Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

|  |  |
| --- | --- |
| Факультет | информационных технологий и управления |
| Кафедра | информационных технологий автоматизированных систем |

*К защите допустить*:

Заведующий кафедрой ИТАС

\_\_\_\_\_\_\_\_\_\_\_А. А. Навроцкий

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

на тему

**АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ**

**СИСТЕМА ТРАНСПОРТНОЙ КОМПАНИИ**

БГУИР ДП 1-53 01 02 06 078 ПЗ

|  |  |
| --- | --- |
| Студент  Руководитель  Консультанты:  *от кафедры ИТАС*  *по экономической части*  Нормоконтролер  Рецензент | М. О. Кохнович  А. М. Ковалёв  Е. Р. Павловская  Т. Л. Слюсарь  Т. С. Боброва |

Минск 2022

**РЕФЕРАТ**

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ТРАНСПОРТНОЙ КОМПАНИИ: дипломный проект / М. О. Кохнович. – Минск : БГУИР, 2022, – п.з. – 63 с., чертежей (плакатов) – 6 л. формата А1.

Объектом проектирования является автоматизированная система для управления заказами и накладными транспортных компаний.

Цель работы – разработка автоматизированной системы для управления заказами и накладными транспортных логистических компаний.

Разработка данной системы позволяет контролировать процесс создания и обработки заказов пользователей, а также упростит работу по сбору их в накладные для дальнейшей доставки. С помощью данной системы сотрудники компании имеют более удобный доступ к сбору информации.

Произведена разработка архитектуры программной системы, а также уточнение используемых технологий и разработка приложения.

Полученные в ходе технико-экономического обоснования результаты доказывают целесообразность разработки системы.

Проект включает в себя серверную и клиентскую часть.

Дипломный проект прошел проверку в системе «Антиплагиат». Процент оригинальности составляет 98,54%.

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc102737692)

[1 Анализ системы управления транспортной компании 7](#_Toc102737693)

[1.1 Описание проблематики темы дипломного проекта 7](#_Toc102737694)

[1.2 Информационная система компании 8](#_Toc102737695)

[1.3 Сравнение аналогов 9](#_Toc102737696)

[1.4 Постановка задач дипломного проекта 13](#_Toc102737697)

[2 Проектирование структуры информационной системы 14](#_Toc102737698)

[2.1 Организационно-экономическая сущность задачи 14](#_Toc102737699)

[2.2 Структура информационной системы 15](#_Toc102737700)

[2.3 Математическое и алгоритмическое обеспечение 16](#_Toc102737701)

[2.4 Информационное обеспечение 17](#_Toc102737702)

[2.5 Техническое и системное программное обеспечение 27](#_Toc102737703)

[2.6 Эргономическое обеспечение 28](#_Toc102737704)

[2.7 Организационное обеспечение 29](#_Toc102737705)

[3 Программная реализация информационной системы 30](#_Toc102737706)

[3.1 Выбор программных средств реализации 30](#_Toc102737707)

[3.2 Структура программного обеспечения 32](#_Toc102737708)

[3.3 Разработка программного кода 33](#_Toc102737709)

[3.4 Тестирование приложения 34](#_Toc102737710)

[3.5 Руководство пользователя 36](#_Toc102737711)

[4 Технико-экономическое обоснование эффективности разработки 53](#_Toc102737712)

[Заключение 62](#_Toc102737713)

[Список использованных источников 63](#_Toc102737714)

[Приложение А (обязательное) Программный код клиента приложения 64](#_Toc102737715)

[Приложение Б (обязательное) Программный код сервера приложения 65](#_Toc102737716)

# Введение

Цель дипломного проектирования заключается в разработке автоматизированной информационной системы транспортной компании.

Автоматизированная информационная система – совокупность программно-аппаратных средств, предназначенных для автоматизации деятельности, связанной с хранением, передачей и обработкой информации. Такая система может быть также определена как комплекс автоматизированных информационных технологий, предназначенных для информационного обслуживания.

Автоматизированная информационная система транспортной компании оптимизирует работу логистических компаний, сводит к минимуму ручные операции сотрудников компаний и предоставляет автоматическое составление накладной для доставки продукции.

Данная пояснительная записка включает в себя несколько разделов, с кратким описание которых можно ознакомиться ниже.

Раздел анализа системы управления транспортной компании описывает предметную область, для которой разработан дипломный проект. В нём присутствует анализирование транспортной компании в целом и существующая в такой компании информационная система. Также в данном разделе формулируется постановка задачи на дипломное проектирование.

Раздел проектирования структуры информационной системы описывает разработку основных компонентов системы, решаются задачи проектирования алгоритмического, информационного, технического, программного, эргономического и организационного видов обеспечения.

Ещё одним разделом является раздел, описывающий программную реализацию информационной системы, а именно процесс разработки программной реализации, отладки, подготовки программной документации системы и контрольных примеров.

Последний раздел включает в себя технико-экономическое обоснование эффективности разработки, а также целесообразность разработки данной системы.

Задачами в дипломном проекте являются: анализ предметной области, анализ технико-экономических показателей организации, проектирование автоматизированной системы, её программная реализация, технико-экономический расчёт внедряемой системы.

1. **Анализ системы управления транспортной компании**
   1. **Описание проблематики темы дипломного проекта**

С экономической точки зрения транспортная логистика – существенный элемент производственного процесса, так как «товар» не считается «готовым», если его не доставили потребителю, который планирует его использовать. Таким образом, ключевой особенностью коммерческой деятельности предприятий грузоперевозок является обеспечение экономически-выгодного процесса перемещения материально-товарных ценностей с использованием оптимального вида транспорта для данного перемещения.

Практически любая транспортная компания сталкивается с целым комплексом проблем, связанных с недостатками управления логистическими задачами, что неминуемо ведет к увеличению издержек и снижению прибыли. Наиболее затратными являются проблемы моделирования маршрутных сетей, взаимодействия с перевозчиками и клиентами, оперативного контроля перевозок и их диспетчеризации, взаимодействия со смежными автоматизированными системами предприятия.

Предприятие, представляет собой сложную систему, состоящую из подразделений и отделов. Для успешного выполнения поставленных задач требуется, чтобы все подразделения компании работали чётко и слаженно. Предприятие имеет следующие структурные подразделения:

* Диспетчерский отдел. Его специалисты занимаются управлением товарами компании, запросами новых товаров, интересующих клиентов.
* Отдел маркетинга и продаж. Основная цель – сбор информации о потенциальных клиентах и их потребностях в транспортных услугах, контроль за выполнением договоров с клиентами.
* Логистический отдел. Работники данного отдела разрабатывают стратегии по оптимизации доставки заказов путём составления накладных.
* Кадровый отдел. Задача специалистов – формировать и развивать корпоративную культуру, помогать с набором штатного персонала, вести учет и анализировать кадровую политику.
* Бухгалтерский отдел. Цель – увеличение эффективности ведения бухучета. Специалисты занимаются учетной политикой, помогают подготовить налоговую отчетность.
* Отдел организации перевозок. Его работники занимаются перевозкой товара по заранее подготовленному маршруту.

Структура компании представлена на рисунке 1.1.

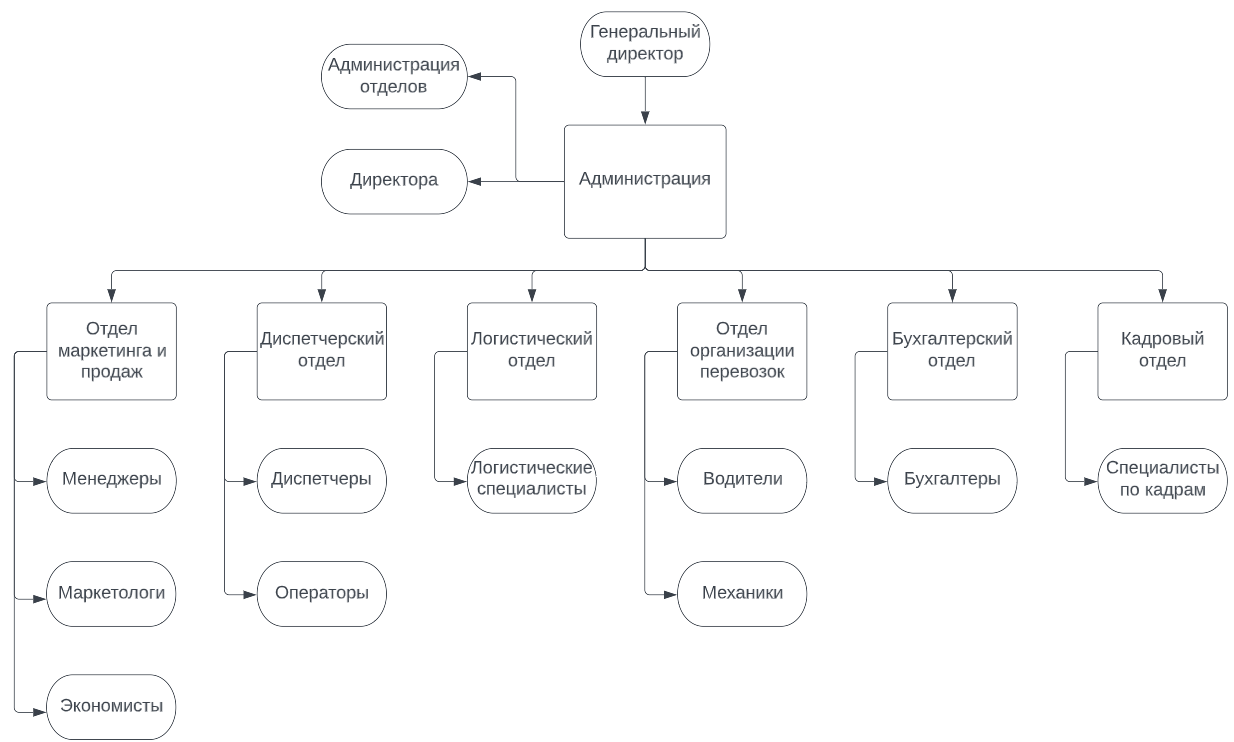


Рисунок 1.1 – Структура компании

* 1. **Информационная система компании**

В классической зарубежной транспортной компании используются *Microsoft* *Word*, *Microsoft* *Excel*, *QuickBooks*.

*Microsoft* *Word* – текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов. Выпускается корпорацией *Microsoft* в составе пакета *Microsoft* *Office*. Первая версия была написана Ричардом Броди (*Richard* *Brodie*) для *IBM* *PC*, использующих *DOS*, в 1983 году. Позднее выпускались версии для *Apple* *Macintosh* (1984), *SCO* *UNIX* и *Microsoft* *Windows* (1989).

*Microsoft* *Excel* (также иногда называется *Microsoft* *Office* *Excel*) – программа для работы с электронными таблицами, созданная корпорацией *Microsoft* для *Microsoft* *Windows*, *Windows* *NT* и *Mac* *OS*, а также *Android*, *iOS* и *Windows* *Phone*. Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением *Excel* 2008 под *Mac* *OS* *X*, язык макропрограммирования *VBA* (*Visual* *Basic* *for* *Application*). Microsoft *Excel* входит в состав *Microsoft* *Office* и на сегодняшний день *Excel* является одним из наиболее популярных приложений в мире.

*QuickBook* – это программа для бухгалтерского учета для малого бизнеса, которую предприятия используют для управления продажами и расходами и отслеживания ежедневных транзакций. Все отчеты встроены в *QuickBooks* и могут быть запущены всего за несколько кликов; отчеты обновляются в режиме реального времени по мере ввода и сохранения транзакций. Также она имеет собственную функцию расчета заработной платы, которая может автоматически рассчитывать и запускать платежную ведомость так часто, как нужно пользователю. Кроме того, данная программа автоматизирует трудоемкие процессы, такие как подготовка банковских выверок.

* 1. **Сравнение аналогов**

Для обзора и анализа функциональности систем взяты следующие приложения:

* *Aurama*;
* *Relog;*
* *Bpm’online sales.*

*Aurama* – сервис автоматизации работы курьерских служб. Состоит из рабочего места логиста и приложения курьера на *Android* устройствах.

Одной из полезных функций приложения является распределение заказов между курьерами. При распределении заказов учитывается местоположение курьера, объем, вес, технические характеристики автомобиля, желаемое время получения. Система предоставляет возможность постоянного отслеживания местоположения курьера, а также контактные данные необходимого курьера для связи с ним.

Также система занимается формированием финансовых отчетов, в том числе в виде наглядных графиков и диаграмм. *Aurama* предоставляет данные о загруженности курьеров, себестоимости доставок, убыточных клиентах и других показателях. [1]

Сервис *Aurama* разработан компанией, которая на протяжении более чем 12 лет занимается организацией электронной торговли, грузоперевозок и доставки. На сегодняшний день пользователями системы стало большое количество служб и сервисов доставки, а также интернет-магазинов.

Интерфейс этого приложения можно увидеть на рисунке 1.2.

Изображение выглядит как карта

Автоматически созданное описание

Рисунок 1.2 – Интерфейс приложения *Aurama*

*Relog* – система для логистических операторов, интернет-магазинов, курьерских служб, а также программа для управления логистикой и дистрибьюцией.

С помощью диспетчерского рабочего места появляется возможность полноценного управления логистическими процессами, насколько сложными бы они ни были. Кроме того, программа позволяет просматривать историю перемещений курьеров за все время, следить онлайн за статусами заказов. Благодаря системе появляется возможность автоматического планирования идеальных траекторий передвижения и полноценной загрузки транспорта.

Мобильный софт для водителей и доставщиков является идеальным в плане функций и эргономики вариантом для работы с клиентскими заказами и маршрутом. Приложение дает доступ к маршрутным листам в формате онлайн, корректировки статусов заявок, оставления комментариев, добавки фотографий и подписей. [2]

*Web*-сервис помогает директорам по логистике и топ-менеджменту компаний получить максимально детальную и глубокую аналитику, видеть картину тенденций и географических особенностей доставки, моделировать исход различных вариантов действий, отслеживать результаты принятия решений.

На рисунках 1.3 и 1.4 можно увидеть интерфейс данного приложения для логистического специалиста и пользователя соответственно.

Изображение выглядит как карта

Автоматически созданное описание

Рисунок 1.3 – Интерфейс приложения *Relog*

для логистического специалиста

Изображение выглядит как карта

Автоматически созданное описание

Рисунок 1.4 – Интерфейс приложения

*Relog* для пользователя

Система *bpm’online* представляет собой единую виртуальную платформу, построенную по модели облачной системы хранения данных, для того чтобы решать вопросы автоматизации основных процессов компании, которые направленны на взаимоотношение с клиентской базой.

Использование данной системы делает доступным управление продажами, маркетингом и сервисом через одно приложение. В *bmp’online* используются шаблонные онлайн-инструменты, которые подстраиваются под различные отрасли бизнеса. [3]

Программный продукт позволяет руководить и отслеживать все заказы организации, группировать их по различным критериям, контролировать стадии выполнения и анализировать эффективность выполнения.

Дополнительной функцией *bpm’online sales* является корпоративная социальная лента, обеспечивающая коммуницирование между сотрудниками организации. Обмен мнениями, профессиональная помощь друг другу, информация о сделках, новых продуктах и тому подобное.

Интерфейс этого приложения можно увидеть на рисунке 1.5.

Изображение выглядит как текст, снимок экрана, компьютер, внутренний

Автоматически созданное описание

Рисунок 1.5 – Интерфейс приложения

*bpm’online sales*

Как видно из всего вышеперечисленного, все системы выполняют практически одни и те же функции и обладают набором одних и тех же компонентов. Эти сервисы хорошо внедрять для современных предприятий. Однако можно выделить следующее улучшение систем: добавление возможности доставки товаров не в одиночку, а партиями по определённому маршруту, оптимальность вычисления которого происходит автоматически. А также добавление возможности поставки товаров в другие страны и расчёт итоговой стоимости с учётом налогов. В связи с этим принято решение о разработке автоматизированной системы.

* 1. **Постановка задач дипломного проекта**

Цель дипломного проектирования заключается в разработке удобной и гибкой автоматизированной системы транспортной компании.

Обеспечить получение объективной информации для принятия управленческих решений и планирования логистических процессов, оптимизацию технологических процессов и повышение производительности труда, сокращение времени простоя транспорта, ускорение процедур расчетов сопутствующих затрат, контроль транспорта в пути и под загрузкой или разгрузкой.

Приложение предусмотрено для логистических компаний, которые хотят систематизировать получение информации о логистических процессах, а именно контроль количества, передвижения и списания продукции. Администратор компании может зарегистрировать свою компанию и весь штат сотрудников и складов с товарами. При отправлении заказов клиентами сотрудниками компании составляется оптимальный план доставки товаров. Так же предоставляется полный отчёт об операциях с товаром в том числе и акты списания.

В рамках данного дипломного проекта ставятся следующие задачи:

* разработать модель автоматизированной системы: произвести анализ предметной области и выбор технических средств для реализации проекта;
* разработать программные модули системы по автоматизации составления накладной по доставке;
* разработать автоматизацию сбора информации о товарах;
* экономически обосновать разработку и внедрение автоматизированной системы.

1. **Проектирование структуры информационной системы**
   1. **Организационно-экономическая сущность задачи**

Разрабатываемая информационная система предназначена для автоматизации составления накладной доставки товаров заказчикам. Сама система представляет собой сайт для транспортных компаний, которые могут зарегистрироваться в системе, добавить своих сотрудников, склады, транспорт и товары. Также в системе для каждой роли сотрудника реализованы соответствующие ему возможности управления.

Реализовано несколько режимов работы информационной системы. Режим работы системного администратора имеет возможность регистрации транспортных компаний, блокировки компаний, а также просмотра информации о сотрудниках.

Режим работы администратора компании имеет возможность добавления новых сотрудников, складов и транспорта, а также просмотр данных о сотрудниках, как и системный администратор.

Для режима работы диспетчера доступно создание заявок на пополнение товара и заявок пользователей, регистрация нового товара, просмотр заказов на склад, к которому прикреплён диспетчер.

Режим работы логистического специалиста имеет возможность, собирания подтверждённых заказов в накладные и просмотр всех активных накладных. Также для режимов работ диспетчера и логистического специалиста предусмотрена возможность просмотра заказов конечных пользователей.

Водители имеют свой режим работы, который позволяет просматривать активные накладные, а также редактирование их и создание актов списания товаров повреждённых или утерянных во время транспортировки либо хранения.

Ещё одним режимом работы является режим диспетчера, который позволяет просматривать информацию о товарах, расположенных на складах, изменение цен товара, а также за его транспортировку. Также у него имеется возможность просмотра актов списания товара.

Также для каждого режима работы предусмотрена авторизация и выход из аккаунта.

На рисунке 2.1 можно увидеть диаграмму вариантов использования для данной системы.

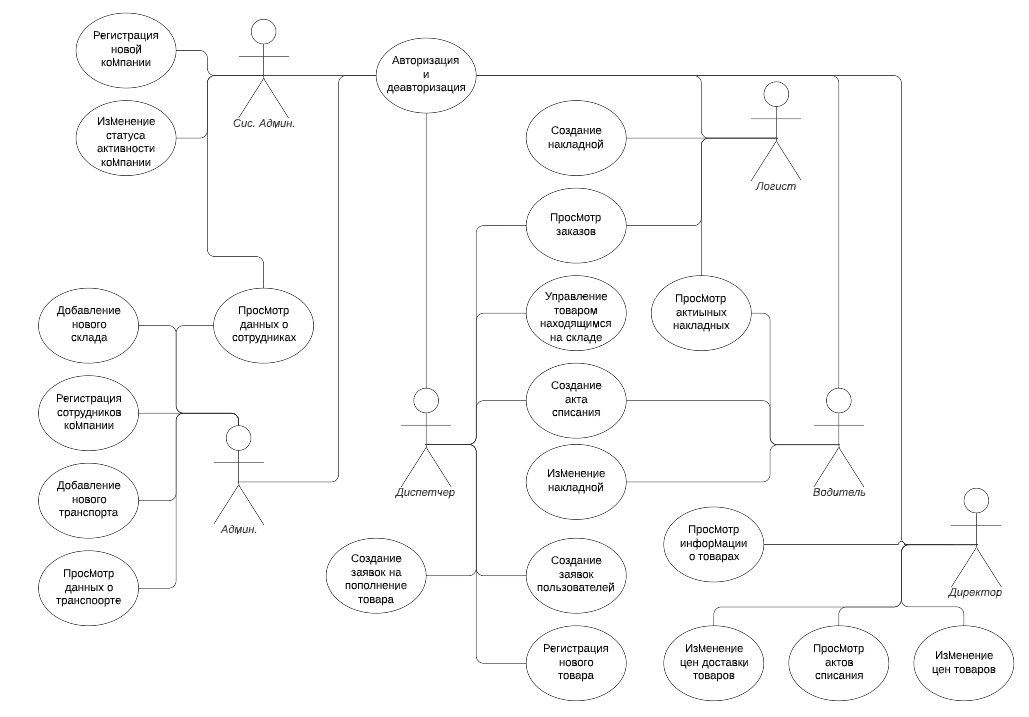


Рисунок 2.1 – Диаграмма вариантов использования

* 1. **Структура информационной системы**

Структуру данной информационной системы можно условно разбить на четыре основных модуля работы.

Одним из них является управление аккаунтом пользователя. Он включает в себя реализацию авторизации и деавторизации пользователя, регистрации нового пользователя, редактирование его данных, как например пароля, а также изменение статуса активности пользователя.

К следующему модулю можно отнести управление имуществом компании. К нему относятся такие возможности, как добавление новых складов, товаров и техники компании, просмотр данных о них, редактирование цен на товары, создание актов списания товаров в связи с определённой причиной непригодности или утери товара, просмотр списка актов списания.

Ещё в один модуль можно выделить управление заказами. В него входит создание заказа, его отклонение либо принятие, а также получение подробной информации о заказах пользователей и пополнении количества товара на складе.

Последним является модуль управления накладными. Здесь реализовано создание новых накладных путём объединения заказов в лист доставки, редактирование накладных, а также получение информации об уже выполненных, активных или накладных, находящихся в режиме ожидания.

На рисунке 2.2 изображена структурная схема разрабатываемой системы.

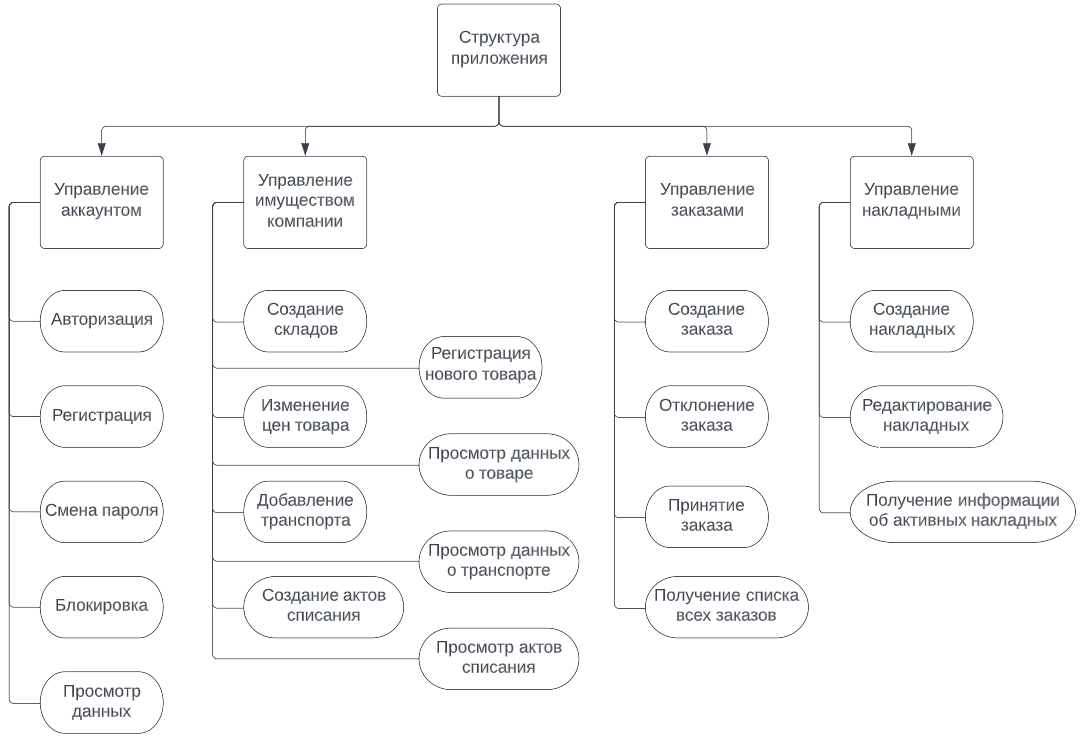


Рисунок 2.2 – Структурная схема приложения

* 1. **Математическое и алгоритмическое обеспечение**

В разрабатываемой информационной системе присутствует автоматизация процесса собирания заказов в накладную путём составления листа доставки. Этот процесс имеет определенный алгоритм работы.

Компании на вход поступают заказы от пользователей на товар, которые в итоге обрабатываются, составляются в накладную и производится их доставка.

Изначально логистический специалист выбирает склад, на который приходили заявки и машину для доставки товара.

Далее он имеет перед собой список принятых диспетчером заявок на этот склад, из которых логист выбирает какие заявки объединить в накладную. При каждом добавлении новой заявки происходит проверка о возможности её добавления путём проверки переполнения вместительности выбранного транспорта.

После выбора заявок происходит составление оптимальной очерёдности доставки. Для этого места доставки сортируются по расстоянию от последнего добавленного места либо начального. После чего место с минимальным расстоянием добавляется в очередь на доставку и удаляется из основного списка. Процесс происходит до тех пор, пока все заказы не добавлены в очередь.

Когда накладная составлена, вычисляется её итоговая выручка путём сложения цен заказов.

Блок-схему алгоритма собирания заказов в товарно-транспортную накладную можно увидеть на рисунке 2.3.

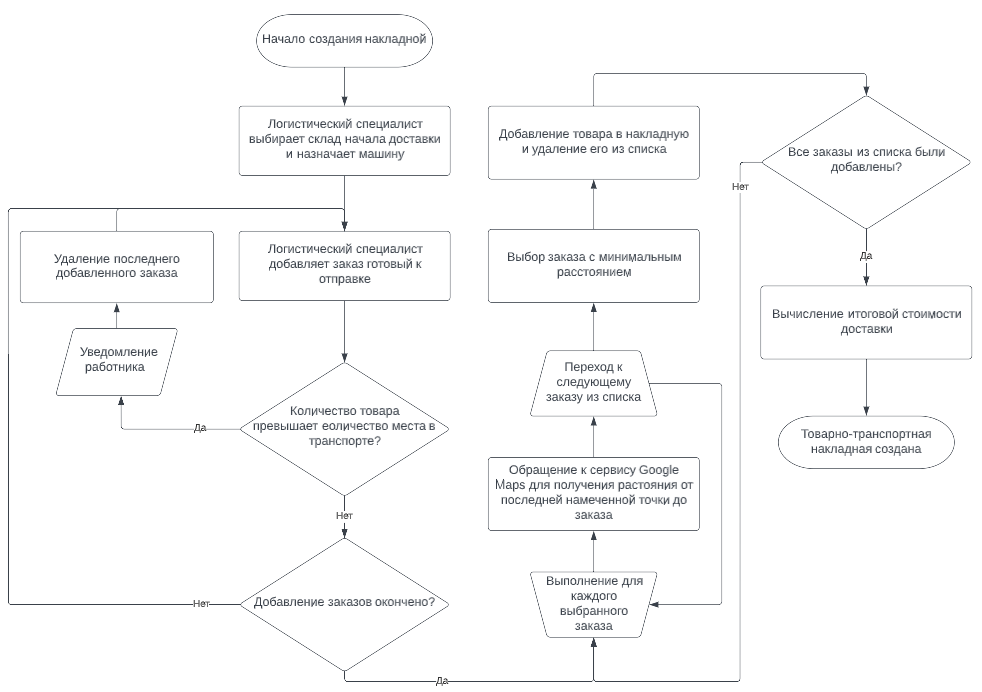


Рисунок 2.3 – Алгоритм создания накладной

* 1. **Информационное обеспечение**

Цель любой информационной системы – обработка данных об объектах реального мира. В широком смысле слова база данных – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.

Определение сущностей и атрибутов: сущность – это субъект, место, вещь, событие или понятие, содержащие информацию. Точнее, сущность – это набор (объединение) объектов, называемых экземплярами. Каждый экземпляр сущности обладает набором характеристик. В логической модели все эти характеристики называются атрибутами сущности.

В таблице 2.1 представлены все сущности, их атрибуты и ключи.

Таблица 2.1 – Сущности, атрибуты и ключи таблиц базы данных

| Сущность | Атрибуты | Ключи |
| --- | --- | --- |
| 1 Адреса | Идентификатор адреса, штат, город, адресная строка 1, адресная строка 2 | Идентификатор адреса |
| 2 Пользователи | Идентификатор пользователя, имя, фамилия, почта, пароль, логин, роль, день рождения, идентификатор адреса, идентификатор компании, статус активности, тип пользователя | Идентификатор пользователя, идентификатор адреса, идентификатор компании |
| 3 Компании | Идентификатор компании, название, дата регистрации, статус активности | Идентификатор компании |
| 4 Склады | Идентификатор склада, название склада, идентификатор адреса, тип склада, максимальная вместительность, текущая вместительность, идентификатор компании | Идентификатор назначения, идентификатор адреса, идентификатор компании |
| 5 Машины | Идентификатор машины, номер, идентификатор последнего адреса, максимальная вместительность, статус, идентификатор компании, текущая вместительность | Идентификатор, машины, идентификатор последнего адреса, идентификатор компании |

Продолжение таблицы 2.1

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибуты | Ключи |
| 6 Категории товаров | Идентификатор категории, название, цена за доставку за километр | Идентификатор категории |
| 7 Товары | Идентификатор товара, уникальный продуктовый код, название, идентификатор категории, занимаемое место, цена, идентификатор компании | Идентификатор товара, идентификатор категории, идентификатор компании |
| 8 Товары на складах | Идентификатор, идентификатор склада, идентификатор товара, количество, статус доступности | Идентификатор, идентификатор склада, идентификатор товара |
| 9 Диспетчеры складов | Идентификатор, идентификатор склада, идентификатор пользователя | Идентификатор, идентификатор склада, идентификатор пользователя |
| 10 Заявки | Идентификатор заявки, номер, идентификатор склада, идентификатор адреса назначения, дата регистрации, дата последнего изменения, идентификатор создателя, идентификатор обновляющего пользователя, статус, идентификатор накладной, направление, очерёдность | Идентификатор заявки, идентификатор склада, идентификатор адреса назначения, идентификатор создателя, идентификатор обновляющего пользователя, идентификатор накладной |
| 11 Товары заявок | Идентификатор, идентификатор заявки, идентификатор товара, количество, цена, занимаемое место | Идентификатор, идентификатор заявки, идентификатор товара |
| 12 Акты списания | Идентификатор акта, дата, идентификатор пользователя, идентификатор склада, идентификатор машины | Идентификатор акта, идентификатор пользователя, идентификатор склада, идентификатор машины |

Продолжение таблицы 2.1

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибуты | Ключи |
| 13 Накладные | Идентификатор накладной, номер, идентификатор склада, идентификатор машины, идентификатор создателя, идентификатор обновляющего пользователя, идентификатор водителя, дата регистрации, дата последнего изменения, состояние | Идентификатор накладной, идентификатор склада, идентификатор машины, идентификатор создателя, идентификатор обновляющего пользователя, идентификатор водителя |
| 14 Товары актов списания | Идентификатор, идентификатор акта, идентификатор товара, количество, причина | Идентификатор, идентификатор акта, идентификатор товара |

Модель является графическим представлением реально реализованной базы данных, то есть каждой сущности соответствует таблица базы данных, а каждому атрибуту – поле таблицы. Ниже приводится описание атрибутов сущностей и их характеристика.

Сущность «Адреса» предназначена для хранения информации об адресах складов компаний, мест доставки, а также адресов последней активности транспорта. Описание атрибутов представлено в таблице 2.2.

Таблица 2.2 – Атрибуты сущности «Адреса»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор адреса | *bigint* | Уникальный ключ |
| 2 Штат | *varchar*(100) | Название штата |
| 3 Город | *varchar*(100) | Название города |
| 4 Адресная строка 1 | *varchar*(255) | Основная адресная информация |
| 5 Адресная строка 2 | *varchar*(255) | Дополнительная информация к адресу |

Сущность «Пользователи» предназначена для хранения информации об всех пользователях, зарегистрированных в системе включая системного администратора. Описание атрибутов представлено в таблице 2.3.

Таблица 2.3 – Атрибуты сущности «Пользователи»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор пользователя | *bigint* | Уникальный ключ |
| 2 Имя | *varchar*(100) | Имя пользователя |
| 3 Фамилия | *varchar*(100) | Фамилия пользователя |
| 4 Почта | *varchar*(100) | Почта пользователя |
| 5 Пароль | *varchar*(255) | Пароль в формате *BCrypt* |
| 6 Логин | *varchar*(50) | Логин пользователя для авторизации |
| 7 Роль | *varchar*(50) | Роль пользователя в компании |
| 8 День рождения | *date* | День рождения пользователя |
| 9 Идентификатор адреса | *bigint* | Идентификатор адреса пользователя |
| 10 Идентификатор компании | *bigint* | Идентификатор компании пользователя |
| 11 Статус активности | *varchar*(50) | Хранит значение активен ли аккаунт |
| 12 Тип пользователя | *varchar*(50) | Системный администратор либо другой пользователь |

Сущность «Компании» предназначена для хранения информации о транспортных компаниях. Описание атрибутов представлено в таблице 2.4.

Таблица 2.4 – Атрибуты сущности «Компании»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор компании | *bigint* | Уникальный ключ |
| 2 Название | *varchar*(200) | Название компании |
| 3 Дата регистрации | *date* | Дата регистрации компании на сайте |
| 4 Статус активности | *varchar*(50) | Хранит значение активна ли компания |

Сущность «Склады» предназначена для хранения информации о складах компании. Описание атрибутов представлено в таблице 2.5.

Таблица 2.5 – Атрибуты сущности «Склады»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор склада | *bigint* | Уникальный ключ |
| 2 Название | *varchar*(45) | Название склада |
| 3 Идентификатор адреса | *bigint* | Идентификатор адреса расположения склада |
| 4 Тип склада | *varchar*(100) | Тип склада |
| 5 Максимальная вместительность | *int* | Хранит количество мест, на которое рассчитан склад |
| 6 Текущая вместительность | *int* | Хранит количество свободных мест |
| 7 Идентификатор компании | *bigint* | Идентификатор владеющей компании |

Сущность «Машины» предназначена для хранения информации о транспорте компании. Описание атрибутов представлено в таблице 2.6.

Таблица 2.6 – Атрибуты сущности «Машины»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор машины | *bigint* | Уникальный ключ |
| 2 Номер | *varchar*(10) | Номер машины |
| 3 Идентификатор последнего адреса | *bigint* | Идентификатор адреса последней активности |
| 4 Максимальная вместительность | *int* | Хранит количество мест, на которое рассчитана машина |
| 5 Текущая вместительность | *int* | Хранит количество свободных мест |
| 6 Идентификатор компании | *bigint* | Идентификатор владеющей компании |

Сущность «Категории товаров» предназначена для хранения информации о категориях товаров компании, размещённых на складах. Описание атрибутов представлено в таблице 2.7.

Таблица 2.7 – Атрибуты сущности «Категории товаров»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор категории | *bigint* | Уникальный ключ |
| 2 Название | *varchar*(50) | Название категории |
| 3 Цена за доставку | *double* | Цена за доставку категории товара за километр |

Сущность «Товары» предназначена для хранения информации о товарах, которые продаёт компания. Описание атрибутов представлено в таблице 2.8.

Таблица 2.8 – Атрибуты сущности «Товары»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор товара | *bigint* | Уникальный ключ |
| 2 Уникальный продуктовый код | *bigint* | Уникальный международный код продукта |
| 3 Название | *varchar*(50) | Название товара |
| 4 Занимаемое место | *int* | Хранит занимаемые единицы места |
| 5 Цена | *double* | Хранит количество свободных мест |
| 6 Идентификатор компании | *bigint* | Идентификатор владеющей компании |

Сущность «Товары на складах» предназначена для хранения информации о товарах, размещённых на конкретных складах. Описание атрибутов представлено в таблице 2.9.

Таблица 2.9 – Атрибуты сущности «Товары на складах»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор | *bigint* | Уникальный ключ |
| 2 Идентификатор склада | *bigint* | Идентификатор склада, где размещён товар |
| 3 Идентификатор товара | *bigint* | Идентификатор размещённого товара |
| 4 Количество | *int* | Текущее количество доступного товара |

Продолжение таблицы 2.9

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание |
| 5 Статус доступности | *varchar*(50) | Хранит информацию доступен ли товар |

Сущность «Диспетчеры складов» предназначена для хранения информации о диспетчерах, закреплёнными за конкретными складами. Описание атрибутов представлено в таблице 2.10.

Таблица 2.10 – Атрибуты сущности «Диспетчеры складов»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор | *bigint* | Уникальный ключ |
| 2 Идентификатор склада | *bigint* | Идентификатор склада компании |
| 3 Идентификатор пользователя | *bigint* | Идентификатор закреплённого диспетчера компании |

Сущность «Заявки» предназначена для хранения информации о входящих и исходящих заявках на продукцию. Описание атрибутов представлено в таблице 2.11.

Таблица 2.11 – Атрибуты сущности «Заявки»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор заявки | *bigint* | Уникальный ключ |
| 2 Номер | *bigint* | Номер заявки внутри компании |
| 3 Идентификатор склада | *bigint* | Идентификатор склада, на который поступила заявка |
| 4 Идентификатор адреса назначения | *bigint* | Идентификатор адреса доставки товаров |
| 5 Дата регистрации | *date* | Дата создания заявки |
| 6 Дата последнего изменения | *date* | Дата последнего редактирования заявки |
| 7 Идентификатор создателя | *bigint* | Идентификатор создателя заявки |
| 8 Идентификатор обновляющего пользователя | *bigint* | Идентификатор пользователя последнего изменения |
| 9 Статус | *varchar*(50) | Статус рассмотрения |

Продолжение таблицы 2.11

|  |  |  |
| --- | --- | --- |
| 10 Идентификатор накладной | *bigint* | Идентификатор накладной |
| 11 Статус направления | *bit*(1) | Является ли заявка исходящей |
| 12 Очерёдность | *int* | Позиция в накладной |

Сущность «Товары заявок» предназначена для хранения информации о товарах, которые содержит конкретная заявка. Описание атрибутов представлено в таблице 2.12.

Таблица 2.12 – Атрибуты сущности «Товары заявок»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор | *bigint* | Уникальный ключ |
| 2 Идентификатор заявки | *bigint* | Идентификатор заявки |
| 3 Идентификатор товара | *bigint* | Идентификатор товара в заявке |
| 4 Количество | *int* | Количество товара в заявке |
| 5 Цена | *double* | Цена с учётом налогов |
| 6 Занимаемое место | *int* | Суммарное количество занимаемых товарами единиц места |

Сущность «Акты списания» предназначена для хранения информации об актах списания товара во время транспортировки. Описание атрибутов представлено в таблице 2.13.

Таблица 2.13 – Атрибуты сущности «Акты списания»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор акта | *bigint* | Уникальный ключ |
| 2 Дата | *date* | Дата списания |
| 3 Идентификатор пользователя | *bigint* | Идентификатор составителя акта |
| 4 Идентификатор склада | *bigint* | Идентификатор склада, с которого был списан товар |
| 5 Идентификатор машины | *bigint* | Идентификатор доставляющего транспорта |

Сущность «Накладные» предназначена для хранения информации о накладных доставок. Описание атрибутов представлено в таблице 2.14.

Таблица 2.14 – Атрибуты сущности «Накладные»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор накладной | *bigint* | Уникальный ключ |
| 2 Номер | *varchar*(45) | Условный номер накладной |
| 3 Идентификатор склада | *bigint* | Идентификатор склада начала накладной |
| 4 Идентификатор машины | *bigint* | Идентификатор машины для доставки |
| 5 Идентификатор создателя | *bigint* | Идентификатор составителя накладной |
| 6 Идентификатор обновляющего пользователя | *bigint* | Идентификатор пользователя последнего изменения |
| 7 Идентификатор водителя | *bigint* | Идентификатор закреплённого водителя |
| 8 Дата регистрации | *date* | Дата составления накладной |
| 9 Дата последнего изменения | *date* | Дата последнего изменения накладной |
| 10 Состояние | *varchar*(50) | Статус накладной |

Сущность «Товары актов списания» предназначена для хранения информации об списанных товарах и соответствующим им актам списания. Описание атрибутов представлено в таблице 2.15.

Таблица 2.15 – Атрибуты сущности «Товары актов списания»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| 1 Идентификатор | *bigint* | Уникальный ключ |
| 2 Идентификатор акта | *bigint* | Идентификатор акта списания |
| 3 Идентификатор товара | *bigint* | Идентификатор списанного товара |
| 4 Количество | *int* | Количество списанного товара |
| 5 Причина | *varchar*(20) | Причина списания товара |

Схему созданной базы данных, включающую в себя таблицы, их подробную информацию и связи между таблицами можно увидеть на рисунке 2.4.

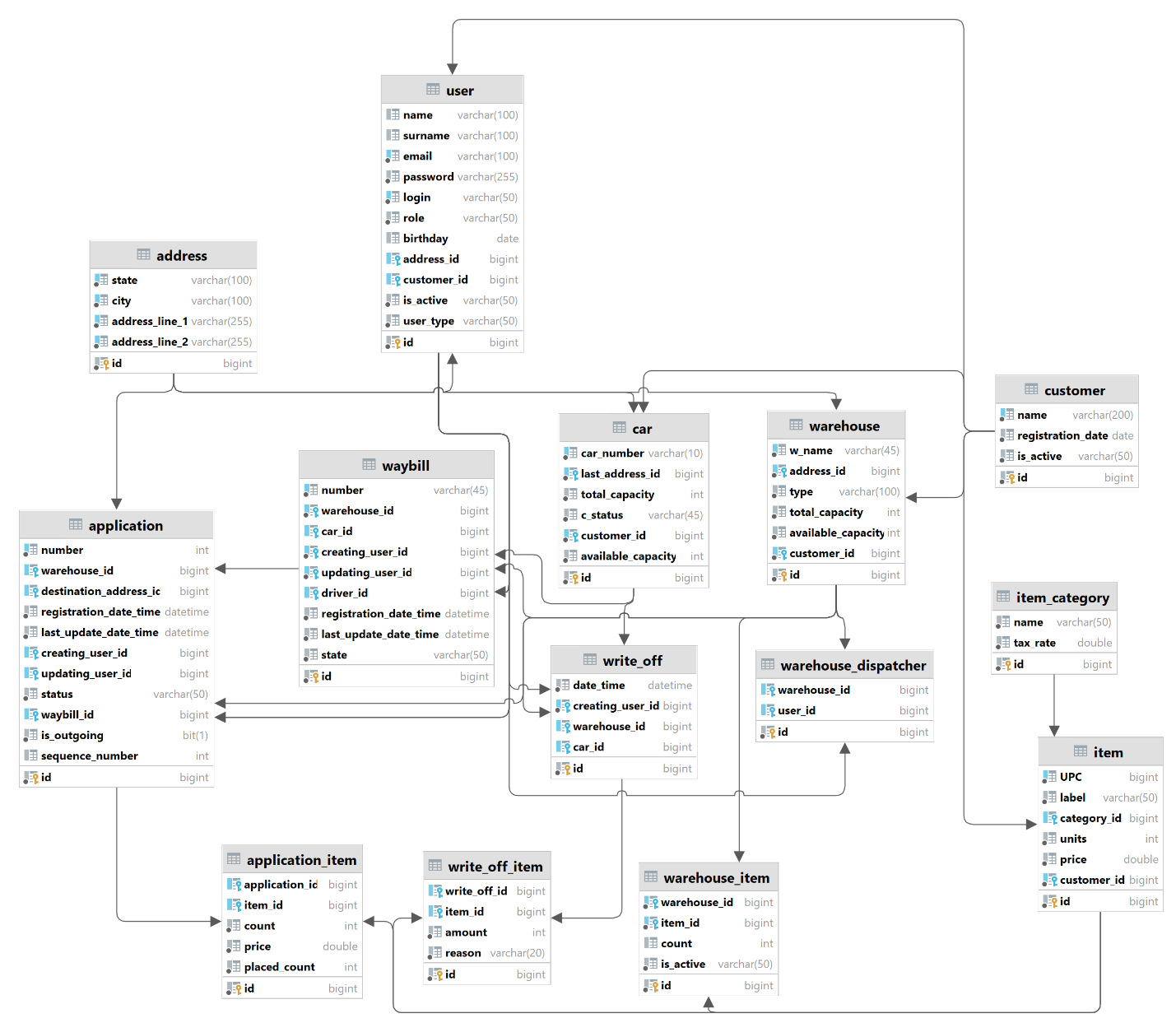


Рисунок 2.4 – Схема базы данных

* 1. **Техническое и системное программное обеспечение**

Для работы с автоматизированной информационной системой тюнинговой компании необходимо знать технические и программные средства, которые необходимы для ее функционирования. Должны присутствовать знания о средствах, как со стороны клиента, так и со стороны сервера.

Для обеих частей приложения необходимо иметь *IntelliJ IDEA* для запуска веб-приложения. Для него есть следующие требования:

* 1 Гб оперативной памяти;
* место на диске не менее 300 Мб и 1 Гб для кэша;
* при установке дополнительных патчей или чего-то подобного может потребовать больше места на диске (больше 1 Гб минимум);
* *Windows* *XP*, 2003, *Vista* 7, 8, 10, 11 (64-*bit* включительно).
* *Mac OS X* 10.5 и выше
* *Linux* *GNOME* или *KDE*
* Минимальное разрешение экрана – 1024*x*768

Самым важным требованием для клиента является браузер с доступом к Интернету, а также установленные пакеты *Node.js* не ниже версии 14.0.0 и *npm* не ниже версии 5.6.

Для работы сервера приложения также необходим дистрибутив технологий *Java Development Kit* версии 17 для компилирования и сборки приложения.

* 1. **Эргономическое обеспечение**

Автоматизированная информационная система имеет вид одностраничного приложения, которое состоит из компонентов. Все страницы имеют один похожий стиль, который подразумевает наличие в самом верху каждой страницы навигационной панели шириной на весь экран, которая содержит ссылки для переходя к другой странице, к которой пользователь имеет доступ, кнопку для перехода на страницу авторизации и, если пользователь уже авторизовался, кнопку для выхода из аккаунта.

После навигационной панели расположена область для расположения всего контента каждой страницы.

Страницы с предоставлением какой-либо информации пользователю включают в себя таблицу с данными. Для удобства поиска в этой таблице предусмотрены соответствующие фильтры данных, а также порядковая нумерация страниц данных. Процесс извлечения данных определённой страницы таблицы происходит со стороны сервера для обеспечения более быстрой работы приложения.

Страницы, предполагающие собой предоставление управления данными, включают в себя формы для соответствующего взаимодействия с системой. Формы открываются во всплывающем модальном окне для более удобного восприятия пользователем. Страницы с возможностью изменения адреса заказов также имеют специальные поля в форме, представляющие собой карту местности. С данной картой пользователь может взаимодействовать для изменения точки доставки, а также изменения масштаба карты для более детального просмотра.

* 1. **Организационное обеспечение**

В автоматизированной информационной системе транспортной компании все пользователи имеют определённые права доступа в зависимости от роли в компании. Им доступны те страницы, через которые можно взаимодействовать непосредственно с необходимыми модулями приложения. Также создан отдельный аккаунт для системного администратора, который имеет наивысший уровень доступа для контроля за компаниями и их сотрудниками.

Одними из основных способов для взлома сайтов являются *SQL*-инъекции, а также неправильное хранение данных паролей пользователей.

*SQL*-инъекция – это попытка изменить запрос к базе данных. Ввести ее можно через форму или ссылку, которая передает параметры методом *GET*. Решением проблемы является использование метода *POST* для работы с данными, так как при передаче через метод *GET* данных они оказываются на виду у всех пользователей, что позволяет злоумышленнику узнать имена переменных и значения, которые присутствуют в них. Так он сможет выбрать оптимальную переменную для ввода инъекции.

Также во всех полях подразумевающих ввод каких-либо данных реализована валидация этих полей с целью недопущения ввода некорректных данных пользователем. При попытке введение некорректных данных пользователь уведомляется соответствующим сообщением.

Проблема хранения паролей пользователей решается при помощи кодирования пароля. Для того, чтобы не заставлять пользователя запоминать ключ – длинную последовательность цифр, разработаны методы преобразования строки символов любой длины (пароля) в его закодированное значение, которое хранится в базе данных. Сама суть здесь заключается в том, чтобы зашифровать данные таким образом, чтобы любой посторонний человек не смог их прочесть, а тому, кто имеет доступ к этим данным, такая возможность все же оставалась.

1. **Программная реализация информационной системы**
   1. **Выбор программных средств реализации**

Для создания сервера используется язык программирования *Java* с использованием библиотека *Spring Framework*.

Данный проект реализуется на языке программирования *Java*. Так же используется язык разметки *HTML* и язык для определения стилей *CSS*. Данный выбор обусловлен тем, что *Java* является одним из ведущих языков для разработки web-ресурсов. Язык имеет очень большое сообщество, за счёт чего большую поддержку и быстрое эффективное развитие. Бизнес логика реализована с помощью *Java*.

*Java* – это язык программирования общего назначения, который следует парадигме объектно-ориентированного программирования и подходу «Написать один раз и использовать везде». *Java* используется для настольных, сетевых, мобильных и корпоративных приложений.

Также *Java* доказала, что *C* – процедурный, управляемый вручную и зависящий от платформы код – это не предел совершенства. Благодаря *Java*, все больше людей начали применять объектно-ориентированное программирование, которое сейчас используется повсеместно.

Плюсы использования *Java*: объектно-ориентированное программирование, является языком высокого уровня с простым синтаксисом, безопасность, автоматическое управление памятью, многопоточность.

Минусы использования: платное коммерческое использование, низкая производительность, отсутствие нативного дизайна, многословный и сложный код.

*Spring* *Framework* (или коротко *Spring*) – универсальный фреймворк с открытым исходным кодом для *Java*-платформы. Он был впервые выпущен под лицензией *Apache 2.0 license* в июне 2003 года. Первая стабильная версия 1.0 была выпущена в марте 2004.

*Spring* может быть рассмотрен как коллекция меньших фреймворков или фреймворков во фреймворке. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании.

В отличие от других платформ, *Spring* фокусируется на нескольких областях функционирования приложений и предоставляет для них широкий спектр дополнительных функций. А в сочетании со *Spring Boot* можно получить готовый каркас для приложения, позволяющий без особых трудностей добавлять в него наиболее часто используемые компоненты, придерживаясь определенной архитектуры, в которую разработчик встраивает свою функциональность.

Для создания клиента используется язык программирования *JavaScript* с библиотекой *React*.

*JavaScript* – язык программирования, поддерживающий объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации *ECMAScript*.

Данный язык программирования обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основными архитектурными чертами являются динамическая типизация и функции как объекты первого класса.

На *JavaScript* оказали влияние многие языки, при разработке была цель сделать язык похожим на *Java*. Языком *JavaScript* не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке. [4]

*React* – библиотека для языка *JavaScript* с открытым исходным кодом для разработки пользовательских интерфейсов.

Основными преимуществами React являются: простота изучения, высокая гибкость и максимальная отзывчивость, устойчивость к высоким нагрузкам, связывание данных от больших к меньшим, малый вес, простота миграции, открытый исходный код.

*React* может использоваться для разработки одностраничных и мобильных приложений.

В качестве среды для разработки клиентской и серверной частей приложения была выбрана *IntelliJ IDEA*.

*IntelliJ IDEA* – [интегрированная среда разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) для многих языков программирования. Несмотря на то, что *IntelliJ IDEA* – в первую очередь *IDE* для *Java*, она понимает и предоставляет интеллектуальную помощь при написании кода на *SQL*, *JPQL*, *HTML*, *JavaScript* и многих других языках и позволяет редактировать код, написанный не на *Java*, внутри строковых литералов *Java*-кода.

Умный редактор кода в сочетании с эргономичным дизайном делают разработку не только эффективной, но и по-настоящему приятной.

* 1. **Структура программного обеспечения**

Данная автоматизированная система разработана с использованием основных модулей *Spring Framework*, такие как: *Spring Builder*, предназначенный для сборки приложения путём инициализации специальных используемых приложением бинов(объектов); *Spring Security*, осуществляющий контроль над пользователями путем создания им полномочий к конкретным частям реализации; *Spring Data*, которая является связующим звеном между базой данных и сервером, а также предоставляет специальные классы для валидации и обработки информации полученной из базы данных.

Для удобства работы разработчиков также использована библиотека *Liquibase*, представляющая возможность поддерживать одну версию базы данных и один набор информации. В отдельном файле с конфигурацией системы прописаны также необходимые переменные для подключения к базе данных.

Также серверная часть разработана с использованием *MVC* схемы разделения данных приложения, что позволяет удобное восприятие разработчикам, быстрое ориентирование в приложении и поиск информации и соответственно ускорение рабочего процесса.

Клиентская часть реализована с использованием одной основной библиотеки *ReactJS*, которая даёт большое количество возможностей разработчику для создания веб-приложения с использованием функционального программирования.

Система поддерживает постоянный контроль об ошибках который передаётся на интерфейс системы, а также отображается в приложении сервера и записывается в файл для долгосрочного хранения.

Также система содержит модуль отправки писем пользователем от главной почты системы, конфигурация которой также прописана в упомянутом выше файле. Отправка писем происходит путём отправки *smtp* протокола.

Приложение условно разбивается на несколько модулей, которые взаимодействую друг с другом и представляют собой конечную систему. Структурную схему данного программного обеспечения, включающего в себя модули приложения, библиотеки и файлы ресурсов можно увидеть на рисунке 3.1.

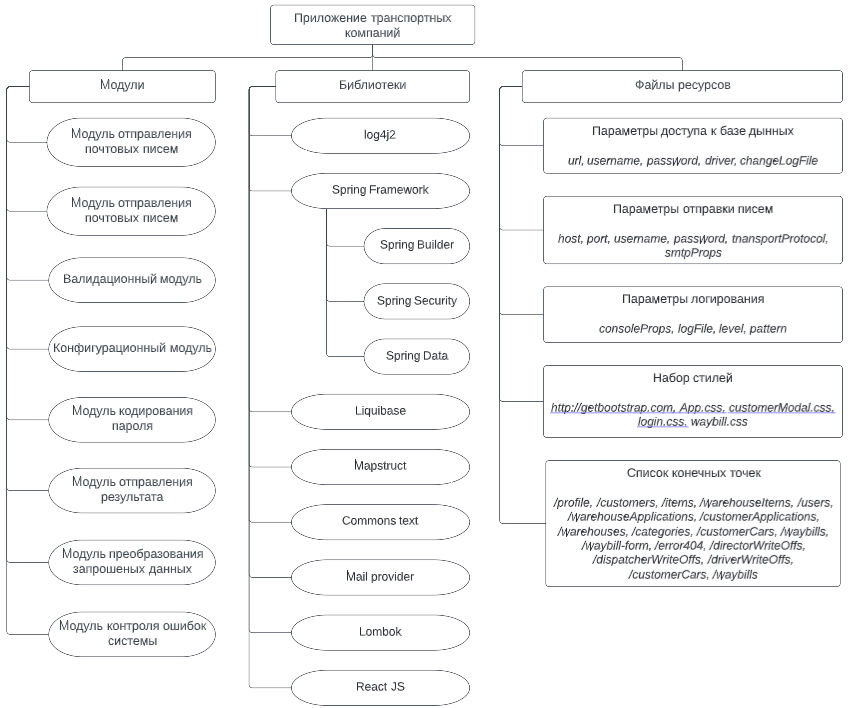


Рисунок 3.1 – Структурная схема программного обеспечения

* 1. **Разработка программного кода**

Рассмотрим разработку программного кода на основе регистрационной части приложения.

Клиентская и серверная части общаются между собой путём отправки запросов на получение определённых данных, которые сервер в свою очередь получает из базы данных и обрабатывает.

Для регистрации новой компании в системе с клиентской части отправляется запрос с формой, содержащей в себе название компании, а также почтовый ящик администратора, на который отправляется пароль для авторизации.

Серверная часть принимает запрос и производит проверку введённых данных. Изначально программный код выгружает данные о зарегистрированных компаниях из базы данных, и проверяет не существует ли уже компания с таким названием либо почтой. Далее производится проверка правильности почтового ящика путем проверки с заранее заготовленным шаблоном. При возникновении какой-либо ошибки или некорректности введённых данных происходит автоматическая отправка ответа клиентской части с передачей текста ошибки, которая в дальнейшем отображается на сайте пользователю.

После проверки правильности введённых данных происходит процесс добавления новой записи в базу данных. Дата регистрации компании имеет значение даты отправления запроса, а статус активности показывает, что компания изначально не заблокирована. В списке сотрудников компании создаётся только администратор компании, логином которого изначально является почтовый ящик компании, а пароль случайно сгенерирован системой.

После успешного добавления данных новой компании и её администратора в базу данных на почтовый ящик, который был указан, произведётся отправка письма, содержащее сгенерированный пароль.

Серверной части в случае безошибочной отработки программного кода передаётся успешное сообщение и происходит перенаправление на станицу со всеми компаниями.

В дальнейшем администратор, пройдя процесс авторизации, может производит аналогичную регистрацию сотрудников компании указывая их персональные данные, а также изменить свои в разделе личного кабинета, в том числе логин и изначально сгенерированный пароль.

* 1. **Тестирование приложения**

В настоящее время, для проверки работоспособности приложения используется тестирование. Тестирование направлено на исследование программного обеспечения, с целью получения информации о надёжности продукта и доказательства того, что программа соответствует заявленным требованиям, что позволяет найти несоответствия разработанного функционала и устранить их.

Тестирование является важным этапом разработки приложения, оно позволяет при написании программного кода обнаружить сценарии, которые могут привести к ошибкам во время работы программы. В некоторых случаях добавление нового функционала ведет к поломкам старого, и в такой ситуации разработчик может об этом даже не узнать. Для предотвращения данной проблемы проведено регрессионное тестирование приложения.

Регрессионное тестирование – это вид тестирования программного обеспечения, позволяющий обнаружить ошибки в уже протестированном ранее функционале. Цикл регрессионного тестирования представлен на рисунке 3.2.



Рисунок 3.2 – Цикл регрессионного тестирования

При написании исходного кода тестировались:

* модули программного средства;
* взаимодействия модулей;
* цельное приложение.

Данные этапы являются важными и связанными между собой, ни один из них невозможно исключить.

Также проведено юнит-тестирование и ручное тестирование веб-приложения.

Юнит-тестирование – процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки.

Ручное тестирование – процесс не автоматического тестирования, проводимый тестировщиком без использования программных средств, для проверки программы или сайта путём моделирования действий пользователя. В роли тестировщиков могут выступать и обычные пользователи, сообщая разработчикам о найденных ошибках.

* 1. **Руководство пользователя**

Впервые зайдя на сайт, пользователь увидит страницу авторизации, которая включает в себя форму для ввода логина и пароля. После введения своих данных и нажатия на кнопку «*Log in*» происходит проверка введённых данных и в случае ошибки пользователь уведомляется соответствующим сообщением. Страницу регистрации с уведомлением о некорректно введённых данных можно увидеть на рисунке 3.3.

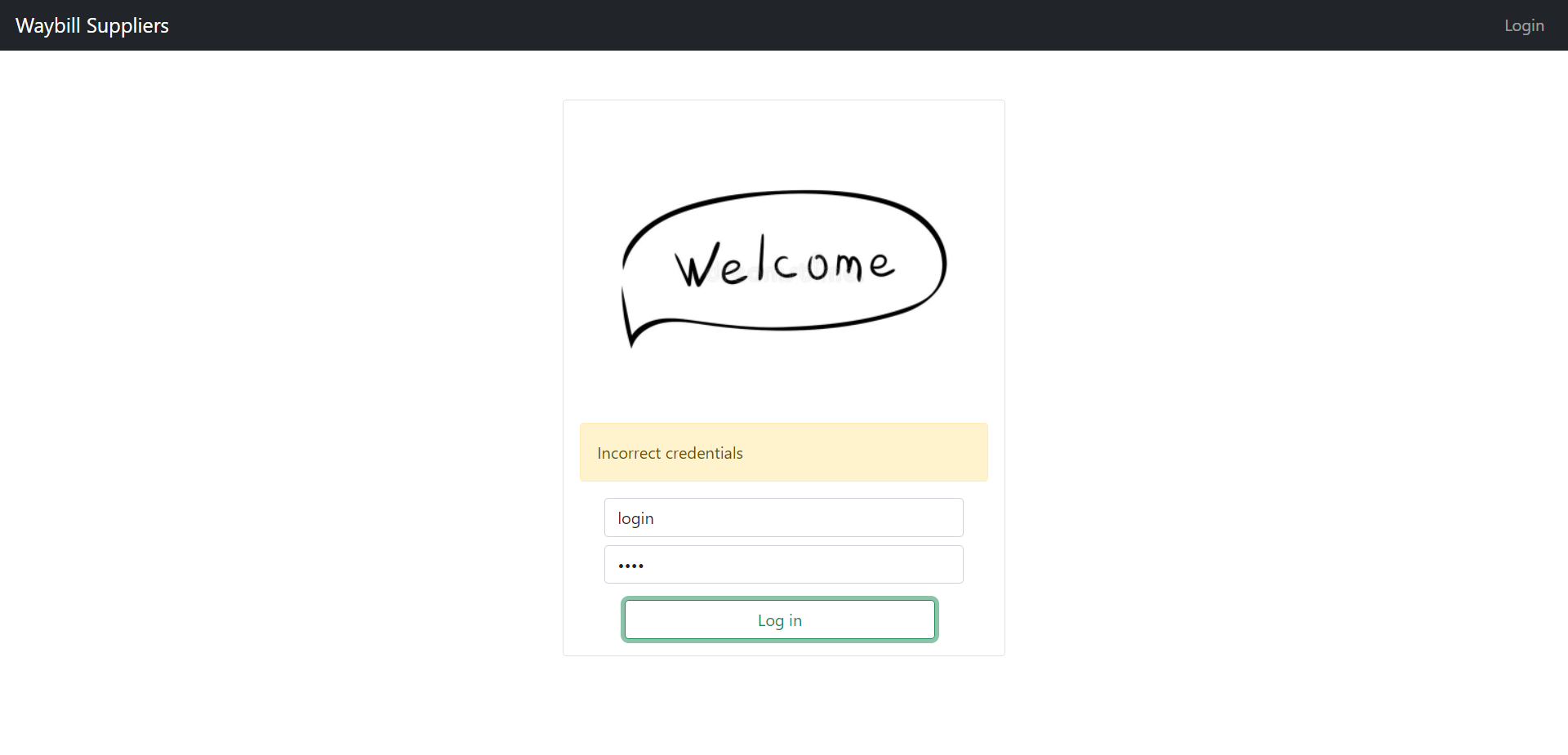


Рисунок 3.3 – Начальная страница после

некорректно введённых данных

В случае же успешного ввода логина и пароля пользователь авторизуется в системе путём создания сессии и сохранения её параметров в сессионном хранилище клиента. Также он перенаправляется на новую страницу в соответствии со своей ролью в системе. Пример данных, находящихся в сессионном хранилище клиента, после успешной авторизации пользователя можно увидеть на рисунке 3.4.

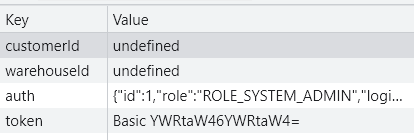


Рисунок 3.4 – Данные созданной сессии пользователя

В любой момент пользователь имеет доступ к выходу из системы. Для этого ему необходимо нажать на соответствующую кнопку в верхнем правом углу. После чего он перенаправляется на страницу с уточнением, действительно ли он хочет выйти из аккаунта. При нажатии на кнопку подтверждения сессия аннулируется, а пользователь перенаправлен на страницу авторизации. Страницу подтверждения выхода из аккаунта можно увидеть на рисунке 3.5.

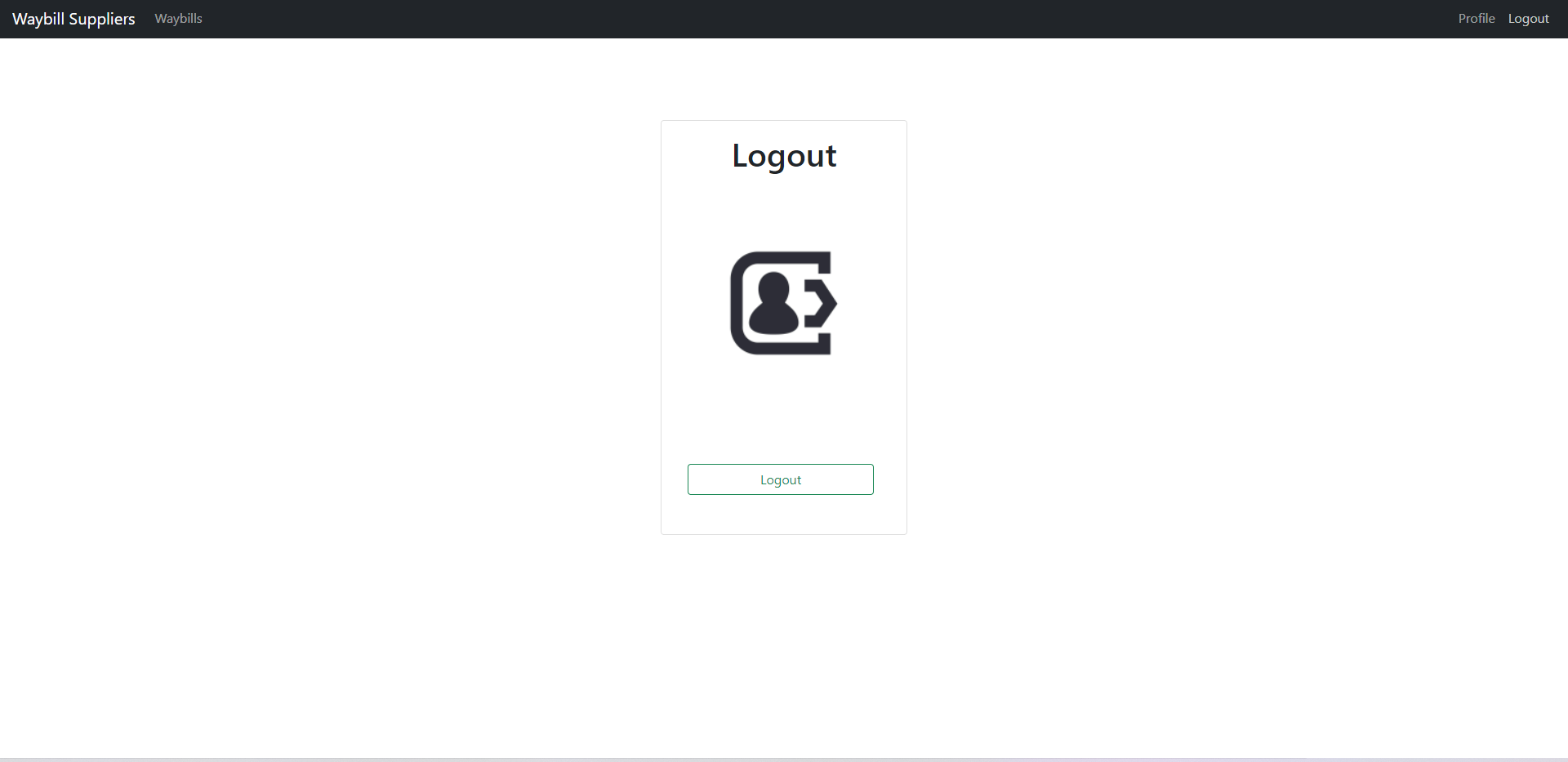


Рисунок 3.5 – Страница подтверждения выхода из системы

Для системного администратора начальной страницей после авторизации является страница с информацией о зарегистрированных компаниях.

Данная страница содержит таблицу данных о компаниях, к которой применён (как и для всех страниц системы) постраничный доступ. Для каждой строки с данными компании существует маркер для пометки, а также под страницей есть кнопка для блокирования либо разблокирования помеченных компаний. Если компания блокируется, то её пользователи автоматически деавторизируются и не имеют возможности к авторизации до момента разблокировки компании.

Страницу с информацией о компаниях можно увидеть на рисунке 3.6

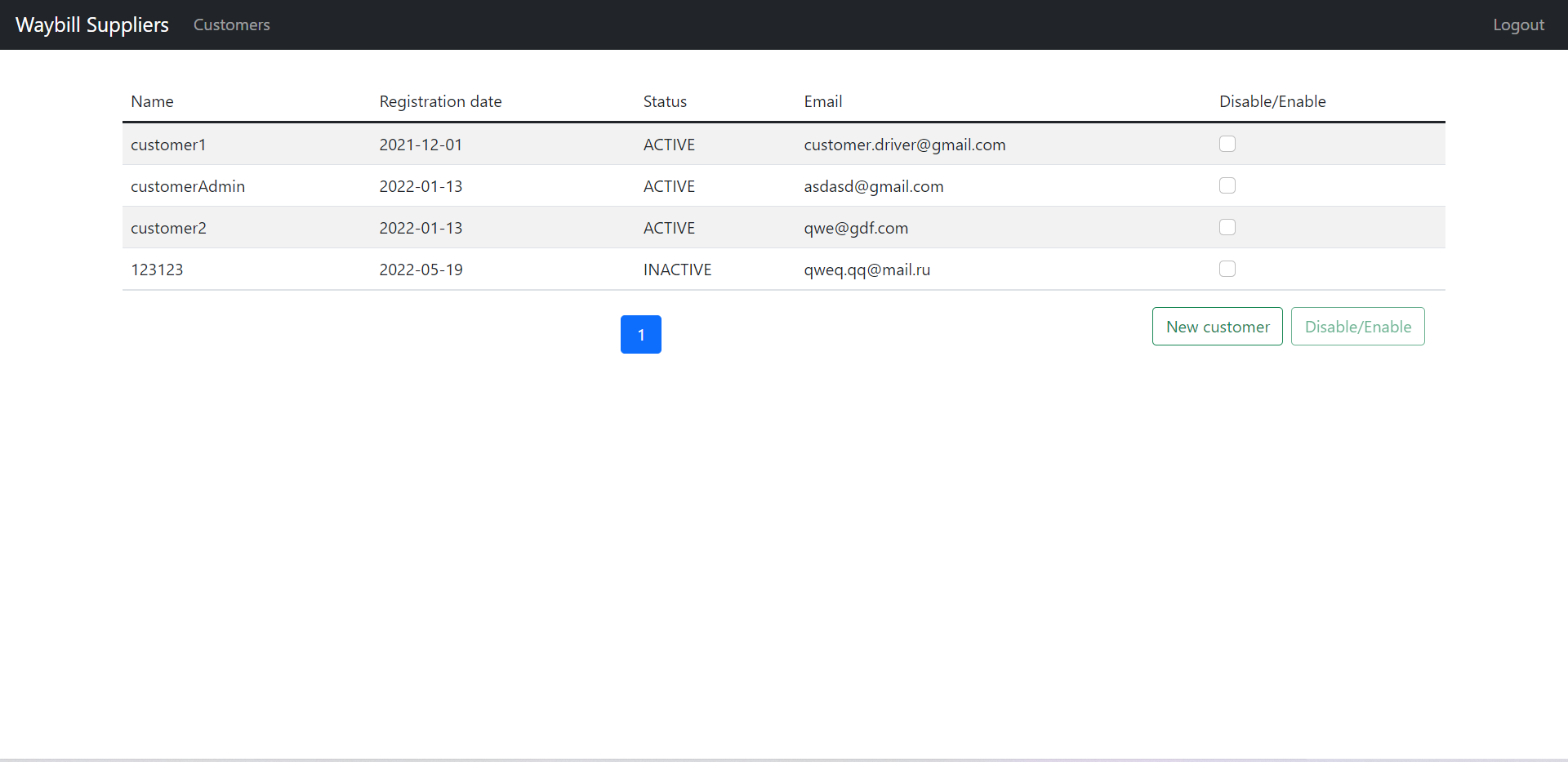


Рисунок 3.6 – Страница данных о компаниях системы

Помимо этого, на странице располагается кнопка для создания нового пользователя. При нажатии на неё появляется модульное окно с формой для ввода данных о новой компании. Данное окно можно увидеть на рисунке 3.7.

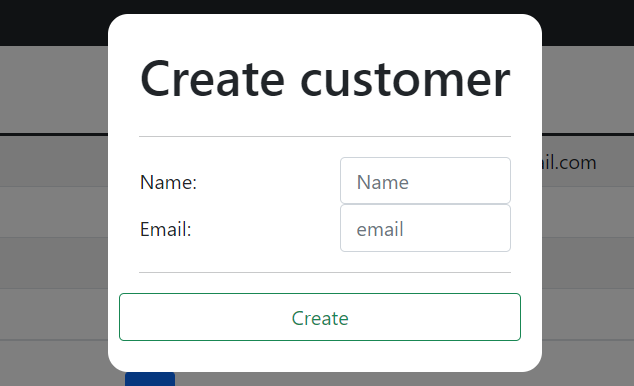


Рисунок 3.7 – Модальное окно для регистрации компании

Любое модальное окно в приложении можно при необходимости закрыть нажав на серую область экрана (вне модального окна).

После ввода данных новой компании и нажатия кнопки «*Create*» в системе регистрируется новая компания, а на указанный почтовый адрес происходит отправка письма с паролем для входа. Пример такого письма можно увидеть на рисунке 3.8.

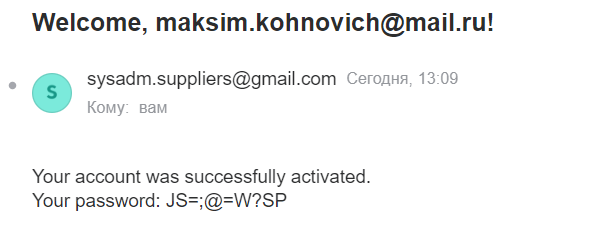


Рисунок 3.8 – Письмо о регистрации компании

Данная почта и пароль являются данными для входа в систему для администратора компании. После входа он перенаправляется на страницу со списком всех сотрудников компании.

На данной страницу присутствует таблица с данными о сотрудниках. Снизу таблицы присутствует специальный маркер для фильтра незаблокированных сотрудников. Также, как и на странице с компаниями для системного администратора, на данной странице для каждого пользователя существуют маркеры для пометки и кнопка под таблицей для блокировки или разблокировки помеченных пользователей.

Под таблицей также присутствует кнопка для регистрации нового сотрудника. После нажатия на неё открывается специальное модальное окно с формой для ввода персональных данных. В данной форме администратор заполняет необходимые поля и нажимает на кнопку «*Create*». При выборе роли диспетчера добавляется специальное поле формы, в котором необходимо выбрать склад, к которому прикрепляется диспетчер. После успешной валидации введённых данных страница обновляется, а на указанную почту высылается письмо с паролем для входа в систему. Письмо является аналогичным письму на рисунке 3.8.

Страницу с данными о сотрудниках, а также форму для регистрации нового сотрудника можно увидеть на рисунках 3.9 и 3.10 соответственно.

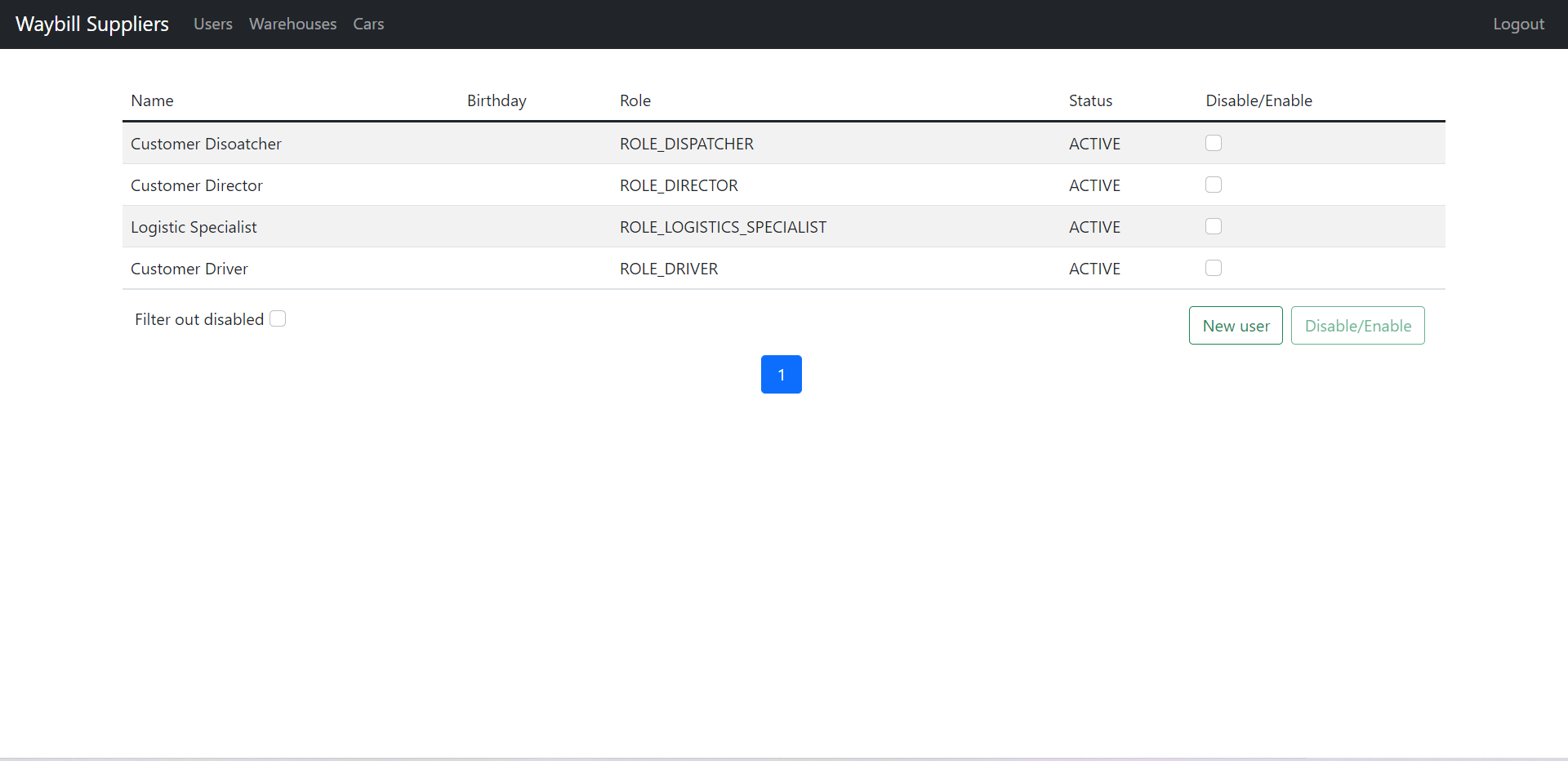


Рисунок 3.9 – Страница с данными о сотрудниках компании

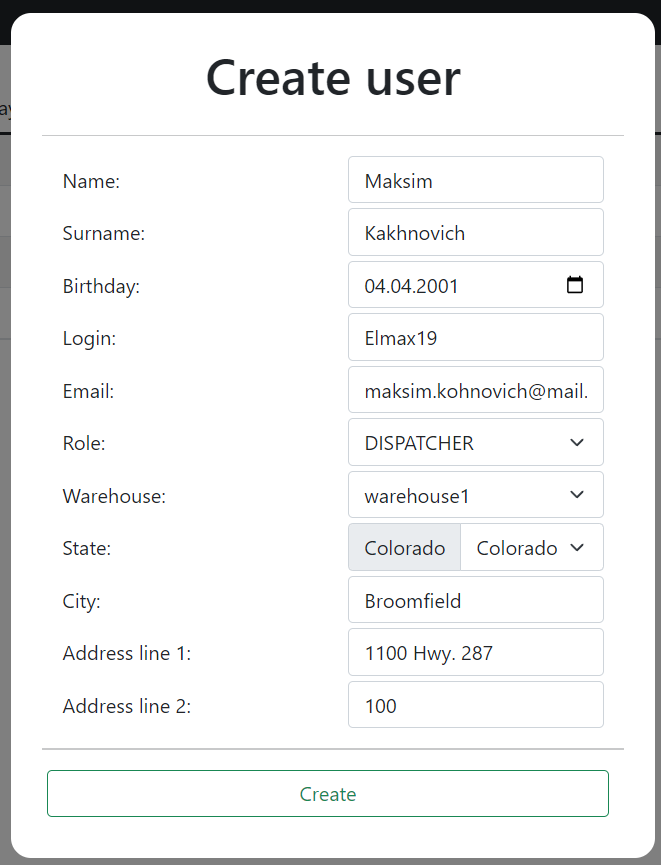


Рисунок 3.10 – Форма создания

нового сотрудника

На панели навигации, расположенной в верхней части странице, администратор может перейти на другие доступные ему страницы.

На странице со складами компании присутствует таблица с данными о складах. Также на странице существуют маркеры для пометки и кнопка под таблицей для удаления помеченных складов компании.

Страницу с данными о складах можно увидеть на рисунке 3.11.

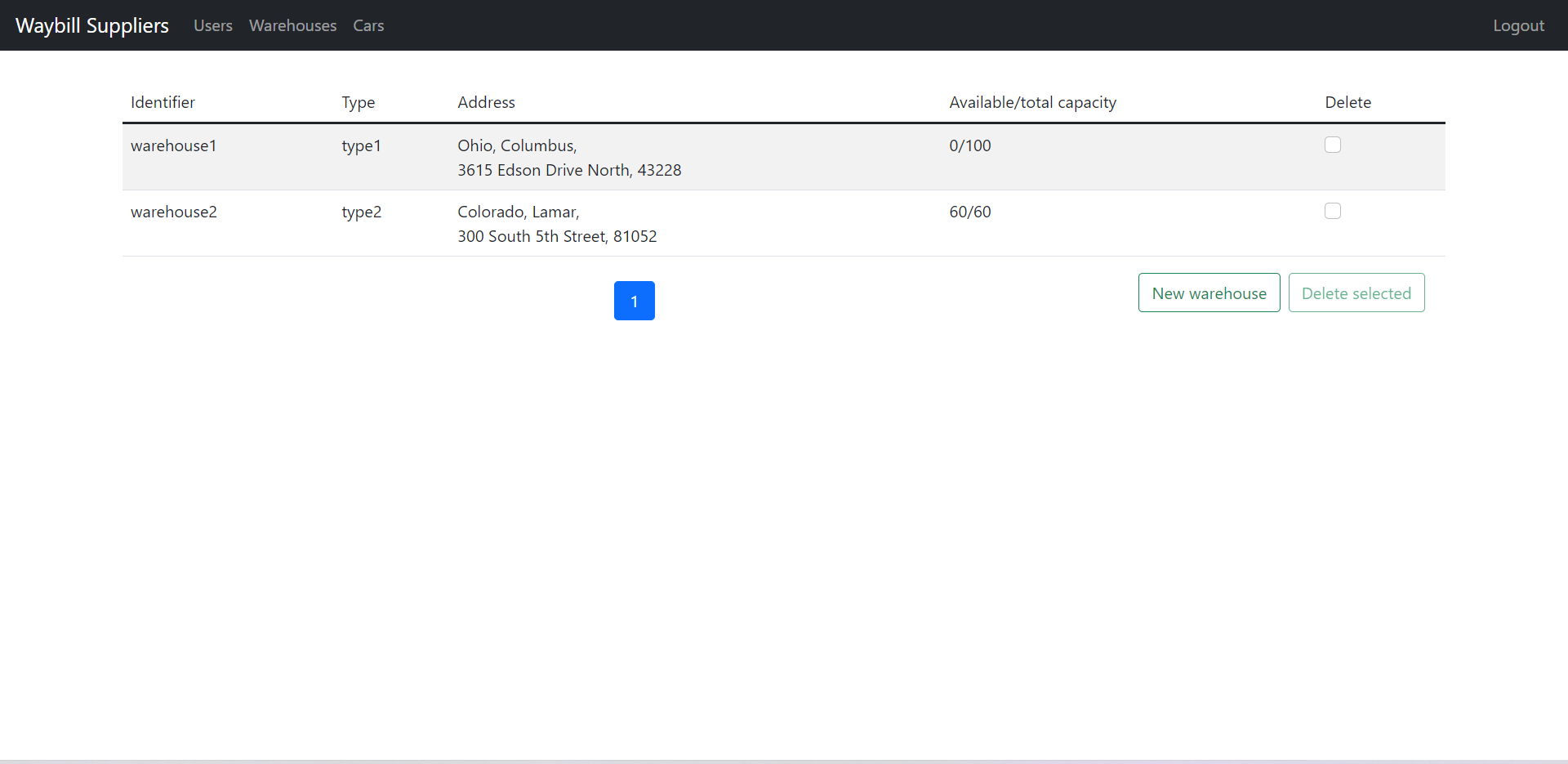


Рисунок 3.11 – Страница с данными о складах компании

Под таблицей также присутствует кнопка для создания нового склада. После нажатия на неё открывается специальное модальное окно с формой для ввода данных. В данной форме администратор заполняет необходимые поля и нажимает на кнопку «*Create*».

Форму для регистрации нового склада можно увидеть на рисунке 3.12.

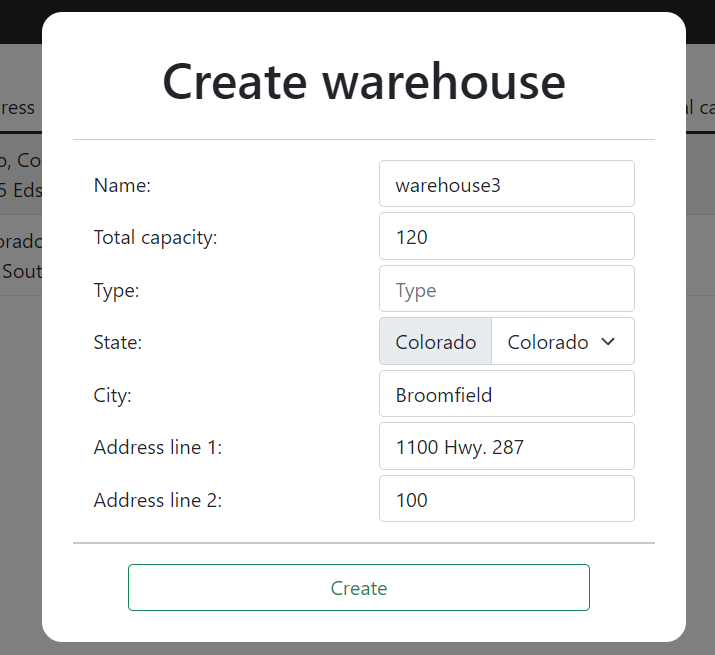


Рисунок 3.12 – Форма регистрации склада

На странице с транспортом компании присутствует таблица с данными о машинах. Также на странице существуют маркеры для пометки и кнопка

под таблицей для удаления помеченного транспорта компании.

Страницу с данными о транспорте можно увидеть на рисунке 3.13.

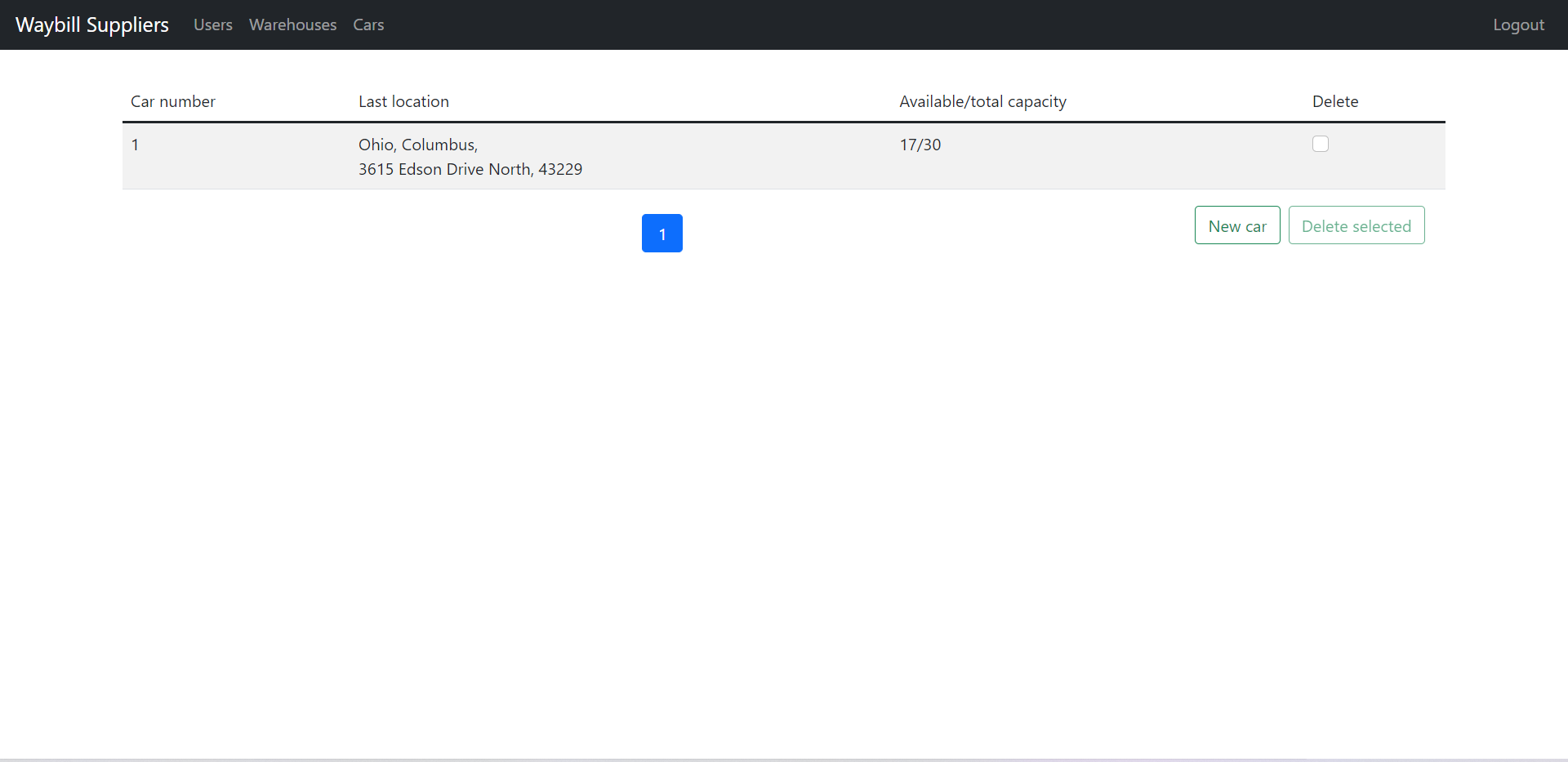


Рисунок 3.13 – Страница с данными о транспорте компании

Под таблицей также присутствует кнопка для создания нового транспорта. После нажатия на неё открывается специальное модальное окно с формой для ввода данных. В данной форме администратор заполняет необходимые поля и нажимает на кнопку «*Create*».

Форму для регистрации нового склада можно увидеть на рисунке 3.14.

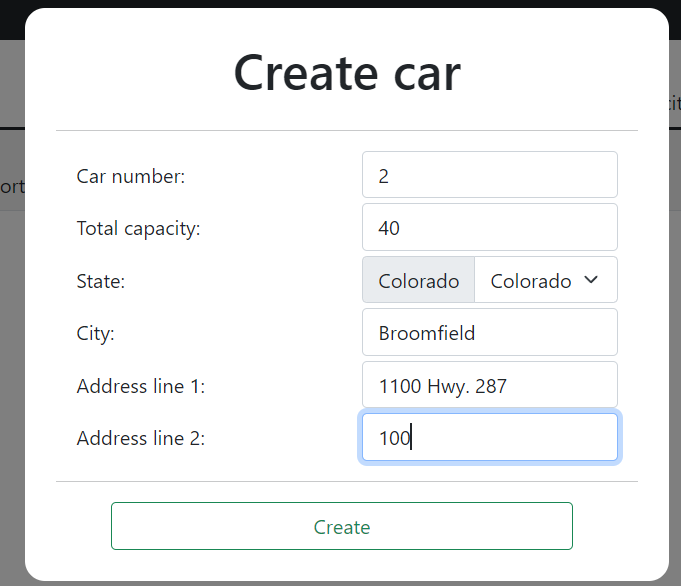


Рисунок 3.14 – Форма регистрации машины

При авторизации сотрудника с ролью диспетчер он перенаправляется на страницу с заявками, приходящими на склад, к которому он прикреплён администратором компании.

Страницу с данными о заявках можно увидеть на рисунке 3.15.

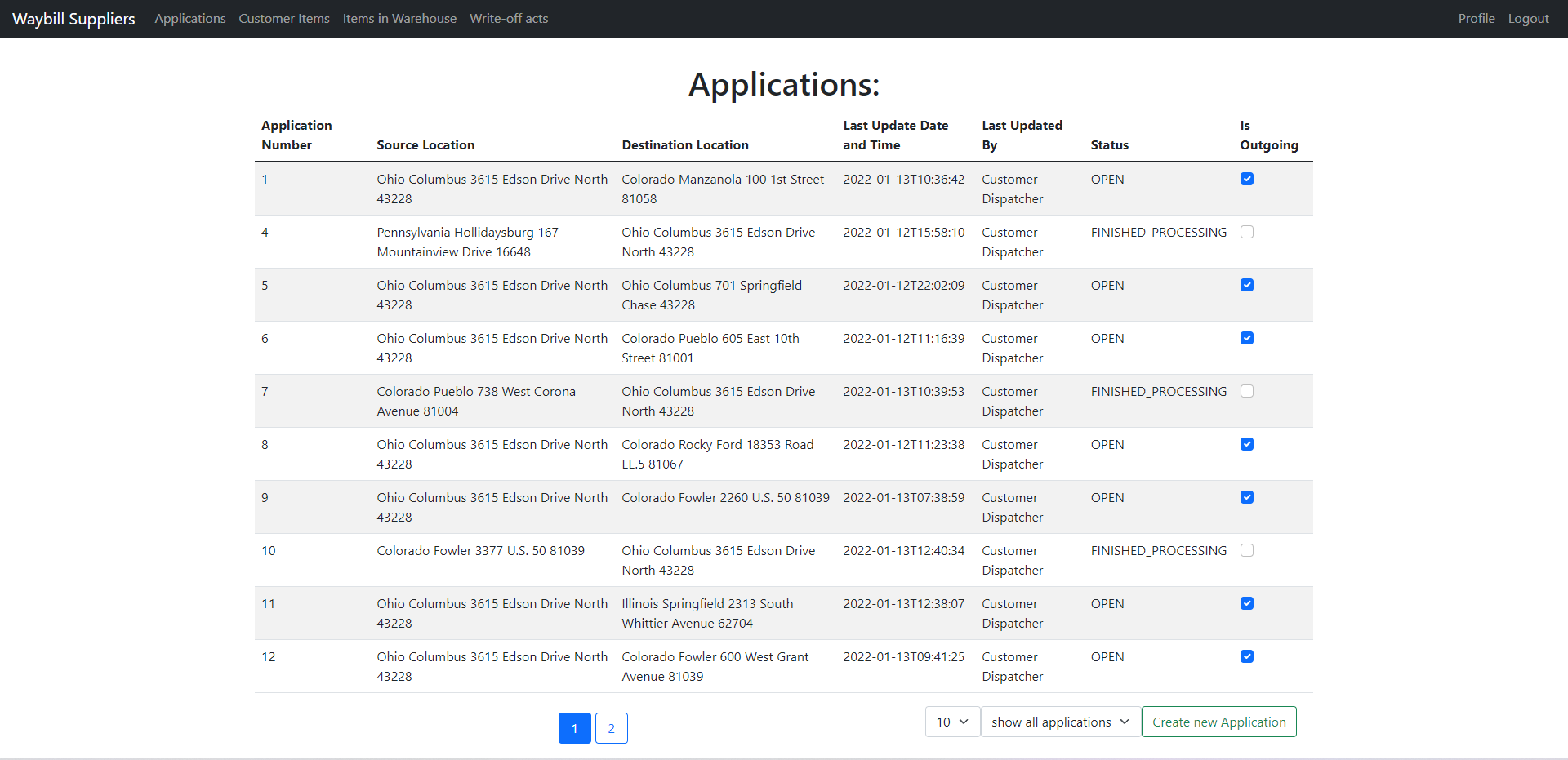


Рисунок 3.15 – Страница с данными о заявках на склад

При нажатии на ячейку таблицы с номером заявки появляется модальное окно с подробной информацией и возможностью её редактировать при условии, что заявка ещё открыта.

Для изменения адреса доставки можно либо найти и нажать на необходимое место на карте, либо ввести его адрес в строке, которая расположена под картой. При попытке ввода неправильного адреса или отметки точки доставки в месте, где нет здания, точка поставлена не будет, а пользователь получит соответствующее уведомление.

Также в данной форме присутствует таблица с товарами, которыми владеет компания. Для добавления товара в заявку необходимо лишь выбрать его уникальный номер, количество и нажать на кнопку «*Add item*». При добавлении нового товара произойдёт автоматический пересчёт количества товаров в заявке, занимаемое место, а также цена с учётом налога доставки в определённый штат.

Последним пунктом в форме является маркер, определяющий является ли заявка на доставку пользователю или на пополнение склада.

После изменение всей информации диспетчер нажимает на кнопку «*Save*». Данные в базе данных обновляются, а страница перезагружается.

Снизу таблицы на странице присутствует кнопка для создания новой заявки, при нажатии на которую появляется форма для создания заявки. Данная форма аналогична форме для редактирования с отличием, что доступно изменение маркера направления доставки (пользователю или на склад).

Пример данной формы можно увидеть на рисунке 3.16.

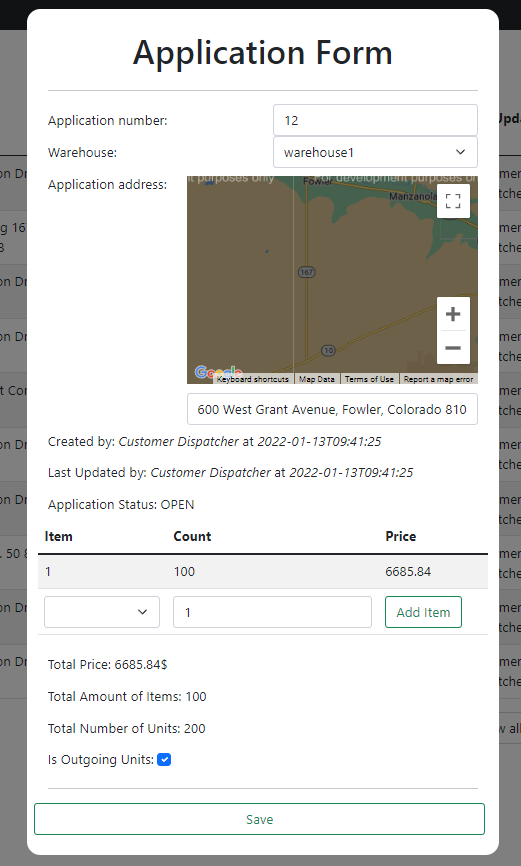


Рисунок 3.16 – Форма

редактирования заявки

Также диспетчер через навигацию может перейти на страницу с данными о товарах, которыми владеет компания. Она также имеет список всех товаров с информацией о ней в виде таблицы, а также маркеры и кнопку для удаления помеченных товаров. Однако товары удаляются лишь в случае, если они не присутствуют в активных заявках.

Для удобства пользователя под таблицей есть выпадающее меню, в котором можно выбрать, сколько строк товара должно располагаться на странице таблицы.

Страницу с информацией о товарах компании можно увидеть на рисунке 3.17.

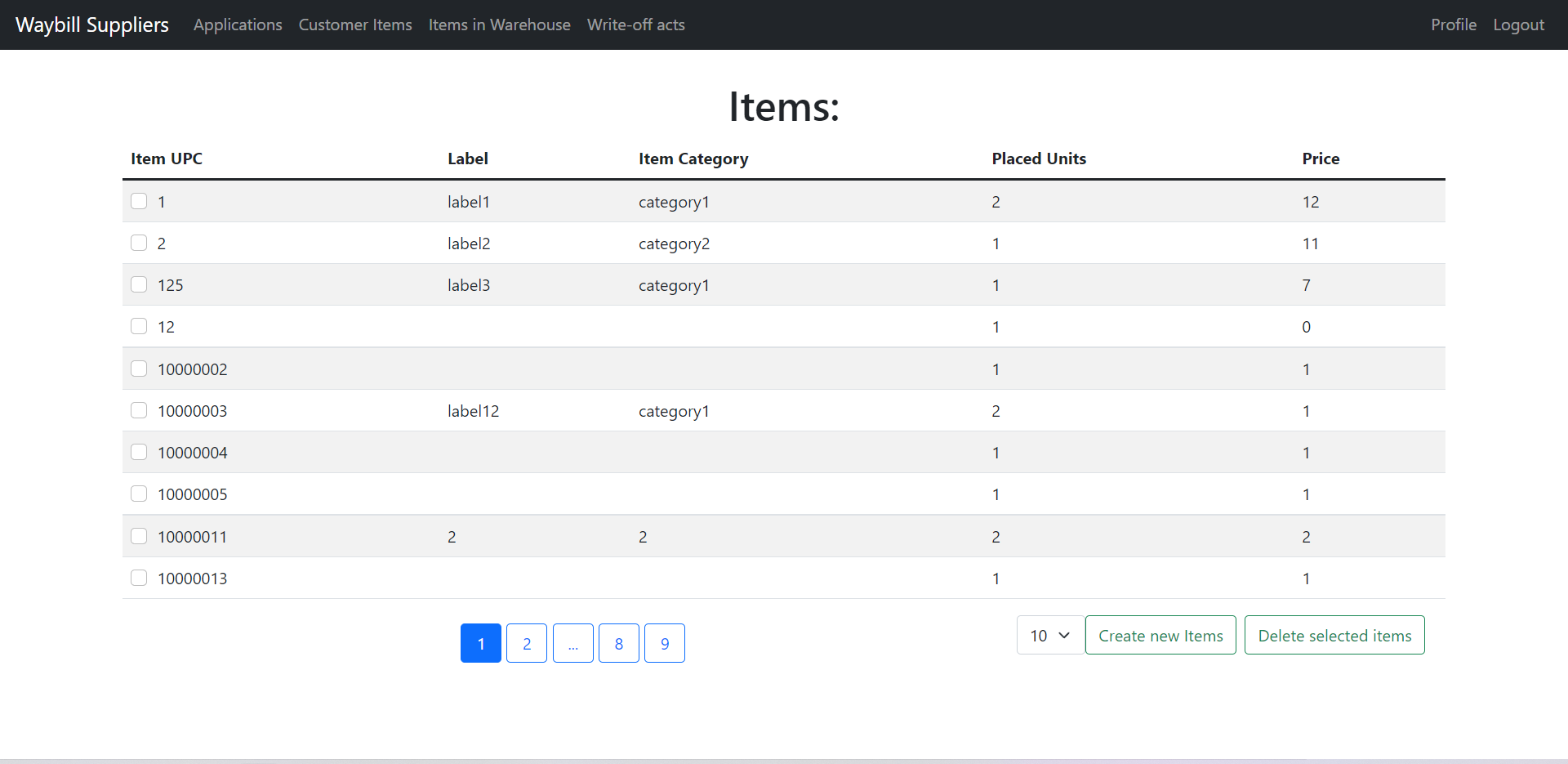


Рисунок 3.17 – Страница с данными о товарах компании

Также при нажатии на ячейку таблицы с названием товара появится модальное окно, на котором находится подробная информация о товаре, которую диспетчер может изменить и нажать на кнопку «*Save*» для сохранения изменений.

Диспетчер может добавлять и новый товар для компании, нажав на соответствующую кнопку, располагающуюся под таблицей. После её нажатия отобразится модальное окно, которое идентично окну редактирования данных о товаре.

Форму для редактирования информации о товаре компании можно увидеть на рисунке 3.18.

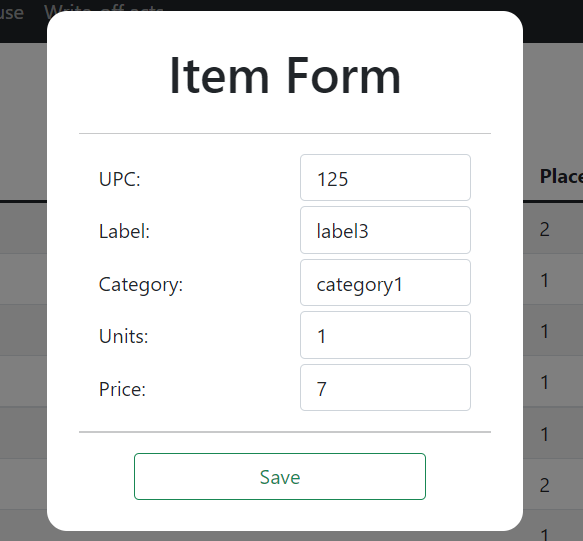


Рисунок 3.18 – Форма

редактирования данных товара

Помимо страницы со списком товаров, которыми владеет компания диспетчер также имеет доступ к товарам, хранящимся или зарегистрированных на складе, к которому он прикреплён. Таблица с данными о товарах на этой странице практически идентична таблице со списком всех товаров компании. Однако в данной таблице добавлены колонку с информацией о текущем количестве товара на складе и специальные маркеры, показывающие доступен ли товар для заказа. Также под таблицей размещена кнопка, при нажатии на которую обновляется информация о доступности товара в соответствии с изменением маркеров.

Кроме того, для данной таблицы, как и для таблицы с данными обо всех товарах компании, существует возможность выбора количества одновременно отображаемых товаров на странице таблицы. А также для удобства пользования под таблицей присутствует выпадающее меню для фильтра доступных, недоступных и всех товаров к показу.

Страницу с информацией о товаре, хранящемся на складе, можно увидеть на рисунке 3.19.

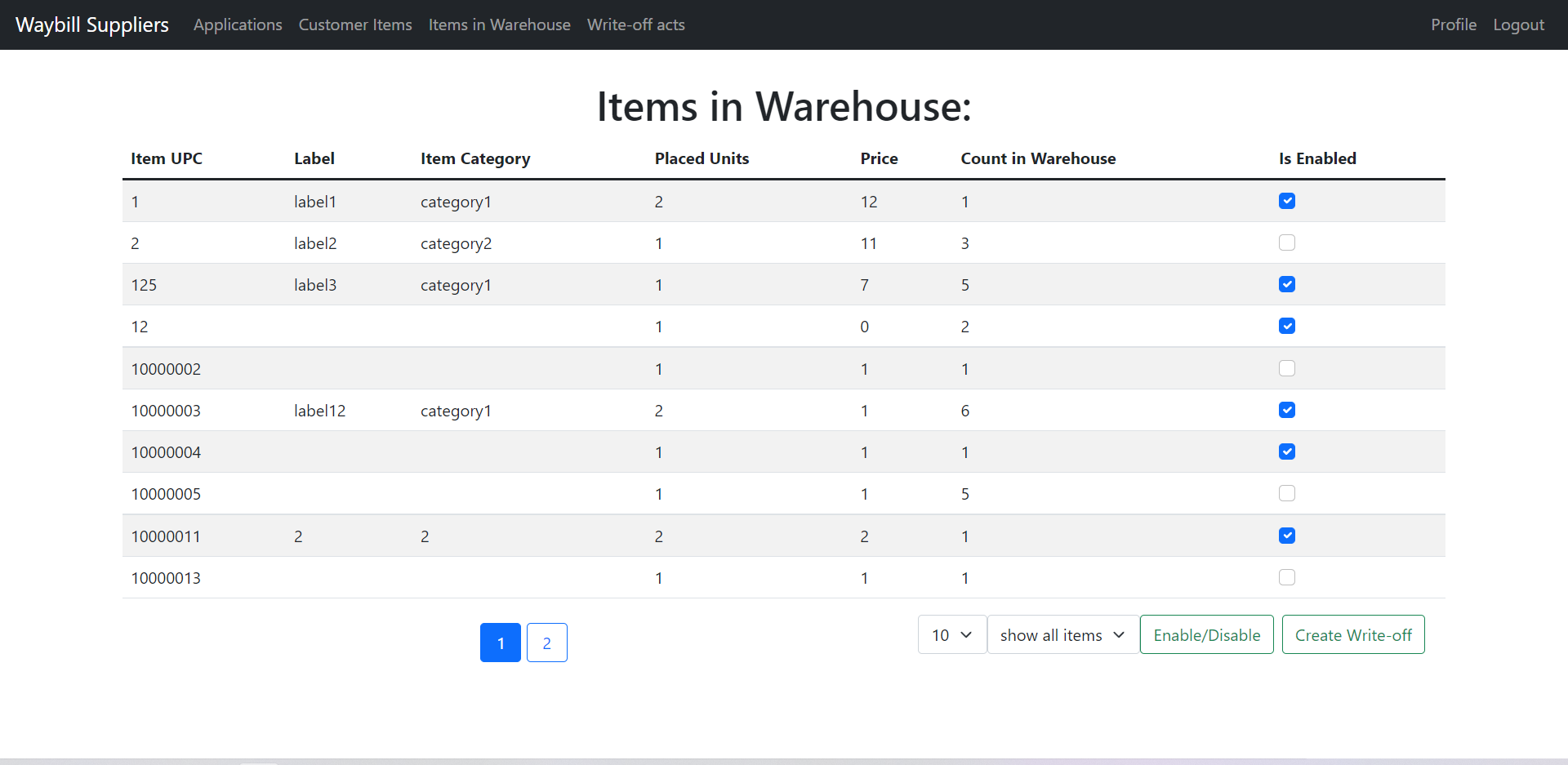


Рисунок 3.19 – Страница с данными о товарах на складе

В случае утери или повреждения товара при доставке или хранении его на кладе, диспетчер может создать акт списания товара, нажав на соответствующую кнопку, также расположенную под таблицей. При нажатии на неё появляется модальное окно с формой для ввода информации о списываемом товаре и нажимает на кнопку «*Create Write-off act*». Добавление товара в список происходит путём ввода уникального номера товара и его количества. После чего выбирается одна из причин списания и товар автоматически добавляется в список при условии, что введённое количество присутствует на складе. Также под списком списываемого товара пересчитывается и отображается итоговое количество единиц товара к списанию.

После отправки формы количество товара на складе уменьшается на количество, указанное диспетчером, а данные об акте списания сохраняются в систему.

Форму для создания актов списания товара можно увидеть на рисунке 3.20.

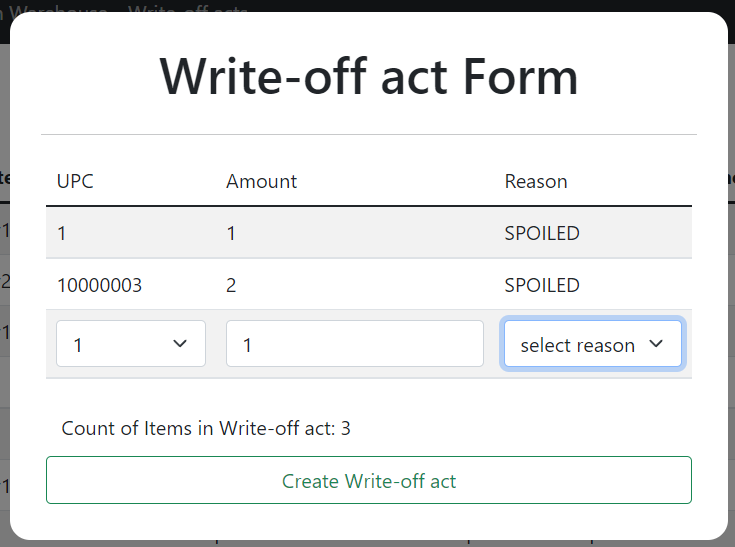


Рисунок 3.20 – Форма создания акта списания

Также диспетчер имеет доступ к просмотру информации об товарах, списанных со склада, к которому он прикреплён. Перейти на страницу с данной информацией он может также с использованием навигационного меню в верхней части приложения.

На странице размещена лишь таблица с данными об актах списания, а также выпадающее меню для контроля количества строк данных, показываемых в таблице на странице.

При нажатии на строку таблицы с актом списания появляется модальное окно с соответствующей информацией о нём. Модальное окно аналогично окну для создания акта списания с отличием в отсутствии возможности редактирования и сохранения информации.

Страницу с данными об актах списания можно увидеть на рисунке 3.21.

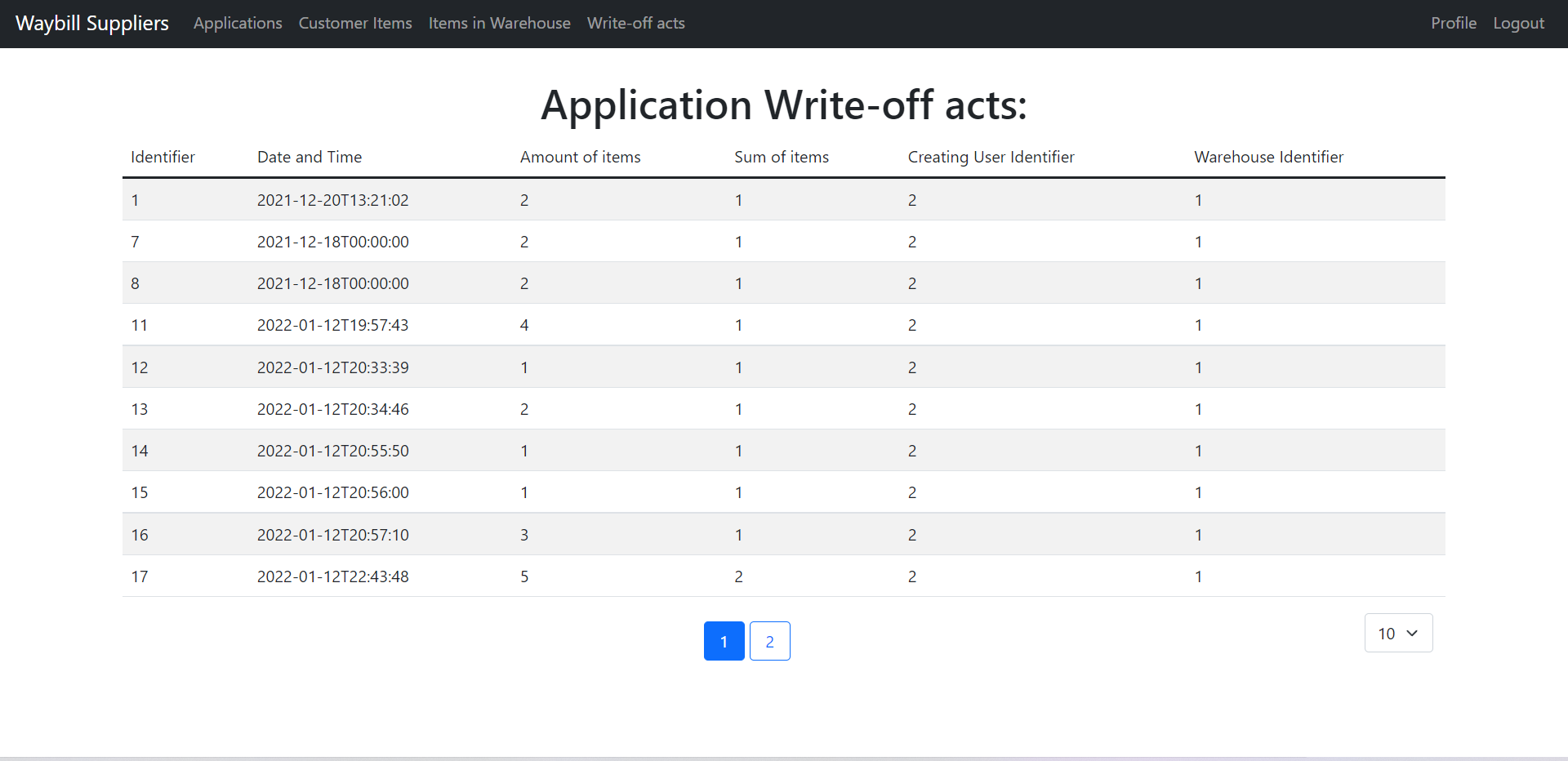


Рисунок 3.21 – Страница с данными о товарах на складе

Все сотрудники компании кроме администраторов около кнопки выхода из аккаунта в панели навигации имеют также кнопку для просмотра своих персональных данных. На ней пользователь может вводить изменения любой информации кроме почтового адреса, логина для входа, а также роли в компании, с которыми он был зарегистрирован её администратором.

Также внизу страницы располагаются кнопки для сохранения в системе, а также сброса введённой в форму информации.

Пример такой страницы можно увидеть на рисунке 3.22.

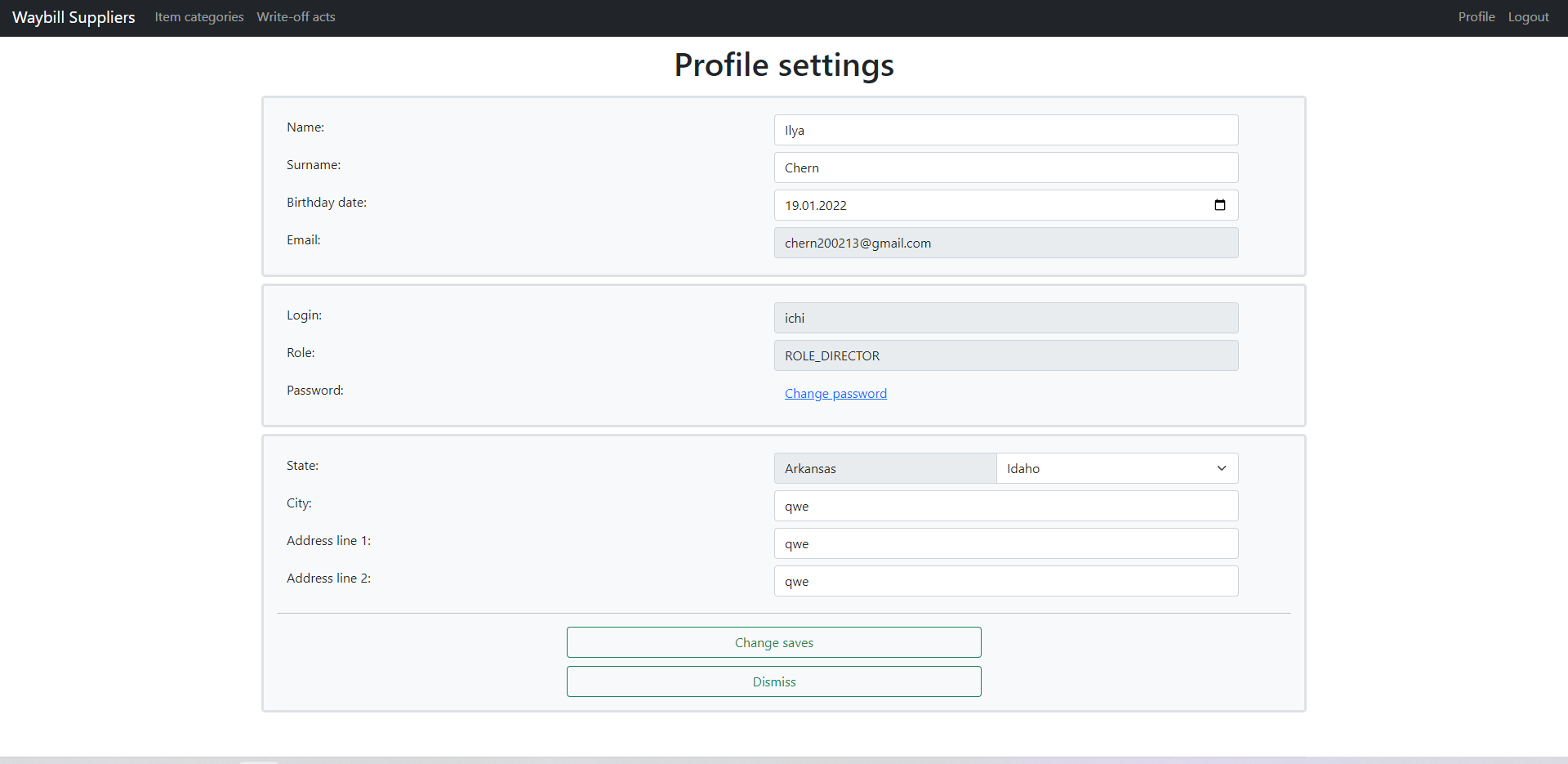


Рисунок 3.22 – Страница с персональными данными сотрудника

При необходимости любой сотрудник может изменить свой пароль для авторизации, нажав на соответствующую ссылку в форме. После нажатия на экране появляется форма, где пользователь должен ввести свой текущий пароль, новый пароль, а также его подтверждение. Повторное введение нового пароля должно совпадать с самим новым паролем. В противном случае пользователь уведомляется соответствующим сообщением. После ввода данных сотрудник нажимает на кнопку «*Change*» и в случае успеха данные обновляются, сессия пользователя аннулируется, а сам он перенаправляется на страницу с авторизацией.

Пример такой формы можно увидеть на рисунке 3.23.

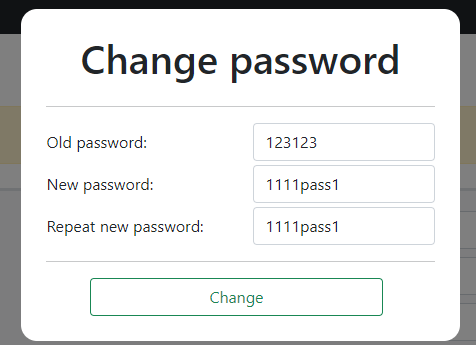


Рисунок 3.23 – Форма смены пароля

При авторизации в систему с ролью логистического специалиста пользователь перенаправляется на страницу с информацией заявок, готовых к сборке в накладную. Данная страница аналогична странице с информацией о заявках для диспетчера, однако логистический специалист не имеет доступа к редактированию информации и создании новых заявок, а также он видит все заявки компании, а не только на конкретный склад, как диспетчер. Также логистическому специалисту предоставляются только заявки на доставку конечному заказчику, а не заявки пополнения товара на складе.

Ознакомившись с данными об одобренных заявках, логист может перейти к странице с информацией о накладных компании с помощью навигационного меню. Данная страница содержит таблицу с информацией о накладных, а также выпадающее меню количество строк данных на странице и кнопки сортировки накладных по их статусу и создателю.

Пример такой страницы продемонстрирован на рисунке 3.24.

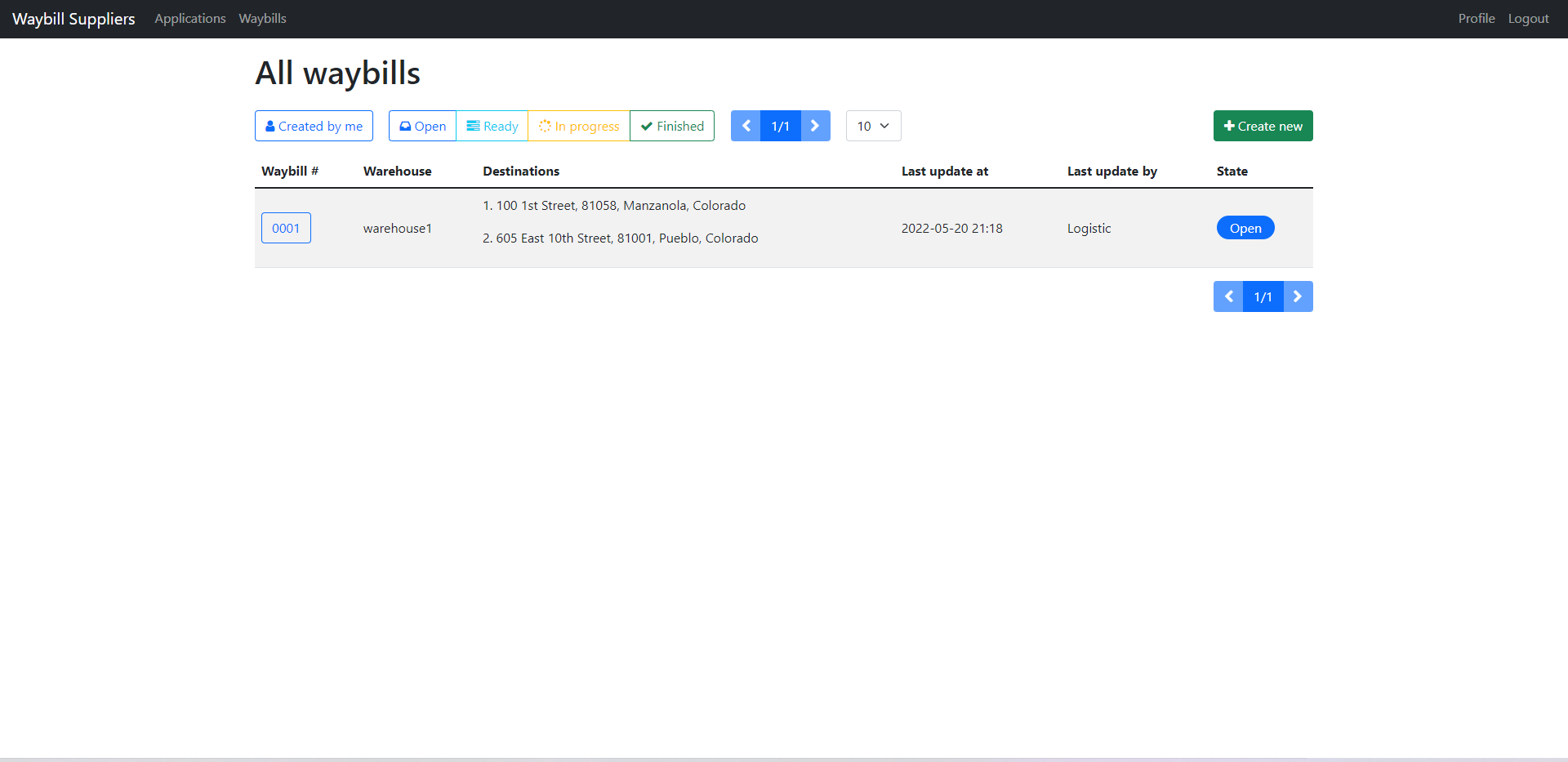


Рисунок 3.24 – Страница с данными накладных компании

Логистический специалист может просмотреть более детальную информацию о накладных либо создать новую нажав на соответствующую кнопку номера накладной или кнопку её создания соответственно. После нажатия на одну из данных кнопок пользователь перенаправляется на страницу редактирования информации накладной.

Страница содержит список заявок, которые включены в накладную. Также присутствует возможность добавления новых заявок в накладную, изменение её номера, склада начала доставки и выбор свободной машины, которая осуществит данную поездку.

Под таблицей имеются специальные кнопки для удаления накладной, сохранения её с статусом готовности к доставке либо для дальнейшего редактирования, а также кнопку для отмены изменений, введённых пользователем.

Пример такой страницы можно увидеть на рисунке 3.25.

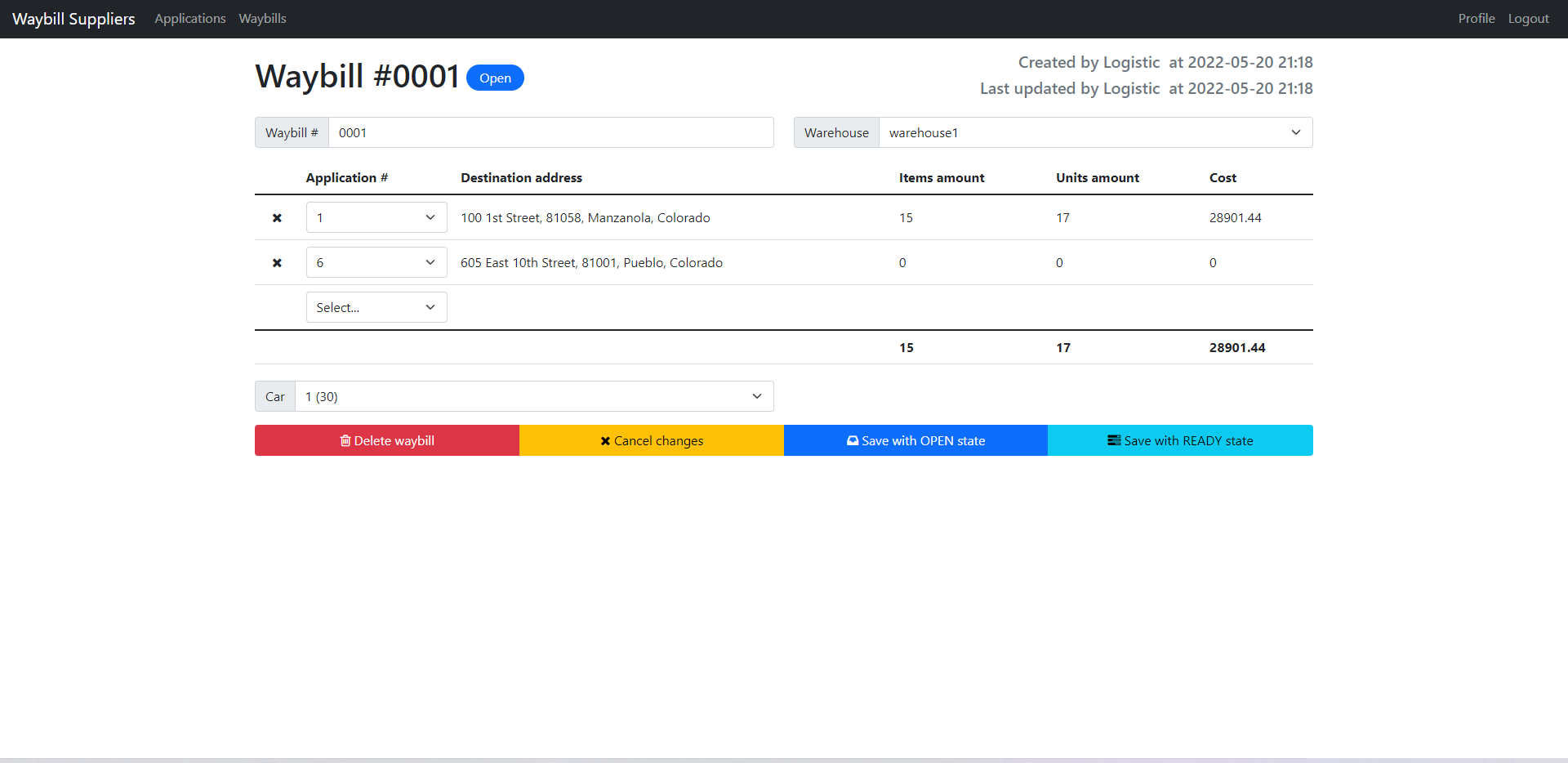


Рисунок 3.25 – Страница редактирования накладной

При авторизации сотрудника с ролью директора компании, он попадает на страницу с информацией о ценах на доставку товаров в зависимости от категории продукции, а также количество товара компании по каждой из категорий.

Пример такой страницы можно увидеть на рисунке 3.26.

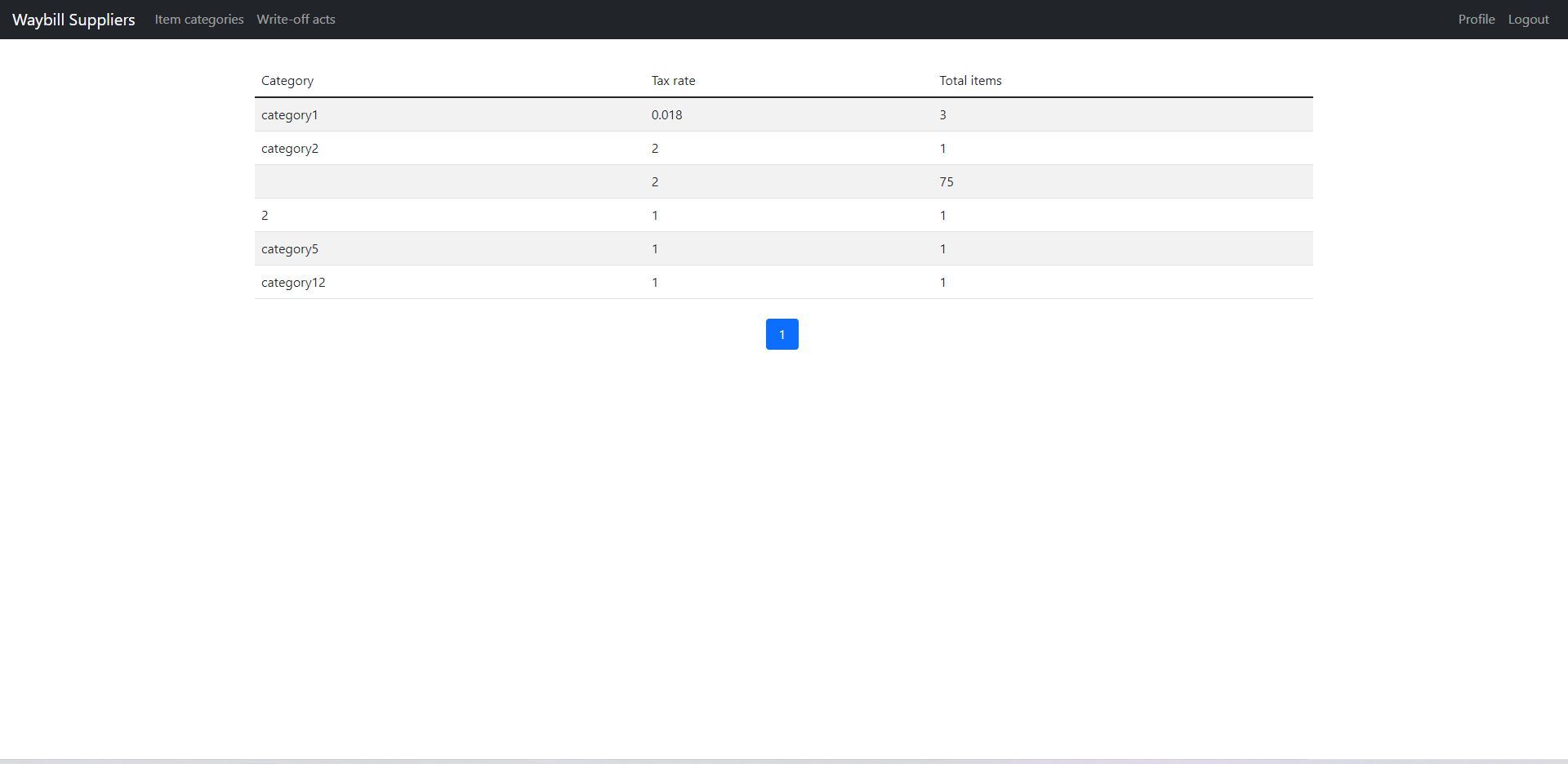


Рисунок 3.26 – Страница с ценами за доставку категорий товаров

В случае, если директор хочет изменить цены за доставку, ему необходимо нажать на строчку информации соответствующей категории в таблице. После чего на страницу открывается модальное окно, в котором он может изменить значение цены и нажать на кнопку «*Create*» для сохранения.

Модальное окно изменения значения цены за доставку категории товара можно увидеть на рисунке 3.27.

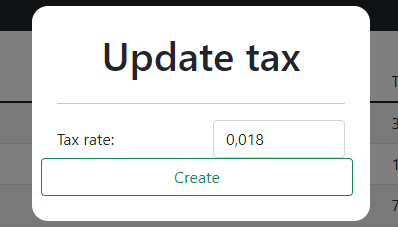


Рисунок 3.27 – Форма изменения

цены за доставку категории товара

Директор компании, так же, как и диспетчер, имеет доступ к странице актов списания, однако он может видеть акты списания товаров со всех складов и транспорта. Он также может просматривать подробную информацию о списанных товарах, нажав на соответствующую строчку таблицы.

При входе в систему с ролью водителя пользователь перенаправляется на страницу с информацией по накладным, где он может ознакомиться с ними, выбрать одну из них и начать доставку. Страница аналогична той, что описана для логистического специалиста.

Также водитель имеет возможность, как и диспетчер, к созданию актов списания товара в случае утери или какого-либо повреждения товара в процессе его доставки, а также просмотреть список актов, созданных водителями. Страницы для взаимодействия с актами списания аналогичны тем, что описаны для диспетчера.

1. **Технико-экономическое обоснование эффективности разработки** 
   1. **Характеристика программного средства**

Целью данного дипломного проекта является разработка веб-приложения для транспортных логистических компаний. Программная часть включает в себя клиентскую часть, которая представляет собой совокупность средств, при помощи которых пользователь взаимодействует с приложением через браузер, и серверную часть. В ходе разработки программной части получен программный продукт, который предоставляет полный функционал для создания накладных из заказов покупателей, предоставляет доступ к управлению имуществом и персоналом компании, а также предоставляет по ним развёрнутую информацию.

Данный продукт подойдет компаниям, которые непосредственно связаны с транспортной деятельностью, например, «*FedEx Corporation*» или «*Wildberries*».

Данный продукт разрабатывается по индивидуальному заказу, что подразумевает, что пользование данным продуктом планируется осуществляться сотрудниками организации-заказчика.

* 1. **Расчет затрат на разработку и производство программного средства**

Для организации-заказчика инвестициями в разработку программного средства является цена программного средства без НДС, которая определяется на основе полных затрат на разработку программного средства организацией-разработчиком и включает в себя следующие затраты: основная заработная плата разработчиков, дополнительная заработная плата разработчиков, отчисления на социальные нужды, прочие расходы, общая сумма затрат на разработку, плановая прибыль, включаемая в цену программного средства, отпускная цена программного средства. Далее рассмотрим каждую инвестицию подробнее.

* + 1. **Основная заработная плата разработчиков**

Расчет осуществляется исходя из состава и численности команды, размера месячной заработной платы каждого участника команды, а также трудоемкости работ, выполняемых при разработке программного средства отдельными исполнителями по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.1) |

|  |  |  |  |
| --- | --- | --- | --- |
| где |  | – | коэффициент премий и иных стимулирующих выплат (составляющий 2); |
|  | *n* | – | категории исполнителей, занятых разработкой программного средства; |
|  |  | – | часовой оклад исполнителя *i*-й категории, р.; |
|  |  | – | трудоемкость работ, выполняемых исполнителем *i*-й категории, ч. |

При определении состава команды были проанализированы требования и анализ предметной области. По итогу принято решение, что на данном проекте должны работать ведущий разработчик, разработчик серверной части продукта, разработчик клиентской части продукта, тестировщик. Команда разработчиков оценила трудоёмкость проекта в три месяца, или 504 рабочих часа (по данным Министерства труда и социальной защиты населения, количество рабочих часов в месяце равно 168 часов). Размеры заработных плат сотрудников указаны по состоянию на 08.05.2022. Ведущий разработчик – 2100 рублей, разработчик серверной части – 1850 рублей, разработчик клиентской части – 1600 рублей. Размер премии равен 100% от размера основной заработной платы. Исходя из вышеперечисленных данных, рассчитаем затраты на основную заработную плату разработчиков. Все данные представлены в таблице 4.1.

Таблица 4.1 – Расчет затрат на основную заработную плату команды

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория исполнителя | Месячный оклад, р | Часовой оклад, р.. | Трудоёмкость работ, ч. | Итого, р. |
| 1 | 2 | 3 | 4 | 5 |
| Разработчик серверной части | 1850 | 11,01 | 504 | 5550 |
| Разработчик клиентской части | 1600 | 9,52 | 504 | 4800 |
| Ведущий разработчик | 2100 | 12,50 | 504 | 6300 |

Продолжение таблицы 4.1

|  |  |
| --- | --- |
| Итого | 16650 |
| Премия и иные стимулирующие выплаты (100%) | 16650 |
| Всего затрат на основную заработную плату разработчиков | 33300 |
|  |  |

* + 1. **Дополнительная заработная плата разработчиков**

Дополнительная заработная плата разработчиков включает выплаты, предусмотренные законодательством о труде, и определяется по нормативу в процентах к основной заработной плате по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.2) |

где – норматив дополнительной заработной платы (составляющий 14%).

После подстановки значений в формулу (4.2) затраты на дополнительную заработную плату команды составят:

* + 1. **Отчисления на социальные нужды**

Расчет размера отчислений в фонд социальной защиты населения и на обязательное страхование определяется в соответствии с действующими законодательными актами Республики Беларусь и вычисляются по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.3) |

где – норматив отчислений на социальные нужды, %.

Согласно законодательству Республики Беларусь, отчисления составляют 34.6%. Сумма отчислений на социальные нужды, согласно формуле (4.3), составит:

* + 1. **Прочие расходы**

Расходы по данной статье включают такие затраты, как приобретение специализированной научно-технической литературы, поддержку со стороны

производителей устройств, определяются по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.4) |

где – норматив прочих затрат в целом по организации.

Приняв значение равным 30% и подставив в формулу (4.4) оставшиеся значения, произведём расчёт прочих затрат:

* + 1. **Общая сумма инвестиций (затрат) на разработку**

Полная сумма затрат на разработку программного обеспечения находится путем суммирования всех рассчитанных статей затрат.

|  |  |  |
| --- | --- | --- |
|  |  | (4.5) |

Учитывая ранее рассчитанные значения затрат, произведём расчёт общей суммы затрат:

* + 1. **Плановая прибыль, включаемая в цену программного средства**

Расчеты по данной статье исходят из нужд на развитие производства и затрат на заработную платы команды.

|  |  |  |
| --- | --- | --- |
|  |  | (4.6) |

где – рентабельность затрат на разработку программного средства.

Приняв значение равным 30% и подставив в формулу (4.6) оставшиеся значения, произведём расчёт плановой прибыли:

* + 1. **Отпускная цена программного средства**

Расчеты по данной статье исходят из затрат на заработную плату и плановой прибыли.

|  |  |  |
| --- | --- | --- |
|  |  | (4.7) |

Расчет итоговых затрат на разработку представлен в таблице 4.2.

Таблица 4.2 – Формирование цены программного средства на основе затрат

|  |  |  |
| --- | --- | --- |
| Наименование  статьи затрат | Расчет по формуле / ссылка на таблицу | Значение, р. |
| 1. Основная заработная плата разработчиков | Табл. 4.1 | 33300,00 |
| 2. Дополнительная заработная плата разработчиков | Формула (4.2) | 4662,00 |
| 3. Отчисления на социальные нужды | Формула (4.3) | 13134,85 |
| 4. Прочие расходы | Формула (4.4) | 9990,00 |
| 5. Общая сумма затрат на разработку | Формула (4.5) | 61086,85 |
| 6. Плановая прибыль, включаемая в цену программного средства | Формула (4.6) | 18326,06 |
| 7. Отпускная цена программного средства | Формула (4.7) | 79412,91 |

* 1. **Расчет результата от разработки и использования программного средства**
     1. **Экономический эффект от разработки программного средства для организации-разработчика**

Экономический эффект для организации-разработчика программного обеспечения заключается в получении прибыли от его реализации, а также его поддержки в будущем, по согласованию с заказчиком. Разрабатываемое программное обеспечение продаётся организации-заказчику в одном экземпляре и планируется поддерживать его в будущем по мере необходимости.

В результате расчет инвестиций в разработку программного средства было выявлено, что стоимость данного продукта составляет 79412,91 рублей.

Прибыль от разработки программного средства можно определить по следующей формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.8) |

где ‒ прибыль, включаемая в цену программного средства, р.

Согласно законодательству Республики Беларусь, резиденты Парка высоких технологий освобождены от налога на добавленную стоимость, а так как компания ООО «Техартгруп» является резидентом Парка высоких технологий, налог на добавленную стоимость берём равным нулю, следовательно, прибыль от продажи программного продукта составляет рублей.

* + 1. **Экономический эффект от разработки программного средства для организации-заказчика**

Для организации-заказчика экономическим эффектом от использования программного средства, разработанного по индивидуальному заказу сторонней организацией, является прирост чистой прибыли, полученный за счёт экономии затрат на заработную плату с начислениями на заработную плату сотрудников за счёт снижения трудоёмкости работ. Указанный вид экономии рассчитывается по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.9) |

|  |  |  |  |
| --- | --- | --- | --- |
| где |  | – | коэффициент премий (принят равным значению 2); |
|  |  | – | трудоемкость выполнения работ сотрудниками до и после внедрения программного средства (принята равной 84 и 28 соответственно), ч; |
|  |  | – | часовой оклад сотрудника, использующего программное средство (принят равным 9,52), р.; |
|  |  | – | плановый объем работ, выполняемых сотрудником (принят равным 96); |
|  |  | – | норматив дополнительной заработной платы (принят равным 20%); |
|  |  | – | ставка отчислений от заработной платы, включаемых в себестоимость, (составляющая 34,6%). |

Подставляя в формулу (4.9) указанные значения получим размер экономии:

Экономическим эффектом при использовании программного средства является прирост чистой прибыли, полученной за счет экономии на текущих затратах предприятия, который рассчитывается по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.10) |

|  |  |  |  |
| --- | --- | --- | --- |
| где |  | – | экономия на текущих затратах при использовании программного средства, р.; |
|  |  | – | прирост текущих затрат (Интернет-трафик и хостинг), связанных с использованием программного средства (принят равным 1620), р.; |
|  |  | – | ставка налога на прибыль согласно действующему законодательству (составляющая 18%). |

Подставив под условные обозначения соответствующие им принятые значения, получим

* 1. **Расчёт показателей экономической эффективности разработки и использования программного средства**
     1. Для организации-разработчика программного средства оценка экономической эффективности разработки осуществляется с помощью расчёта рентабельности инвестиций на разработку программного средства по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.11) |

После подстановки значений и в формулу (4.11) получим значение рентабельности инвестиций:

* + 1. Поскольку сумма инвестиций меньше суммы годового экономического эффекта, то есть инвестиции окупятся менее чем за год, для организации-заказчика оценка экономической эффективности инвестиций в разработку программного средства осуществляется с помощью расчёта рентабельности инвестиций по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.12) |

где ‒ ставка налога на добавленную стоимость в соответствии с законодательством, %.

Так как компания ООО «Техартгруп» является резидентом Парка высоких технологий, налог на добавленную стоимость берём равным нулю.

В соответствии с формулой (4.12), рентабельности инвестиций равна:

**7.5 Выводы по технико-экономическому обоснованию**

В ходе технико-экономического обоснования разрабатываемого программного продукта произведён расчёт затрат на разработку программного обеспечения, выполнена оценка экономического эффекта от его продажи, произведён расчёт эффективности инвестиций в разработку, а также показана его экономическая целесообразность. Чистая прибыль от реализации программного продукта составит 18326,06 рублей. Оценки экономической эффективности для организации-разработчика и для организации-заказчика составили 30% и 69,04% соответственно. Полученные результаты считаются хорошими показателями. Исходя из произведённых расчётов и выполненных оценок, можно сделать вывод, что данный проект является экономически выгодным.

**Заключение**

В ходе выполнения дипломного проекта разработана автоматизированная система, предназначенная для работы с заказами, поступающими в транспортные компании, которое позволит оптимизировать рабочий процесс. Также изучена теория и предметная область, изучен процесс разработки веб-приложения, его компонентов и подходов к созданию.

Функциональность сервера дипломного проекта реализована на языке *Java*, в качестве основной библиотеки использовался *Spring Framework*. Функциональность клиента дипломного проекта реализована на языке *JavaScript*, с использованием библиотеки *React*.

Разработанная система обладает такими преимуществами, как:

− простота в использовании и освоении;

− невысокие системные требования;

− низкая стоимость приобретения;

− простой и минималистичный интерфейс.

После проведения расчёта экономических показателей эффективности, исходя из анализа эффективности можно сделать вывод о том, что разработка и реализация данного программного продукта, являются целесообразными.

Таким образом, разработка данной системы является успешной и экономически выгодной.

Проект разработан в полном объеме в соответствии с поставленной целью. Одним из дальнейших направлений по продолжению работы с данной системой является добавления нового функционала, а также улучшение интерфейса для работы с проектом с помощью мобильного телефона. Кроме того, присутствует возможность реализации взаимодействия с сервисами поставки новых товаров в компании.

**Список использованных источников**

1. *Aurama* – cервис автоматизации курьерских служб [Электронный ресурс]. – Режим доступа : *https://www.aurama.ru*.
2. Программа для маршрутизации – *RELOG* [Электронный ресурс]. – Режим доступа : *https://getrelog.com*.
3. **Обзор системы *bpm’online*** [Электронный ресурс]. – Режим доступа : *https://habr.com/ru/company/trinion/blog/*281192.
4. *MDN Web Docs JavaScript*. Материалы для обучения языку программирования [Электронный ресурс]. – Электронные данные. – Режим доступа : *https://developer.mozilla.org/ru/docs/Web/JavaScript*.
5. Официальный сайт *React JS* [Электронный ресурс]. – Электронные данные. – Режим доступа : *https://ru.reactjs.org*.
6. *React*. Введение в хуки [Электронный ресурс]. – Электронные данные. – Режим доступа : *https://ru.reactjs.org/docs/hooks-intro.html*.
7. *Habr JavaScript*. Прототипно-ориентированный язык программирования [Электронный ресурс]. – Электронные данные. – Режим доступа : *https://habr.com/ru/hub/javascript*.
8. *Proselyte*. Руководство по *SQL*. Введение. [Электронный ресурс]. – Электронные данные. – Режим доступа : *https://spring.io/projects/spring-framework*.
9. *MSDN*. Полное руководство по работе с базой данных *MsSql* [Электронный ресурс]. – Электронные данные. – Режим доступа : *https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/ms-sql*.
10. *React*. Справочник *API* хуков [Электронный ресурс]. – Электронные данные. – Режим доступа : *https://ru.reactjs.org/docs/hooks-reference.html*.
11. Доманов, А.Т. СТП 01-2017. Стандарт предприятия. Дипломные проекты(работы). Общие требования. / А. Т. Доманов, Н. И. Сорока – Минск : БГУИР, 2017. – 170 с.

# Приложение А (обязательное) Программный код клиента приложения

# Приложение Б (обязательное) Программный код сервера приложения