МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет-изданий)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«База данных для службы по ремонту техники с использованием технологии резервного копирования и восстановления»

Выполнил студент Сятковская Е. Д.

(Ф.И.О.)

Руководитель проекта ассистент Нистюк О. А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В .

(учен. степень, звание, должность, Ф.И.О., подпись)

Консультант: ассистент Нистюк О. А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: ассистент Нистюк О. А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2022

**Содержание**

[Введение 3](#_Toc121137066)

[1 Аналитический обзор аналогов 4](#_Toc121137067)

[1.1 Программное средство «РемОнлайн» 4](#_Toc121137068)

[1.2 Программное средство «RepairCenter» 5](#_Toc121137069)

[2 Разработка архитектуры проекта 6](#_Toc121137070)

[2.1 Обобщенная структура управления приложением 6](#_Toc121137071)

[2.2 Проектирование модели базы данных 7](#_Toc121137072)

[3 Использование Oracle 12c 11](#_Toc121137073)

[3.1 Создание таблиц 11](#_Toc121137074)

[3.2 Создание ролей для разграничения доступа 11](#_Toc121137075)

[3.3 Создание пакетов и процедур для работы с базой данных 12](#_Toc121137076)

[3.4 Описание процедур экспорта и импорта 13](#_Toc121137077)

[3.5 Тестирование производительности базы данных 15](#_Toc121137078)

[4 Описание технологии резервного копирования и восстановления 18](#_Toc121137079)

[5 Руководство пользователя 19](#_Toc121137080)

[5.1 Область клиента 19](#_Toc121137081)

[5.2 Область сотрудника 21](#_Toc121137082)

[5.3 Тестирование клиентской области 23](#_Toc121137083)

[5.4 Тестирование области работника 25](#_Toc121137084)

[Заключение 27](#_Toc121137085)

[Cписок источников 28](#_Toc121137086)

[Приложение А 29](#_Toc121137087)

[Приложение Б 30](#_Toc121137088)

[Приложение В 31](#_Toc121137089)

# **Введение**

В данной записке приведено описание реляционной базы данных и desktop-приложения, разработанных в соответствии с заданием на курсовое проектирование по теме «База данных для службы по ремонту техники с использованием технологии резервного копирования и восстановления».

Цель разработки базы данных – спроектировать максимально удобную для встраивания в программное средство базу данных, внутри которой будет осуществлена вся работа с таблицами с помощью процедур.

Данная база данных должна находится на удалённом сервере и быстро разворачиваться на любом другом сервере, посредством последовательного выполнения скриптов.

В базе данных должны присутствовать разграничения для различных пользователей, таких как клиент и работник. Это было сделано для того, чтобы пользователь не смог повредить базу данных.

В качестве интерфейса выбран API Windows Forms. Для работы с данным интерфейсом использовался объектно-ориентированный язык программирования C#.

Приложение будет обладать следующими функционалом:

* регистрация клиентов;
* авторизация клиентов и сотрудников;
* формирование заказа клиентом;
* просмотр заказов клиентов сотрудниками;
* изменение статуса заказа;
* заказ деталей сотрудником.

Для разработки и управления базой данных курсового проекта использовалась система управления реляционными базами данных Oracle 12с.

# **1 Аналитический обзор аналогов**

В наши дни множество аналогов программных средств можно найти на различных веб-ресурсах, связанных с данной тематикой. Перед тем как приступить к разработке требовалось провести анализ существующих аналогов.

1.1 Программное средство «РемОнлайн»

На рисунке 1.1 предствлено окно приложения «РемОнлайн»

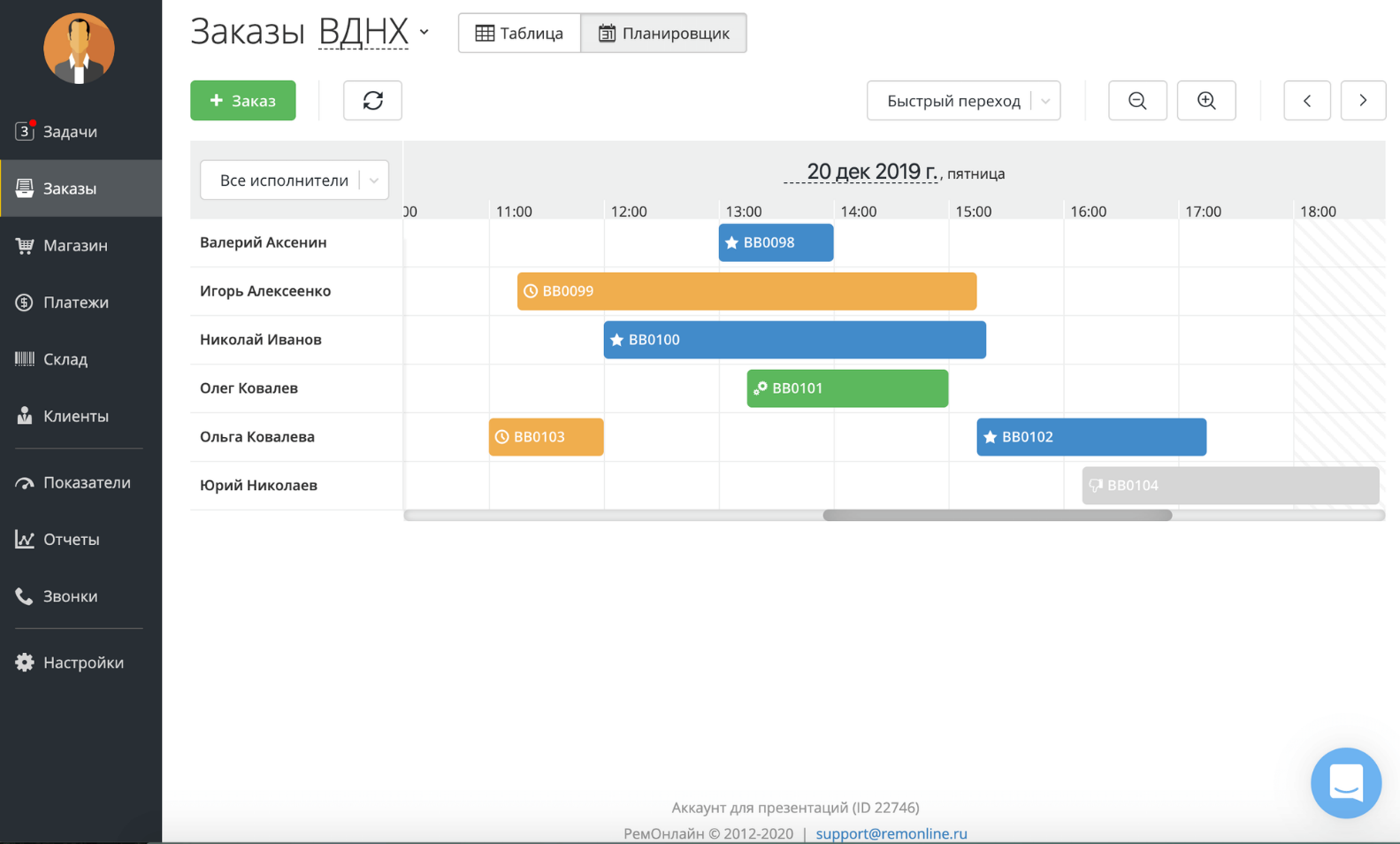


Рисунок 1.1 – Окно аналога приложения «РемОнлайн»

Ознакомившись с программных средством, был сделан вывод об основных функциональных особенностях:

* использование списка для представления множества данных;
* возможность просматривать клиентов, заказы, склад;
* возможность осуществлять sms и email рассылку;
* просмотр отчётов;
* возможность обновлять данные в списке;
* просмотр итого по выборке.

1.2 Программное средство «RepairCenter»

На рисунке 1.2 представлено главное окно приложения «RepairCenter».

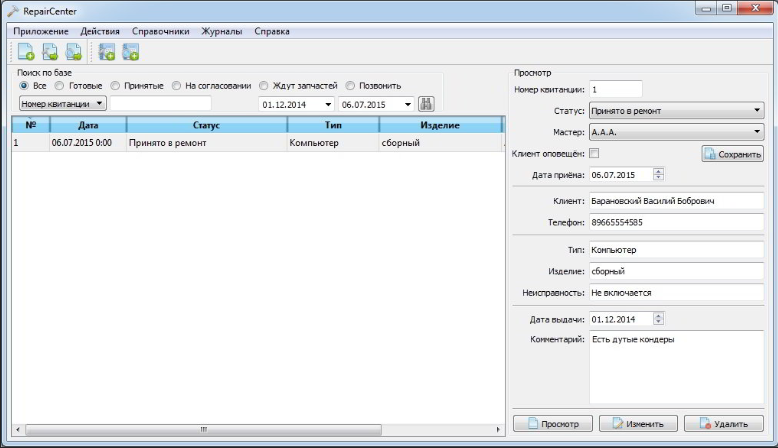


Рисунок 1.2 – Окно аналога приложения «RepairCenter»

Ознакомившись с программных средством, был сделан вывод об основных функциональных особенностях:

* наличие формы приёмной квитанции с множеством различных полей для заполнения;
* возможность просмотра истории;
* возможность просмотра планируемых работ;
* возможность формирования квитанции.

# **2 Разработка архитектуры проекта**

## **2.1** **Обобщенная структура управления приложением**

В приложении имеется разделение пользователей на клиентов и работников. Роль администратора в приложении не предусмотрена, она возлагается на администратора базы данных.

Для наглядности демонстрации возможностей пользователей воспользуемся UML-диаграммами. UML (Unified Modeling Language – уникальный язык моделирования) – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем.

Диаграмма вариантов использования приложения для клиента представлена на рисунке 2.1.

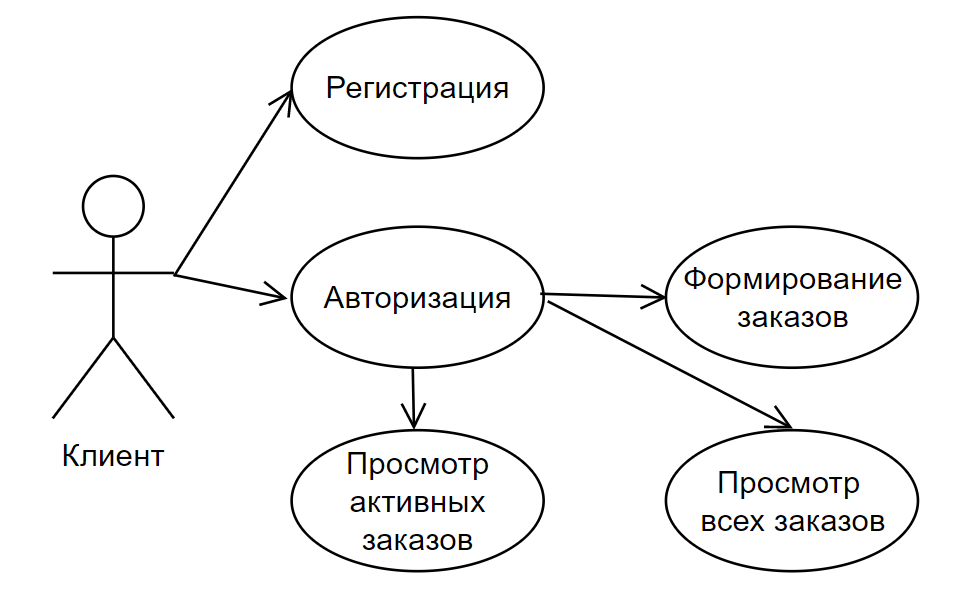


Рисунок 2.1 – Диаграмма использования приложения для клиента

Клиентское приложение – сразу после открытия предоставляет возможность авторизоваться либо же зарегистрироваться. В клиентском приложении предоставляется возможность работы с:

* формой для формирования заказа;
* списком активных заказов;
* списком всех обработанных заказов.

Диаграмма вариантов использования приложения для работника представлена на рисунке 2.2.

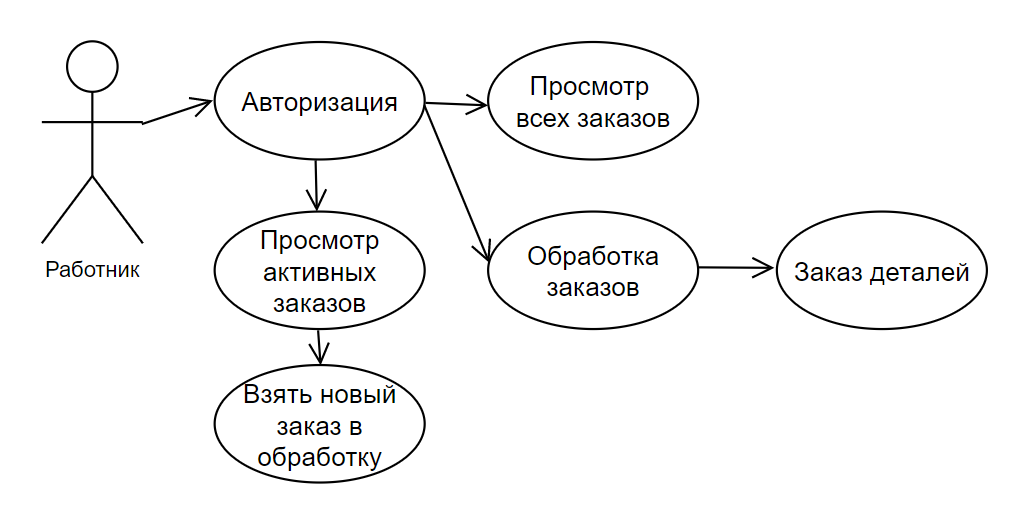


Рисунок 2.2 – Диаграмма использования приложения для работника

Приложение работника – сразу после открытия предоставляет возможность авторизоваться. В данном приложении предоставляется возможность работы с:

* списком активных заказов;
* добавлять заказ из списка активных заказов в список обработки;
* списком всех выполненных заказов.

Область работы администратора – SqlDeveloper или SqlPlus. Он ответственен за инициализацию, контроль и мониторинг информационной системы.

## **2.2 Проектирование модели базы данных**

Первым этапом выполнения курсовой работы будет создание логически взаимосвязанных таблиц. Чтобы составить визуальную взаимосвязанную структуры нашей базы данных, нам необходимо продумать, какая информация будет храниться в этих таблицах, после этого создать связи с помощью первичных и внешних ключей.

База данных состоит из 9 таблиц, перечисленных ниже (рисунок 2.3).

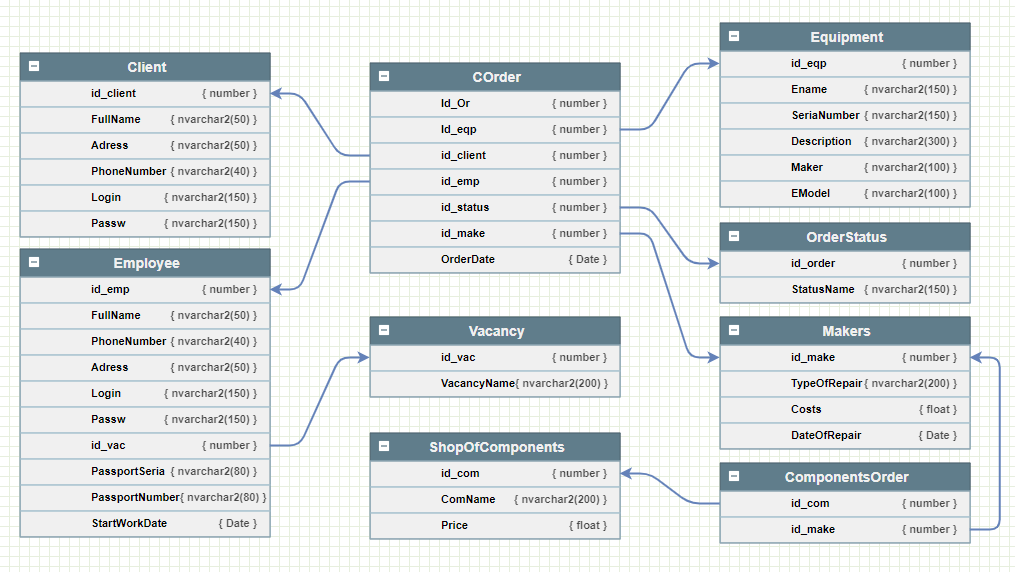


Рисунок 2.3 – Взаимосвязь таблиц данных

Таблица «Client» содержит информацию о клиентах. Описание ее полей приведено в таблице 2.1.

Таблица 2.1 – Структура таблицы «Client»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Client | number | Идентификатор клиента |
| FullName | nvarchar2 | ФИО |
| Adress | nvarchar2 | Адрес |
| PhoneNumber | nvarchar2 | Номер телефона |
| Login | nvarchar2 | Логин |
| Passw | nvarchar2 | Пароль |

Таблица «Employee» содержит информацию о сотрудниках. Описание ее полей приведено в таблице 2.2.

Таблица 2.2 – Структура таблицы «Employee»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Emp | number | Идентификатор сотрудника |
| FullName | nvarchar2 | ФИО |
| Id\_Vac | number | Идентификатор должности |
| PassportSeria | nvarchar2 | Серия паспорта |
| PassportNumber | nvarchar2 | Номер паспорта |

Продолжение таблицы 2.2

|  |  |  |
| --- | --- | --- |
| Adress | nvarchar2 | Адрес |
| PhoneNumber | nvarchar2 | Номер телефона |
| StartWorkDate | date | Дата начала работы |
| Login | nvarchar2 | Логин |
| Passw | nvarchar2 | Пароль |

Таблица «OrderStatus» содержит информацию о статусе заказа. Описание ее полей приведено в таблице 2.3.

Таблица 2.3 – Структура таблицы «OrderStatus»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Order | number | Идентификатор заказа |
| StatusName | nvarchar2 | Имя статуса |

Таблица «Equipment» содержит информацию о сломанном оборудовании. Описание ее полей приведено в таблице 2.4.

Таблица 2.4 – Структура таблицы «Equipment»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Eqp | number | Идентификатор оборудования |
| EName | nvarchar2 | Наименование оборудования |
| SeriaNumber | nvarchar2 | Серия оборудования |
| Description | nvarchar2 | Описание проблемы |
| Maker | nvarchar2 | Мастер |
| EModel | nvarchar2 | Модель |

Таблица «Makers» содержит информацию о проделанном ремонте. Описание ее полей приведено в таблице 2.5.

Таблица 2.5 – Структура таблицы «Makers»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Make | number | Идентификатор ремонта |
| TypeOfRepair | nvarchar2 | Тип ремонта |
| Costs | number | Стоимость |
| DateOfRepair | date | Дата исполнения |

Таблица «ComponentsOrder» содержит информацию об использованных комплектующих мастером. Описание ее полей приведено в таблице 2.6.

Таблица 2.6 – Структура таблицы «ComponentsOrder»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Com | number | Идентификатор компонента |
| Id\_Make | number | Идентификатор мастера |

Таблица «ShopOfComponents» содержит информацию о комплектующих. Описание ее полей приведено в таблице 2.7.

Таблица 2.7 – Структура таблицы «ShopOfComponents»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Com | number | Идентификатор детали |
| ComName | nvarchar2 | Имя детали |
| Price | number | Цена |

Таблица «Vacancy» содержит информацию о должностях. Описание ее полей приведено в таблице 2.8.

Таблица 2.8 – Структура таблицы «Vacancy»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Vac | number | Идентификатор должности |
| VacancyName | nvarchar2 | Имя должности |

Таблица «COrder» содержит основную информацию о заказах. Описание ее полей приведено в таблице 2.9.

Таблица 2.9 – Структура таблицы «COrder»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id\_Or | number | Идентификатор заказа |
| Id\_Eqp | number | Идентификатор оборудования |
| Id\_Client | number | Идентификатор клиента |
| Id\_Emp | number | Идентификатор работника |
| Id\_Status | number | Идентификатор статуса |
| Id\_Make | number | Идентификатор ремонта |
| OrderDate | date | Дата заказа |

Данный таблицы представляют логическую структуру базы данных.

# **3 Использование Oracle 12c**

## **3.1 Создание таблиц**

Для работы в базе данных мы должны создать таблицы и организовать между ними связи, которые будут представлять логическую структуру базы данных. В нашу базу будут включены такие таблицы:

* Client;
* Vacancy;
* Employee;
* OrderStatus;
* Equipment;
* Makers;
* ComponentsOrder;
* ShopOfComponents;
* COrder.

Создание этих таблиц проходит исполнение команд блока PL/SQL.

Все таблицы создаются в отдельном файле Tables.sql, для дальнейшей возможности запускать скрипт в целом файле в новой базе данных.

## **3.2 Создание ролей для разграничения доступа**

Для правильной работы десктоп приложения с базой данных необходимо разграничение ролей. Для выполнения данной задачи в СУБД Oracle предусмотрены встроенные функции, такие как пользователи, роли и привилегии.

В данном случае были созданы две роли им были выданы привилегии, также было создано два пользователя и назначены им роли (рисунок 3.1).

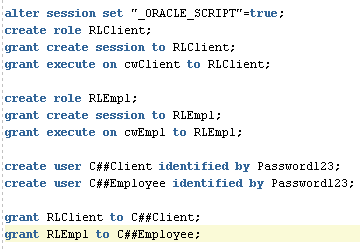


Рисунок 3.1 – Создание пользователей и выдача им привилегий

## **3.3 Создание пакетов и процедур для работы с базой данных**

После добавления ролей и создания всех необходимых таблиц приступаем к созданию процедур, позволяющих работать с данными.

Пакет Oracle PL/SQL – это объект схемы, который группирует логически связанные типы, элементы и подпрограммы. Пакеты обычно состоят из двух частей: спецификации и тела, хотя иногда тело не нужно.

Хранимая процедура (stored procedure) – это программа, которая выполняет некоторые действия с информацией в базе данных, и при этом сама хранится в базе данных.

Для логического разделения процедур были созданы три пакета:

* cwClient – пакет процедур для клиента;
* cwEmpl – пакет процедур для работника;
* cwAdmin – пакет процедур для администратора базы данных.

Заголовок пакета, содержащий процедуры для работы клиента представлен на рисунке 3.2. Тело пакета расположено в приложении А.

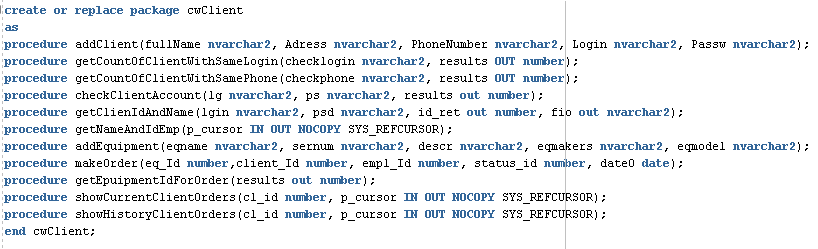


Рисунок 3.2 – Заголовок пакета cwClient

Данный пакет содержит процедуру addClient для регистрации клиентов, checkClientAccount отвечающую за авторизацию, makeOrder для оформления заказа, showCurrentClientOrders осуществляющая просмотр текущих заказов клиента, showHistoryClientOrders – просмотр выполненных заказов.

Заголовок пакета, содержащий процедуры для работы сотрудника представлен на рисунке 3.3. Тело пакета расположено в приложении Б.

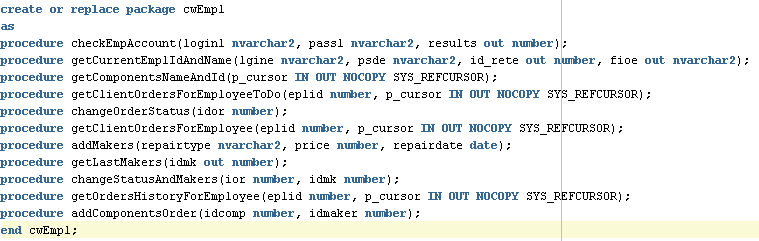


Рисунок 3.3 – Заголовок пакета cwEmpl

Данный пакет содержит процедуру checkEmpAccount отвечающая за авторизацию сотрудника в системе, getClientOrdersForEmployeeToDo для отображения новых заказов, getOrdersHistoryForEmployee выводящая список выполненных заказов.

Заголовок пакета, содержащий процедуры для работы администратора базы данных представлен на рисунке 3.4.

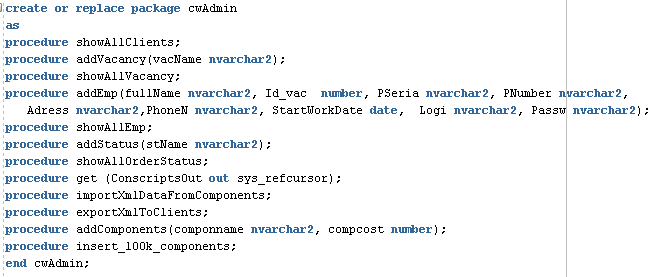


Рисунок 3.4 – Заголовок пакета cwAdmin

Тело пакета администратора расположено в приложении В.

Данный пакет содержит процедуры addVacancy, addEmp, addStatus, addComponents для заполнения таблиц Vacancy, Employee, OrderStatus, ShopOfComponents соответственно. Процедуры showAllClients, showAllVacancy, showAllEmp, showAllOrderStatus для вывода информации из соответствующих таблиц. Процедуры exportXmlToClients, importXmlDataFromComponents предназначены для экспорта/ импорта информации.

**3.4 Описание процедур экспорта и импорта**

Для комфортного администрирования базы данных в ней необходимо реализовать возможность экспорта и импорта данных в удобный для администратора формат. В нашем случае это формат xml;

Для доступа к папке из СУБД Oracle нужно должен быть создан directory.

Объект directory является логической ссылкой в базе данных на каталог файловой системы сервера, где установлена база данных Oracle. Владельцем всех объектов directory в базе данных является пользователь sys, даже если объект directory создан другим пользователем. Имена объектов directory уникальны внутри всей базы данных. (рисунок 3.5).



Рисунок 3.5 – Создание объекта directory

На рисунке 3.6 представлена процедура импорта данных из файла components\_import.xml в базу данных.

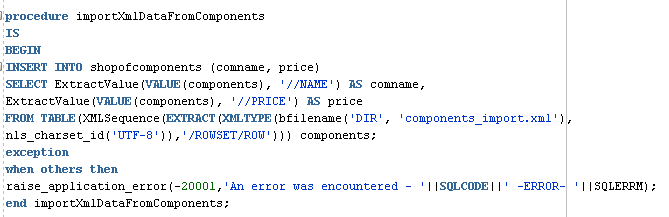


Рисунок 3.6 – Процедура импорта данных

На рисунке 3.7 представлена процедура экспорта данных из базы данных в файл client\_export.xml.

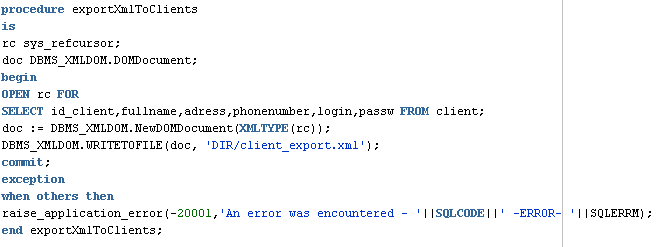


Рисунок 3.7 – Процедура экспорта данных

Данные процедуры расположены в пакете администратора cwAdmin.

## **3.5 Тестирование производительности базы данных**

Для проверки производительности базы данных необходимо заполнить ее большим количеством различных данных и узнать время выполнения одного запроса. Для этого с помощью процедуры insert\_100k\_components, которая представлена на рисунке 3.8, было добавлено 100000 строк.

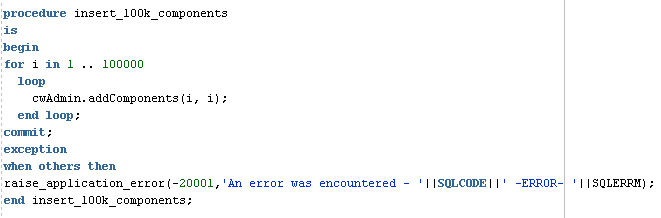


Рисунок 3.8 – Процедура заполнения таблицы

На рисунке 3.9 демонстрируется количество записей в заполненной процедурой таблице.

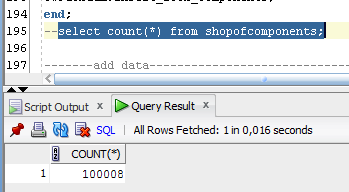


Рисунок 3.9 – Количество записей в таблице

Построим план запроса по всем элементам таблицы и посмотрим время его выполнения без индекса (рисунок 3.10).

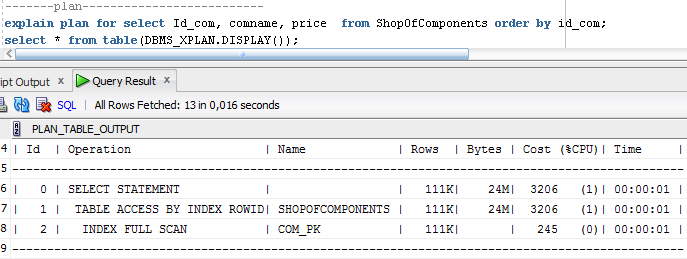


Рисунок 3.10 – План запроса без индекса

Для оптимизации работы с поисковыми запросами необходимо использовать индексы. Поэтому для этой таблицы был реализован индекс по наиболее используемой выборке (рисунок 3.11).



Рисунок 3.11 – Индекс для таблицы ShopOfComponents

Построим план запроса по всем элементам таблицы и посмотрим время его выполнения вместе с индексом (рисунок 3.12).

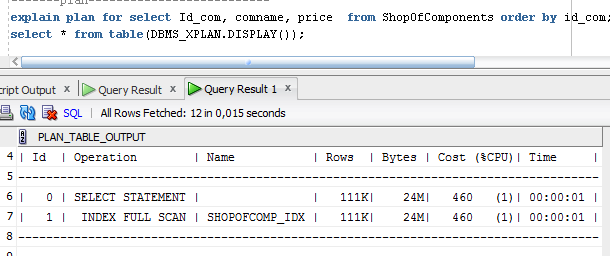


Рисунок 3.12 – План запроса с индексом

Исходя из приведенных выше результатов запросов, можно увидеть, что добавление индекса привело к уменьшению стоимости запроса примерно в 7 раз.

# **4 Описание технологий**

## **4.1 Основная технология резервного копирования и восстановления**

Традиционный пользовательский метод резервного копирования состоит в применении команд операционной системы для копирования необходимых файлов в другое место и/или на ленточное устройство. В случае применения утилиты RMAN резервное копирование файлов базы данных Oracle выполняться внутри базы данных посредством самого сервера баз данных. RMAN умеет делать резервные копии и копии образов файлов данных, управляющих файлов, архивных журналов повторного выполнения, файлов SPFILE и фрагментов резервных копий RMAN. Поэтому компания Oracle рекомендует применять для резервного копирования баз данных именно интерфейс RMAN.

Утилита RMAN упрощает процедуры резервного копирования, позволяя использовать мощные и в то же время легкие в написании сценарии резервного копирования и восстановления. Вдобавок RMAN обладает функциям наподобие обнаружения повреждений внутри блоков данных и способностью выполнять резервное копирование только изменившихся блоков в базе данных. Сценарии RMAN можно сохранять в базе данных и использовать прямо оттуда, что избавляет от необходимости писать сценарии уровня операционной системы. RMAN автоматически обеспечивает создание резервных копий всех файлов базы данных, что исключает вероятность допущения человеческой ошибки, которая присутствует в методах резервного копирования, основанных на применении команд операционной системы.

Для автоматизации процесса резервного копирования и восстановления были дополнительно разработаны файлы с расширением .rman и вызывающие их batch-скрипты.

В файле copy.rman (рисунок 4.1) содержатся команды, которые производят резервное копирование управляющего файла и табличных пространств базы данных.

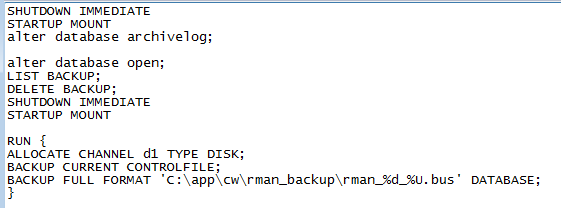


Рисунок 4.1 – Файл copy.rman

Команда List Backup предназначена для вывода списка всех имеющихся копий базы данных. Чтобы удалить старые копии используется команда delete backup. Сами функции копирования размещены в блоке run. Первым делом происходит выделение канала (соединение с сервером), затем происходит резервное копирование текущего управляющего файла, после происходит копирование файлов табличных пространств в указанном формате.

После можно удалить файлы и восстановить базу данных.

Для восстановления базы данных использую файл restore.rman (рисунок 4.2).

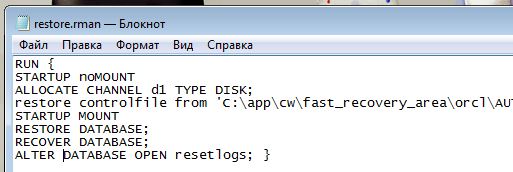


Рисунок 4.2 – Файл restore.rman

При помощи команды restore происходит восстановление файлов, командой recover восстанавливаем базу данных.

## **4.2 Технология шифрования**

# **5 Руководство пользователя**

## **5.1 Область клиента**

Для начала работы в приложении пользователю требуется зарегистрироваться. Для этого ему требуется корректно заполнить все поля в форме регистрации (рисунок 5.1).

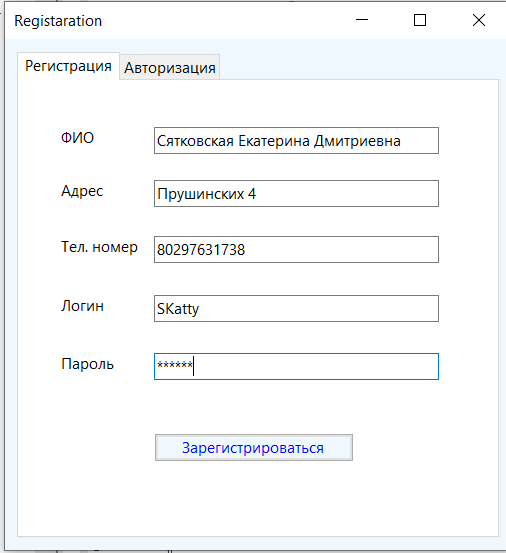


Рисунок 5.1 – Меню регистрации

Далее ему требуется авторизоваться. Для этого ему требуется прейти в меню авторизация и ввести свои логин и пароль (рисунок 5.2).

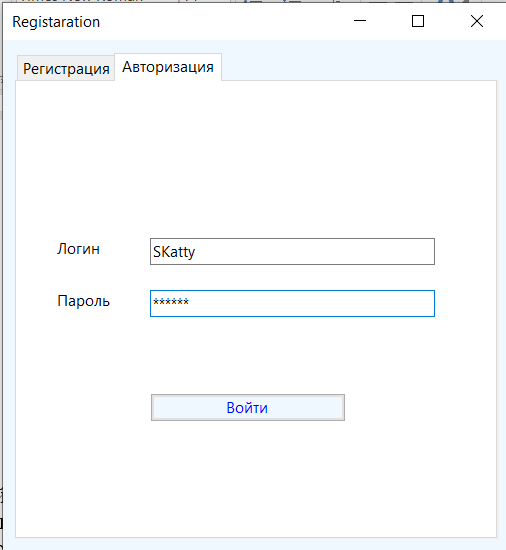


Рисунок 5.2 – Меню авторизации

После авторизации его встретит окно, в котором он сможет сформировать заказ. Для этого ему потребуется заполнить все поля и нажать на кнопку оформить заказ (рисунок 5.3).

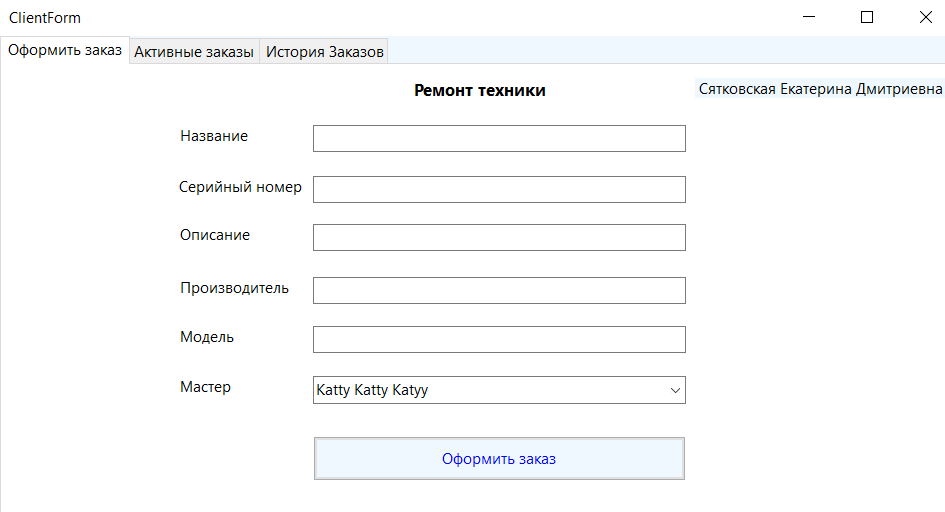


Рисунок 5.3 – Меню оформления заказов

После этого он сможет увидеть свой заказ в меню активных заказов (рисунок 5.4).

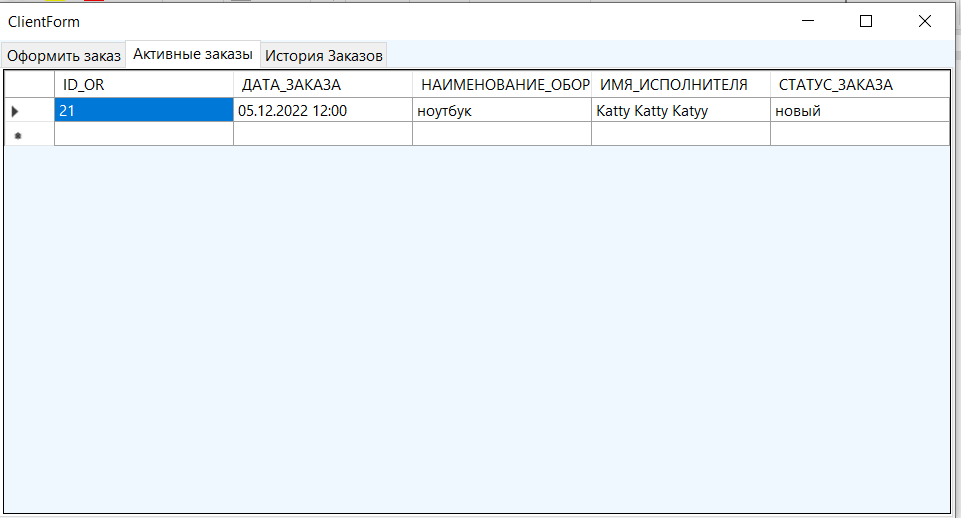


Рисунок 5.4 – Меню просмотра активных заказов

После того как работник обработает заказ клиента, клиент сможет увидеть свой заказ в истории заказов (рисунок 5.5).

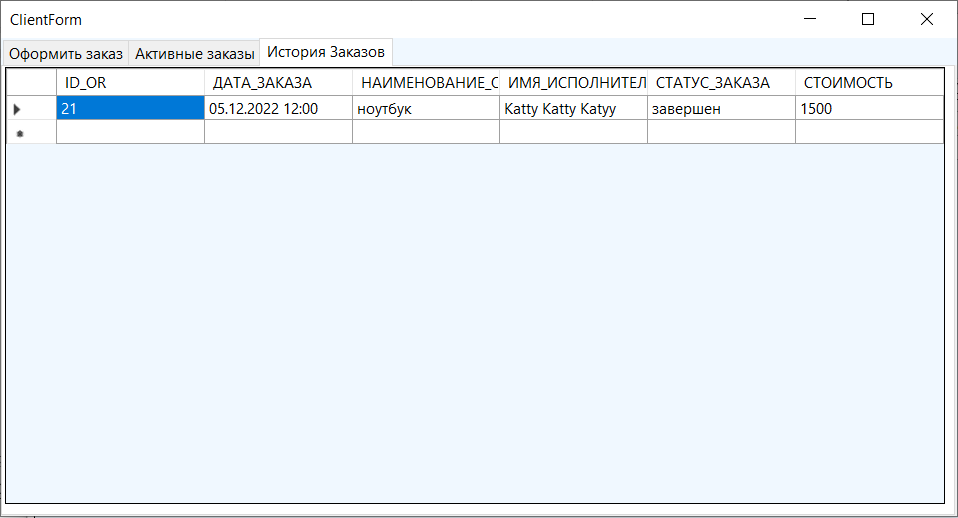


Рисунок 5.5 – Меню просмотра истории заказов

Окно истории заказов схоже с окном активных заказов. Различие заключается в статусе заказа и некоторой информацией, доступной лишь в истории заказов.

## **5.2 Область сотрудника**

Сотрудник имеет похожее окно авторизации, как и у клиента (рисунок 5.6).

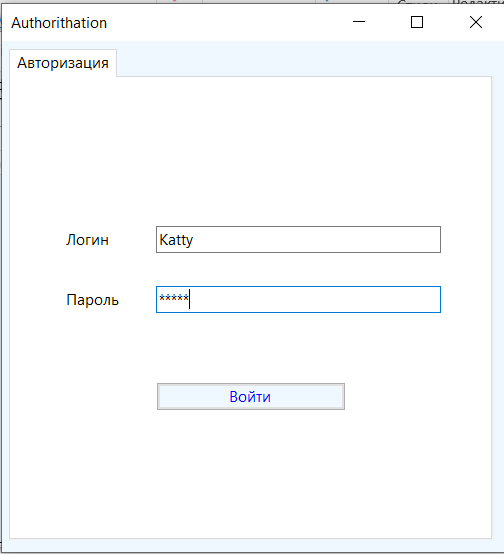


Рисунок 5.6 – Меню авторизации

После авторизации сотруднику будет доступно окно с просмотром активных заказов. Он может выбрать любой на обработку. Для этого ему потребуется нажать на один заказ и таблицы и добавить его в обработку (рисунок 5.7).

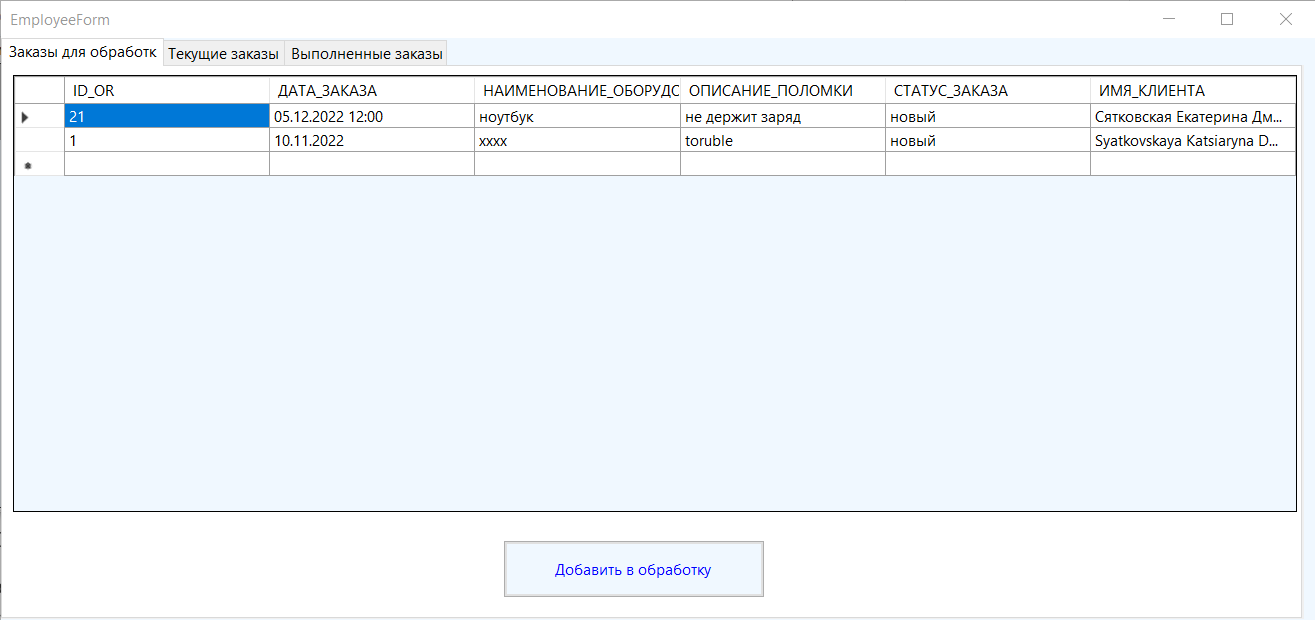


Рисунок 5.7 – Меню просмотра заказов для обработки

После того как заказ попал в обработку работник должен его обработать. Для этого ему требуется перейти в меню текущих заказов, выбрать заказ, заполнить необходимые поля, добавить детали, если они необходимы и выполнить заказ (рисунок 5.8).

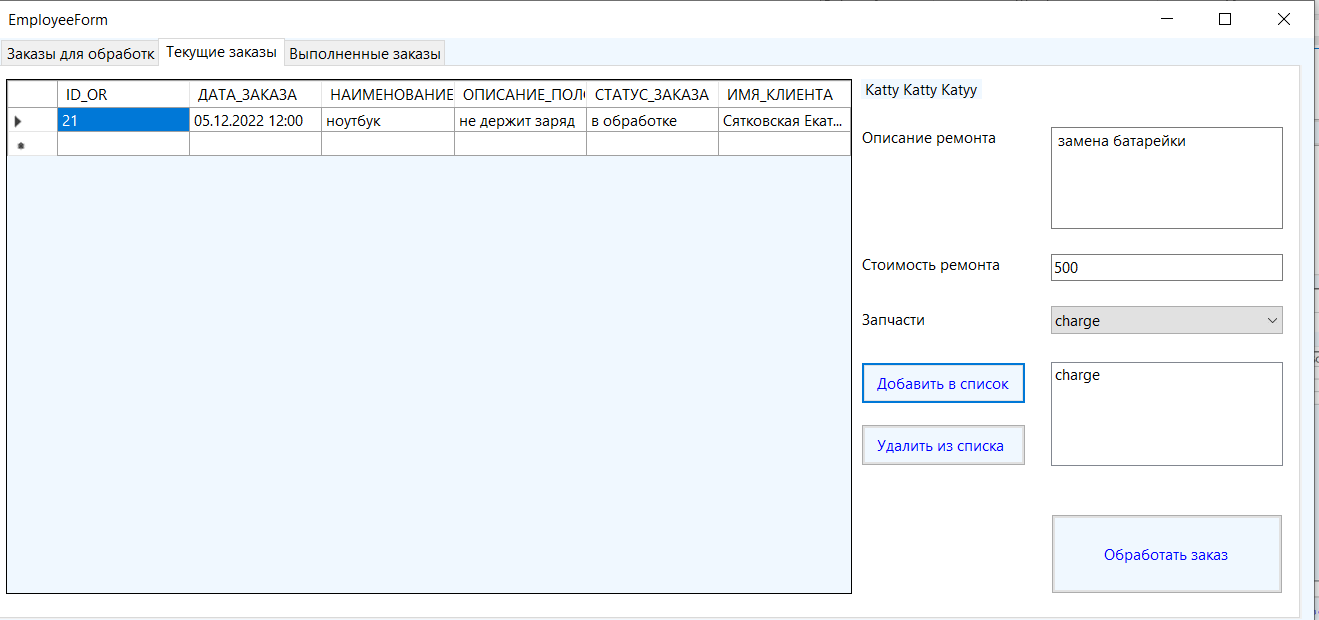


Рисунок 5.8 – Меню просмотра текущих заказов

После этого заказ поменяет статус и перейдёт в историю заказов клиента и работника (рисунок 5.9).

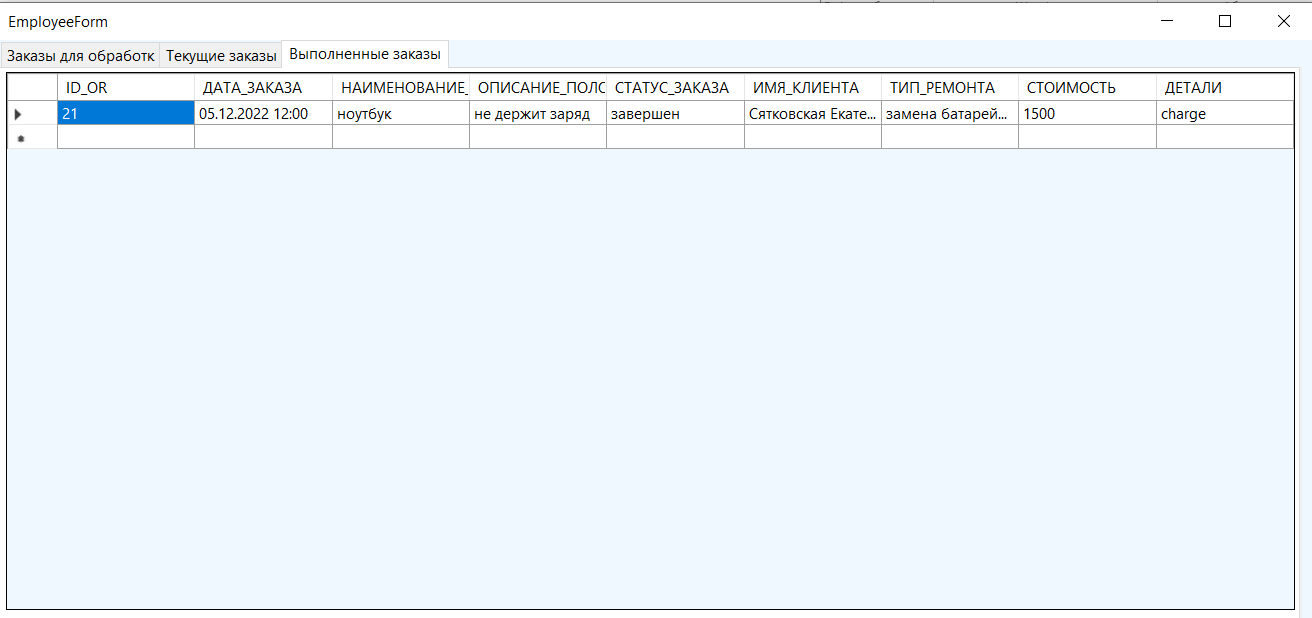


Рисунок 5.9 – Меню просмотра истории заказов

Также статус заказа меняется и при добавлении заказа в обработку.

## **5.3 Тестирование клиентской области**

Приложение проектировалось и разрабатывалось в направлении минимизации человеческих ошибок. В приложении присутствует минимальное количество мест, где клиент может вызвать ошибку программы. Все варианты использования приложения продуманы и не допускают возможности сделать некорректное действие.

При регистрации пользователя нельзя оставлять поля незаполненными. Если попробовать зарегистрироваться, не заполнив поля, появятся предупреждения, то есть у пользователя не получится зарегистрироваться пока он не заполнит все поля (рисунок 5.10).

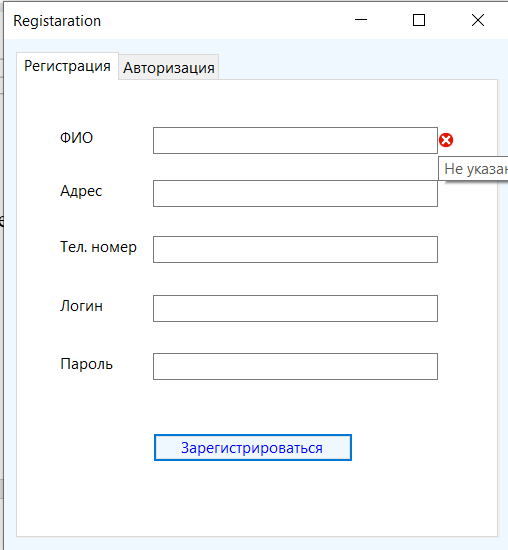


Рисунок 5.10 – Меню регистрации

При авторизации также происходит проверка полей на пустоту. При отсутствии зарегистрированного аккаунта у пользователя не получится авторизоваться (рисунок 5.11).

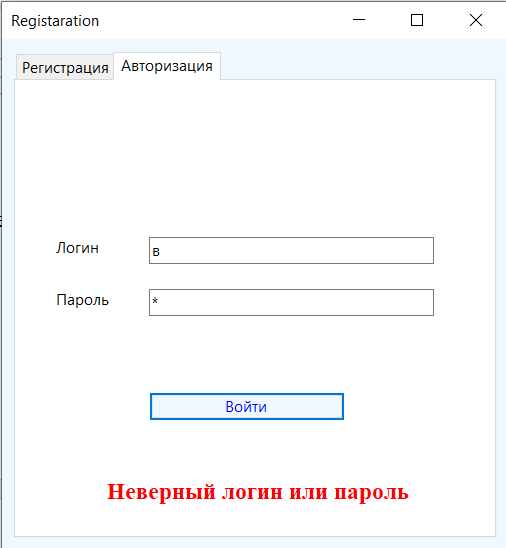


Рисунок 5.11 – Меню авторизации

После авторизации клиент может оформить заказ. При этом у него это получится в том случае, если он заполнит все поля (рисунок 5.12). В случае попытки осуществить заказ с незаполненными полями у него появятся предупреждения.

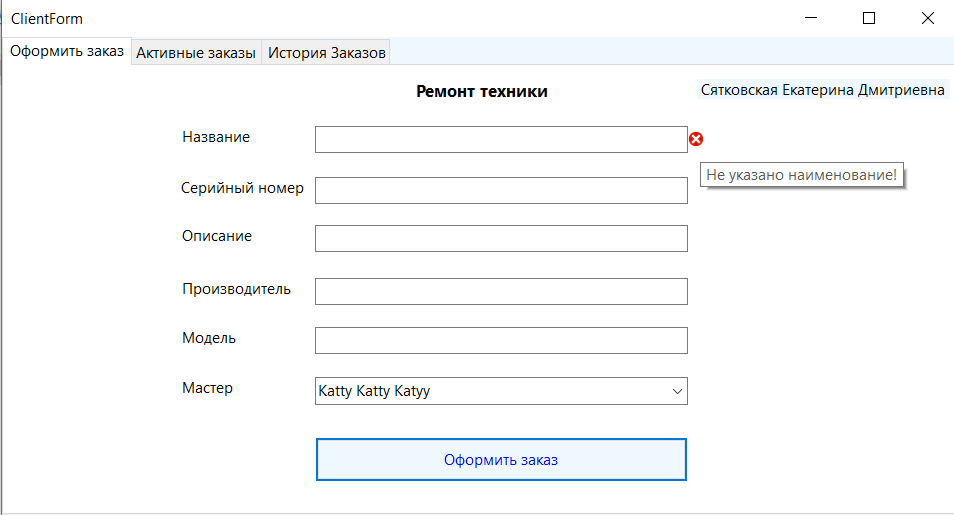


Рисунок 5.12 – Меню оформления заказов

В клиентском приложении это были все места, где он мог бы допустить ошибку.

## **5.4 Тестирование области работника**

У работника значительно больше мест, где можно было бы допустить ошибки. При авторизации также происходит проверка полей на пустоту, а также проверка на наличие аккаунта в базе данных (рисунок 5.13).

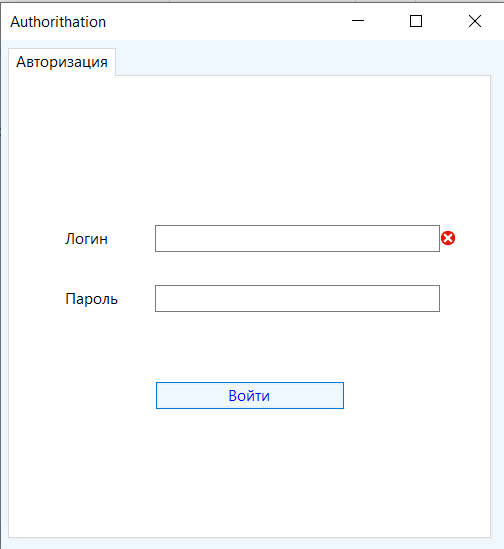


Рисунок 5.13 – Меню авторизации работника

После авторизации сотрудник попадает в окно с выбором заказа для обработки. Для этого ему требуется нажать на требуемый заказ. Если заказов не будет, и работник нажмёт на кнопку обработки, то он получит предупреждение, что заказов нет (рисунок 5.14).

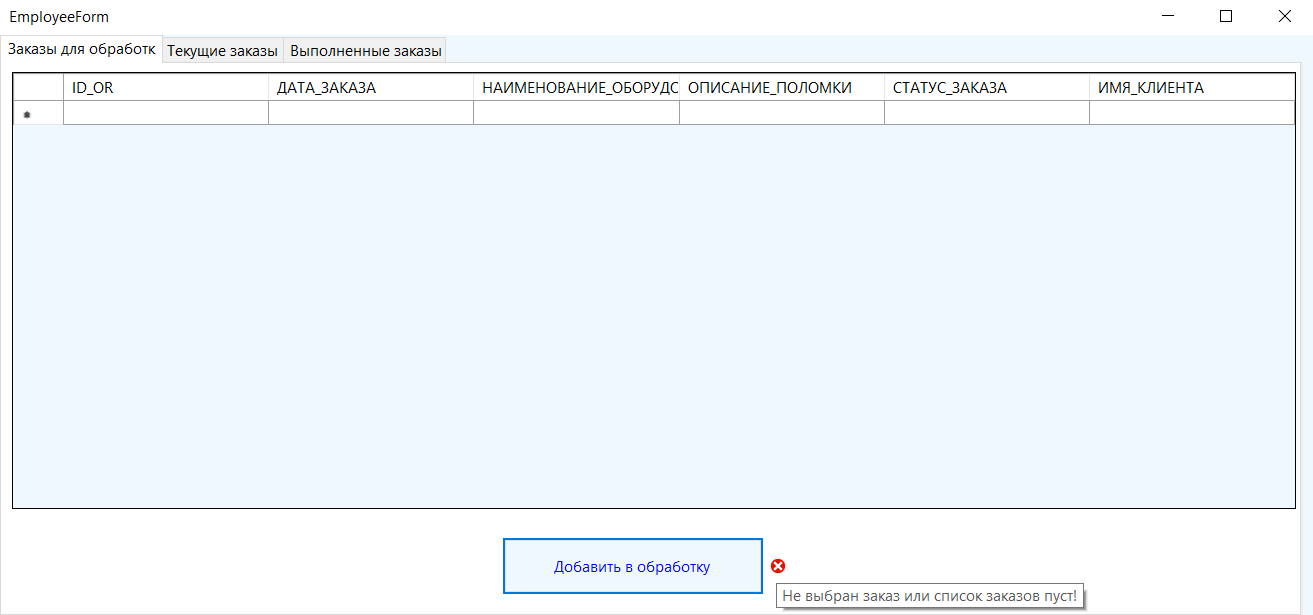


Рисунок 5.14 – Меню обработки заказов

В текущих заказах также стоит проверка на пустые поля и наличие заказов (рисунок 5.15).

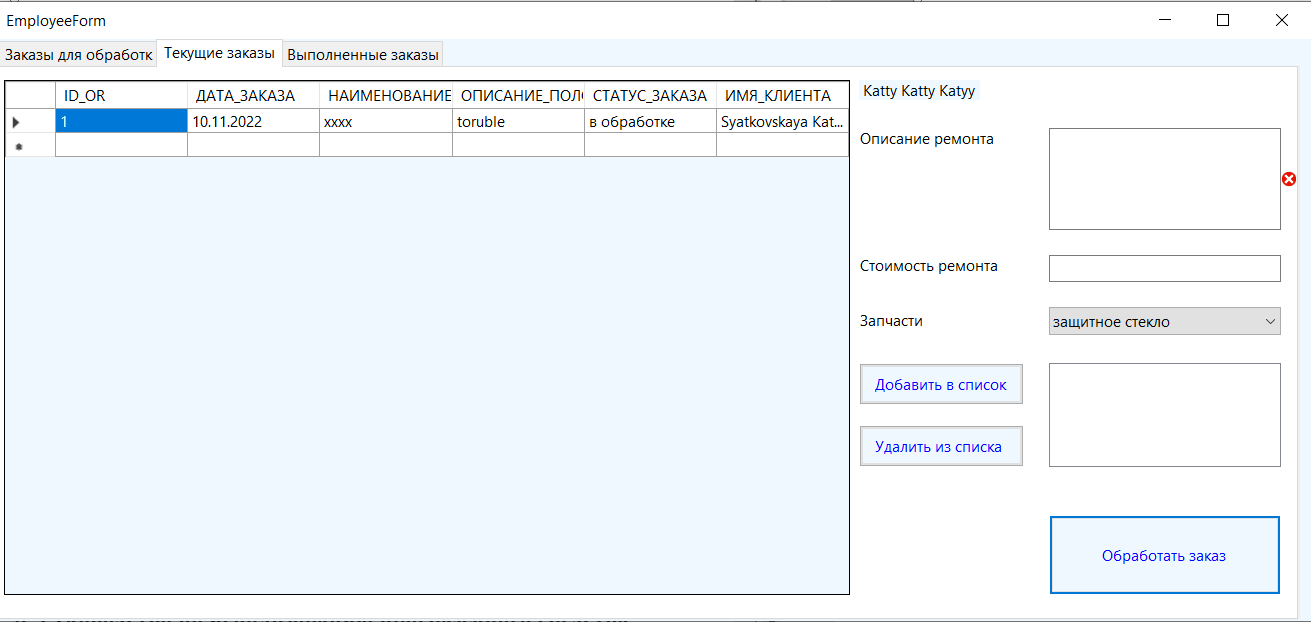


Рисунок 5.15 – Меню текущих заказов

Это были единственные места, где работник мог совершить ошибку.

# **Заключение**

Данный курсовой проект был направлен на развитие навыков администрирования базы данных, а также на улучшение понимания взаимодействия программного средства с БД Oracle 12C на удалённом сервере.

В ходе выполнения работы были достигнуты следующие результаты:

* исследованы обучающие приложения-аналоги;
* сформулирована модель данных и приведено её графическое представление;
* подготовлена база данных Oracle с использованием технологии резервного копирования и восстановления;
* подготовлены процедуры для осуществления импорта и экспорта данных с форматом xml;
* были созданы пользователи с различными привилегиями для работы с базой данных;
* были созданы скрипты для быстрого развертывания начальных таблиц компонентов и начальных таблиц пользователей базы данных с дальнейшим их заполнением;
* разработано приложение для взаимодействия с базой данных.

Приложение прошло тестирование при использовании в БД большого количество данных.

Реализация базы данных будет хорошим решением для любого сервиса данного направления, поскольку позволяет охватить необходимый минимум требований для администрирования и использования.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объёме.

# **Cписок источников**

1. Язык C# и .NET Framework [Электронный ресурс] – http://professorweb.ru/my/csharp/charp\_theory/level1/infonet.php.
2. Руководство по ADO.NET и работе с базами данных [Электронный ресурс] – https://metanit.com/sharp/adonet/.
3. Документация Oracle [Электронный ресурс] / Foundation, Inc. https://docs.oracle.com/cd/B28359\_01/server.111/b31222/toc.htm
4. Официальный сайт Oracle [Электронный ресурс] / Foundation, Inc. https://www.oracle.com/database/database-vault/index.html
5. Продукты Oracle [Электронный ресурс] [Электронный ресурс] / Foundation, Inc. http://www.interface.ru/home.asp?artId=24678
6. Документы Oracle [Электронный ресурс] / Foundation, Inc. http://www.oracle.com/technetwork/database/security/database-vault-ds-12c-1898877.pdf

# **Приложение А**

# **Приложение Б**

# **Приложение В**