

ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГА
КОМИТЕТ ПО НАУКЕ И ВЫСШЕЙ ШКОЛЕ

Санкт-Петербургское государственное бюджетное
профессиональное образовательное учреждение
«ПЕТРОВСКИЙ КОЛЛЕДЖ»
(СПб ГБПОУ «Петровский колледж»)

Отделение информационных технологий

ДИПЛОМНЫЙ ПРОЕКТ

РАЗРАБОТКА ПЛАТФОРМЫ УЧЁТА ОПТОВОЙ ТОРГОВЛИ

Выполнил:

Коршунов Артём Леонидович

обучающегося 3802 группы

Специальность 09.02.07 Информационные системы и
программирование

Руководитель:

А.В.Бережков

(Личная подпись)

Нормоконтролер:

А.С.Вагин

(Личная подпись)

Санкт-Петербург, 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Аналитическая часть.....	5
1.1 Описание предметной области	5
1.2 Анализ существующих аналогов.....	6
1.3 Требования к разрабатываемой информационной системе.....	8
1.4 Обоснование выбора технологий и инструментальных средств проектирования и разработки	10
2 Проектная часть.....	13
2.1 Разработка модели базы данных.....	13
2.2 Проектирование информационной системы	14
2.3 Проектирование интерфейсов.....	16
3 Реализация и тестирование программного обеспечения	20
3.1 Реализация основных функций программного обеспечения.....	20
3.2 Тестирование программного обеспечения	39
ЗАКЛЮЧЕНИЕ	45
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	46
ПРИЛОЖЕНИЕ А	48
ПРИЛОЖЕНИЕ Б.....	52
ПРИЛОЖЕНИЕ В	53

ВВЕДЕНИЕ

Оптовая торговля – вид бизнеса, при котором организация закупает товары у поставщиков, а затем, за установленный процент, продаёт товары клиентам. Организации заключают договоры с поставщиками и клиентами, чтобы на основе составленных договоров составлять заявки на поставки товаров.

Оптовая торговля является связующим звеном между поставщиками и клиентами [7]. Поставщики занимаются производством товаров и, в связи с большой конкуренцией, у них слабо развита цепочка клиентов. Продажа продукции организациям оптовой торговли позволяет обойти это препятствие, и клиенты, работающие с организацией, смогут увидеть и приобрести предоставленные поставщиком товары.

Работа организаций оптовой торговли невозможна без использования информационных систем, так как растёт количество заказов от клиентов, расширяется база поставщиков и ассортимент товаров. Информационные системы минимизируют «человеческий фактор» в работе предприятия, позволяют манипулировать хранимой в базе данных информацией, а также дают защититься от сильных конкурентов [13].

Целью работы является разработка платформы для учёта оптовой торговли предприятия.

Платформа должна решать следующие задачи:

- регистрация администраторов, агентов, поставщиков и клиентов;
- заключение договоров с поставщиками и клиентами (с возможностью печати составленного договора);
- работа с заявками на поставки клиентов и поставщиков;

Работу в платформе может осуществлять четыре различных вида пользователей: администратор, агент, клиент, поставщик. Деятельность каждого пользователя на платформе отличается.

Для реализации платформы необходимо разработать приложения:

- для предприятия (администраторы и агенты);

- для пользователей (поставщики и клиенты).

Для достижения поставленной цели необходимо решить следующие задачи:

- описать предметную область;
- провести обзор аналогов;
- выделить основные требования к системе;
- разработать модель базы данных;
- спроектировать информационную систему;
- реализовать основные функции программного обеспечения;
- протестировать программное обеспечение.

1 Аналитическая часть

1.1 Описание предметной области

Оптовая фирма занимается закупкой большого количества товара у поставщиков и продаёт купленный товар клиентам, получая процент суммы сделки. Каждому поставщику и клиенту, перед сотрудничеством с оптовой фирмой, необходимо оформить договор, в котором указывается: данные клиента/поставщика; товары, которые хотелось бы приобретать/продавать и время действия договора (количество лет). Поставщик или клиент составляют договор, а затем отправляют его на рассмотрение агенту предприятия.

Агенты оптовой фирмы занимаются рассмотрением договоров поставщиков и клиентов. Каждый договор имеет статус, который агент может изменять в зависимости от ситуации. Статусы договоров могут быть следующие:

- в обработке (договор создан и не рассмотрен агентом);
- одобрен (договор рассмотрен и одобрен агентом);
- отклонён (договор рассмотрен и отклонён агентом);
- приостановлен (договор приостановлен клиентом или поставщиком);
- истёк (время действия договора истекло);

При каждом изменении статуса договора, об этом узнает поставщик или клиент.

Помимо рассмотрения договоров, агенты занимаются рассмотрением заявок клиентов и оформлением заявок поставщикам. В заявке указывается: данные о поставщике/клиенте, перечень товаров на заказ с указанием количества и цены. Заявка оформляется на основе составленного договора, при этом, в заявке можно выбирать не все товары, перечисленные в договоре. Каждая заявка содержит статус. Статусы заявок могут быть следующие:

- в обработке (заявка создана и не рассмотрена);
- одобрена (заявка одобрена);
- отклонена (заявка отклонена);

- в пути (товары, указанные в заявке, отправлены);
- доставлено (товары, указанные в заявке, доставлены в пункт назначения).

Заявки создаются клиентами и агентами. Заявки клиентов рассматривают агенты, а поставщики рассматривают заявки клиентов.

В фирме присутствует администратор, который имеет возможность управлять пользователями платформы оптовой торговли.

1.2 Анализ существующих аналогов

Существует множество аналогов информационных систем для оптовой торговли, некоторые из которых будут рассмотрены в этом пункте.

1.2.1 Программа 1С:Предприятие

1С:предприятие является одной из самых известных программ для оптовой торговли в России. Она состоит из прикладных решений («конфигураций»), благодаря которым существует возможность расширения системы. Интерфейс программы представлен на рисунке 1.

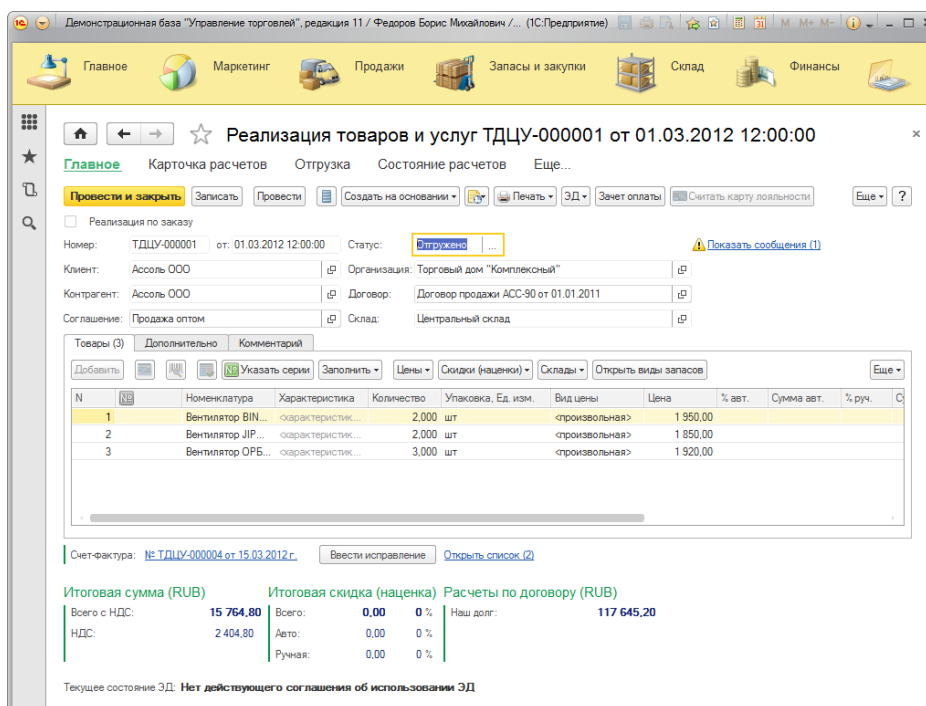


Рисунок 1 – Интерфейс программы «1С:Предприятие»

Преимущества данного программного продукта являются:

- расширение системы за счёт добавления новых «конфигураций»;
- просмотр товаров на предприятии с указанием их количества и цены;
- формирование статистики;
- возможность печати документов;

Помимо плюсов, существуют и минусы программного продукта, а именно:

- непредсказуемость системы при обновлениях;
- нагромождённый интерфейс;
- требование профессиональных навыков для работы в системе;
- цена программного продукта.

1.2.2 Программа Битрикс24

Битрикс24 позволяет автоматизировать работу внутри компании. Система отлично подходит для оптовой торговли, так как позволяет отслеживать товары на складах предприятия, а также работать напрямую с контрагентами через встроенный мессенджер и видеозвонки. Интерфейс программы представлен на рисунке 2.

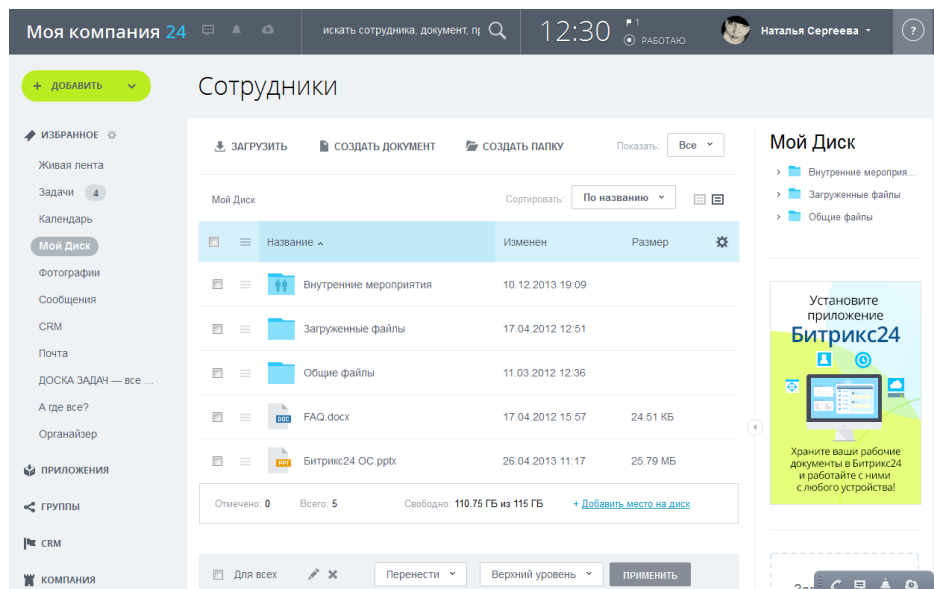


Рисунок 2 – Интерфейс программы «Битрикс24»

Преимущества программы «Битрикс24» являются:

- удобный интерфейс;
- отслеживание товаров на предприятии;

- работа с клиентами и поставщиками через чаты;
- возможность создания и загрузки документов.

Недостатками программы «Битрикс24» являются:

- ограниченные возможности бесплатного тарифа;
- частое обновление интерфейса, из-за которого сложно ориентироваться в системе;
- требование профессиональных навыков для работы в системе.

1.3 Требования к разрабатываемой информационной системе

1.3.1 Функциональные требования

Разрабатываемая платформа оптовой торговли состоит из пяти различных модулей. Каждый модуль включает в себя свои функции.

1.3.1.1 Требования к модулю «Администратор»

Модуль «Администратор» должен иметь следующие возможности:

- создание, редактирование, удаление работников предприятия (администраторов, кладовщиков, агентов);
- создание, редактирование поставщиков;
- создание, редактирование клиентов;
- получение отчётности деятельности организации оптовой торговли (заработок за определённый период, список самых популярных товаров, список самых ценных поставщиков и клиентов и т.д.) с возможностью печати;
- изменение своих данных (логин, пароль и т.д.).

Администратор не может удалить самого себя, а также не может удалять поставщиков и клиентов, в связи с хранением информации о договорах, заявках и товарах.

1.3.1.2 Требования к модулю «Агент»

Модуль «Агент» должен иметь следующие возможности:

- получение заявок на заключения договоров, изменение статуса на принято/отклонено;

- просмотр содержимого договора (данные о поставщике/клиенте и товарах) с возможностью печати;
- создание заявок поставщикам на поставки товаров на предприятие оптовой торговли;
- получение заявок на поставки товаров клиентов, изменение статуса на принято/отклонено;
- просмотр содержимого заявки клиента (данные о клиенте, товары, общая сумма);
- изменение своих данных (логин, пароль и т.д.).

1.3.1.3 Требования к модулю «Поставщик»

Модуль «Поставщик» должен иметь следующие возможности:

- создание договоров на поставку продукции (на предприятие оптовой торговли);
- просмотр содержимого своих договоров;
- получение заявок от предприятия оптовой торговли на поставки своей продукции, изменение статуса на принято/отклонено;
- регистрация;
- изменение своих данных (логин, пароль и т.д.).

1.3.1.4 Требования к модулю «Клиент»

Модуль «Клиент» должен иметь следующие возможности:

- создание договоров на поставку продукции (себе);
- просмотр содержимого своих договоров;
- создание заявок на поставку продукции (себе);
- регистрация;
- изменение своих данных (логин, пароль и т.д.).

1.3.2 Нефункциональные требования

Проанализировав преимущества и недостатки ведущих аналогов для учёта оптовой торговли, были выделены несколько требований, которые должны быть учтены при разработке:

- удобный и понятный интерфейс;
- печать документов;
- отслеживание товаров на предприятии;
- работа с поставщиками и клиентами.

1.4 Обоснование выбора технологий и инструментальных средств проектирования и разработки

1.4.1 Выбор технологий и инструментальных средств проектирования

1.4.1.1 Инструмент StarUML

StarUML – программный инструмент моделирования, который поддерживает UML (унифицированный язык моделирования) [10]. Программа поддерживает 11 различных типов диаграмм.

Выбор данного инструмента связан со следующими плюсами:

- широкий выбор диаграмм;
- удобство использования;
- бесплатный программный продукт.

1.4.2 Выбор технологий и инструментальных средств разработки

1.4.2.1 Язык программирования C#

Язык программирования C# был разработан в 1998-2001 годах. На данный момент, является одним из самых популярных языков программирования современности [4].

Язык программирования C# является объектно-ориентированным языком программирования (далее – ООП), имеет схожий синтаксис со своими «старшими братьями» - языками C и C++, перенимая их достижения и исправляя недостатки, удобный в использовании (приложения разрабатываются быстрее и

легче), имеет встроенный «сборщик мусора», благодаря которому, система сама понимает, когда нужно очистить память от лишнего мусора [14].

Исходя из перечисленных преимуществ и полученных знаний по ходу обучения в колледже, мной был выбран этот язык программирования.

1.4.2.2 Среда разработки Visual Studio 2022

Visual Studio является интегрированной средой разработки (далее – IDE), которая выпускается под эгидой Microsoft с 1997 года. Она имеет встроенный компилятор, позволяет отлаживать написанные программы, а также позволяет публиковать приложения. На данный момент, выпущена последняя версия – Visual Studio 2022, которая и будет использоваться в разработке платформы учёта оптовой торговли.

1.4.2.3 Реляционная база данных

Реляционные базы данных (далее – РБД) подразумевают хранение большого количества таблиц с различными типами связей между ними. Они работают с небольшими по размерам транзакциями, но идущими большим потоком [12].

Для разработки платформы оптовой торговли была выбрана РБД, так как в базе данных необходимо хранить множество информации о товарах, поставщиках и клиентах, их договорах и заявках на поставку продукции. Такие данные связываются между собой разными видами ключей (один-к-одному, один-ко-многим, многие-ко-многим), которые позволяют получать информацию из разных таблиц в одну общую.

1.4.2.4 Платформа .NET Entity Framework Core

Для взаимодействия приложений и базы данных используются технологии доступа к данным. Одна из таких технологий – .NET Entity Framework Core.

.NET Entity Framework Core является более свежей и поддерживаемой разработчиками, кроссплатформенной (может запускаться на разных

операционных системах), имеет открытый исходный код и является более производительной, чем .NET Framework [15].

Для разработки платформы учёта оптовой торговли был выбран .NET Core 6.

1.4.2.5 Технология Windows Presentation Foundation (WPF)

Windows Presentation Foundation (WPF) – технология, созданная Microsoft, как замена уже устаревшей Windows Forms технологии. Включает в себя язык разметки XAML, благодаря которому можно разделить интерфейсы приложений от реализаций их функций [5]. Пример окна приложения, написанного на технологии WPF, представлен на рисунке 3.

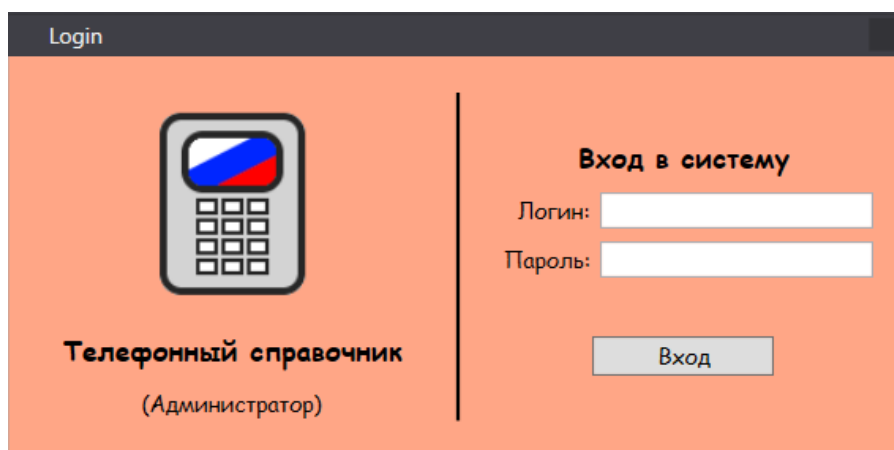


Рисунок 3 – Окно авторизации на технологии WPF

Для работы с WPF, требуется любой .NET язык программирования. Как было описано в п. 1.4.2.1, С# является частью .NET, поэтому технология WPF будет использована для разработки платформы оптовой торговли.

2 Проектная часть

2.1 Разработка модели базы данных

Модель разработанной базы данных представлена на рисунке 4.

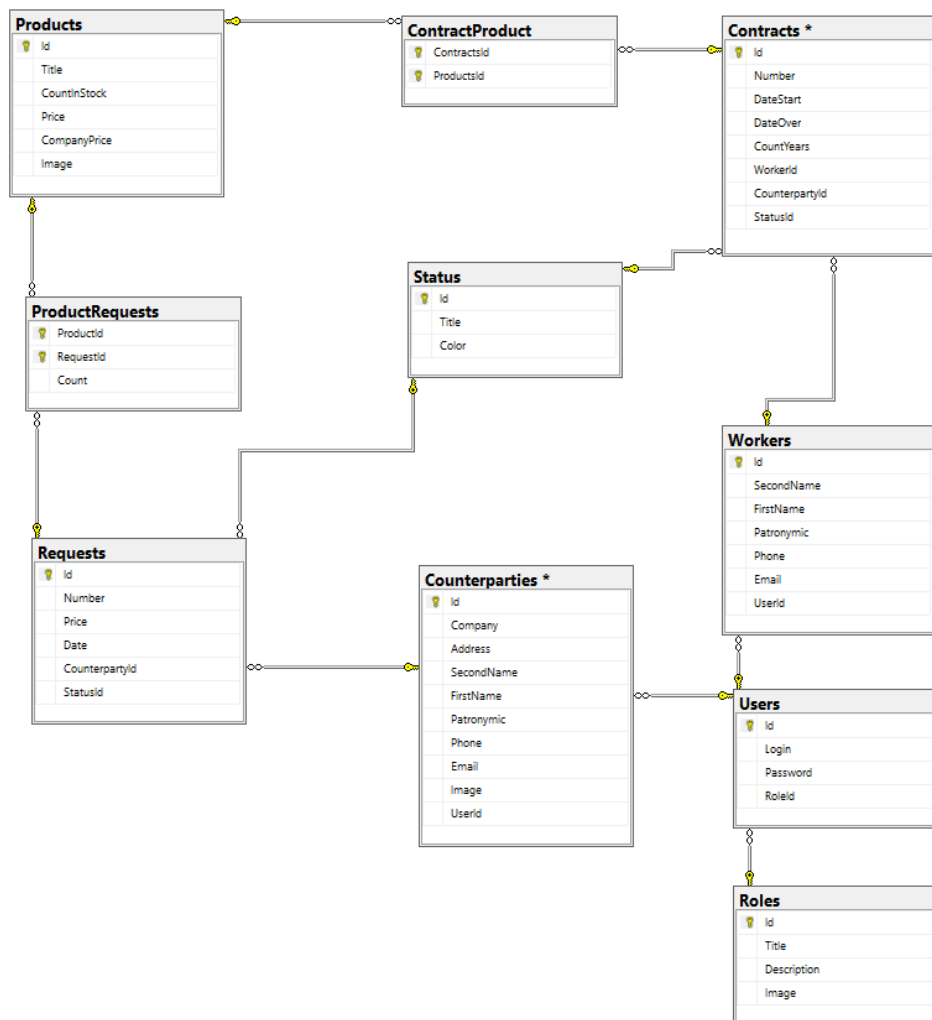


Рисунок 4 – Модель базы данных платформы учёта оптовой торговли

Исходя из построенной модели, можно выделить таблицы:

- Roles (роли пользователей);
- Users (пользователи системы);
- Workers (работники предприятия);
- Counterparties (поставщики и клиенты предприятия);
- Status (статусы договоров и заявок);
- Products (товары предприятия);
- Contracts (договоры клиентов и поставщиков);

- ContractProduct (товары в договорах);
- Requests (запросы поставок поставщиков и клиентов);
- ProductRequest (товары в запросах поставок).

2.2 Проектирование информационной системы

2.2.1 Use-case диаграмма

Use-case диаграмма позволяет показать, чем занимается каждая роль в системе, какие прецеденты используются. Use-case диаграмма платформы учёта оптовой торговли «Покупочка» изображена на рисунке 5.

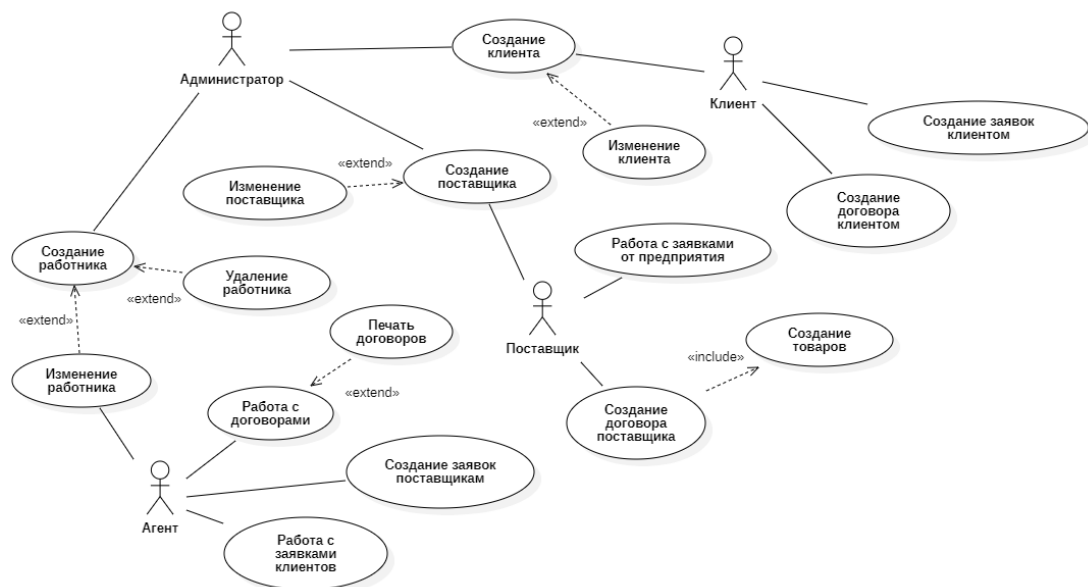


Рисунок 5 – Use-case диаграмма платформы оптовой торговли «Покупочка»

Подробное описание каждого прецедента диаграммы находится в таблице

1.

Таблица 1 – Подробное описание каждого прецедента

Прецедент	Краткое описание
Администратор	
Создание работника	Позволяет создавать работника предприятия с возможностью изменения и удаления
Создание поставщика	Позволяет создавать поставщика с возможностью изменения
Создание клиента	Позволяет создавать клиента с возможностью изменения
Агент	
Работа с договорами	Позволяет работать с договорами поставщиков и клиентов, меняя их статус. Существует возможность печати договора
Создание заявок поставщикам	Позволяет создавать заявки поставщикам на основе ранее составленных договоров
Работа с заявками клиентов	Позволяет работать с заявками клиентов, меняя их статус
Поставщик	
Создание поставщика	Позволяет создавать поставщика с возможностью изменения
Создания договора поставщика	Позволяет создавать договор с возможностью создания новых товаров
Работа с заявками от предприятия	Позволяет работать с заявками предприятия, меняя их статус
Клиент	
Создание клиента	Позволяет создавать клиента с возможностью изменения
Создания договора клиента	Позволяет создавать договор с указанием доступных товаров
Создание заявок клиентом	Позволяет создавать заявки клиентом на основе ранее составленных договоров с предприятием

2.3 Проектирование интерфейсов

2.3.1 Окно авторизации приложения «PokupochkaCompany»

Окно авторизации в приложении «PokupochkaCompany» должно содержать:

- логотип приложения;
- название приложения;
- поле ввода логина и пароля;
- кнопку «запомнить меня» (для запоминания пользователя и автоматического ввода данных при следующих авторизациях);
- кнопку входа.

Примерный интерфейс окна представлен на рисунке 6.

The diagram illustrates the layout of the authorization window for the application «PokupochkaCompany». It consists of several components arranged in a structured manner:

- Logo:** A large rectangular box on the left side containing the text "[ЛОГОТИП]".
- Application Name:** A rectangular box below the logo containing the text "[Название приложения]".
- Authorization Form:** A group of elements on the right side:
 - Title:** The word "Авторизация" (Authorization) is centered at the top of the form area.
 - Login Field:** Labeled "Логин:" followed by a rectangular input box containing "[Поле ввода]".
 - Password Field:** Labeled "Пароль:" followed by a rectangular input box containing "[Поле ввода]".
 - Remember Me:** A checkbox followed by the text "["Запомнить меня"]".
 - Login Button:** A rectangular button at the bottom containing the text "[Кнопка входа]".

Рисунок 6 – Интерфейс окна авторизации «PokupochkaCompany»

2.3.2 Окно авторизации приложения «PokupochkaCounterparty»

Окно авторизации в приложении «PokupochkaCounterparty» должно содержать:

- логотип приложения;

- название приложения;
- поле ввода логина и пароля;
- кнопку «запомнить меня» (для запоминания пользователя и автоматического ввода данных при следующих авторизациях);
- кнопку входа;
- кнопку регистрации.

Примерный интерфейс окна представлен на рисунке 7.

The diagram illustrates the layout of the authorization window. On the left, there is a large rectangular area containing a placeholder for a logo, labeled "[ЛОГОТИП]", and below it, the application name placeholder, labeled "[Название приложения]". To the right of this area, the title "Авторизация" is centered at the top. Below the title, there are two input fields: the first is labeled "Логин:" and the second "Пароль:", both with a placeholder "[Поле ввода]". Under the password field is a checkbox followed by the text "["Запомнить меня"]". At the bottom of the right section, there are two buttons: "[Кнопка входа]" and "[Кнопка регистрации]".

Рисунок 7 – Интерфейс окна авторизации «PokupochkaCounterparty»

2.3.3 Окно регистрации приложения «PokupochkaCounterparty»

Окно регистрации в приложении «PokupochkaCounterparty» должно содержать:

- изображение (с возможностью изменения);
- поля для ввода;
- кнопку отмены;
- кнопку регистрации.

Примерный интерфейс окна представлен на рисунке 8.

Рисунок 8 – Интерфейс окна регистрации «РokupochkaCounterparty»

2.3.4 Основное окно приложений

Основное окно приложений должно содержать:

- главные кнопки, которые будут открывать вторичные кнопки и содержимое;
- вторичные кнопки, которые будут менять область содержимого;
- область содержимого, в котором будет отображена информация;
- данные о пользователе (ФИО, роль в системе);
- кнопку изменения данных пользователя;
- кнопку выхода.

Примерный интерфейс основного окна приложений изображён на рисунке

9.

[Главная кнопка 1]		[Главная кнопка 2]	[Главная кнопка N]
[Вторичная кнопка 1]	[СОДЕРЖИМОЕ]		
[Вторичная кнопка 2]			
[Вторичная кнопка N]			
[Данные о пользователе]		[Кнопка "Изменить"]	[Кнопка "Выход"]

Рисунок 9 – Интерфейс основного окна приложений

3 Реализация и тестирование программного обеспечения

3.1 Реализация основных функций программного обеспечения

3.1.1 Авторизация в приложении «РокирочкаСompany»

Листинг приложения «РокирочкаСompany» описан в приложении А. Окно авторизации изображено на рисунке 10.

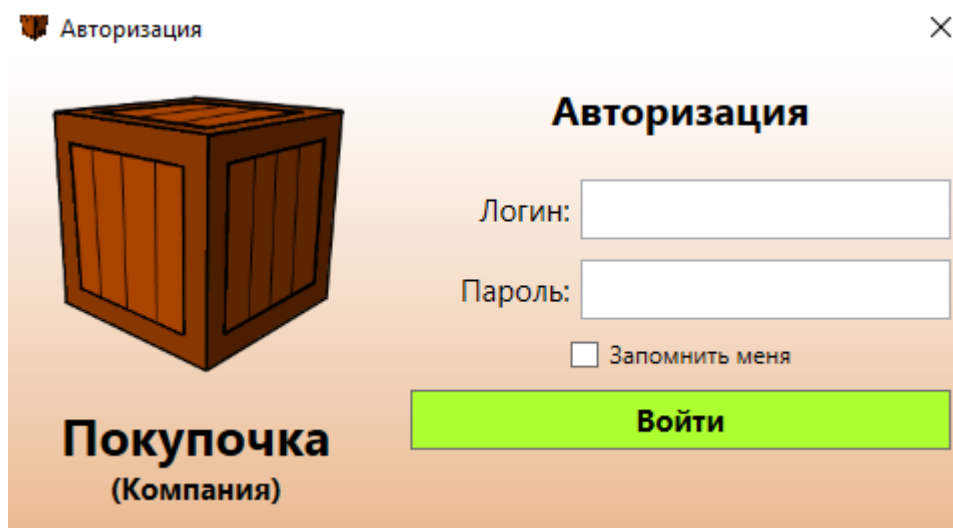


Рисунок 10 – Окно авторизации в «РокирочкаСompany»

Окно авторизации предназначено для входа в приложение. Оно содержит поля ввода логина, пароля, функцию «запомнить меня» и кнопку входа.

При нажатии кнопки «Войти», система ищет пользователя с введённым логином и паролем, при этом показывая «загрузку». При этом, блокируются все поля и кнопка входа в приложение (Рисунок 11).

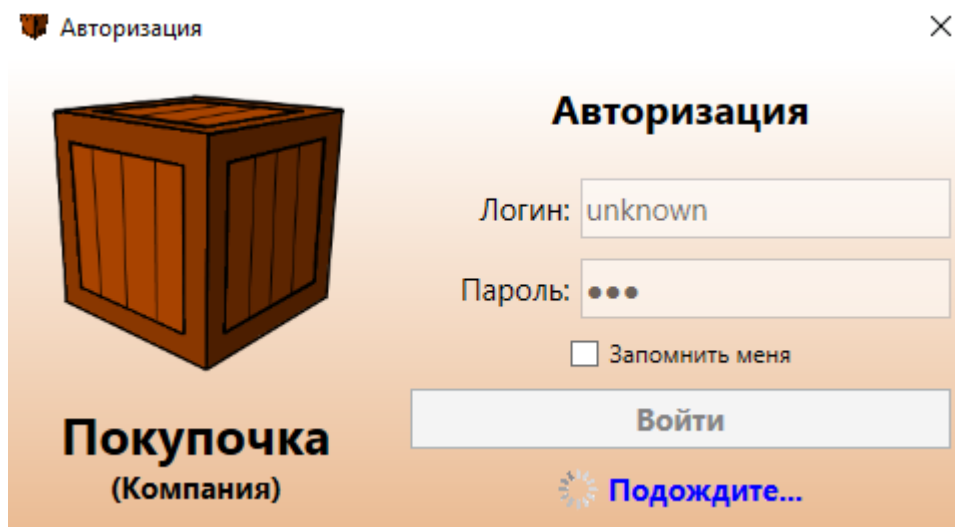


Рисунок 11 – Ожидание входа в приложение

При неудачной попытке входа, система уведомит об этом сообщением под кнопкой входа. Демонстрация такого поведения изображена на рисунке 12.

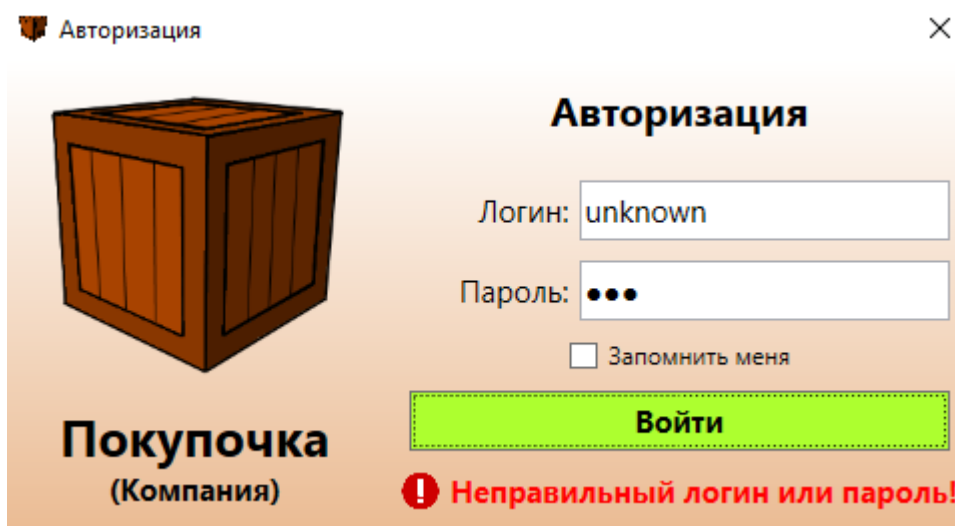


Рисунок 12 – Неудачная попытка входа в приложение

3.1.2 Авторизация в приложении «PokupochkaCounterparty»

Листинг приложения «PokupochkaCompanу» описан в приложении Б. Окно авторизации изображено на рисунке 13.

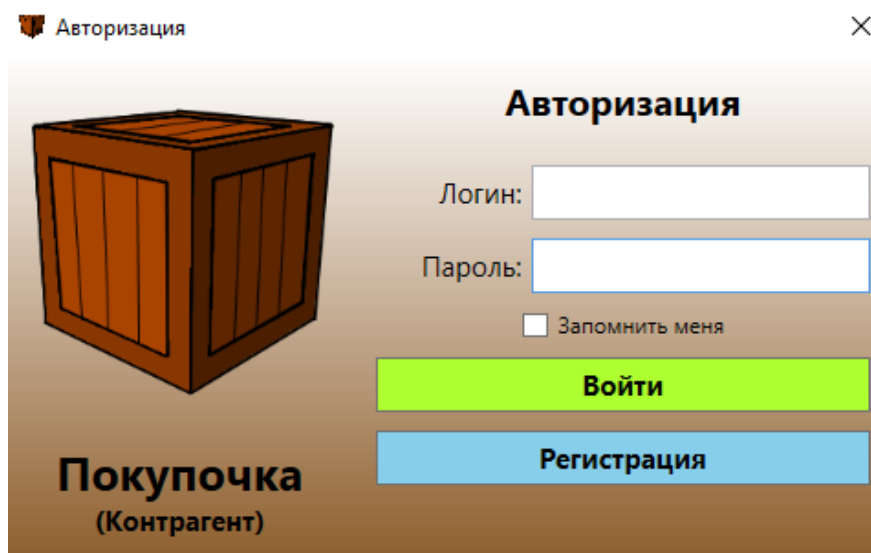


Рисунок 13 – Окно авторизации в «PokupochkaCounterparty»

Окно имеет такое же поведение, что и в приложении «PokupochkaCompany» (п. 3.1.1). Единственное отличие – наличие кнопки регистрации.

3.1.3 Регистрация в приложении PokupochkaCounterparty

После нажатия на кнопку «Регистрация», пользователю открывается окно выбора роли (Рисунок 14).



Рисунок 14 – Окно выбора роли

После выбора роли, открывается окно с регистрацией (Рисунок 15).

Рисунок 15 – Окно регистрации клиента

Клиент заполняет все необходимые данные, по желанию выбирает фотографию и после нажатия на кнопку «Добавить клиента», сможет войти под своей учётной записью.

3.1.4 Модуль «Администратор»

3.1.4.1 Главное окно администратора

В распоряжении администратора разработано окно, представленное на рисунке 16.

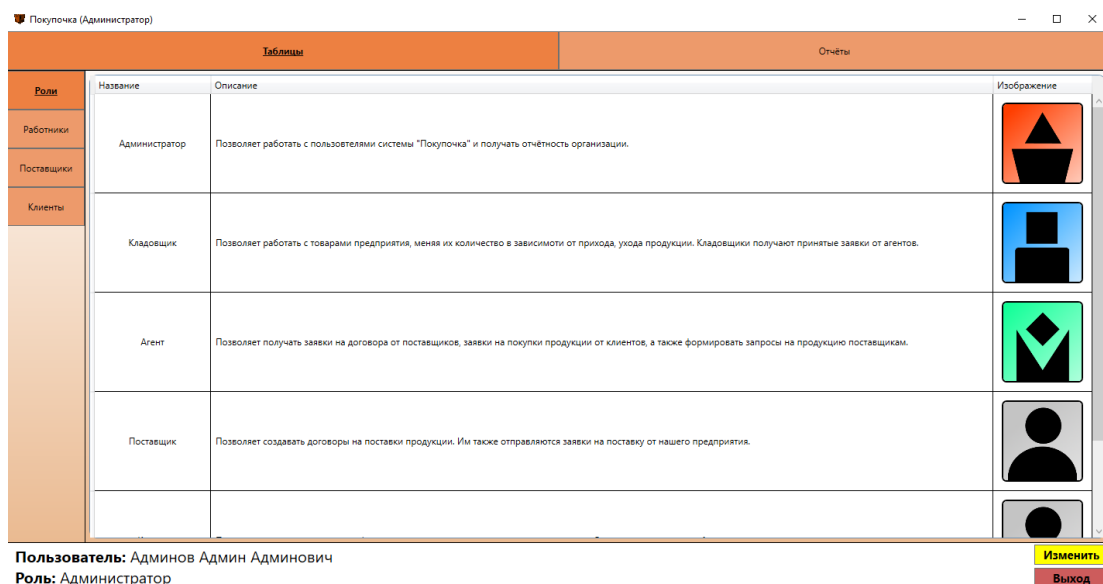


Рисунок 16 – Окно администратора

Из этого окна, администратор может перейти:

- раздел «Таблицы»;
- раздел «Отчёты».

Раздел таблицы предназначен для просмотра и редактирования некоторых таблиц в БД:

- таблица «Роли» с информацией о ролях в системе;
- таблица «Работники» с информацией о работниках и возможностью их редактирования;
- таблица «Поставщики» с информацией о поставщиках предприятия с возможностью редактирования;
- таблица «Клиенты» с информацией о клиентах предприятия с возможностью редактирования.

Раздел отчёты разработан для просмотра деятельности организации.

При нажатии на любую из кнопок, её фон меняется, а текст внутри становится жирным и подчёркивается. Это сделано для понятности, на какой вкладке находится администратор.

Если администратор введёт данные в странице, перейдёт на другую вкладку, а затем вернётся обратно, вся его введённая информация сохранится.

3.1.4.2 Таблица «Работники»

Интерфейс таблицы «Работники» показан на рисунке 17.

Информация о работнике	Роль
ФИО: Адинов Админ Админович Телефон: +8 (800) 555 35 21 Почта: aasdasdasdas Логин: ad Пароль: ad	Администратор
ФИО: Кладов Клад Кладович Телефон: +7 (777) 777 77 77 Почта: klad2022@yandex.ru Логин: st Пароль: st	Кладовщик
ФИО: Иванов Иван Иванович Телефон: +9 (999) 999 99 99 Почта: agent007@yandex.ru Логин: ag Пароль: ag	Агент

Рисунок 17 – Интерфейс таблицы «Работники»

На странице расположено поле поиска по ФИО и логину, присутствует фильтрация по ролям, а также кнопка сброса, чтобы вся введённая информация в поле поиска и фильтре сбросилась. Также присутствуют кнопки добавления, изменения и удаления работника.

3.1.4.3 Окно добавления работника

Окно добавления работника представлено на рисунке 18.

* Фамилия:

* Имя:

Отчество:

* Телефон: + _ () _ _ _

* Почта:

* Логин:

* Пароль:

Роль: Администратор

Отмена

Рисунок 18 – Окно добавления работника

Администратор должен ввести данные во все обязательные поля (отмечены красной звёздочкой), а также выбрать роль для работника. В случае, если какое-то из обязательных полей не заполнено, система выдаст предупреждение (Рисунок 19).

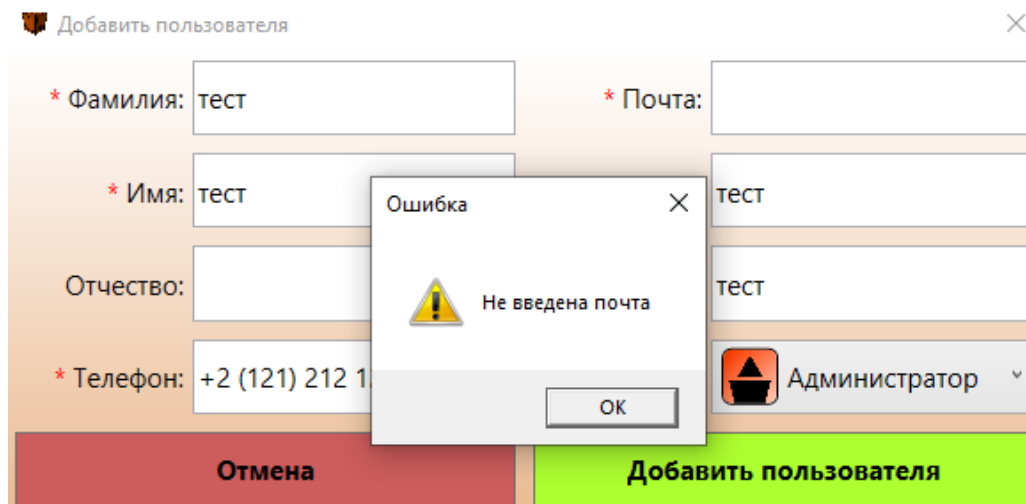


Рисунок 19 – Предупреждение о незаполненных обязательных полях

3.1.4.4 Таблица «Поставщики»

Интерфейс таблицы «Поставщики» показан на рисунке 20.

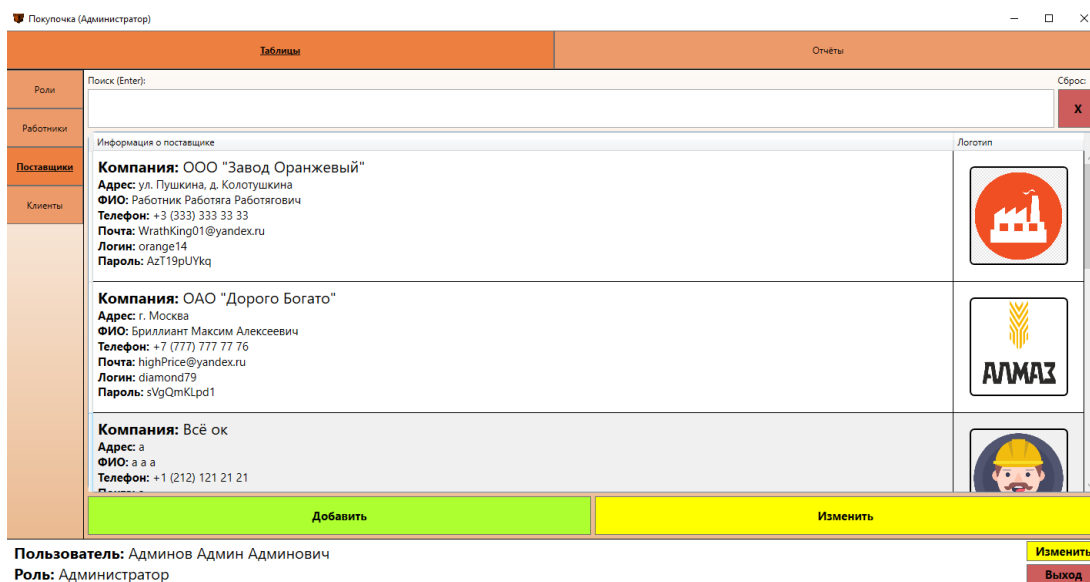


Рисунок 20 – Интерфейс таблицы «Поставщики»

На странице расположено поле поиска по названию компании и кнопка сброса, чтобы вся введенная информация в поле поиска сбросилась. Также присутствуют кнопки добавления и изменения поставщика.

3.1.4.5 Окно добавления поставщика

Окно добавления поставщика изображено на рисунке 21.

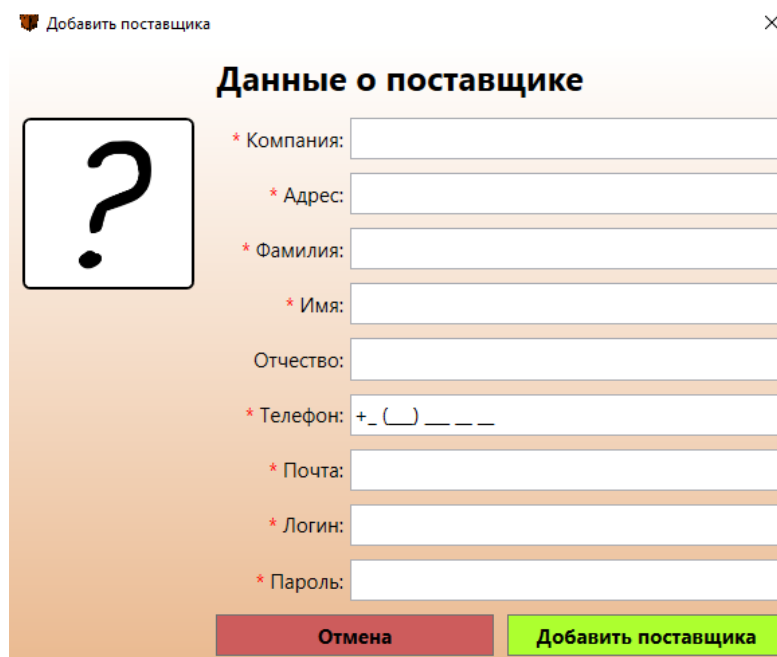


Рисунок 21 – Окно добавления поставщика

Администратор должен ввести данные во все обязательные поля (отмечены красной звёздочкой) для создания поставщика. В случае, если какое-то из обязательных полей не заполнено, система выдаст предупреждение.

3.1.4.6 Таблица «Клиенты»

Таблица «Клиенты» имеет схожий интерфейс, что и таблица «Поставщики». Переход на эту таблицу запускает другой запрос БД, благодаря которому выводятся только клиенты.

3.1.4.7 Окно добавления клиента

Окно добавления клиента схоже с окном добавления поставщика.

3.1.5 Модуль «Агент»

3.1.5.1 Главное окно агента

Для агента разработано окно, которое представлено на рисунке 22.

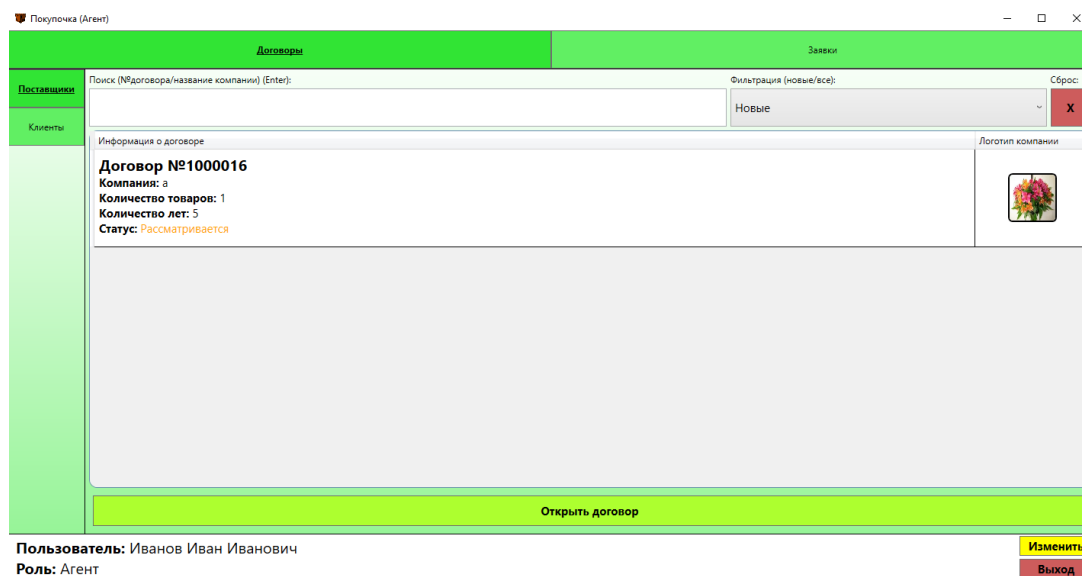


Рисунок 22 – Главное окно агента

Из этого окна, агент может перейти в:

- раздел «договоры»;
- раздел «заявки».

Раздел «договоры» предназначен для просмотра договоров поставщиков и клиентов. Поставщикам и клиентам разработаны разные разделы.

Раздел «заявки» предназначен для создания заявок для поставщиков и просмотра заявок клиентов. Заявкам разработаны разные разделы.

3.1.5.2 Договоры поставщиков

Страница договоров поставщиков изображена на рисунке 23.


Поиск (№договора/название компании) (Enter):		Фильтрация (новые/все):	Сброс:
		Новые	X
Информация о договоре Договор №1000016 Компания: а Количество товаров: 1 Количество лет: 5 Статус: Рассматривается		Логотип компании 	
Открыть договор			

Рисунок 23 – Страница договоров поставщиков

Агент может искать договоры по номеру договора и названию организации, фильтровать договоры по статусу (новые, активные/отклонённые), сбрасывать строку поиска и фильтрацию и открывать выбранный договор для просмотра.

3.1.5.3 Договоры клиентов

Страница договоров клиентов имеет схожий интерфейс со страницей договоров поставщиков.

3.1.5.4 Просмотр договора

Страница просмотра договора схожа у поставщиков и клиентов. Интерфейс страницы изображен на рисунке 24.

Договор №1000016
 Компания: а
 Дата начала: 30.05.2022
 Дата окончания: 30.05.2027

Данные о товаре	Изображение
Название: asdada Цена за ед.: 122121 руб.	

<
Отклонить
Принять

Рисунок 24 – Страница просмотра договоров агентом

Страница содержит номер договора, информацию о поставщике, товары, которые будет поставлять поставщик и кнопки «назад», «отклонить» и «принять». Агент принимает решение о отклонении или принятии договора, после чего статус договора меняется на выбранный.

3.1.5.5 Заявки поставщиков

Агенты создают заявки для поставок продукции от поставщиков. Страница всех заявок поставщикам представлена на рисунке 25.

Новый

#100000
38000,00Р
Доставлено

#100001
14000,00Р
В обработке

Рисунок 25 – Страница заявок поставщикам

На странице отображается кнопка «новый», которая ведёт к странице добавления новой заявки. Также на странице кнопками отображаются все заявки агента с указанием номера, общей стоимости и статуса.

3.1.5.6 Создание новой заявки

Страницы создания новой заявки представлен на рисунке 26.

Создание новой заявки

Доступные товары

Поиск (Enter): X

	Творог "Простоквашино" (5%) Количество: 0	40.00Р
	Молоко "Простоквашино" (3%) Количество: 100	40.00Р

> <

Выбранные товары

	Кол-во	
	Сметана "Простоквашино" (20%)	500 30000,00Р

< >

Итого: 30000,00 Р

< **Создать заявку**

Рисунок 26 – Страница создания новой заявки

Если товара на складе нет, товар подсвечивается красным. Так агент поймёт, какие товары нужно покупать в первую очередь. При выборе товара, он перемещается в правый список, в котором появляется возможность указания количества товара. При изменении количества товара, справа считается общая стоимость выбранного товара и общая стоимость всех товаров. После выбора всех необходимых товаров, агент нажимает на кнопку «создать заявку» и заявка добавляется в БД.

3.5.1.7 Заявки клиентов

Страница заявок клиентов представлена на рисунке 27.

Информация о заявке		Общая сумма
Заявка №100002 Количество товаров: 2 Статус: <i>Доставлено</i>		49400.00Р

Открыть заявку

Рисунок 27 – Страница заявок клиентов

Агент может искать заявки по номеру заявки и названию организации, фильтровать заявки по статусу (новые, в пути/доставлены), сбрасывать строку поиска и фильтрацию и открывать выбранную заявку для просмотра.

3.5.1.8 Просмотр заявки

Страница просмотра заявки представлена на рисунке 28.

Изображение	Данные о товаре	Общая сумма
	Название: Молоко "Простоквашино" (3%) Цена за ед.: 40.00 Р Количество: 200 шт.	8000.00Р
	Название: Сметана "Простоквашино" (20%) Цена за ед.: 60.00 Р Количество: 500 шт.	30000.00Р

Итого: 38000.00 Р

Рисунок 28 – Страница просмотра заявки

Страница содержит всю информацию о поставщике, товарах, их стоимости и общей стоимости товаров.

3.1.6 Модуль «Поставщик»

3.1.6.1 Главное окно поставщика

Поставщику разработано окно, которое изображено на рисунке 29.

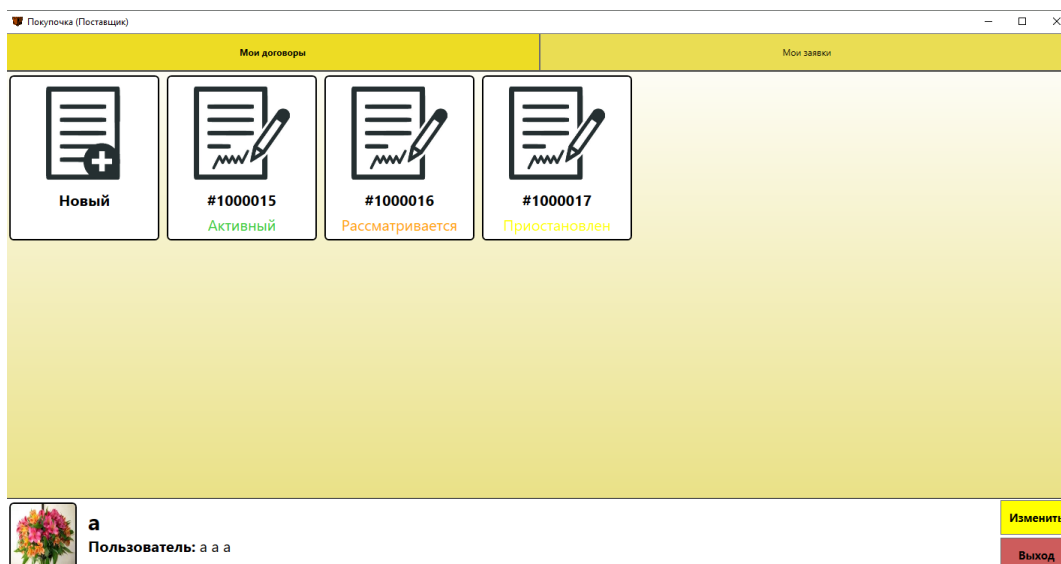


Рисунок 29 – Главное окно поставщика

Главное окно поставщика содержит два раздела:

- раздел «мои договоры», в котором отображаются договоры авторизованного поставщика;
- раздел «мои заявки», в котором отображаются заявки от предприятия оптовой торговли.

3.1.6.2 Мои договоры

Страница договоров поставщика представлена на рисунке 30.



Рисунок 30 – Страница договоров поставщика

Страница содержит перечисление всех договоров поставщика с отображением номера и статуса. Помимо этого, первым элементом страницы

является кнопка «новый», которая открывает страницу создания нового договора.

3.1.6.3 Новый договор

Страница создания нового договора изображена на рисунке 31.

Создание нового договора

? Название: Новый товар 1 Цена: 100 X

? Название: Новый товар 2 Цена: 500 X

+

Количество лет:

Отмена Создать

Рисунок 31 – Страница создания нового договора

Поставщик может добавлять товары в свой договор при нажатии на кнопку «+». После нажатия, появляется новый шаблон товара с картинкой, полями названия и цены и кнопкой «X», которая удалит шаблон.

После нажатия на кнопку «Создать», система проверит корректность введённых в поля ввода данных, после чего создаст новый договор со всеми товарами, указанными поставщиком на странице. Система сгенерирует автоматически номер договора, дату начала и окончания.

3.1.6.4 Просмотр договора

Страница просмотра договора поставщиком представлена на рисунке 32.

Договор №1000015

Дата начала: 30.05.2022
Дата окончания: 30.05.2027

Товары в договоре

?	sdadasda Цена: 212121
?	asdasdasd Цена: 2212121

<

Приостановить

Рисунок 32 – Просмотр договора поставщиком

Страница содержит номер договора, дату начала и окончания действия договора, список товаров, кнопки «<» (назад) и «приостановить». Нажатие на кнопку «приостановить», изменит статус договора на «приостановлен». Это полезно, если поставщик понимает, что указанного товара в договоре мало для продажи.

3.1.6.5 Мои заявки

Страница заявок поставщика представлена на рисунке 33.

Поиск (Enter):

Фильтрация:

Сброс:

Новые

X

Информация о заявке

Общая сумма

Заявка №100001
Количество товаров: 3
Статус: В обработке

14000.00Р

Открыть заявку

Рисунок 33 – Страница заявок поставщика

Сюда попадают заявки от агентов предприятия оптовой торговли. Поставщик может искать заявки по номеру заявки, фильтровать заявки по статусу (новые, в пути/доставлены), сбрасывать строку поиска и фильтрацию и открывать выбранную заявку для просмотра.

3.1.6.6 Просмотр заявки

Страница просмотра заявки схожа с агентом (п. 3.1.5.8).

3.1.7 Модуль «Клиент»

3.1.7.1 Главное окно клиента

Клиенту разработано окно, изображенное на рисунке 34.

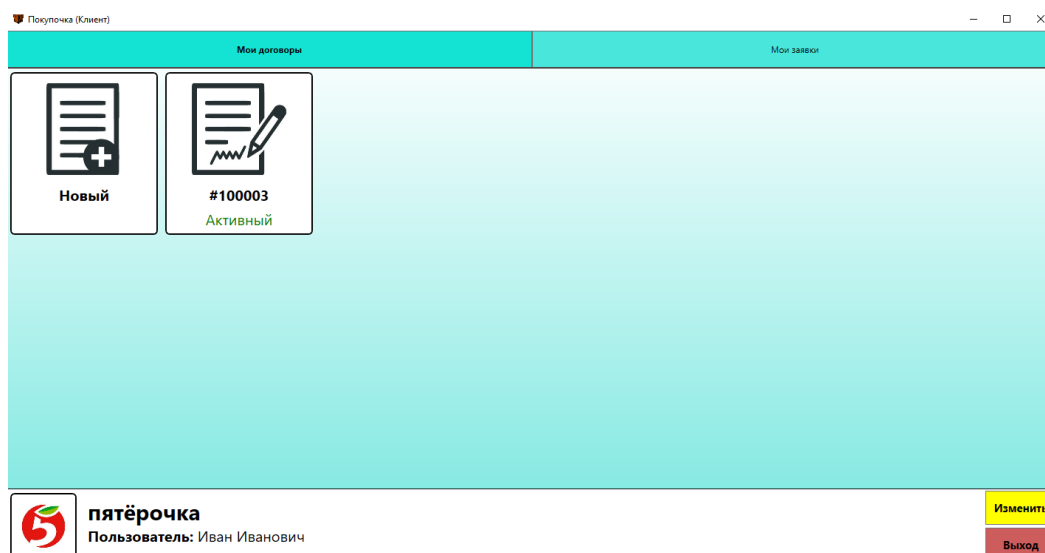


Рисунок 34 – Главное окно клиента

Главное окно клиента содержит два раздела:

- раздел «мои договоры», в котором отображаются договоры авторизованного клиента;
- раздел «мои заявки», в котором отображаются заявки авторизованного клиента.

3.1.7.2 Мои договоры

Страница договоров клиента изображена на рисунке 35.

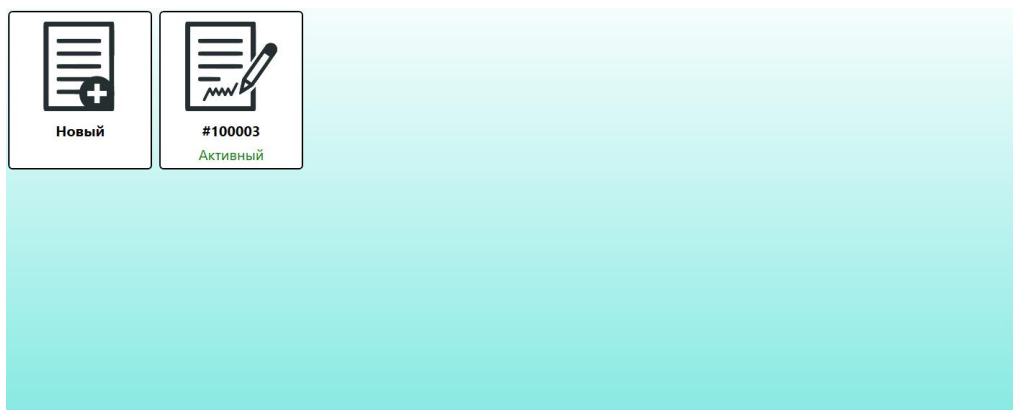


Рисунок 35 – Страница договоров клиента

Страница содержит все договоры клиента с указанием номера и статуса. Также представлена возможность создания нового договора.

3.1.7.3 Новый договор

Страница создания договора клиентом представлена на рисунке 36.

Доступные товары		Выбранные товары	
Поиск (Enter):			
Сметана "Простоквашино" (20%)	78.00Р	Молоко "Простоквашино" (3%)	52.00Р
Творог "Простоквашино" (5%)	52.00Р		

Количество лет: 5

Создать договор

Рисунок 36 – Страница создания договора

Слева отображается список товаров, которые на данный момент находятся на складах предприятия. Цены на товары отображаются с наценкой от предприятия.

3.1.7.4 Просмотр договора

Страница просмотра договора схожа с поставщиком (п. 3.1.6.4).

3.1.7.5 Мои заявки

Страница просмотра заявок схожа агентом (п. 3.1.5.5).

3.1.7.6 Создание заявки

Страница создания заявки клиентом представлена на рисунке 37.

Создание новой заявки

Доступные товары

Поиск (Enter):

Сметана "Простоквашино" (20%)
Количество: 1100 78.00Р

Выбранные товары

Молоко "Простоквашино" (3%)
Макс. кол-во: 200
Количество: 200 10400,00Р

Итого: 10400,00 Р

Создать заявку

Рисунок 37 – Страница создания заявки клиентом

Клиент может выбрать товары из составленных ранее договоров. После выбора нужного товара, он перемещается в правый список, в котором можно указать количество товара. При этом, количество товара не может быть больше количества, которое сейчас находится на складе предприятия.

3.1.7.7 Просмотр заявки

Страница просмотра заявки схожа с агентом (п. 3.1.5.8).

3.2 Тестирование программного обеспечения

Тестирование – набор операций, проводимых для обеспечения выявления и/или оценки свойств одного или более элементов тестирования. Тестирование помогает оценить качество ПО, снизить количество ошибок и получить информацию о состоянии ПО на данный момент [9].

На сегодняшний день, одним из основных методов оценки качества ИТ-продуктов является тестирование [8]. Разработчикам важно предоставить свой продукт клиентам с минимальным количеством ошибок, чтобы клиент не думал перейти к конкурентам.

Существует два вида тестирования:

- ручное тестирование (тест-кейсы);
- автоматическое тестирование (unit-тесты).

Для тестирования платформы учёта оптовой торговли выбраны оба вида тестирования.

3.2.1 Ручное тестирование

Ручное тестирование – это выполнение шагов, которые должны привести к какому-либо ожидаемому результату. Шаги выполнения расписываются в тест-кейсах [8]. Преимущество тест-кейсов в том, что их может выполнять даже тот, кто не знаком с системой. Достаточно выполнить написанные в тест-кейсе шаги, написать фактический результат и ручное тестирование на этом закончится.

Типичный шаблон тест-кейса изображен на рисунке 38.

Тестовый пример #1:

Тестовый пример #	
Приоритет тестирования	
Заголовок/название теста	
Краткое изложение теста	
Этапы теста	
Тестовые данные	
Ожидаемый результат	
Фактический результат	
Статус	
Предварительное условие	
Постусловие	
Примечания/комментарии	

Рисунок 38 – Шаблон тест-кейса

3.2.1.1 Тест-кейс #1

Тест-кейс проверки авторизации под администратором указан в таблице 2.

Таблица 2 – Тест-кейс проверки авторизации под администратором

Тестовый пример #	1
Приоритет тестирования	Высокий
Заголовок/название теста	Проверка входа в систему под администратором
Краткое изложение теста	Необходимо проверить корректность входа в систему под логином и паролем администратора
Этапы теста	1. Ввести в поля логина и пароля тестовые данные 2. Нажать кнопку «Войти»
Тестовые данные	Логин: ad Пароль: ad
Ожидаемый результат	Открыто окно администратора с названием «Покупочка (Администратор)»
Фактический результат	Открыто окно администратора с названием «Покупочка (Администратор)»

Продолжение таблицы 2

Статус	Пройден
Предварительное условие	Запустить приложение «РокироскаCompany»
Постусловие	Открыто окно администратора с названием «Покупочка (Администратор)»
Примечания/комментарии	-

3.2.1.1 Тест-кейс #2

Тест-кейс проверки добавления работника указан в таблице 3.

Таблица 3 – Тест-кейс проверки добавления работника

Тестовый пример #	2
Приоритет тестирования	Высокий
Заголовок/название теста	Проверка добавления работника
Краткое изложение теста	Необходимо проверить корректность добавления работника
Этапы теста	<ol style="list-style-type: none"> 1. Перейти в раздел «Работники» 2. Нажать на кнопку «Добавить» 3. Ввести в поля ввода тестовые данные 4. Нажать кнопку «Добавить пользователя»
Тестовые данные	<p>Фамилия: Иванов</p> <p>Имя: Иван</p> <p>Телефон: 12345678910</p> <p>Почта: ivan.ivanovich@yandex.ru</p> <p>Логин: ivan</p> <p>Пароль: qwerty</p> <p>Роль: Агент</p>
Ожидаемый результат	Работник успешно добавлен и отображен в списке работников

Продолжение таблицы 3

Фактический результат	Работник успешно добавлен и отображен в списке работников
Статус	Пройден
Предварительное условие	1. Запустить приложение «РокироскаCompany» 2. Войти под «Администратором»
Постусловие	Работник успешно добавлен и отображен в списке работников
Примечания/комментарии	-

3.2.1.1 Тест-кейс #3

Тест-кейс проверки удаления работника указан в таблице 4.

Таблица 4 – Тест-кейс проверки удаления работника

Тестовый пример #	3
Приоритет тестирования	Высокий
Заголовок/название теста	Проверка удаления работника
Краткое изложение теста	Необходимо проверить корректность удаления работника
Этапы теста	1. Перейти в раздел «Работники» 2. Выбрать любого работника 3. Нажать кнопку «Удалить» 4. Подтвердить удаление
Тестовые данные	-
Ожидаемый результат	Работник успешно удалён и не отображается в списке работников
Фактический результат	Работник успешно удалён и не отображается в списке работников
Статус	Пройден

Продолжение таблицы 4

Предварительное условие	1. Запустить приложение «РокироскаCompanу» 2. Войти под «Администратором»
Постусловие	Работник успешно удалён и не отображается в списке работников
Примечания/комментарии	-

3.2.2 Автоматическое тестирование

Автоматическое тестирование позволяет исключить человеческий фактор в прохождении тестов. Автоматизация – наилучший способ сэкономить время, повысить эффективность тестов, а также точность тестирования в случае, если тесты необходимо повторить [6]

Преимущества и недостатки автоматического тестирования представлены в таблице 5. [6]

Таблица 5 – Плюсы и минусы автоматизированных тестов

Плюсы	Минусы
Меньше времени на выполнение тестов в сравнении с ручным тестированием	Любое изменение в работе системы может потребовать переделать все тесты
Возможность проведения тестов, которые нельзя провести без использования автоматизированных тестов	Ошибка в одном из тестов может привести к ошибочным результатам остальных тестов
Исключается человеческий фактор	Риск ошибок в самом автоматическом тесте
Возможность проводить тестирование в любое время	Не все функциональные способности можно покрыть автоматизированными тестами

3.2.2.1 Unit-тест #1

Необходимо проверить корректное добавление работника в БД. Результат тестирования изображен на рисунке 39. Листинг теста и объяснение описано в приложении В1.

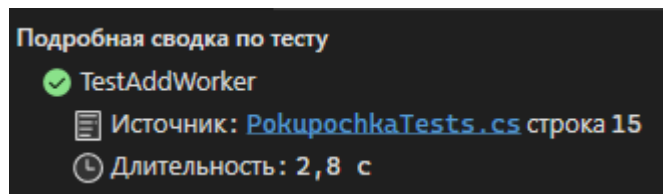


Рисунок 39 – Результат unit-теста добавления работника

3.2.2.2 Unit-тест #2

Необходимо проверить корректное изменение работника в БД. Результат тестирования изображен на рисунке 40. Листинг unit-теста и объяснение описано в приложении В2.

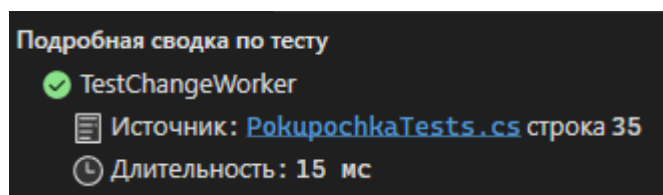


Рисунок 40 – Результат unit-теста изменения работника

3.2.2.3 Unit-тест #3

Необходимо проверить корректное удаление работника в БД. Результат тестирования изображен на рисунке 41. Листинг unit-теста и объяснение описано в приложении В3.

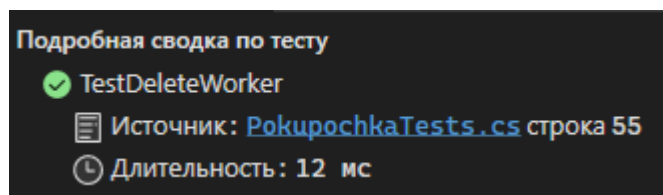


Рисунок 41 – Результат unit-теста удаления работника

ЗАКЛЮЧЕНИЕ

В результате выполнения дипломного проекта, разработана платформа учёта оптовой торговли с модулями «Администратор», «Агент» и «Поставщик».

Были выполнены следующие задачи:

- описана предметная область;
- проведён обзор аналогов;
- выделены основные требования к системе;
- разработана модель базы данных;
- спроектирована информационная система;
- реализованы основные функции программного обеспечения;
- протестировано программное обеспечение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Басова Т.А. РАЗРАБОТКА БАЗЫ ДАННЫХ ПО ТЕХНОЛОГИИ КЛИЕНТ-СЕРВЕР // Вестник магистратуры. 2019. №11-4 (98). URL: <https://cyberleninka.ru/article/n/razrabotka-bazy-dannyh-po-tehnologii-klient-server>.
- 2 Вальдивия Х.А., Лора-Гонсалес А., Лимон К., Кортес-Вердин К., Очаран-Эрнандес Х.О. ПАТТЕРНЫ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ: МНОГОПРОФИЛЬНЫЙ ОБЗОР ЛИТЕРАТУРЫ // Труды ИСП РАН. 2021. №1. URL: <https://cyberleninka.ru/article/n/patterny-mikroservisnoy-arhitektury-mnogoprofilnyy-obzor-literatury>.
- 3 Журавлев Александр Александрович СОЗДАНИЕ ПРИЛОЖЕНИЯ ДЛЯ УЧЕТА ОЦЕНОК СТУДЕНТОВ ЗА ДИПЛОМНУЮ РАБОТУ С ПОМОЩЬЮ WINDOWS PRESENTATION FOUNDATION // StudNet. 2021. №6. URL: <https://cyberleninka.ru/article/n/sozдание-prilozheniya-dlya-ucheta-otsenok-studentov-za-diplomnuyu-rabotu-s-pomoschyu-windows-presentation-foundation>.
- 4 Карандеева Ирина Юрьевна, Карандеев Денис Юрьевич Рейтинг языков программирования RedMonk как оптимальный инструмент для выявления тенденций в популярности языков программирования // E-Scio. 2021. №1 (40). URL: <https://cyberleninka.ru/article/n/rejting-yazykov-programmirovaniya-redmonk-kak-optimalnyy-instrument-dlya-vyyavleniya-tendentsiy-v-populyarnosti-yazykov>
- 5 Ковальчук А. А., Павленко Е. П., Айвазов В. А. Применение программных технологий WPF и Silverlight при разработке информационных систем // ВЕЖПТ. 2020. №2 (37). URL: <https://cyberleninka.ru/article/n/primenenie-programmnyh-tehnologiy-wpf-i-silverlight-pri-razrabotke-informatsionnyh-sistem>
- 6 Криволятева Мария Сергеевна, Гайкова Любовь Вадимовна ПРОБЛЕМЫ ВЫБОРКИ ТЕСТ-КЕЙСОВ ДЛЯ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ ИТ-ПРОДУКТОВ // НК. 2020. №1. URL: <https://cyberleninka.ru/article/n/problemy-vyborki-test-keysov-dlya-avtomaticheskogo-testirovaniya-it-produktov>

- 7 Мазанкова Татьяна Васильевна ОЦЕНКА ЗАНЯТОСТИ В СФЕРЕ УСЛУГ // Власть и управление на Востоке России. 2021. №1 (94). URL: <https://cyberleninka.ru/article/n/otsenka-zanyatosti-v-sfere-uslug>.
- 8 Моисеев Даниил Александрович Методология и процесс ручного тестирования // НиКСС. 2020. №3 (19). URL: <https://cyberleninka.ru/article/n/metodologiya-i-protsess-ruchnogo-testirovaniya>
- 9 Мостипака Анастасия Евгеньевна ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ // Наука и образование сегодня. 2020. №12 (59). URL: <https://cyberleninka.ru/article/n/testirovanie-programmnogo-obespecheniya>
- 10 Руководство пользователя StarUML. — Текст : электронный // staruml.sourceforge.net : [сайт]. — URL: [http://staruml.sourceforge.net/docs/user-guide\(ru\)/user-guide.pdf](http://staruml.sourceforge.net/docs/user-guide(ru)/user-guide.pdf)
- 11 Система программ 1С:Предприятие, официальный сайт - поддержка и обучение, приобретение и внедрение // 1С:Предприятие URL: <https://v8.1c.ru/>
- 12 Что такое реляционная база данных. — Текст : электронный // Oracle : [сайт]. — URL: <https://www.oracle.com/cis/database/what-is-a-relational-database/>
- 13 Шарипова Н.А., Вентер Ю.А. Информатизация Российской оптовой торговли // СТЭЖ. 2020. №2 (23). URL: <https://cyberleninka.ru/article/n/informatizatsiya-rossiyskoy-optovoy-torgovli>.
- 14 Шикуть Алла Васильевна, Аристов Борис Константинович, Просуков Евгений Алексеевич Особенности. Достоинства и преимущества использования с# в приложениях // Символ науки. 2021. №11-3. URL: <https://cyberleninka.ru/article/n/osobennosti-dostoinstva-i-preimuschestva-ispolzovaniya-s-v-prilozheniyah>
- 15 .NET (и .NET Core) — введение и обзор // Microsoft URL: <https://docs.microsoft.com/ru-ru/dotnet/core/introduction>

ПРИЛОЖЕНИЕ А

Листинг приложения РокирошкaCompany

А1 Листинг окна авторизации

При открытии окна авторизации приложения, устанавливаются настройки этого окна (Листинг 1).

Листинг 1 – Установка настроек окна авторизации

```
private void SetWinSettings()
{
    UserStyles.SetUserStyles("CompanyStyles.xaml");
    Style = UserStyles.WindowSyle;

    UserDataReader.Set();
    if (UserDataReader.UserData.Remember)
    {
        TbLogin.Text = UserDataReader.UserData.Login;
        PbPassword.Password =
UserDataReader.UserData.Password;
        ChbMemory.IsChecked = true;
    }
}
```

Первым делом устанавливается стиль окна. Метод класса UserStyles SetUserStyles() устанавливает стили из указанного файла в поля класса (Листинг 2).

Листинг 2 – Установка настроек окна авторизации

```
public static void SetUserStyles(string xamlFile)
{
    string xamlUri = "/UserStyles/" + xamlFile;

    ResourceDictionary resource = new();
    resource.Source = new Uri(xamlUri, UriKind.Relative);

    WindowSyle = resource["WindowStyle"] as Style;
    DefaultButtonStyle = resource["DefaultButtonStyle"] as
Style;
    SelectButtonStyle = resource["SelectButtonStyle"] as
Style;
}
```

Метод находит необходимые стили в указанном XAML-файле. Далее, необходимый стиль (в поле WindowStyle) присваивается окну авторизации.

Вернёмся к листингу 1. После установки стилей, вызывается метод Set() класса UserDataReader, который ищет JSON-файл с данными пользователя (Листинг 3).

Листинг 3 – Поиск данных пользователя

```
private static string jsonUri = "user.json";

public static UserData UserData = null!;

public static void Set()
{
    if (File.Exists(jsonUri))
    {
        UserData =
JsonConvert.DeserializeObject<UserData>(File.ReadAllText(jsonUri))
!;
    }
    else UserData = new();
}
```

Если пользователь до этого заходил в систему и ставил галочку «запомнить меня», то метод найдёт данные пользователя и подставит их в поля Логина и Пароля, а также снова поставит галочку «запомнить меня».

После проведения всех операций с настройками окна, оно открывается для пользователя.

Пользователь вводит данные (логин, пароль) и нажимает на кнопку «Войти» (Листинг 4).

Листинг 4 – Кнопка входа

```
private async void BtnEnter_Click(object sender,
RoutedEventArgs e)
{
    BtnEnter.IsEnabled = false;
    TbLogin.IsEnabled = false;
    PbPassword.IsEnabled = false;

    CustomMessage message = new();
    await message.ShowMessage(SpMessage,
MessageType.Loading, "Подождите...");

    List<User> users = DbConnect.Db.Users.Include(u =>
u.Role).Include(u => u.Worker).ToList();

    User? findUser = users.FirstOrDefault(user =>
    user.Login == TbLogin.Text &&
    user.Password == PbPassword.Password &&
```

```

        (user.RoleId == (int)RoleKey.Administrator ||
         user.RoleId == (int)RoleKey.Storekeeper ||
         user.RoleId == (int)RoleKey.Agent));

        if (findUser != null) LoginToTheApp(findUser);
        else await message.ShowMessage(SpMessage,
MessageTypes.Error, "Неправильный логин или пароль!");

        BtnEnter.IsEnabled = true;
        TbLogin.IsEnabled = true;
        PbPassword.IsEnabled = true;
    }

```

Первым делом, блокируются поля ввода и кнопка входа. Затем, метод ShowMessage выводит на экран сообщение об ожидании (Листинг 5).

Листинг 5 – Вывод сообщения

```

public async Task ShowMessage(StackPanel sp, MessageTypes
type, string text)
{
    string imgUrl = "/Images/Messages/";
    Brush brush = null!;
    SetImgAndBrush(type, ref imgUrl, ref brush);

    if (sp.Children.Count > 0)
        sp.Children.Clear();

    BitmapImage bitmap = new();
    bitmap.BeginInit();
    bitmap.UriSource = new Uri(imgUrl, UriKind.Relative);
    bitmap.EndInit();

    Image img = new();
    ImageBehavior.SetAnimatedSource(img, bitmap);
    img.Width = 20;
    img.Height = 20;
    sp.Dispatcher.Invoke(() => sp.Children.Add(img));

    TextBlock tbl = new TextBlock();
    tbl.VerticalAlignment = VerticalAlignment.Center;
    tbl.Margin = new Thickness(5, 0, 0, 0);
    tbl.Foreground = brush;
    tbl.FontWeight = FontWeights.Bold;
    tbl.FontSize = 15;
    tbl.Text = text;
    sp.Dispatcher.Invoke(() => sp.Children.Add(tbl));

    await Task.Delay(2000);
}

```

Данный метод позволяет вывести картинку и сообщение к ней на экран.

Введённые данные пользователем используются для поиска пользователя в БД. При этом, ищутся только те пользователи, роль которых «Администратор», «Кладовщик» или «Агент».

Если пользователь найден, вызывается метод `LoginToTheApp()` с передачей найденного пользователя в БД (Листинг 6).

Листинг 5 – Авторизация в приложении

```
private void LoginToTheApp(User user)
{
    CurrentUser.User = user;

    UserDataReader.UserData.Login = user.Login;
    UserDataReader.UserData.Password = user.Password;
    UserDataReader.UserData.Remember =
(bool) ChbMemory.IsChecked!;
    UserDataReader.Save();

    PokupCompWin window = new();
    window.Show();
    Close();
}
```

Метод сохраняет пользователя в память, а затем сохраняет введённые данные в JSON файл. После этого, открывается главное окно приложения `PokuposkaCompany`.

ПРИЛОЖЕНИЕ Б

Листинг приложения PokurochkaCounterparty

Б1 Листинг окна авторизации

Окно авторизации PokurochkaCounterparty схоже с окном авторизации PokurochkaCompany. Листинг окна авторизации PokurochkaCompany описан в приложении А1.

Единственное отличие – кнопка входа в окне авторизации (Листинг 6).

Листинг 6 – Кнопка входа

```
private async void BtnEnter_ClickAsync(object sender,
RoutedEventArgs e)
{
    BtnEnter.IsEnabled = false;
    TbLogin.IsEnabled = false;
    PbPassword.IsEnabled = false;

    CustomMessage message = new();
    await message.ShowMessage(SpMessage,
MessageType.Loading, "Подождите...");

    List<User> users = DbConnect.Db.Users.Include(u =>
u.Role).Include(u => u.Counterparty).ToList();

    User? findUser = users.FirstOrDefault(user =>
        user.Login == TbLogin.Text &&
        user.Password == PbPassword.Password &&
        (user.RoleId == (int)RoleKey.Supplier ||
        user.RoleId == (int)RoleKey.Client));

    if (findUser != null) LoginToTheApp(findUser);
    else await message.ShowMessage(SpMessage,
MessageType.Error, "Неправильный логин или пароль!");

    BtnEnter.IsEnabled = true;
    TbLogin.IsEnabled = true;
    PbPassword.IsEnabled = true;
}
```

Введённые данные пользователем используются для поиска пользователя в БД. При этом, ищутся только те пользователи, роль которых «Поставщик» или «Клиент».

ПРИЛОЖЕНИЕ В

Листинг unit-тестов

В1 Листинг unit-теста #1

Unit-тест вызывает метод `TestAddWorker()`, который отвечает за проверку корректного добавления работника в БД (Листинг 7).

Листинг 7 – Тестирование добавления работника в БД

```
[TestMethod]
public void TestAddWorker()
{
    Worker worker = new();
    worker.SecondName = "TEST";
    worker.FirstName = "TEST";
    worker.Patronymic = "TEST";
    worker.Phone = "+7(777) 777-77-77";
    worker.Email = "TEST@yandex.ru";
    worker.User!.Login = "test";
    worker.User!.Password = "test";
    worker.User!.RoleId = (int)RoleKey.Administrator;
    worker.AddOrChange();

    Worker? findWorker =
DbConnect.Db.Workers.FirstOrDefault(w => w.User!.Login ==
worker.User!.Login);
    Assert.AreEqual(worker, findWorker, "Работник не
найден в БД");

    worker.User!.Delete();
}
```

Первым делом, создаётся экземпляр класса `Worker` (работник) и заполняются все необходимые данные. Далее, вызывается метод `AddOrChange()` класса `Worker`, который отвечает за добавление или изменение работника в БД (Листинг 8)

Листинг 8 – Добавление работника в БД

```
public void AddOrChange()
{
    if (Id == 0) DbConnect.Db.Workers.Add(this);
    DbConnect.Db.SaveChanges();
}
```

После добавления работника в БД, происходит поиск этого самого работника в БД, путём вызова метода `FirstOrDefault()`. Далее, новый работник сверяется с найденным в БД.

После прохождения теста, работник удаляется из БД, так как он был создан для теста.

В2 Листинг unit-теста #2

Unit-тест вызывает метод `TestChangeWorker()`, который отвечает за проверку корректного изменения работника в БД (Листинг 9).

Листинг 9 – Изменение работника в БД

```
[TestMethod]
public void TestChangeWorker()
{
    Random rnd = new();
    List<Worker> workers = DbConnect.Db.Workers.ToList();

    Worker rndWorker = workers[rnd.Next(workers.Count)];
    string oldLogin = rndWorker.User!.Login;
    string newLogin = "TEST";

    rndWorker.User!.Login = newLogin;
    rndWorker.AddOrChange();

    Worker? findWorker =
    DbConnect.Db.Workers.FirstOrDefault(w => w.Id == rndWorker.Id);
    Assert.AreEqual(newLogin, findWorker!.User!.Login,
    "Данные работника не изменились");

    rndWorker.User!.Login = oldLogin;
    rndWorker.AddOrChange();
}
```

Первым делом, создаётся новый объект класса `Random` (для выбора случайного работника в БД), далее, достаются все работники из БД и объекту `rndWorker` присваивается случайный работник. Затем, объявляется переменная `oldLogin`, в который сохраняется текущий логин работника.

После сохранения старого логина работника, ему присваивается новый ("TEST"). Запись сохраняется вызовом метода `AddOrChange()` (Листинг 8), после чего происходит поиск нужного работника в БД.

Далее сравнивается логин работника с новым логином. Если они одинаковы, тест пройден. После проверки, старый логин работника возвращается.

В3 Листинг unit-теста #3

Unit-тест вызывает метод `TestChangeWorker()`, который отвечает за проверку корректного удаления работника в БД (Листинг 10).

Листинг 10 – Удаление работника в БД

```
public void TestDeleteWorker()
{
    Worker worker = new();
    worker.SecondName = "TEST";
    worker.FirstName = "TEST";
    worker.Patronymic = "TEST";
    worker.Phone = "+7(777) 777-77-77";
    worker.Email = "TEST@yandex.ru";
    worker.User!.Login = "test";
    worker.User!.Password = "test";
    worker.User!.RoleId = (int)RoleKey.Administrator;
    worker.AddOrChange();
    worker.User!.Delete();

    Worker? findWorker =
    DbConnect.Db.Workers.FirstOrDefault(w => w.User!.Login ==
    worker.User!.Login);
    Assert.AreEqual(null, findWorker, "Работник не удалён
из БД");
```

Первым делом, создаётся новый работник с данными. Созданный работник добавляется в БД, а затем удаляется. После, происходит поиск созданного работника в БД и, если он не найден, тест считается пройденным.