответе на запрос пользователя (например, если текущий JWT истек или его требуется обновить по другим причинам), и клиент должен обновить сохраненный JWT на новый.

6. Выход из аккаунта и удаление JWT. При выходе пользователя из системы (logout) или при истечении срока действия токена, JWT должен быть удален из хранилища на клиентской стороне для безопасности.

Пример работы с токеном представлен в листинге 1.

Листинг 1 – Пример работы с токеном

```
function setLoggedOut() {
  tokenProvider.setTokenPair(null, null);
  function getAuthToken() {
    return tokenStorage.getTokenValue();
  }
  function getRefreshToken() {
    return refreshTokenStorage.getTokenValue();
  }
}
```

В данном примере отображены удаление токена, его получение и обновление.

3.3. Взаимодействие с серверной частью

Для реализации клиентской части веб-системы сервером предоставляется API системы. Запрос к серверу формируется в виде объекта формата JSON и отправляется асинхронно с помощью ключевого слова `await`, чтобы программа дождалась завершения запроса перед продолжением выполнения других действий в функции. Была использована библиотека `Axios`, предоставляющая удобный интерфейс для отправки запросов. Пример запроса представлен в листинге 2.

Листинг 2 – Пример запроса к серверной части

```
async function loadProductsAsync () {
  try {
    const {
      data,
    } = await api.get(
      loadProductsUrl,);
    setProducts(data);
  } catch (error: any) {
    errorBoundary.setErrorInfo({
      code: error.response.status, });
  }
}
```

Приведенный код реализует запрос к серверной части для получения списка продукции.

Общий вид запроса к серверу представлен в листинге 3.

Листинг 3 – Запрос на получение информации о продукции и ответ сервера

```
GET/POST Метод API
header: токен
RESPONSE: JSON-объект
```

Методы API

Все методы поделены на следующие блоки:

- 1) `products` – методы блока «продукция»;
- 2) `products log` – методы блока «зарегистрированная продукция»;
- 3) `shops` – методы блока «цеха»;
- 4) `tracked products` – методы блока «отслеживаемая продукция»;
- 5) `accounts` – методы блока «аккаунты»;
- 6) `roles` – методы блока «роли»;
- 7) `tenants` – методы блока «организации».

Всего используется 4 метода блока `products`, 3 метода блока `products log`, 3 метода блока `shops`, 4 метода блока `tracked products`, 6 методов блока `accounts`, 5 методов блока `roles`, 3 метода блока `tenants`.

Представлены примеры некоторых запросов к серверу и ответов. В приложении В представлен листинг запроса на получение информации о продукции и ответ сервера.

На рисунке 8 представлен скриншот с данными этого запроса.

Наименование	Штучность	Код	Цех	Срок годности в минутах
Булочка с маком	штучный	4343435567	Хлебобулочный цех	720
Чеснок	весовой	3432434244	Овощной цех	1080
Оливье	весовой	6575765678	Салатный цех	720
Булочка с повидлом	штучный	1234567896	Хлебобулочный цех	1080
Лук	весовой	2132343556	Овощной цех	7200
Хлеб "Карталинский"	штучный	4535345433	Хлебобулочный цех	1440
Конфеты "Славянка"	весовой	2324545344	Кондитерский цех	7200
Перец	весовой	4543534545	Овощной цех	5040

Рисунок 8 – Страница добавления новой продукции

В листинге 4 представлен запрос на добавление продукции и ответ сервера.

Листинг 4 – Запрос на регистрацию продукции и ответ сервера

```
POST /tracked-products
{
  header: JSON Web Token,
  body:{
    "productId": 6398,
    "lifetimeInMinutes": 720,
    "barcode": "1234567896",
    "trackedProductCount": 10,
    "temperature": 100,
    "organolepticProps": {
      "color": 3,
      "appearance": 2,
      "smell": 5,
      "consistency": 4}}
RESPONSE:{"id":6}
```

На рисунке 9 представлен скриншот с данными этого запроса.

19:09

Булочка с маком

Вид	1	2	3	4	5
Цвет	1	2	3	4	5
Запах	1	2	3	4	5
Консистенция	1	2	3	4	5
Количество	10				
Срок годности в минутах:	−	720	+		
− Скрыть температуру					
Температура в градусах цельсия:	−	100	+		

Сохранить

Рисунок 9 – Страница регистрации продукции

В заголовке также передается токен клиента. Тело запроса содержит информацию об id типа регистрируемой продукции, ее сроке годности, коде, температуре, количестве и органолептических свойствах, таких как запах, цвет, консистенция и вид.

3.4. Информационная архитектура

Для наглядного представления страничной структуры приложения была составлена диаграмма, изображенная на рисунке 10.

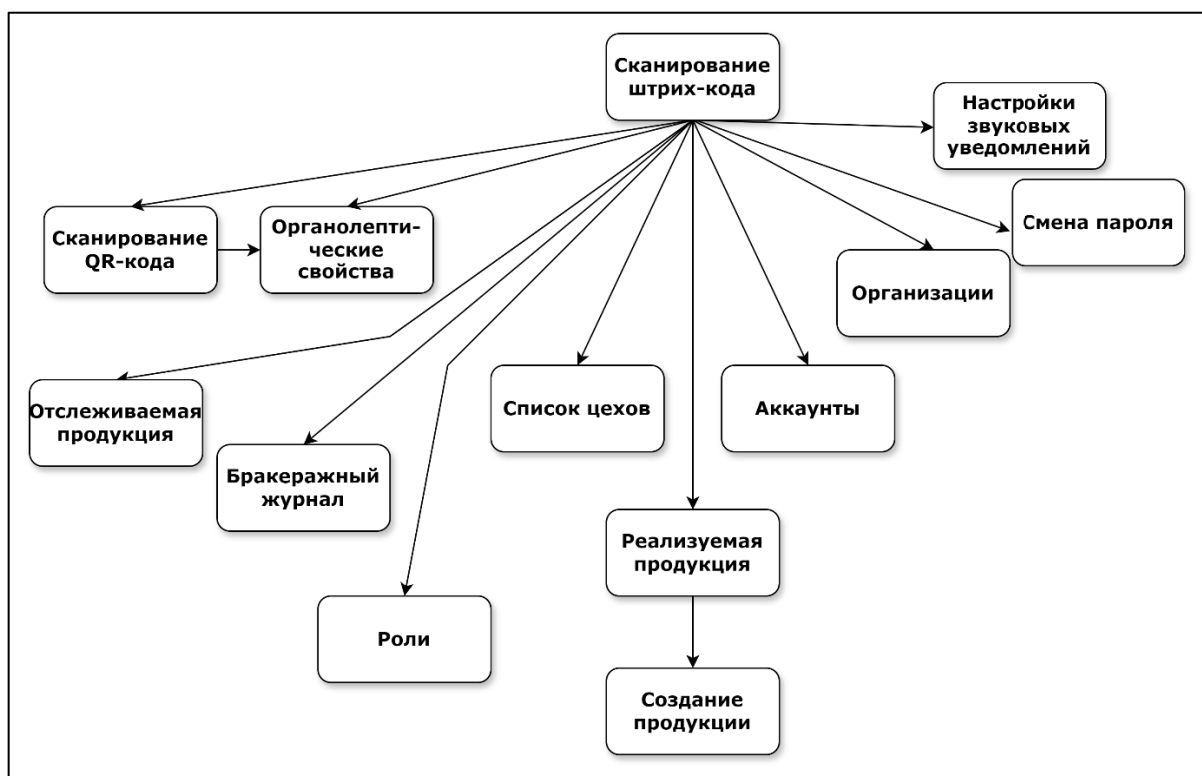


Рисунок 10 – Информационная архитектура

В информационной структуре представлены такие страницы как:

- 1) сканирование штрих-кода – страница для сканирования штрих-кода сканнером;
- 2) сканирование QR-кода – страница для сканирования QR-кода камерой;
- 3) смена пароля – страница, отображающая форму смены пароля текущего аккаунта;

- 4) отслеживаемая продукция – страница для отслеживания зарегистрированной продукции, предоставляющая информацию о свежей, испорченной продукции и продукции, у которой срок годности скоро истечет;
- 5) бракеражный журнал – страница, отображающая всю информацию о зарегистрированной когда-либо продукции, предоставляющая возможность фильтровать ее по цехам, периоду и экспортировать в файл Excel;
- 6) список цехов – страница, отображающая информацию о цехах или поставщиках продукции, с возможностью создания нового цеха;
- 7) реализуемая продукция – страница, отображающая информацию о всей реализуемой продукции предприятия, позволяющая фильтровать ее по цехам и создавать новую;
- 8) создание продукции – страница, позволяющая создать новый тип реализуемой продукции;
- 9) аккаунты – страница, отображающая информацию об аккаунтах, зарегистрированных в системе, с возможностью блокировки, редактирования и создания нового аккаунта;
- 10) организации – страница, отображающая информацию об организациях, зарегистрированных в системе, с возможностью создавать новые организации;
- 11) роли – страница, отображающая информацию о ролях, доступных в системе, с возможностью создать новые роли и задавать разрешения, доступные этим ролям;
- 12) настройки звуковых уведомлений – страница, отображающая и позволяющая редактировать настройки звуковых уведомлений, такие как включение или отключение уведомлений и время их срабатывания до истечения срока годности продукции;
- 13) органолептические свойства – страница для регистрации продукции, с возможностью задать оценки свойств продукции для таких параметров как запах, цвет, внешний вид и консистенция, температура и время хранения.

3.5. React разработка

Одностраничное приложение (SPA) загружает только одну HTML-страницу и обновляет ее содержимое динамически, что улучшает пользовательский опыт. В React для этого создаются компоненты, которые отображают различные части интерфейса в зависимости от состояния приложения. Перерисовка компонентов автоматически происходит при изменении состояния или пропсов. Разработка на React основана на создании компонентов, которые управляются состоянием (state) и пропсами (props), что позволяет легко управлять данными и обновлять интерфейс без перезагрузки страницы.

В листинге 5 представлен пример стейта для хранения информации о коде продукции.

Листинг 5 – Пример стейта для хранения информации о коде продукции

```
import { makeAutoObservable } from "mobx";

type BarcodeInfoType = {
  id: number;
  name: string;
  barcodes: Array<number>;
};

class BarcodeInfoState {
  private _barcodeInfo: Array<BarcodeInfoType> | null = null;
  private _barcodeNumber: string = ``;
  private _isOpen = false;
  private _productName: string = ``;
  constructor() {
    makeAutoObservable(this);
  }
  get isOpenModal() {
    return this._isOpen;
  }
  set productName(ProductName: string) {
    this._productName = ProductName;
  }
  get productName() {
    return this._productName;
  }
  setOpenModal(isOpen: boolean) {
    this._isOpen = isOpen;
  }
  get barcodeInfo() {
    return this._barcodeInfo;
  }
  get barcodeNumber() {
    return this._barcodeNumber;
  }
  get infoBarcode(info: Array<BarcodeInfoType> | null) {
    this._barcodeInfo = info;
  }
  updateBarcodeNumber(updatedNumber: string) {
    this._barcodeNumber = updatedNumber;
  }
}
export default BarcodeInfoState;
```

Данный пример кода представляет класс `BarcodeInfoState`, который используется для управления состоянием информации о штрих-кодах. В классе определены приватные поля `_barcodeInfo`, `_barcodeNumber`, `_isOpen` и `_productName`, представляющие информацию о штрих-кодах, номер штрих-кода, статус модального окна и имя продукта соответственно.

Класс также содержит методы для доступа к этим полям (getters) и для их обновления (setters). Метод `getInfoBarcode` используется для получения информации о штрих-кодах, а `updateBarcodeNumber` - для обновления номера штрих-кода.

Конструктор класса использует функцию `makeAutoObservable` из пакета `mobx` для автоматического создания геттеров и сеттеров для всех полей класса, что облегчает управление состоянием.

3.6. Адаптивный интерфейс

Для реализации адаптивного интерфейса использовались различные технологии и подходы:

- 1) HTML и CSS медиа-запросы, позволяющие настраивать стили в зависимости от характеристик устройства, таких как размер экрана или ориентация;
- 2) Flexbox и Grid Layout, предоставляющие удобные инструменты для управления расположением элементов на веб-странице, что помогает создать адаптивный макет;
- 3) Mobile-first подход, являющийся стратегией разработки, начинающей с оптимизации для мобильных устройств и затем улучшающей отображение для более крупных экранов;
- 4) разделение на компоненты, являющееся принципом разработки, предполагающий разбиение интерфейса на независимые компоненты для легкого масштабирования и повторного использования кода;

5) ВЕМ подход, являющийся методологией именования классов CSS, основанной на блоках, элементах и модификаторах, для создания понятной и поддерживаемой структуры стилей.

В листинге 6 представлен пример адаптивной верстки страницы типов продукции.

Листинг 6 – Пример адаптивной верстки страницы типов продукции

```
.all-products-list-page {
  $this: &;
  padding: 20px 20px 0;
  &__head {
    display: flex;
    flex-direction: column;
    justify-content: flex-end;
    margin-bottom: 30px;
  }
  &__button {
    margin-top: 20px;
    margin-left: auto;
    width: 120px;
    text-align: center;
    @include tablet {
      width: auto;}}}
```

Данный пример кода представляет собой пример адаптивной верстки компонента страницы типов продукции, к которой далее подключаются другие компоненты, такие как таблицы и фильтры. В этом примере используются Mobile-first подход, Flexbox и ВЕМ подход.

Скриншоты интерфейсов веб-системы представлены в приложении Г.

Вывод по третьей главе

Разработана веб-система для отслеживания сроков годности продукции с использованием Visual Studio Code, React, Node.js, MobX, Axios, Clsx, Lodash, Moment, React-table, Vite, ZXing, HTML, Sass и TypeScript. Обеспечена безопасная аутентификация с JWT и адаптивная верстка с Mobile-first, Flexbox и ВЕМ. Axios использован для передачи данных, а ZXing для сканирования штрих-кодов.

4. ТЕСТИРОВАНИЕ

Клиентская часть тестируется Unit, E2E и компонентными тестами в инструменте автоматизации внешнего тестирования Cypress [25].

Unit-тесты: для тестирования отдельных модулей или компонентов используются юнит-тесты. Целью этих тестов является проверка корректности работы отдельных функций, классов или модулей в изоляции от других частей приложения.

E2E-тесты: для проверки работы приложения на уровне пользовательского интерфейса используются E2E-тесты (end-to-end). Они имитируют действия пользователя в браузере и проверяют, что приложение ведет себя ожидаемым образом в реальных условиях.

Компонентные тесты: тесты, для проверки отдельных компонентов пользовательского интерфейса. Они проверяют, что компоненты правильно отображаются и взаимодействуют с внешними данными или событиями.

Разработка веб-системы проводилась с использованием метода TDD [26], который предполагает написание тестов перед написанием кода приложения. Это позволяет обеспечить полное покрытие кода тестами и улучшить его качество за счет раннего выявления ошибок.

Таким образом, вся клиентская часть веб-системы покрыта автоматизированными тестами, включая юнит-тесты для отдельных модулей, E2E-тесты для проверки работы приложения в целом, и, возможно, компонентные тесты для проверки отдельных компонентов пользовательского интерфейса.

Представлены примеры тестов. В листинге 7 представлен Unit-тест для тестирования работы компонента `productsState`.

Листинг 7 – Unit-тест для тестирования правильной инициализации продуктов в `productsState`

```
const INITIAL_DATA: ProductsInfo = {
  expiredTrackedProductsCount: 0,
  expiringTrackedProductCount: 0,
  list: [
    {
      id: 123456,
```

```

        name: `Салат`,
        createdAtUtc: `2023-12-05T10:00:00`,
        expiresAtUtc: 2023-12-05T14:00:00,
        minutesBeforeExpiration: 60,
        status: 0,
        weight: 0,
    },
],
];
it(`SHOULD return all products WHEN initialized`, () => {
    productsState.initialize({
        productsInfo: INITIAL_DATA,
    });
    expect(productsState.allProductsInfo.expiredTrackedProductsCount)
        .to.deep.eq(0);
    expect(productsState.allProductsInfo.list)
        .to.has.lengthOf(1);
    expect(productsState.allProductsInfo)
        .to.deep.eq(INITIAL_DATA);
}
)
;

```

Данный тест использует подготовленные данные для инициализации в компонент `productsState`. Далее проводится проверка данных, что в них нет просроченных продуктов, что загружены данные только об одном продукте и что загруженные данные полностью соответствуют предоставленным.

В листинге 8 представлен E2E-тест для тестирования страницы всей реализуемой продукции.

Листинг 8 – E2E-тест для тестирования страницы реализуемой продукции

```

describe(`Create a product and make sure it appears in the products table`,
() => {
    const code = generateProductCode();
    beforeEach(`Authorize`, () => {
        cy.authByApi();
        cy.clearProductsByApi();
    });

    after(`Clean`, () => {
        cy.clearProductsByApi();
    });

    it(`GIVEN allProducts page
        WHEN creating a product
        THEN make sure it's in the products table and remove it from the table`,
() => {
        AllProductsPage.visit();
        AllProductsPage.createNewProduct(code);
        AllProductsPage.checkNewProduct(code);
    });
});

```

Данный тест проверяет корректность работы всех функций страницы реализуемой продукции, такие как создание нового типа реализуемой продукции и отображение его в таблице.

В листинге 9 представлен компонентный тест для проверки компонента страницы всей продукции.

Листинг 9 – E2E-тест для тестирования страницы реализуемой продукции.

```
describe(`AllProductsListPage`, () => {  
  
  it(`SHOULD render all products list WHEN open page`, () => {  
    mountComponent();  
  
    cy.getByData(`all-products-list-page`)  
    .should(`exist`);  
  });  
  
  it(`SHOULD render no products message WHEN there are no products`, () => {  
    cy.intercept(`GET`, (`${API_ROOT}/products`), []).as(`call-1`);  
    mountComponent();  
  
    cy.getByData(`no-products-message`)  
    .contains(`Продукции нет`);  
  }  
});  
  
  it(`SHOULD render add button WHEN open page`, () => {  
  
    mountComponent();  
  
    cy.getByData(`all-products-list-page-button`)  
    .should(`exist`);  
  }  
});  
});
```

Данный тест проверяет корректность работы компонента страницы реализуемой продукции. Просматривается правильное отображение элементов страницы.

При изменениях кода запускаются все тесты для проверки работоспособности изменений. В таблице 5 приведена часть отчета по прохождению тестов.

Таблица 5 – Часть отчета по прохождению тестов

Тесты	Время	Выполнено	Не выполнено	Пропущено
BarcodeInfoState	44 ms	2	0	0
BarcodeScanner	10 ms	7	0	0
Clock	69 ms	2	0	0
ErrorModal	27 ms	2	0	0

Тесты	Время	Выполнено	Не выполнено	Пропущено
FilterMenu	178 ms	4	0	0
ProductsList	10 ms	18	0	0
AllProductsListPage	106 ms	3	0	0
CreateProductPage	20 ms	10	0	0
ProductsLogPage	20 ms	5	0	0
ShopsListPage	85 ms	7	0	0
AllProductsTable	110 ms	4	0	0
CharacteristicsList	165 ms	5	0	0
OrganolepticCharacteristics	143 ms	5	0	0
ShopsFilterContainer	59 ms	2	0	0
ProductsState	104 ms	10	0	0
SettingsAccountPage	10 ms	4	0	0
SettingsAlarmPage	769 ms	2	0	0
MobileControlsPanel	61 ms	1	0	0
ProductsLogExportButton	53 ms	2	0	0
ClockNumber	73 ms	3	0	0

На данный момент разработки написан 151 тест, суммарное время выполнения которых равно 21 секунде. В конце отчетов, если все тесты выполнены, приводится итоговая строка, изображенная на рисунке 11.

✓ All specs passed!	00:21	151	0	0
---------------------	-------	-----	---	---

Рисунок 11 – Итоговая строка отчета выполнения тестов

Вывод по четвертой главе

При разработке веб-системы с использованием метода TDD (Test-Driven Development) применялись юнит-тесты и E2E-тесты для проверки модулей и взаимодействия компонентов, а также тестировались интерфейсы для обеспечения их корректности. Автоматизация тестирования с помощью Cypress повысила качество, сократила время ручного тестирования и обеспечила надежность системы.

ЗАКЛЮЧЕНИЕ

В рамках данной работы была разработана система, которая автоматизирует отслеживание сроков годности продукции, ведение учета произведенной продукции, а также предоставляет возможность разделения функциональности в соответствии с ролями сотрудников.

Были выполнены поставленные задачи.

1. Выполнен анализ предметной области, описаны требования к системе отслеживания срока годности продукции.
2. Спроектирована структура клиентской части системы.
3. Выполнена реализация клиентской части системы.
4. Выполнено тестирование клиентской части системы.

В настоящий момент производится интеграция разработанной системы на предприятия. В будущем планируется продолжить разработку и усовершенствование системы, внедряя дополнительные функции и улучшая существующие. Это может включать в себя расширение возможностей мониторинга и аналитики, добавление новых инструментов для управления продукцией и сроками годности, а также улучшение интерфейса пользователя для более удобного и интуитивно понятного использования. Кроме того, планируется обновление системы с учетом изменяющихся потребностей бизнеса и требований рынка, чтобы обеспечить ее актуальность и конкурентоспособность в долгосрочной перспективе.

ЛИТЕРАТУРА

1. Контур.Маркет. [Электронный ресурс] URL: <https://kontur.ru/market/features/otslezhivayte-sroki-godnosti> (дата обращения: 15.03.2024 г.).
2. S-Mobile. [Электронный ресурс] URL: <https://data-krat.ru/s-mobile> (дата обращения: 15.03.2024 г.).
3. CloudShop. [Электронный ресурс] URL: <https://cloudshop.ru> (дата обращения: 15.03.2024 г.).
4. Бэнкс А., Порселло Е. React и Redux: функциональная веб-разработка. – СПб.: «Питер», 2018. – С. 336.
5. Макфарланд, Д. JavaScript и jQuery: исчерпывающее руководство. – М.: «Диалектика», 2015. – 880 с.
6. Лабберс П., Олберс Б., Салим Ф. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений – М.: «Вильямс», 2011. – 272 с.
7. Макфарланд Д. Новая большая книга CSS = CSS: The Missing Manual. – Санкт-Петербург: Питер, 2017. – 720 с.
8. Работа с JSON – Изучение веб-разработки – MDN Web Docs. [Электронный ресурс] URL: https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/JSON (дата обращения: 05.04.2024 г.).
9. Callegati F., Cerroni W., Ramilli M. Man-in-the-Middle Attack to the HTTPS Protocol // IEEE Security Privacy. – 2009. – Январь (т. 7, вып. 1). – С. 78–81.
10. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. – 2-е изд. – М., СПб.: ДМК Пресс, Питер, 2004. – 432 с.
11. Visual Studio Code. [Электронный ресурс] URL: <https://code.visualstudio.com> (дата обращения: 12.04.2024 г.).
12. React. [Электронный ресурс] URL: <https://reactjs.org/> (дата обращения: 15.04.2024 г.).

13. Node.js. [Электронный ресурс] URL: <https://nodejs.org/> (дата обращения: 15.04.2024 г.).
14. MobX. [Электронный ресурс] URL: <https://mobx.js.org/README.html> (дата обращения: 15.03.2024 г.).
15. Axios. [Электронный ресурс] URL: <https://axios-http.com/> (дата обращения: 15.04.2024 г.).
16. Clsx. [Электронный ресурс] URL: <https://github.com/lukeed/clsx> (дата обращения: 15.04.2024 г.).
17. Lodash. [Электронный ресурс] URL: <https://lodash.com/> (дата обращения: 15.04.2024 г.).
18. Moment. [Электронный ресурс] URL: <https://momentjs.com/> (дата обращения: 15.04.2024 г.).
19. React-table. [Электронный ресурс] URL: <https://react-table.tanstack.com/> (дата обращения: 15.04.2024 г.).
20. Vite. [Электронный ресурс] URL: <https://vitejs.dev/> (дата обращения: 15.04.2024 г.).
21. ZXing. [Электронный ресурс] URL: <https://github.com/zxing/zxing> (дата обращения: 15.04.2024 г.).
22. Фримен Э., Фримен Э. Изучаем HTML, XHTML и CSS. – П.: «Питер», 2010. – 656 с.
23. Sass. [Электронный ресурс] URL: <https://sass-lang.com/documentation/> (дата обращения: 15.03.2024 г.).
24. TypeScript. [Электронный ресурс] URL: <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html> (дата обращения: 15.04.2024 г.).
25. Cypress: Testing Frameworks for Javascript. [Электронный ресурс] URL: <https://www.cypress.io> (дата обращения: 28.04.2024 г.).
26. Бек К. Экстремальное программирование: разработка через тестирование = Test-driven Development. – Питер, 2003. – 224 с.

ПРИЛОЖЕНИЯ

Приложение А. Диаграмма вариантов использования

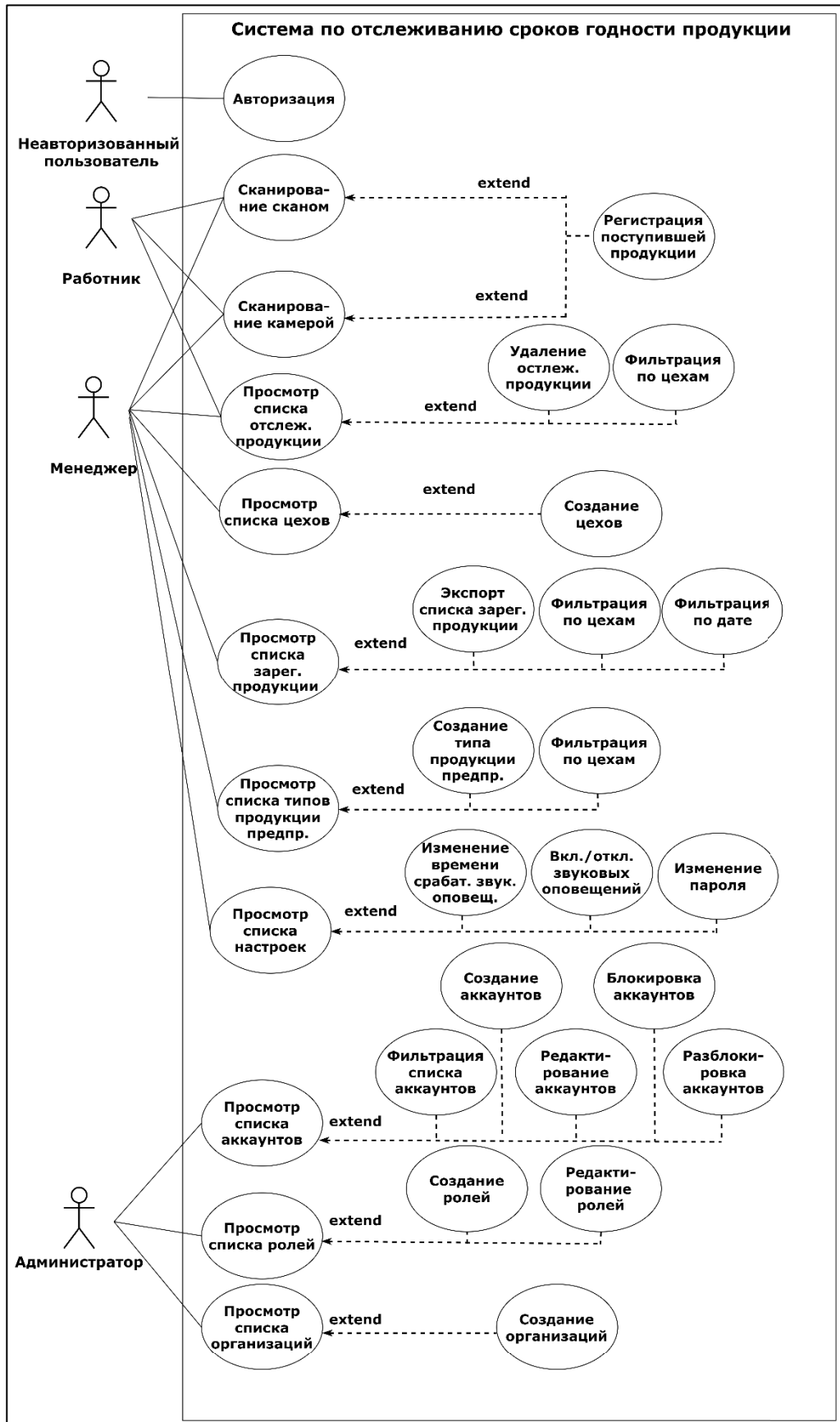


Рисунок 1 – Диаграмма вариантов использования

Приложение Б. Спецификация вариантов использования

Спецификация вариантов использования (ВИ) системы приведена в таблицах 1–31.

Таблица 1 – Спецификация ВИ «Авторизация»

Прецедент: Авторизация
ID: 1
Краткое описание: Вход в систему
Главные актеры: Гость
Второстепенные актеры: Нет
Предусловия: В базу данных внесены данные о пользователях
Основной поток: 1. Прецедент начинается, когда гость открывает вкладку входа в систему. Система отображает страницу входа в систему. 2. Гость вводит данные в форму и нажимает на кнопку «Войти». 3. Система находит пользователя в базе и авторизует его, предоставляя права пользователя.
Постусловия: Гость перешел в статус пользователя и имеет учетную запись в веб-системе.
Альтернативные потоки: I. Пользователь не был найден в базе. Система не нашла пользователя с соответствующими данными. 1. Система переходит к шагу 2 с уведомлением об ошибке.

Таблица 2 – Спецификация ВИ «Просмотр списка аккаунтов»

Прецедент: Просмотр списка аккаунтов
ID: 2
Краткое описание: Просмотр списка аккаунтов администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: Был выполнен прецедент с ID: 1 и администратор авторизовался в системе.
Основной поток: 1. Прецедент начинается, когда администратор открывает вкладку просмотра списка аккаунтов. 2. Система отображает страницу просмотра списка аккаунтов с полученными от сервера данными.
Постусловия: Список аккаунтов отображается на странице.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 3 – Спецификация ВИ «Создание аккаунта»

Прецедент: Создание аккаунта
ID: 3
Краткое описание: Создание аккаунта администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и администратор авторизовался в системе. 2. Был выполнен прецедент с ID: 2 и нажата кнопка «Создать аккаунт».
Основной поток: 1. Прецедент начинается, когда администратор открывает вкладку создания аккаунта. 2. Система отображает страницу создания аккаунта. 3. Администратор вводит в форму данные о новом пользователе и нажимает на кнопку «Создать аккаунт». 4. Система добавляет информацию о новом пользователе в базу данных.
Постусловия: 1. Новый пользователь добавлен в систему. 2. Система отображает уведомление об успешном создании аккаунта.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. Новые данные пользователя не прошли валидацию. 1. Система выводит сообщение об ошибке.

Таблица 4 – Спецификация ВИ «Редактирование аккаунта»

Прецедент: Редактирование аккаунта
ID: 4
Краткое описание: Редактирование аккаунта администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и администратор авторизовался в системе. 2. Был выполнен прецедент с ID: 2 и нажата кнопка «Редактировать аккаунт».
Основной поток: 1. Прецедент начинается, когда администратор открывает вкладку редактирования аккаунта. 2. Система отображает страницу редактирования аккаунта. 3. Система получает данные выбранного аккаунта от сервера и отображает их в форме. 4. Администратор изменяет в форме данные о пользователе и нажимает на кнопку «Сохранить». 5. Система обновляет информацию о новом пользователе.
Постусловия: 1. Данные о пользователе изменены. 2. Система отображает уведомление об успешном изменении данных аккаунта.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. Новые данные пользователя не прошли валидацию. 1. Система выводит сообщение об ошибке.

Таблица 5 – Спецификация ВИ «Блокировка аккаунта»

Прецедент: Блокировка аккаунта
ID: 5
Краткое описание: Блокировка аккаунта администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и администратор авторизовался в системе. 2. Был выполнен прецедент с ID: 2.
Основной поток: 1. Прецедент начинается, когда администратор нажимает на кнопку «Блокировать аккаунт» на странице просмотра списка аккаунтов. 2. Система обновляет статус аккаунта на «Заблокирован» и обновляет данные на странице.
Постусловия: Пользователь с заблокированным аккаунтом утрачивает доступ к системе.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 6 – Спецификация ВИ «Разблокировка аккаунта»

Прецедент: Разблокировка аккаунта
ID: 6
Краткое описание: Разблокировка аккаунта администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и администратор авторизовался в системе. 2. Был выполнен прецедент с ID: 2. 3. Был выполнен прецедент с ID: 5.
Основной поток: 1. Прецедент начинается, когда администратор нажимает на кнопку «Разблокировать аккаунт» на странице просмотра списка аккаунтов. 2. Система обновляет статус аккаунта на «Разблокирован» и обновляет данные на странице.
Постусловия: Пользователь с разблокированным аккаунтом восстанавливает доступ к системе.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 7 – Спецификация ВИ «Фильтрация списка аккаунтов»

Прецедент: Фильтрация списка аккаунтов
ID: 7
Краткое описание: Фильтрация списка аккаунтов администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и администратор авторизовался в системе. 2. Был выполнен прецедент с ID: 2.
Основной поток: 1. Прецедент начинается, когда администратор начинает вводить в форму фильтра имя пользователя. 2. Система отображает отфильтрованный список аккаунтов.
Постусловия: Администратор видит отфильтрованный список аккаунтов.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 8 – Спецификация ВИ «Просмотр списка ролей»

Прецедент: Просмотр списка ролей
ID: 8
Краткое описание: Просмотр списка ролей администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: Был выполнен прецедент с ID: 1 и администратор авторизовался в системе.
Основной поток: 1. Прецедент начинается, когда администратор открывает вкладку просмотра списка ролей. 2. Система отображает страницу просмотра списка ролей с полученными от сервера данными.
Постусловия: Список ролей отображается на странице.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 9 – Спецификация ВИ «Создание роли»

Прецедент: Создание аккаунта
ID: 9
Краткое описание: Создание роли администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и администратор авторизовался в системе. 2. Был выполнен прецедент с ID: 8 и нажата кнопка «Новая роль».
Основной поток: 1. Прецедент начинается, когда администратор открывает вкладку создания роли. 2. Система отображает страницу создания роли. 3. Администратор вводит в форму наименование роли, выбирает разрешения на доступ для этой роли и нажимает на кнопку «Создать роль». 4. Система добавляет информацию о новой роли в базу данных.
Постусловия: 1. Новая роль добавлена в систему. 2. Система отображает уведомление об успешном создании роли.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 10 – Спецификация ВИ «Редактирование роли»

Прецедент: Редактирование роли
ID: 10
Краткое описание: Редактирование роли администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и администратор авторизовался в системе. 2. Был выполнен прецедент с ID: 8 и нажата кнопка «Редактировать роль».
Основной поток: 1. Прецедент начинается, когда администратор открывает вкладку редактирования роли. 2. Система отображает страницу редактирования роли. 3. Система получает данные выбранной роли от сервера и отображает их в форме. 4. Администратор изменяет в форме данные о роли и нажимает на кнопку «Сохранить». 5. Система обновляет информацию о новой роли.
Постусловия: 1. Данные роли изменены. 2. Система отображает уведомление об успешном изменении данных роли.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 11 – Спецификация ВИ «Просмотр списка организаций»

Прецедент: Просмотр списка организаций
ID: 11
Краткое описание: Просмотр списка организаций администратором
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: Был выполнен прецедент с ID: 1 и администратор авторизовался в системе.
Основной поток: 1. Прецедент начинается, когда администратор открывает вкладку просмотра списка организаций. 2. Система отображает страницу просмотра списка организаций с полученными от сервера данными.
Постусловия: Список организаций отображается на странице.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 12 – Спецификация ВИ «Создание организации»

Прецедент: Создание организации
ID: 12
Краткое описание: Создание организации администратором.
Главные актеры: Администратор.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и администратор авторизовался в системе. 2. Был выполнен прецедент с ID: 11 и нажата кнопка «Новая организация».
Основной поток: 1. Прецедент начинается, когда администратор открывает вкладку создания организации. 2. Система отображает страницу создания организации. 3. Администратор вводит в форму наименование организации и нажимает на кнопку «Создать организацию». 4. Система добавляет информацию о новой организации в базу данных.
Постусловия: 1. Новая организация добавлена в систему. 2. Система отображает уведомление об успешном создании организации.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 13 – Спецификация ВИ «Просмотр списка цехов»

Прецедент: Просмотр списка цехов.
ID: 13
Краткое описание: Просмотр списка цехов.
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе.
Основной поток: 1. Прецедент начинается, когда пользователь открывает вкладку просмотра списка цехов. 2. Система отображает страницу просмотра списка цехов с полученными от сервера данными.
Постусловия: Список цехов отображается на странице.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 14 – Спецификация ВИ «Создание цеха»

Прецедент: Создание цеха
ID: 14
Краткое описание: Создание цеха.
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 13.
Основной поток: 1. Прецедент начинается, когда пользователь нажимает на кнопку «Добавить». 2. Система отображает форму создания цеха. 3. Пользователь вводит в форму наименование цеха и нажимает на кнопку «Сохранить». 4. Система добавляет информацию о новом цехе в базу данных.
Постусловия: 1. Новый цех добавлен в систему. 2. Система обновляет список цехов.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 15 – Спецификация ВИ «Просмотр списка типов продукции»

Прецедент: Просмотр списка типов продукции
ID: 15
Краткое описание: Просмотр списка типов продукции
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе.
Основной поток: 1. Прецедент начинается, когда пользователь открывает вкладку просмотра списка типов продукции. 2. Система отображает страницу просмотра списка типов продукции с полученными от сервера данными.
Постусловия: Список типов продукции отображается на странице.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. Сервер прислал пустой список типов продукции. 1. Система выводит сообщение что типов продукции нет.

Таблица 16 – Спецификация ВИ «Создание типа продукции»

Прецедент: Создание типа продукции
ID: 16
Краткое описание: Создание типа продукции.
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 14. 4. Был выполнен прецедент с ID: 15.
Основной поток: 1. Прецедент начинается, когда пользователь нажимает на кнопку «Добавить». 2. Система отображает форму создания типа продукции. 3. Пользователь вводит в форму наименование данные о продукции и нажимает на кнопку «Сохранить». 4. Система добавляет информацию о новом типе продукции в базу данных.
Постусловия: 1. Новый тип продукции добавлен в систему. 2. Система обновляет список типов продукции.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 17 – Спецификация ВИ «Фильтрация списка типов продукции по цехам»

Прецедент: Фильтрация списка типов продукции по цехам
ID: 17
Краткое описание: Фильтрация списка типов продукции по цехам
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
<p>Предусловия:</p> <ol style="list-style-type: none"> 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 14. 3. Был выполнен прецедент с ID: 15.
<p>Основной поток:</p> <ol style="list-style-type: none"> 1. Прецедент начинается, когда пользователь нажимает на кнопку с названием цеха, по которому хочет отфильтровать список. 2. Система обновляет список типов продукции.
<p>Постусловия:</p> <p>Пользователь видит отфильтрованный список типов продукции.</p>
<p>Альтернативные потоки:</p> <p>I. Ошибка при обращении к серверу.</p> <ol style="list-style-type: none"> 1. Система выводит сообщение об ошибке. <p>II. Сервер прислал пустой список цехов.</p> <ol style="list-style-type: none"> 1. Система выводит сообщение что цехов нет.

Таблица 18 – Спецификация ВИ «Сканирование сканом»

Прецедент: Сканирование сканом
ID: 18
Краткое описание: Сканирование сканом
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
<p>Предусловия:</p> <ol style="list-style-type: none"> 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 16 или был выполнен прецедент с ID: 20.
<p>Основной поток:</p> <ol style="list-style-type: none"> 1. Прецедент начинается, когда пользователь переходит на страницу сканирования сканом. 2. Система отображает страницу сканирования сканом. 3. Пользователь сканирует продукцию или вводит ее код в поле ввода. 4. Система находит в базе информацию о нужной продукции и переходит к прецеденту 19.
<p>Постусловия:</p> <p>Пользователь переходит на страницу регистрации поступившей продукции.</p>
<p>Альтернативные потоки:</p> <p>I. Ошибка при обращении к серверу.</p> <ol style="list-style-type: none"> 1. Система выводит сообщение об ошибке. <p>II. В базе не была найдена продукция.</p> <ol style="list-style-type: none"> 1. Система выводит сообщение что продукция не найдена.

Таблица 19 – Спецификация ВИ «Регистрация поступившей продукции»

Прецедент: Регистрация поступившей продукции
ID: 19
Краткое описание: Регистрация поступившей продукции
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 18.
Основной поток: 1. Прецедент начинается, когда пользователь переходит на страницу регистрации поступившей продукции. 2. Система отображает страницу регистрации поступившей продукции и форму регистрации. 3. Пользователь вводит информацию о поступившей продукции. 4. Система сохраняет данные о поступившей продукции.
Постусловия: Система отображает уведомление об успешном добавлении продукции.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 20 – Спецификация ВИ «Сканирование камерой»

Прецедент: Сканирование камерой
ID: 20
Краткое описание: Сканирование камерой
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 16.
Основной поток: 1. Прецедент начинается, когда пользователь переходит на страницу сканирования камерой. 2. Система отображает страницу сканирования камерой. 3. Пользователь сканирует продукцию или вводит ее код в поле ввода. 4. Система находит в базе информацию о нужной продукции и переходит к прецеденту 19.
Постусловия: Пользователь переходит на страницу регистрации поступившей продукции.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. В базе не была найдена продукция. 1. Система выводит сообщение что продукция не найдена.

Таблица 21 – Спецификация ВИ «Просмотр списка отслеживаемой продукции»

Прецедент: Просмотр списка отслеживаемой продукции
ID: 21
Краткое описание: Просмотр списка отслеживаемой продукции
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 19.
Основной поток: 1. Прецедент начинается, когда пользователь открывает вкладку просмотра списка отслеживаемой продукции. 2. Система отображает страницу просмотра списка отслеживаемой продукции с полученными от сервера данными.
Постусловия: Список отслеживаемой продукции отображается на странице.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. Сервер прислал пустой список типов продукции. 1. Система выводит сообщение что типов продукции нет.

Таблица 22 – Спецификация ВИ «Удаление отслеживаемой продукции»

Прецедент: Удаление отслеживаемой продукции
ID: 22
Краткое описание: Удаление отслеживаемой продукции из списка отслеживаемой продукции
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 21.
Основной поток: 1. Нажимает на кнопку «Реализовано» или «Списано». 2. Система обновляет данные продукции и обновляет список отслеживаемой продукции.
Постусловия: Пользователь не видит удаленную отслеживаемую продукцию в списке.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 23 – Спецификация ВИ «Фильтрация списка отслеживаемой продукции по цехам»

Прецедент: Фильтрация списка отслеживаемой продукции по цехам
ID: 23
Краткое описание: Фильтрация списка отслеживаемой продукции по цехам
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 14. 3. Был выполнен прецедент с ID: 21.
Основной поток: 1. Прецедент начинается, когда пользователь нажимает на кнопку с названием цеха, по которому хочет отфильтровать список. 2. Система обновляет список отслеживаемой продукции.
Постусловия: Пользователь видит отфильтрованный список отслеживаемой продукции.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. Сервер прислал пустой список цехов. 1. Система выводит сообщение что цехов нет.

Таблица 24 – Спецификация ВИ «Просмотр списка зарегистрированной продукции»

Прецедент: Просмотр списка зарегистрированной продукции
ID: 24
Краткое описание: Просмотр списка зарегистрированной продукции
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 19.
Основной поток: 1. Прецедент начинается, когда пользователь открывает вкладку просмотра списка зарегистрированной продукции. 2. Система отображает страницу просмотра списка зарегистрированной продукции с полученными от сервера данными.
Постусловия: Список зарегистрированной продукции отображается на странице.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. Сервер прислал пустой список зарегистрированной продукции. 1. Система выводит сообщение что зарегистрированной продукции нет.

Таблица 25 – Спецификация ВИ «Фильтрация списка зарегистрированной продукции по цехам»

Прецедент: Фильтрация списка зарегистрированной продукции по цехам
ID: 25
Краткое описание: Фильтрация списка зарегистрированной продукции по цехам
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 14. 3. Был выполнен прецедент с ID: 24.
Основной поток: 1. Прецедент начинается, когда пользователь нажимает на кнопку с названием цеха, по которому хочет отфильтровать список. 2. Система обновляет список зарегистрированной продукции.
Постусловия: Пользователь видит отфильтрованный список зарегистрированной продукции.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. Сервер прислал пустой список цехов. 1. Система выводит сообщение что цехов нет.

Таблица 26 – Спецификация ВИ «Фильтрация списка зарегистрированной продукции по дате»

Прецедент: Фильтрация списка зарегистрированной продукции по дате
ID: 26
Краткое описание: Фильтрация списка зарегистрированной продукции по дате
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 24.
Основной поток: 1. Прецедент начинается, когда пользователь нажимает на поле ввода даты и вводит или выбирает дату. 2. Система обновляет список зарегистрированной продукции.
Постусловия: Пользователь видит отфильтрованный список зарегистрированной продукции.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 27 – Спецификация ВИ «Экспорт списка зарегистрированной продукции»

Прецедент: Экспорт списка зарегистрированной продукции
ID: 27
Краткое описание: Экспорт списка зарегистрированной продукции
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 24.
Основной поток: 1. Прецедент начинается, когда пользователь нажимает на кнопку «Экспортировать». 2. Система создает файл со списком зарегистрированной продукции для скачивания и отправляет пользователю.
Постусловия: Пользователь скачивает файл со списком зарегистрированной продукции.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 28 – Спецификация ВИ «Просмотр списка настроек»

Прецедент: Просмотр списка настроек
ID: 28
Краткое описание: Просмотр списка настроек
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе.
Основной поток: 1. Прецедент начинается, когда пользователь открывает вкладку просмотра списка настроек. 2. Система отображает страницу просмотра настроек с полученными от сервера данными.
Постусловия: Список настроек отображается на странице и виден пользователю.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 29 – Спецификация ВИ «Изменение времени срабатывания звукового оповещения»

Прецедент: Изменение времени срабатывания звукового оповещения
ID: 29
Краткое описание: Изменение времени срабатывания звукового оповещения
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 28.
Основной поток: 1. Прецедент начинается, когда пользователь изменяет данные в поле ввода времени срабатывания звукового оповещения и нажимает на кнопку «Сохранить». 2. Система обновляет параметры настроек.
Постусловия: Выводится сообщение об успешном изменении настроек.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 30 – Спецификация ВИ «Включение или выключение звуковых оповещений»

Прецедент: Включение или выключение звуковых оповещений.
ID: 30
Краткое описание: Включение или выключение звуковых оповещений.
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 28.
Основной поток: 1. Прецедент начинается, когда пользователь изменяет данные в поле ввода включения звуковых оповещений и нажимает на кнопку «Сохранить». 2. Система обновляет параметры настроек.
Постусловия: Выводится сообщение об успешном изменении настроек.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке.

Таблица 31 – Спецификация ВИ «Изменение пароля»

Прецедент: Изменение пароля.
ID: 30
Краткое описание: Изменение пароля текущего аккаунта в настройках.
Главные актеры: Пользователь.
Второстепенные актеры: Нет.
Предусловия: 1. Был выполнен прецедент с ID: 1 и пользователь авторизовался в системе. 2. Был выполнен прецедент с ID: 28.
Основной поток: 1. Прецедент начинается, когда пользователь вводит текущий и новый пароли и нажимает на кнопку «Сохранить». 2. Система обновляет пароли.
Постусловия: Выводится сообщение об успешном изменении пароля.
Альтернативные потоки: I. Ошибка при обращении к серверу. 1. Система выводит сообщение об ошибке. II. Пользователь ввел неверный текущий пароль. 1. Система выводит сообщение об ошибке в данных.

Приложение В. Листинг запроса

Листинг 1 – Запрос на получение информации о продукции и ответ сервера

```
GET /products {
header: JSON Web Token
RESPONSE:
[
  {
    "id": 6413,
    "name": "Булочка с маком",
    "shopName": "Хлебобулочный цех",
    "isPiece": true,
    "barcodes": [4343435567]
    "lifetimeInMinutes": 720
  },
  {
    "id": 6560,
    "name": "Чеснок",
    "shopName": "Овощной цех",
    "isPiece": false,
    "barcodes": [3432434244],
    "lifetimeInMinutes": 1080
  },
  {
    "id": 6414,
    "name": "Оливье",
    "shopName": "Салатный цех",
    "isPiece": false,
    "barcodes": [6575765678],
    "lifetimeInMinutes": 720
  },
  {
    "id": 6398,
    "name": "Булочка с повидлом",
    "shopName": "Хлебобулочный цех",
    "isPiece": true,
    "barcodes": [1234567896],
    "lifetimeInMinutes": 1080
  },
  {
    "id": 6399,
    "name": "Лук",
    "shopName": "Овощной цех",
    "isPiece": false,
    "barcodes": [2132343556],
    "lifetimeInMinutes": 7200
  },
  {
    "id": 6402,
    "name": "Хлеб \"Карталинский\"",
    "shopName": "Хлебобулочный цех",
    "isPiece": true,
    "barcodes": [4535345433],
    "lifetimeInMinutes": 1440
  },
  {
    "id": 6400,
    "name": "Конфеты \"Славянка\"",
    "shopName": "Кондитерский цех",
    "isPiece": false,
    "barcodes": [2324545344],
    "lifetimeInMinutes": 7200
  },
  {
    "id": 6401,
    "name": "Перец",
    "shopName": "Овощной цех",
    "isPiece": false,
    "barcodes": [4543534545],
    "lifetimeInMinutes": 5040}
]
```

Приложение Г. Скриншоты разработанной системы

Скриншоты разработанной системы приведены на рисунках 2–13.

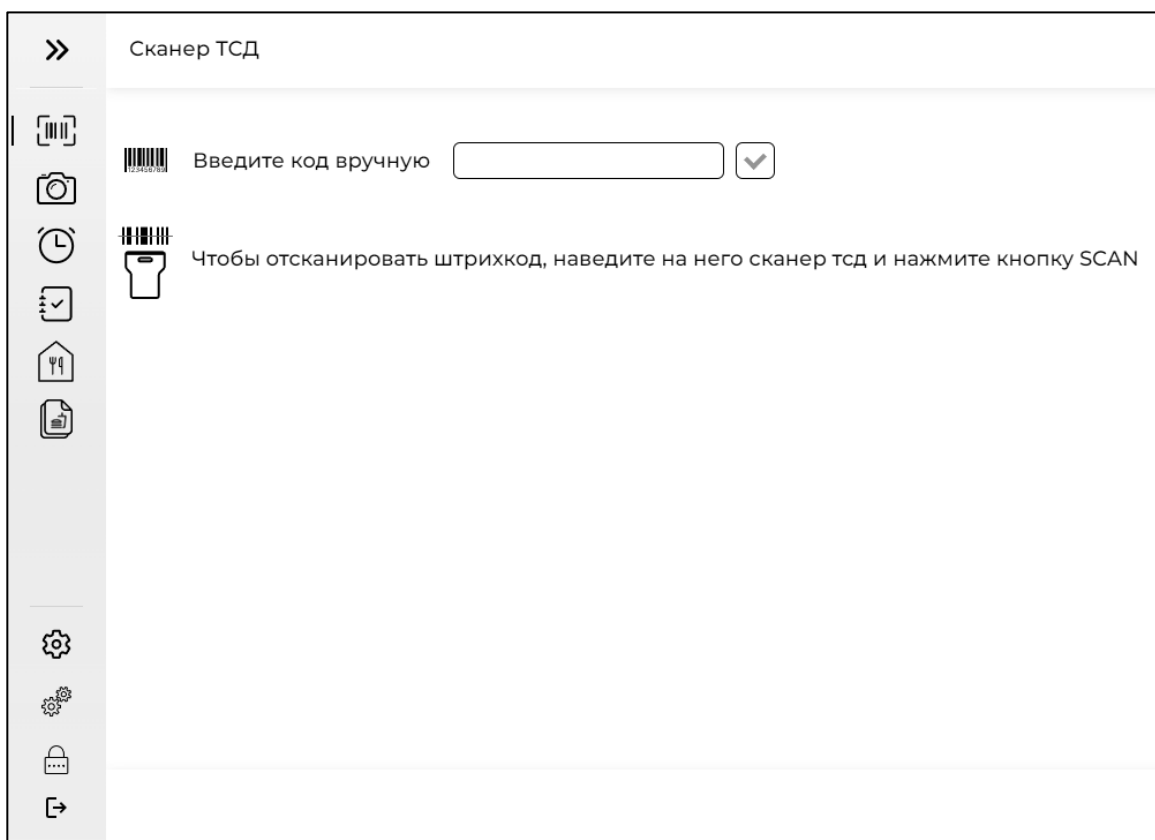


Рисунок 2 – Страница сканирования штрих-кода сканом

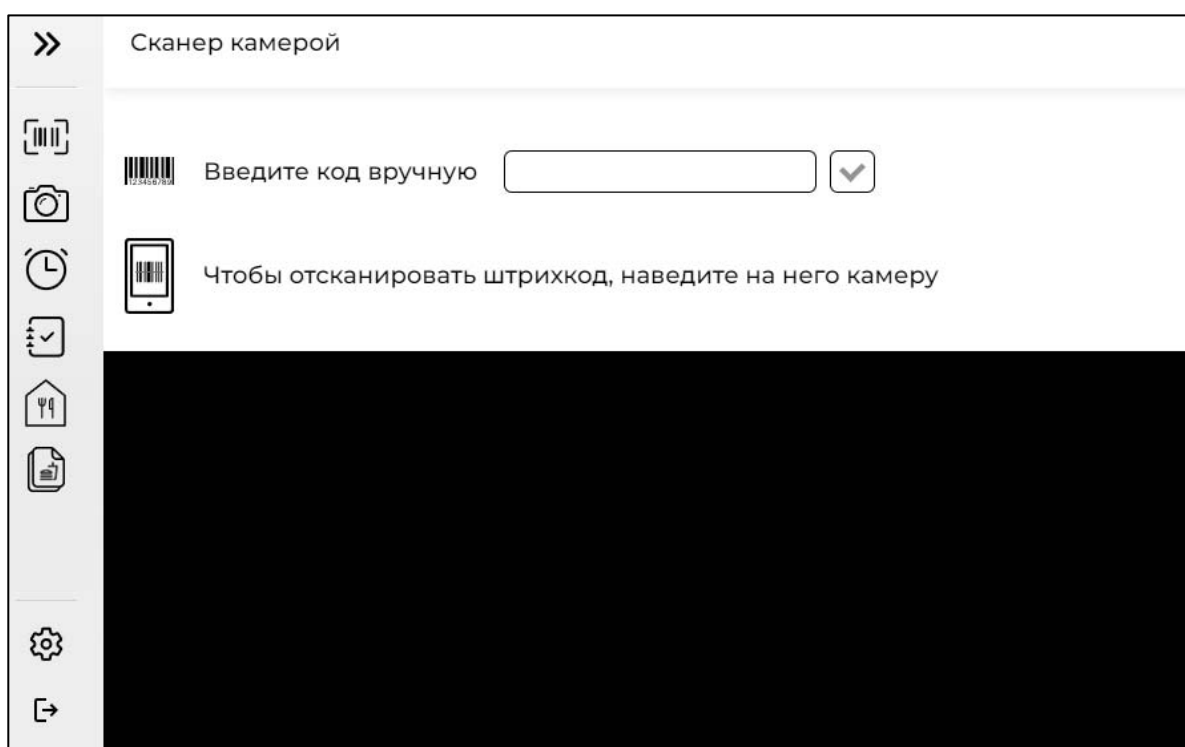


Рисунок 3 – Страница сканирования штрих-кода камерой

Реализуемая продукция

Наименование	Штучность	Код	Цех	Срок годности в минутах
Булочка с маком	штучный	4343435567	Хлебобулочный цех	720
Чеснок	весовой	3432434244	Овощной цех	1080
Оливье	весовой	6575765678	Салатный цех	720
Булочка с повидлом	штучный	1234567896	Хлебобулочный цех	1080
Лук	весовой	2132343556	Овощной цех	7200
Хлеб "Карталинский"	штучный	4535345433	Хлебобулочный цех	1440
Конфеты "Славянка"	весовой	2324545344	Кондитерский цех	7200
Перец	весовой	4543534545	Овощной цех	5040

Рисунок 4 – Страница реализуемой продукции

Добавить новую продукцию

Наименование

Тип Штучный Весовой

Цех

Код

Срок годности в минутах

Рисунок 5 – Страница создания типа продукции

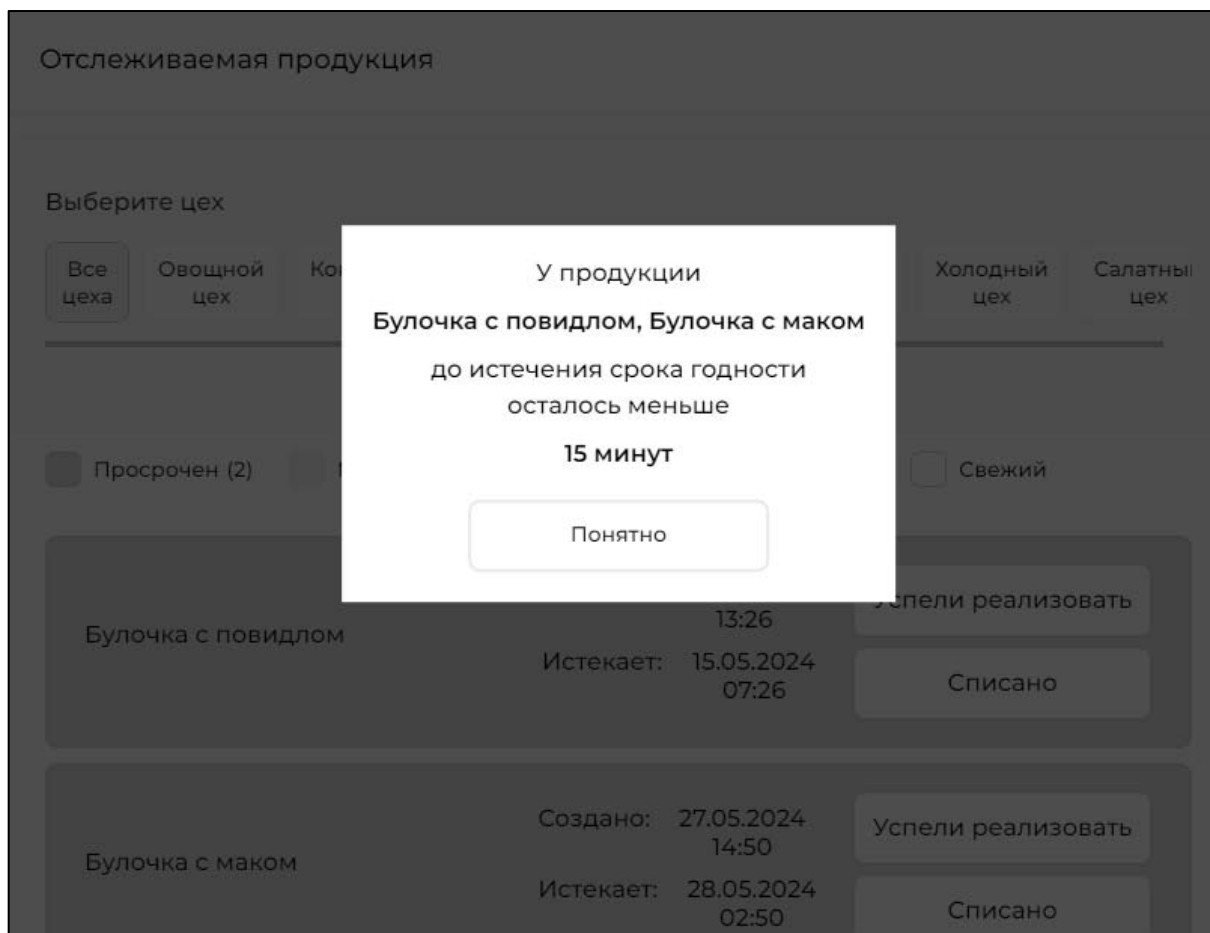


Рисунок 6 – Страница отслеживаемой продукции

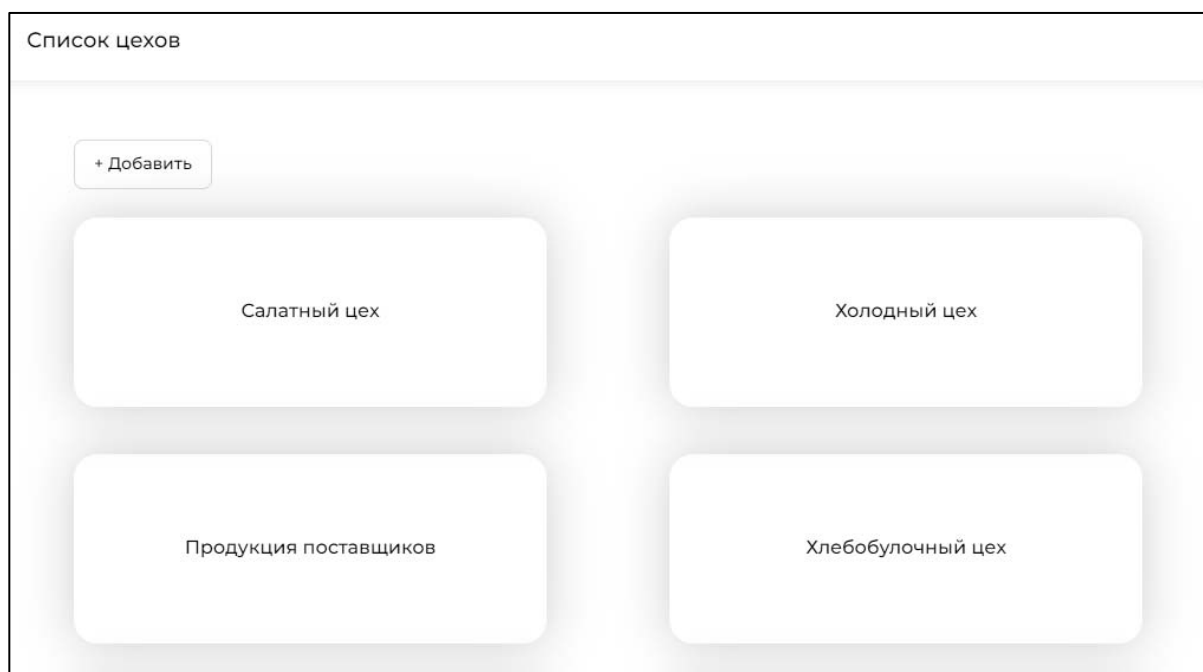


Рисунок 7 – Страница цехов

Бракеражный журнал

Журнал органолептической оценки качества полуфабрикатов, блюд и кулинарных изделий

дд.мм.гггг × 📄 Экспорт

Выберите цех, для которого хотите экспортировать бракеражный журнал

Дата и время ...	Наименование	К...	В...	Т...	В...	Ц...	З...	К...	Р...
27.05.2024 14:50	Булочка с маком	10	—	100	2	3	5	4	—
15.05.2024 09:32	Перец	—	53.454	—	4	5	5	4	Реализов
15.05.2024 09:32	Оливье	—	76.567	—	2	3	5	2	Просроч
14.05.2024 13:26	Булочка с повидлом	1	—	—	2	3	4	5	—

Рисунок 8 – Страница бракеражного журнала

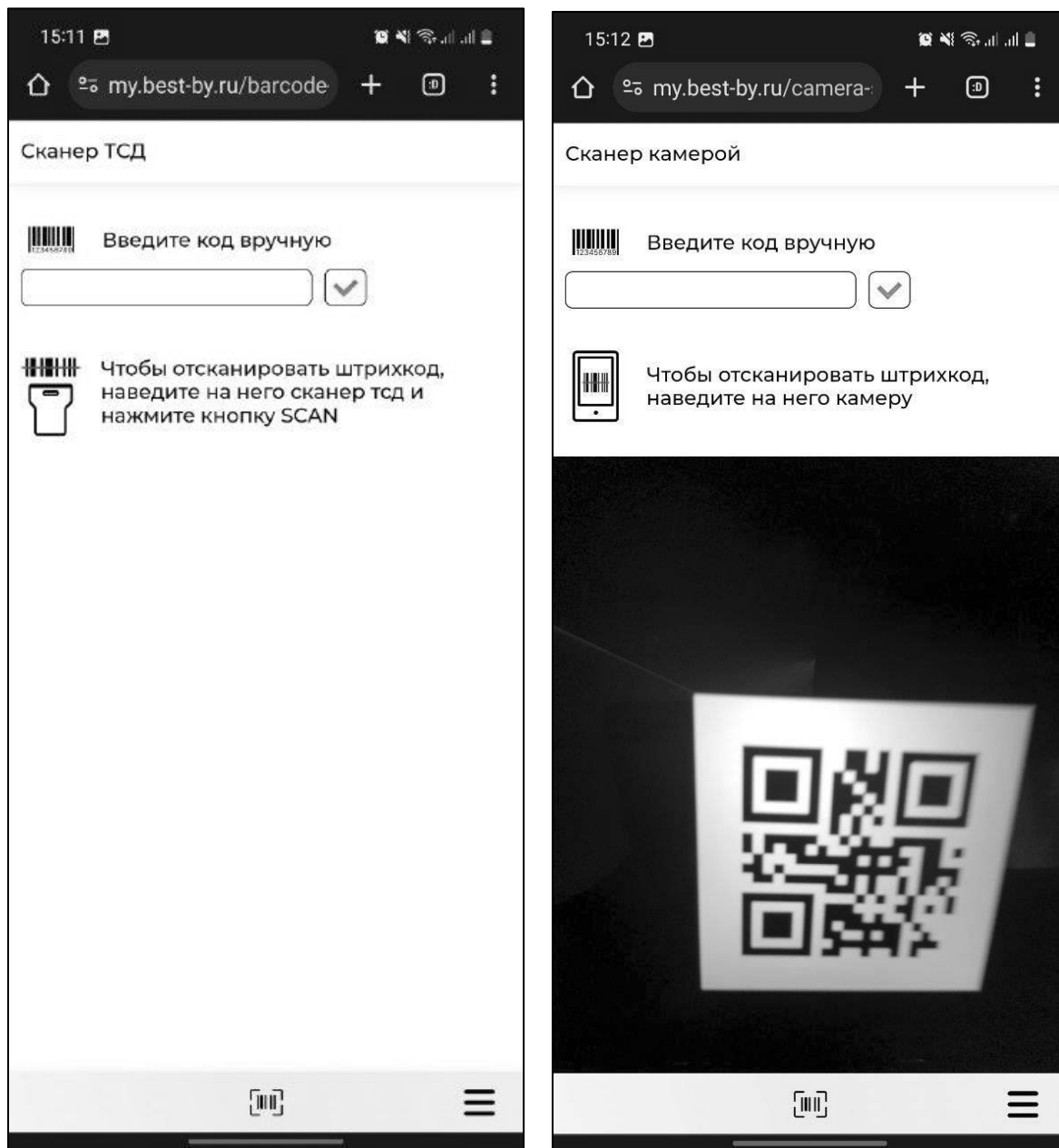


Рисунок 9 – Скриншоты страницы сканирования с мобильного устройства сканером (левый) и камерой (правый)

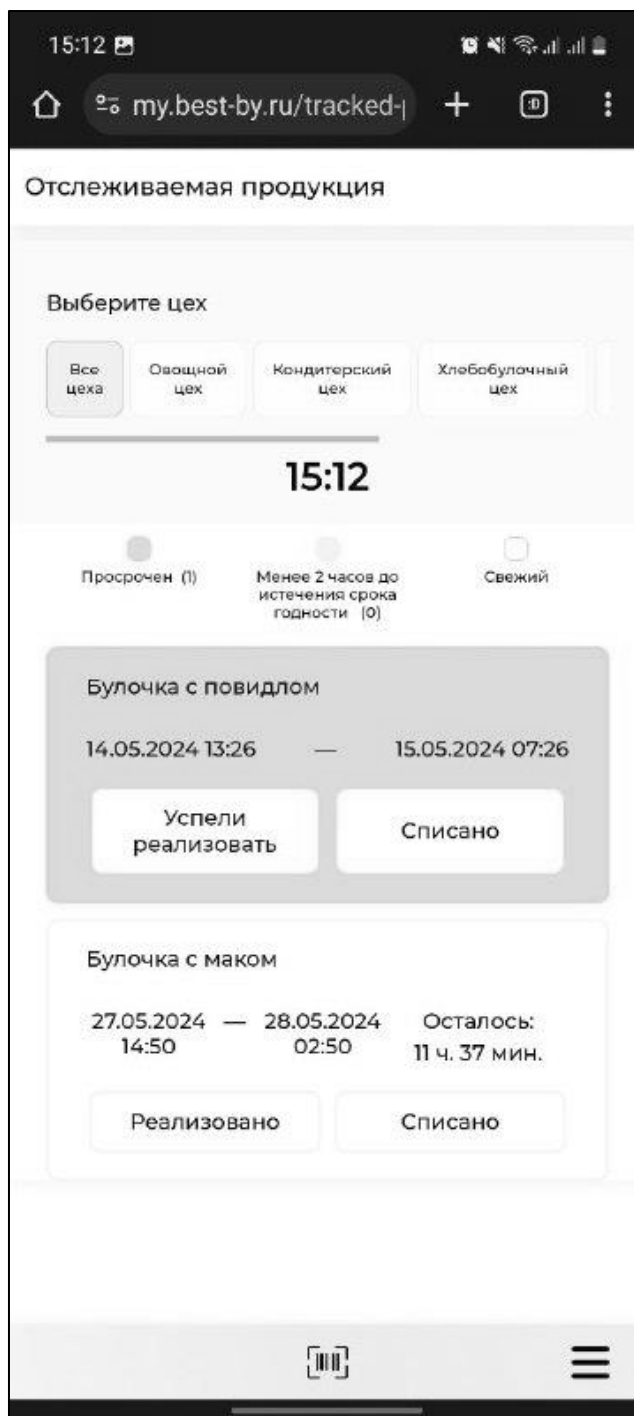


Рисунок 10 – Скриншот страницы отслеживаемой продукции с мобильного устройства

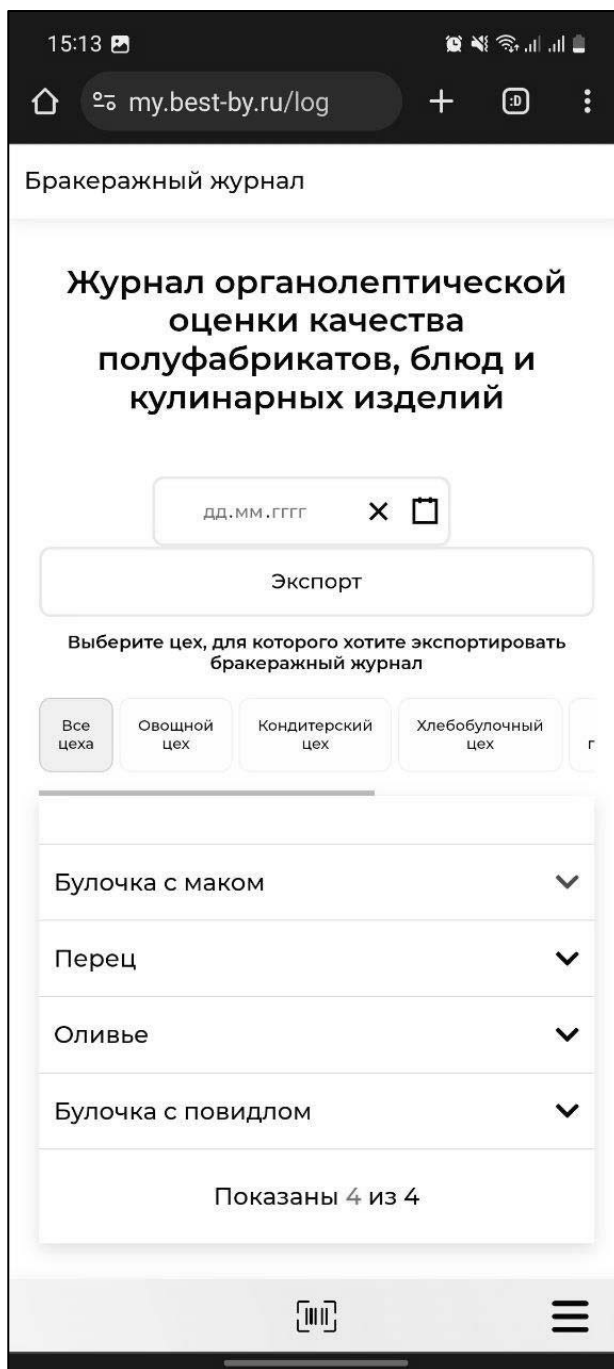


Рисунок 11 – Скриншот страницы бракеражного журнала с мобильного устройства

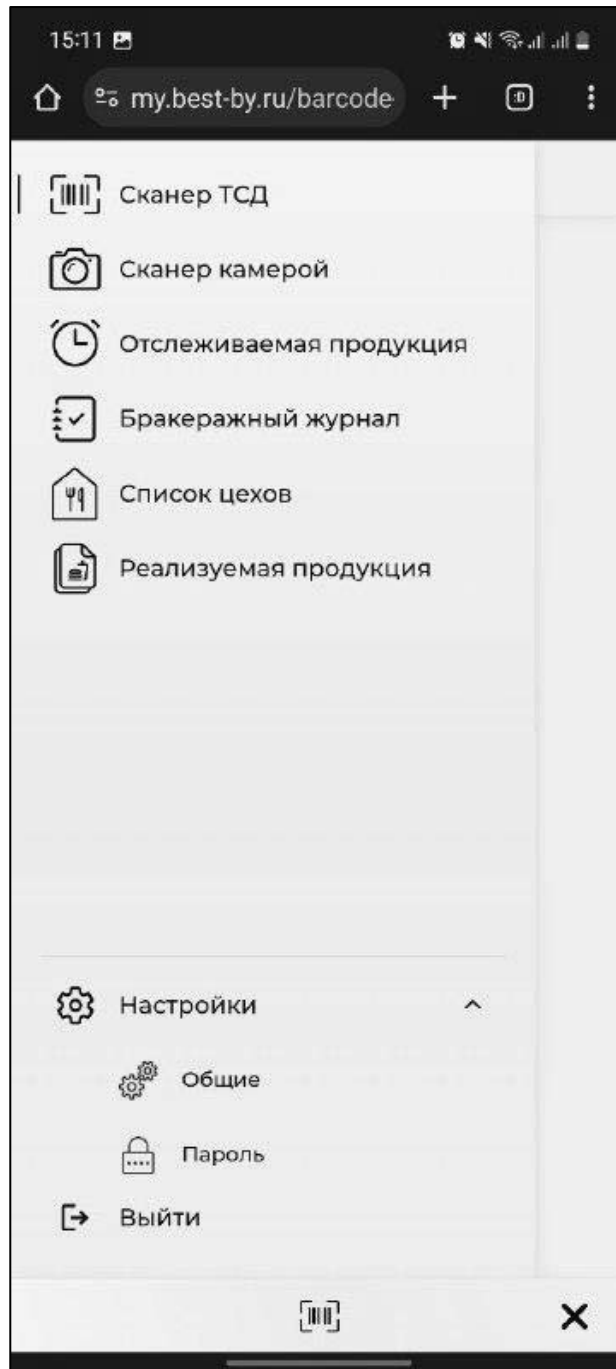


Рисунок 12 – Скриншот панели навигации в мобильной версии

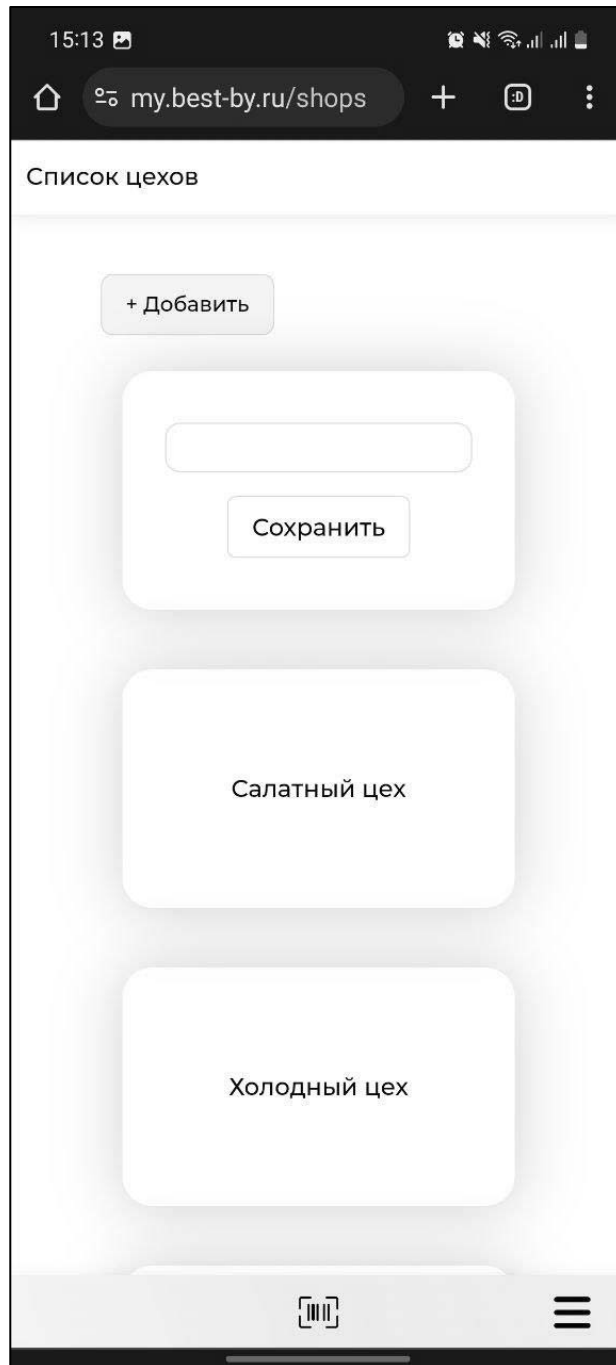


Рисунок 13 – Скриншот страницы цехов с мобильного устройства