Министерство ОБРАЗОВАНИЯ Республики Беларусь

Инв. № подл.

*37*

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Учреждение образования «бобруйский государственный

аграрно-экономический колледж»

**Разработка и отладка БД и СУБД клиент-серверного типа «Учет материалов на складе»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине   
«Базы данных и системы управления базами данных»

**КП 21П.013102.081 ПЗ**

**Руководитель:**

Преподаватель Т.О. Парфененко

**Разработчик**:

учащийся группы 21-П       Д. В. Наврозов

БОБРУЙСК 2021

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc87961668)

[1 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc87961669)

[1.1 Техническое задание 6](#_Toc87961670)

[1.2 Анализ предметной области 11](#_Toc87961671)

[1.3 Функциональная модель приложения 11](#_Toc87961672)

[1.4 Логическая модель данных 13](#_Toc87961673)

[1.5 Физическая модель данных 15](#_Toc87961674)

[1.5.1 Описание структуры таблиц и связей между ними 16](#_Toc87961675)

[1.5.2 Описание индексов и ограничений 17](#_Toc87961676)

[2 ОПИСАНИЕ ПРОГРАММЫ 19](#_Toc87961677)

[2.1 Выбор и обоснование средств программирования для решения поставленной задачи, их краткая характеристика 19](#_Toc87961678)

[2.2 Описание интерфейса, форм ввода и вывода данных 20](#_Toc87961679)

[2.3 Описание запросов и представлений данных 27](#_Toc87961680)

[2.4 Описание триггеров и хранимых процедур 29](#_Toc87961681)

[3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 30](#_Toc87961682)

[4 СПРАВОЧНАЯ СИСТЕМА 38](#_Toc87961683)

[5 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ 39](#_Toc87961684)

[ЗАКЛЮЧЕНИЕ 41](#_Toc87961685)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 43](#_Toc87961686)

[ПРИЛОЖЕНИЕ А. Листинг программы 44](#_Toc87961687)

[ПРИЛОЖНИЕ Б. Структурная схема 70](#_Toc87961688)

[ПРИЛОЖНИЕ В. Диаграмма компонентов 71](#_Toc87961689)

[ПРИЛОЖЕНИЕ Г. Диаграмма вариантов использования 72](#_Toc87961690)

[ПРИЛОЖЕНИЕ Д. Диаграмма «сущность-связь» 73](#_Toc87961691)

ВВЕДЕНИЕ

Сферой применения высокоуровневых методов информатики и программирования является решение комплекса задач, связанных с приходом и расходом товара, а также с необходимостью его учета. Этот комплекс решается в несколько этапов, каждому из которых соответствуют собственные задачи. Результатом решения задач высокоуровневых методов информатики и программирования является обеспечение автоматизированного учета товара на складе.

Основная часть материально-производственных запасов используется в качестве предметов труда в производственном процессе. Они целиком потребляются в каждом производственном цикле и полностью переносят свою стоимость на стоимость производственной продукции.

Материалы - вид запасов. К материалам относятся сырье, основные и вспомогательные материалы, покупные полуфабрикаты и комплектующие изделия, топливо, тара, запасные части, строительные и прочие материалы.

Учет производственных запасов осуществляется на основании следующих первичных документов: товарно-транспортной накладной, требований-накладной, карточки складского учета материалов, ведомости учета остатков материалов на складе [14].

Исходя из этих направлений, становится актуальным создание и использование конкурентно способной базы данных и информационно-поисковой системы «Учет материалов на складе».

Объектом курсового проектирования является программное обеспечение информационных технологий в сфере складского учета.

Предметом курсового проекта является автоматизация процесса учета материалов на складе.

Цель курсового проекта – разработка и отладка базы данных и СУБД клиент-серверного типа «Учет материалов на складе».

Для достижения поставленной цели необходимо решить следующие задачи:

выполнить анализ предметной области, на основании которого будет подобран материал;

разработать структуру и определить принципы управления программой;

создать, отладить и протестировать программное средство;

разработать сопроводительную документацию.

Создания системы обусловлена тем, что с её помощью можно облегчить труд введения различной документации в сфере складского учета. Ведь всё это будет сохраняться в удобном, для пользователя, электронном виде.  Также это позволит ускорить работу всей системы склада, за счёт ускоренного поиска необходимых данных, за счёт простого ввода новых данных и формирования необходимых документов. И самое главное — это удобный способ хранения информации.

В качестве языка программирования выбран язык C#. Данный язык программирование, имеет различные средства контроля и достаточно прост в изучении. Язык отражает наиболее важные и фундаментальные концепции (идеи) алгоритмов в очевидной и легко воспринимаемой форме, что предоставляет средства, помогающие проектировать программу; СУБД будет спроектирована на языке MySQL [7].

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Техническое задание

**1.1.1 Общие положения**

Тема курсовой работы – создание и отладка информационно-поисковой системы клиент-серверного типа «Учет материалов на складе».

Программный продукт предназначен для учета и контроля движения материалов на складе.

Программное средство разрабатывается в рамках курсового проекта по дисциплине «Базы данных».

Основанием для разработки служит задание на курсовое проектирование.

Разработчик – Наврозов Даниил Вячеславович

Плановые сроки начала работы – 10.09.2021 г.

**1.1.2. Назначение и цели создания системы**

«Учет материалов на складе» – информационно-поисковой система клиент-серверного типа, предназначенная для:

оптимизации рабочих процессов;

повышение скорости работы работника;

повышение качества работы;

предоставления камфорных условий работы со складом.

Основными целями внедрения системы является:

создание функционально полного механизма подготовки, согласования и хранения различных данных;

создание единого механизма по учету материалов.

**1.1.3 Требования к системе**

Минимальная конфигурация:

тип процессора: Pentium и выше; MySQL WorkbenchMySQL WorkbenchMySQL WorkbenchMySQL WorkbenchMySQL WorkbenchMySQL WorkbenchMySQL WorkbenchMySQL WorkbenchMySQL WorkbenchMySQL Workbench

объем оперативного запоминающего устройства: 32 Мб и более;

объем свободного места на жестком: 2 Гб;

Рекомендуемая конфигурация:

тип процессора: Pentium II 400;

объем оперативного запоминающего устройства: 128 Мб;

объем свободного места на жестком диске: 2 Гб;

Требования к программной совместимости.

Программа должна работать под управлением семейства опе­рационных систем Windows (Windows XP/7/8/10/11).

**1.1.5 Состав и содержание работ по созданию системы**

Осуществление всего комплекса работ по созданию должно осуществляться в несколько очередей. Спецификация работ по созданию первой очереди «Учет материалов на складе» в объеме требований настоящего ТЗ (см. таблицу 1.1).

Таблица 1.1 – Состав работ по созданию системы

|  |  |  |
| --- | --- | --- |
| **Стадия работ** | **Выполняемые работы** | **Итого выполнения работы** |
| **1** | **2** | **3** |
| Формирование требований | Обследование объектов автоматизации.  Разработка частного ТЗ на создание подсистемы. | Отчет о результатах обследования.  Утверждение заказчиком ЧТЗ на создание подсистемы. |
| Проектирование | Разработка технического проекта.  Разработка прототипа подсистемы.  Разработка проектов организационно-распорядительной, программной и эксплуатационной документации на подсистему | Технический проект на подсистему.  Спецификации программно-аппаратных средств подсистемы. |
| Поставка программно-технических средств для опытной эксплуатации | Поставка программно-технических средств (лицензионного ПО) для опытной эксплуатации на объектах автоматизации, входящих в состав опытной зоны | Акты |
| Разработка программных средств | Разработка, отладка и тестирование программных средств подсистемы | Программные средства на машиночитаемых носителях.  Комплект проектов организационно-распорядительной, программной и эксплуатационной документации на подсистему |
| Приемка работ | Проведение предварительных испытаний | Протоколы испытаний  Акт готовности подсистемы к развертыванию в опытной зоне |

**1.1.6 Порядок контроля и приёмки**

Испытания Подсистемы должны проводиться в соответствии с требованиями ГОСТ 34.603-92 "Информационная технология. Виды испытаний автоматизированных систем". При реализации Подсистемы в рамках настоящего ТЗ устанавливаются предварительные испытания на стенде Исполнителя по созданию Подсистемы.

Испытания Подсистемы должны осуществляться в соответствии с документом "Программа и методика испытаний", который должен устанавливать необходимый и достаточный объем испытаний, обеспечивающий требуемый уровень достоверности получаемых результатов. Программа и методика испытаний утверждается Заказчиком.

Предварительные испытания заканчиваются подписанием приемочной комиссией протокола испытания с указанием в нем перечня необходимых доработок программного обеспечения, конструкторской, программной и эксплуатационной документации и сроков их выполнения.

После устранения замечаний, осуществляются повторные предварительные испытания Подсистемы. На повторные предварительные испытания Исполнителем предъявляются доработанные по результатам ранее выполненных испытаний материалы. Испытания завершаются оформлением Акта готовности Подсистемы к развертыванию в опытной зоне.

Отдельные пункты ТЗ могут изменяться и уточняться по согласованию сторон.

**1.1.7 Работы по вводу системы в действие**

Для подготовки АИС к вводу в эксплуатацию необходимо:

назначить должностное лицо в организации заказчика, ответственное за приемку системы;

установить комплекс технических средств, удовлетворяющие требованиям соответствующего ТЗ;

совместно с исполнителем выполнить инсталляцию системного ПО в соответствии с руководством администратора;

провести ввод данных справочной информации и настройку системы в соответствии с руководством по установке ПО;

составить совместно с исполнителем документ «Программа испытаний»;

провести испытания в соответствии с документом «Программа испытаний».

При удовлетворительном результате испытаний – подписать акт технической готовности системы к опытной эксплуатации. При наличии замечаний выпустить документ «Перечень предложений и замечаний для доработки системы».

**1.1.8 Документирование**

Документы должны быть представлены на бумажном виде (оригинал) и на магнитном носителе (копия). Исходные тексты программ - только на магнитном носителе (оригинал). Возможно предоставление комплекта документации и текстов программ на компакт-дисках.

Все документы должны быть оформлены на русском языке. Состав документов на общее программное обеспечение, поставляемое в составе ПО «Учет материалов на складе», должен соответствовать требованиям заказчика.

**1.1.9 Источники разработки**

Настоящее ТЗ разработано на основании следующих стандартов и нормативных документов:

ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;

ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение документов при создании автоматизированных систем;

РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.

ГОСТ 21.404-85 Автоматизация технологических процессов.

ГОСТ 24.701-86 ЕСС АСУ. Надёжность автоматизированных систем управления.

1.2 Анализ предметной области

В настоящее время складской учет является неотъемлемой частью экономики любого предприятия, производящего продукцию. Альтернативным способом учета материалов на складе является учет в электроном виде для упрощения и ускорение работы всего складского учета.

База данных – это совокупность сведений о конкретных объектах реального мира в какой-то предметной области или разделе предметной области.

В данном определении база данных представлена как совокупность данных, а не сведений, потому что понятия «информация» и «данные» в нашем случае различимы. Информация, зафиксированная в определенной форме, пригодной для последующей обработки, хранения и передачи, представляет собой данные.

Довольно часто под базой данных понимают множество данных, которые управляются специальной программной системой. Такие программные системы называются системами управления базами данных СУБД.

В неё можно вносить информацию: о материалах, клиентах, поставщиках продукции, поставках материалов, отпуске материалов клиентам, сотрудниках работающих с клиентами и поставщиками. А также можно добавлять и удалять записи.

1.3 Функциональная модель приложения

Функциональная модель – это модель, которая показывает, какие функции у проектируемой модели и как они взаимодействуют между собой. Данная модель создавалась в программе BPwin (рисунок 1.3.1).

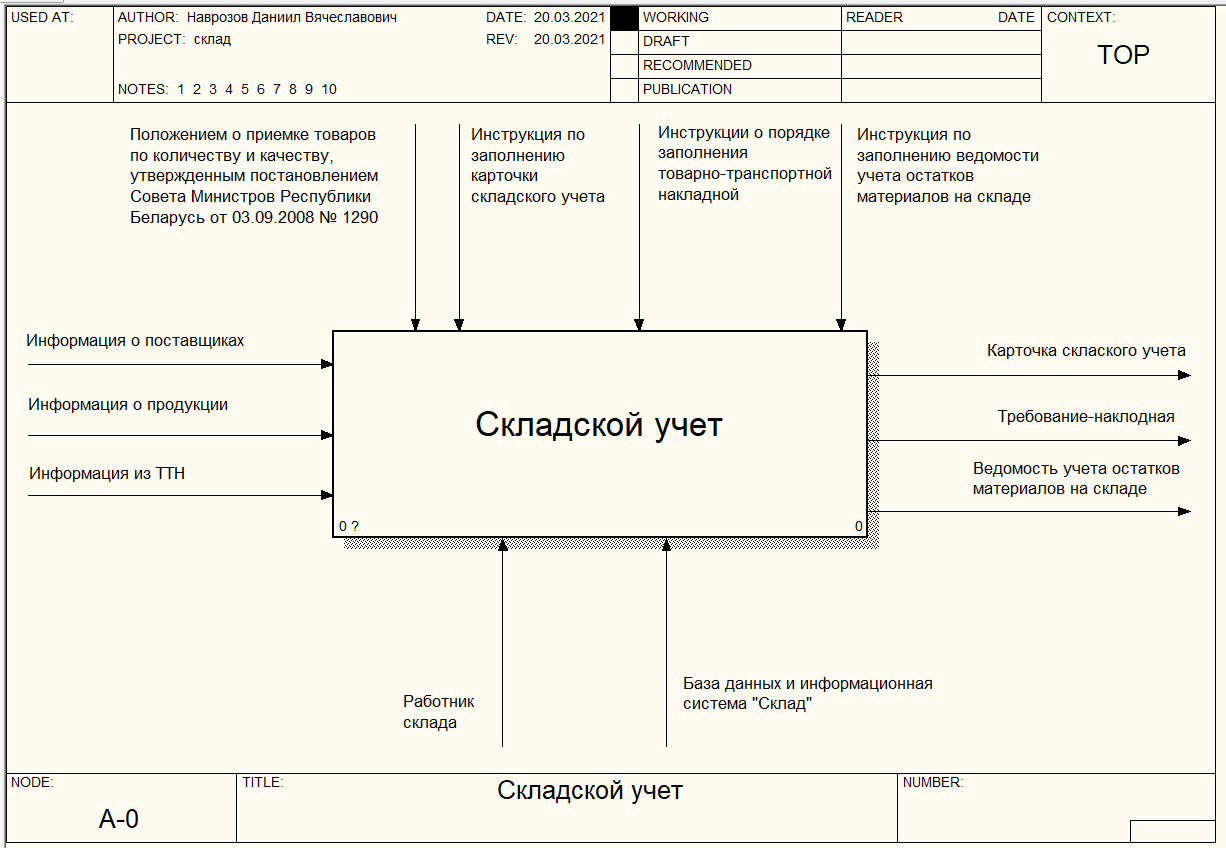


Рисунок 1.3.1 – Контекстная диаграмма

На диаграмме декомпозиции видно, что программу можно условно разделить на 4 блока: ввод или корректировка данных справки, составление требований-накладной, заполнение карточки складского учета, составление ведомости учета остатков материалов (рисунок 1.3.2).

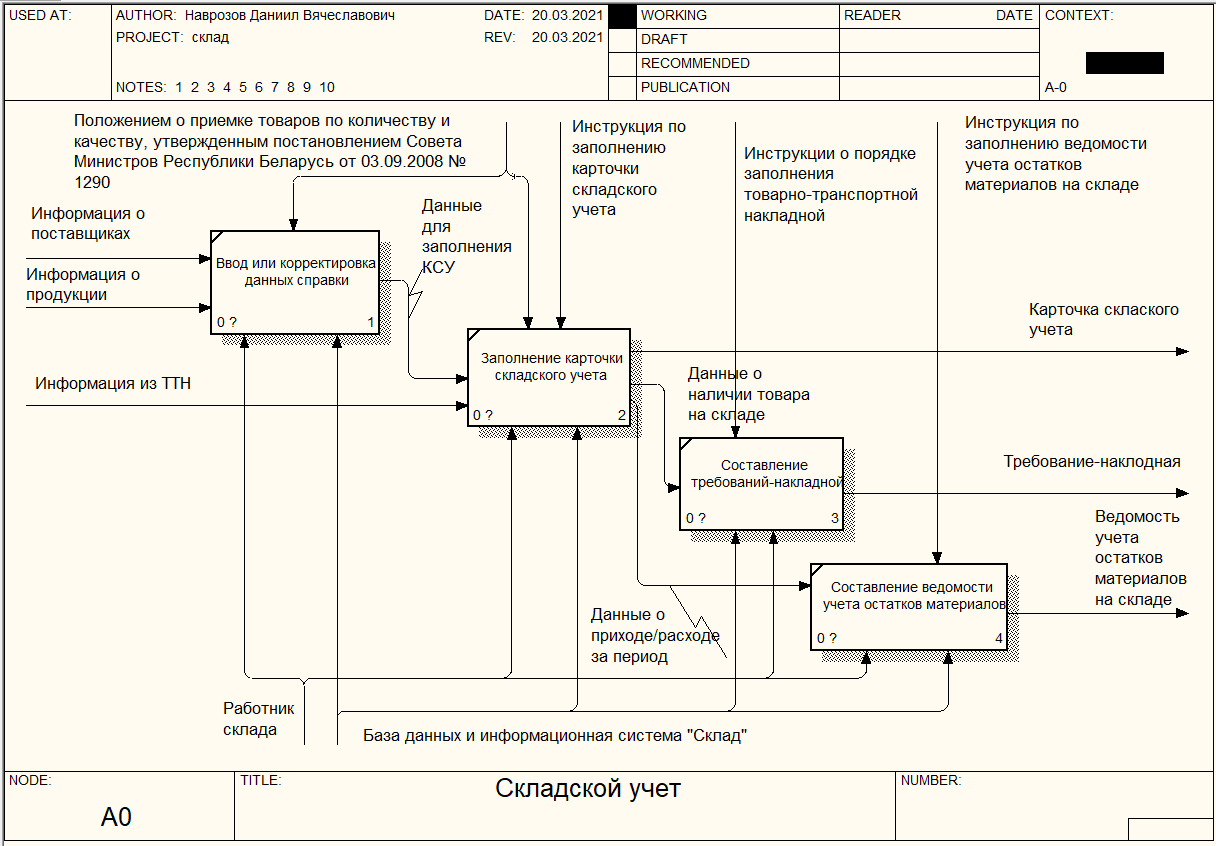


Рисунок 1.3.2 – Функциональная модель

1.4 Логическая модель данных

Логическая модель – графическое представление структуры базы данных с учетом принимаемой модели данных (иерархической, сетевой, реляционной и т.д.), независимое от конечной реализации базы данных и аппаратной платформы (рисунок 1.4.1).

Сущности «client\_help» и «country» связаны связью «один ко многим», т.к. одна страна, может быть, у нескольких клиентов, а клиент одной страны.

Сущности «client\_help» и «ntn\_one» связаны связью «один ко многим», т.к. один клиент может быть в нескольких требование-накладных, а требование-накладная с одним клиентом.

Сущности «country» и «provider\_help» связаны связью «один ко многим», т.к. одна страна может быть у нескольких поставщиков, а у поставщика только одна страна.

Сущности «ntn\_one» и «ntn\_two» связаны связью «один ко многим», т.к. одна шапка требование-накладной может быть у нескольких записей требование-накладной, а у записи только одна шапка.

Сущности «ntn\_two» и «produkt\_help» связаны связью «один ко многим», т.к. один продукт имеет несколько записей в требование-накладной, а требование-накладная только один продукт.

Сущности «ntn\_two» и «worker\_help» связаны связью «один ко многим», т.к. один работник имеет несколько записей в требование-накладной, а требование-накладная только одного сотрудника.

Сущности «ntn\_two» и «ksu\_one» связаны связью «один ко многим», т.к. одна карточка складского учета имеет несколько записей в требование-накладной, а требование-накладная только одну каточку складского учета.

Сущности «ttn» и «ksu\_one» связаны связью «один ко многим», т.к. одна карточка складского учета имеет несколько записей в товарно-транспортной накладной, а товарно-транспортная накладная только одну каточку складского учета.

Сущности «ttn» и «worker\_help» связаны связью «один ко многим», т.к. один работник имеет несколько записей в товарно-транспортной накладной, а товарно-транспортная накладная только одного сотрудника.

Сущности «ttn» и «produkt\_help» связаны связью «один ко многим», т.к. один продукт имеет несколько записей в товарно-транспортной накладной, а товарно-транспортная накладная только один продукт.

Сущности «ttn» и «provider\_help» связаны связью «один ко многим», т.к. один поставщик имеет несколько записей в товарно-транспортной накладной, а товарно-транспортная накладная только одного поставщика.

Сущности «produkt\_help» и «unit\_produkt» связаны связью «один ко многим», т.к. одна величина имеет несколько записей продуктах, а продукт может иметь только одну величину.

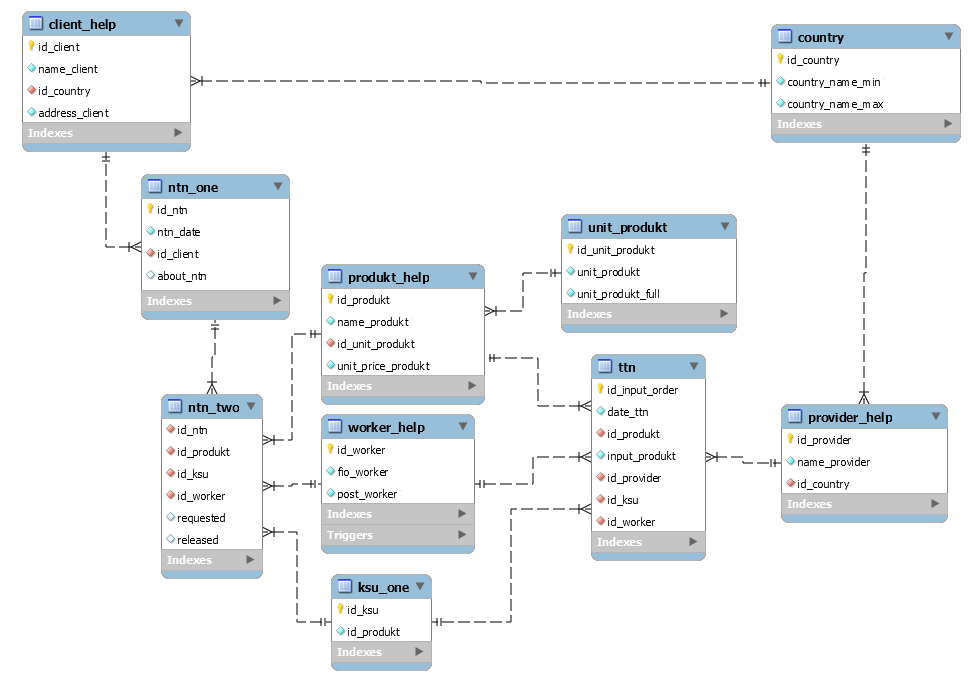


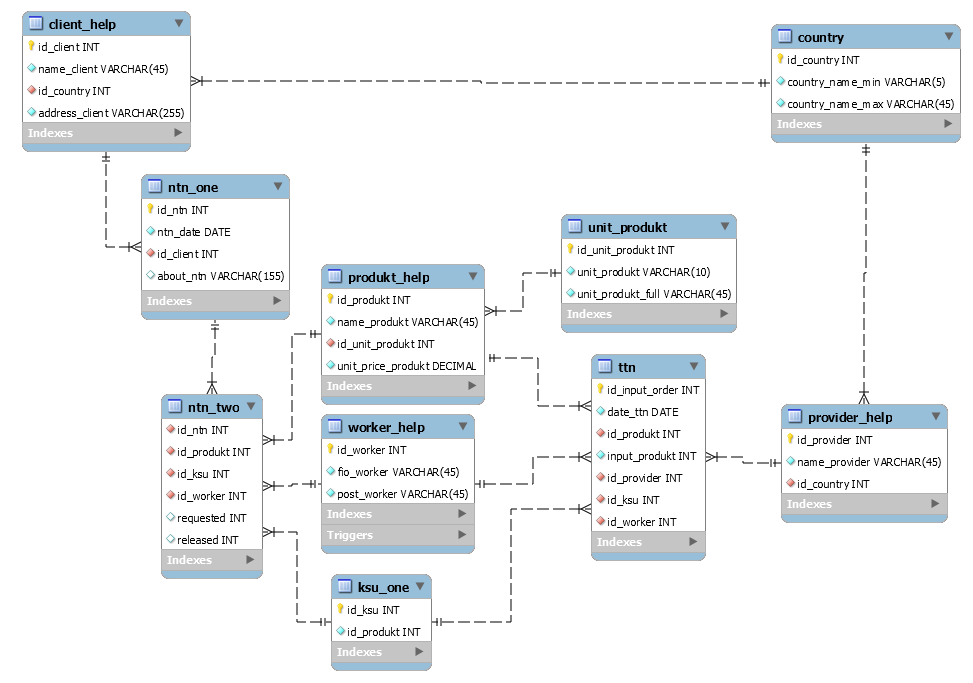


Рисунок 1.4.1 – Логическая модель

1.5 **Физическая модель данных**

Физическая модель данных – модель, определяющая размещение данных на внешних носителях, методы доступа и технику индексирования. Она так же называется внутренней моделью системы (рисунок 1.5.1).

Для каждого атрибута необходимо определить тип данных его значений, а также для всех сущностей должны определятся связи различного вида, в зависимости от надобности.



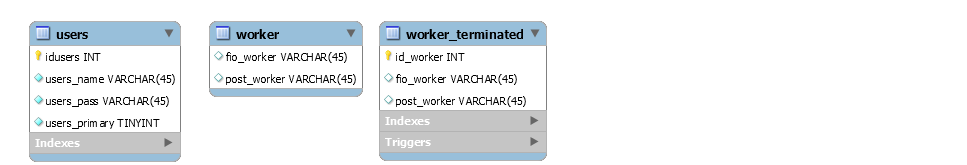


Рисунок 1.5.1 – Физическая модель

1.5.1 Описание структуры таблиц и связей между ними

Для проектирования базы данных была выбрана СУБД MySQL Workbench.

Разрабатываемая база данных будет иметь 13 таблиц:

client\_help, включает в себя следующие поля: id\_client(int), name\_client(VARCHAR(45)), id\_country(int), address\_client(VARCHAR(255));

country, включает в себя следующие поля: id\_country(int), country\_name\_min(VARCHAR(5)), country\_name\_max(VARCHAR(45));

ksu\_one, включает в себя следующие поля: id\_ksu(int), id\_produkt(int);

ntn\_one, включает в себя следующие поля: id\_ntn(int), ntn\_date(DATE), id\_client(int), about\_ntn(VARCHAR(155));

ntn\_two, включает в себя следующие поля: id\_ntn(int), id\_produkt(int), id\_ksu(int), id\_worker(int), requested(int), released(int);

produkt\_help, включает в себя следующие поля: id\_produkt(int), name\_produkt(VARCHAR(45)), id\_unit\_produkt(int), unit\_price\_produkt(DECIMAL);

provider\_help, включает в себя следующие поля: id\_provider(int), name\_provider(VARCHAR(45)), id\_country(int);

ttn, включает в себя следующие поля: id\_input\_order(int), date\_ttn(DATE), id\_produkt(int), input\_produkt(int), id\_provider(int), id\_ksu(int), id\_worker(int);

unit\_produkt, включает в себя следующие поля: id\_unit\_produkt(int), unit\_produkt(VARCHAR(10)), unit\_produkt\_full(VARCHAR(45));

users, включает в себя следующие поля: idusers(int), users\_name(VARCHAR(45)), users\_pass(VARCHAR(45)), users\_primary(TINYINT);

worker, включает в себя следующие поля: fio\_worker(VARCHAR(45)), post\_worker(VARCHAR(45));

worker\_help, включает в себя следующие поля: id\_worker(int), fio\_worker(VARCHAR(45)), post\_worker(VARCHAR(45));

worker\_terminated, включает в себя следующие поля: id\_worker(int), fio\_worker(VARCHAR(45)), post\_worker(VARCHAR(45));

1.5.2 **Описание индексов и ограничений**

Индекс представляет собой часть базы данных, в которой содержится информация об организации данных в таблицах БД [7].

В отличие от ключей, которые просто идентифицируют отдельные записи, индексы занимают дополнительные объемы памяти (довольно значительные) и могут храниться как совместно с таблицами, так и в виде отдельных файлов. Индексы создаются вместе с таблицей и обновляются при модификации данных. При этом работа по обновлению индекса для большой таблицы может отнимать много ресурсов, поэтому имеет смысл ограничить число индексов для таких таблиц, где происходит частое обновление данных.

Индекс содержит в себе уникальные идентификаторы записей и дополнительную информацию об организации данных. Поэтому если при выполнении запроса сервер или локальная СУБД обращается для отбора записи к индексу, то это занимает значительно меньше времени, т.к. понятно, что идентификатор, гораздо меньше самой записи. Кроме этого, индекс "знает", как организованы данные и может ускорять обработку за счет группирования записей по сходным значениям параметров.

Набор данных может работать и без применения индексов, но для этого соответствующая таблица БД не должна иметь первичного ключа - случай довольно редкий. Поэтому по умолчанию в наборе данных используется первичный индекс. При открытии набора данных все записи отсортированы в соответствии с первичным ключом [11].

В данной базе данных используются следующие индексы:

В таблице «client\_help» индекс – name\_client;

В таблице «produkt\_help» индекс – название name\_produkt;

В таблице «provider\_help» индекс – название name\_provider;

В таблице «users» индекс – название users\_name.

2 ОПИСАНИЕ ПРОГРАММЫ

2.1 Выбор и обоснование средств программирования для решения поставленной задачи, их краткая характеристика

Для реализации программы курсового проекта, выбрана среда программирования Microsoft Visual Studio.

Microsoft Visual Studio обеспечивает высокое качество кода на протяжении всего цикла жизни ПО, от проектирования до внедрения. Какие бы приложения не разрабатывались: для SharePoint, Интернет, Windows, Windows Phone и других платформ, Visual Studio – это универсальное всеобъемлющее решение. Сегодня это основное и самое эффективное средство разработки решений для платформы Microsoft.

Visual Studio олицетворяет собой представление корпорации Microsoft об интеллектуальных клиентских приложениях и позволяет быстро создавать подключаемые к базам данных приложения, способные обеспечить широчайшие возможности для работы пользователей. С помощью Visual Studio 2012 можно собирать и анализировать информацию проще, чем когда бы то ни было раньше, что способствует принятию эффективных бизнес-решений. С помощью Visual Studio 2019 возможно быстро создавать более безопасные, управляемые и надежные приложения, использующие преимущества Windows Vista и системы Microsoft Office.

А такие новинки, как поддержка ASP.NET AJAX и надстройка Silverlight для Visual Studio, помогут создавать целый спектр интерактивных веб-приложений. С выходом Windows Server 2019 и SQL Server 2019 возможности платформы Майкрософт еще больше расширятся.

Studio продолжает традиции, заложенные предыдущими версиями продукта, и содержит множество новинок, предназначенных как для упрощения создания приложений для различных платформ отдельными, так и для повышения производительности всей проектной команды. Поэтому мною была выбрана именно эта среда разработки [10].

2.2 Описание интерфейса, форм ввода и вывода данных

На данной форме вводятся регистрационные данные пользователя (рисунок 2.2.1).

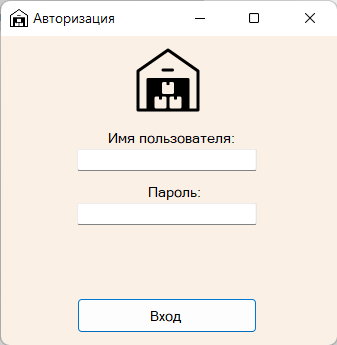


Рисунок 2.2.1 –­­ Форма авторизации

На данной форме можно полную информацию об остатках на складе: название, итоговую сумму, количество. А также просмотреть и сравнить товары на диаграмме остатков (рисунок 2.2.2).

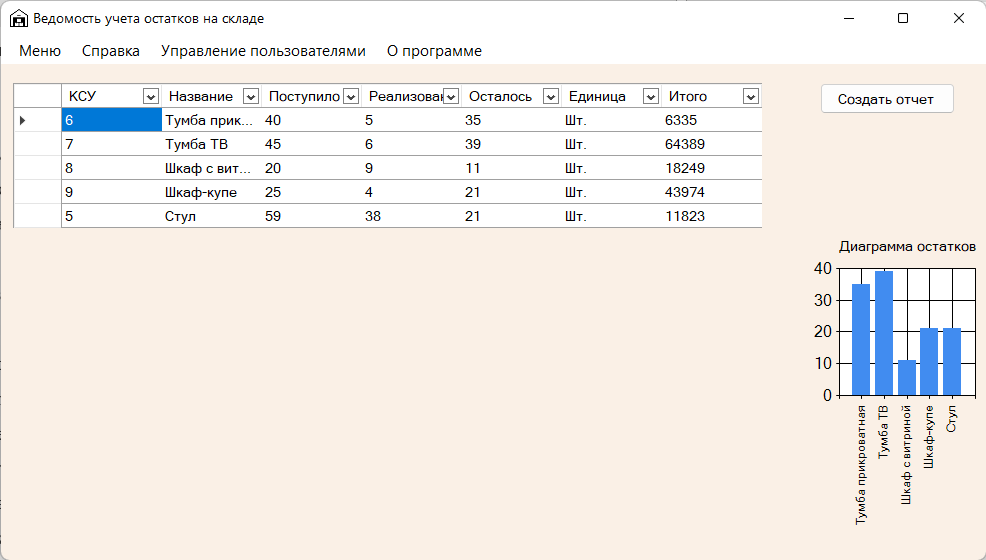


Рисунок 2.2.2 – Главная форма программы

На данной форме отображается проведенные товарно-транспортные накладные, а также предусмотрено добавление новых данных (рисунок 2.2.3).

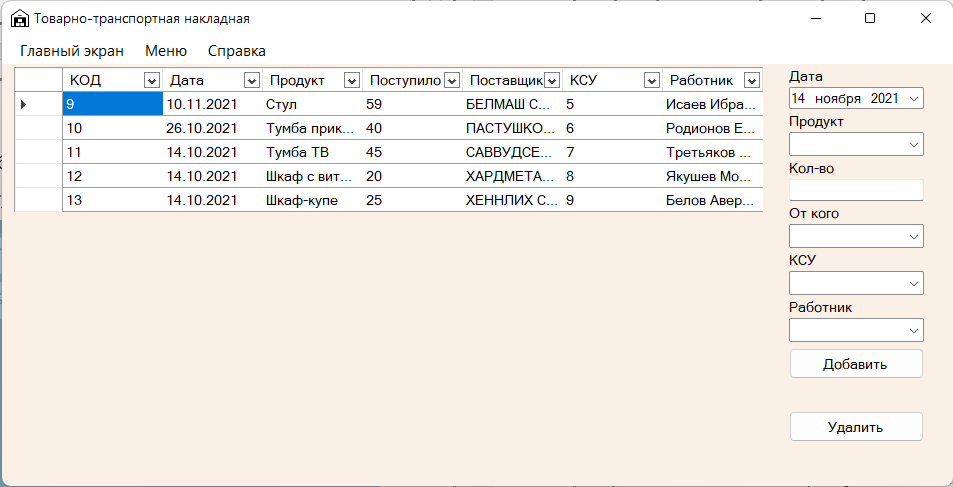


Рисунок 2.2.3 – Форма товарно-транспортных накладных

На данной форме отображается проведенные требование-накладные и их записи, а также пользователь может перейти к формам добавления (рисунок 2.2.4).

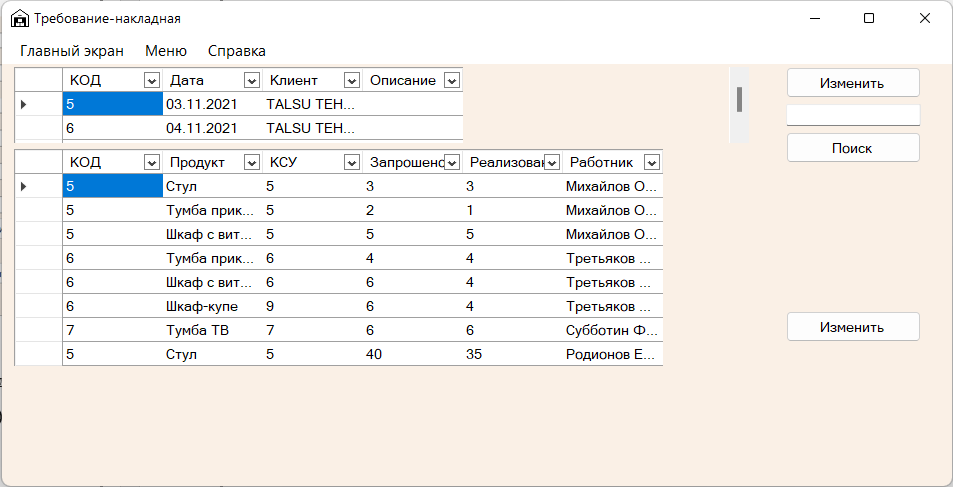


Рисунок 2.2.4 – Форма просмотра требование-накладных

На данной форме пользователь может добавлять новые товарно-накладные (рисунок 2.2.5).

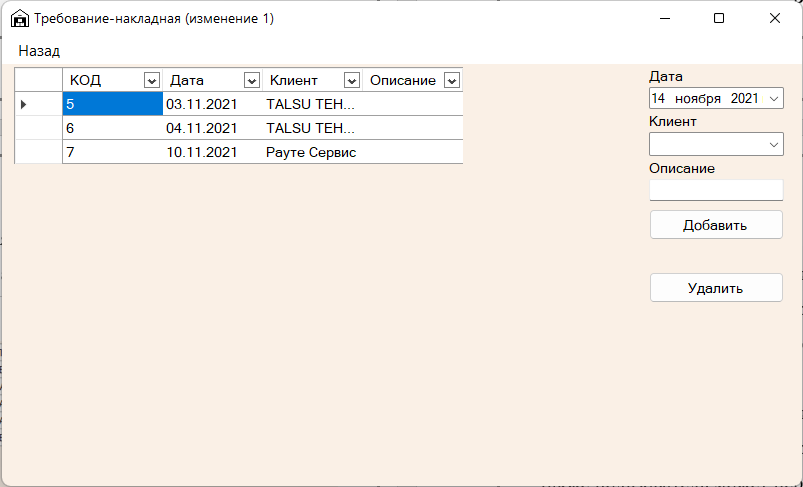


Рисунок 2.2.5 – Форма добавления требований-накладных

На данной форме пользователь может проводить материалы по требование-накладным (рисунок 2.2.6).

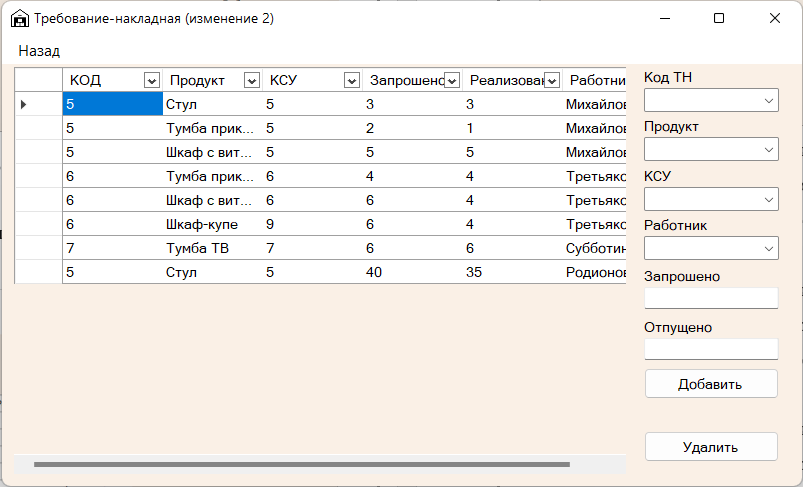


Рисунок 2.2.6 – Форма добавления материалов в требование-накладную

На данной форме отображается проведенные карточки складского учета, а также пользователь может перейти к форме добавления (рисунок 2.2.7).

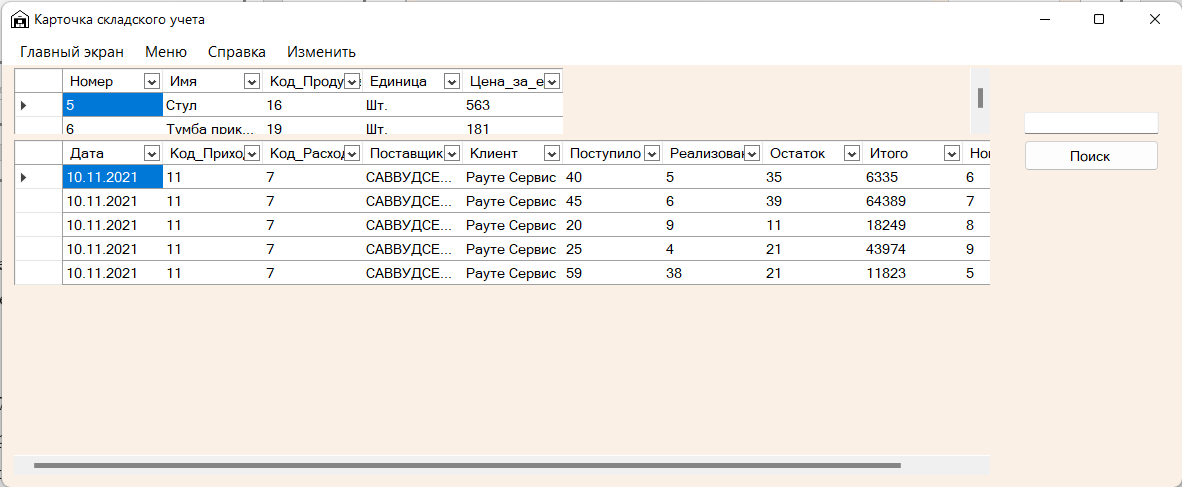


Рисунок 2.2.7 – Форма просмотра карточки складского учета

На данной форме пользователь может добавить новую карточку складского учета (рисунок 2.2.8).

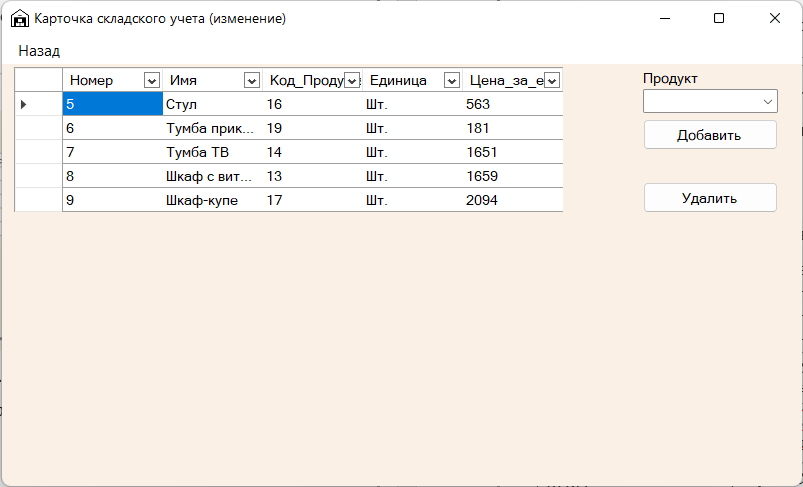


Рисунок 2.2.8 – Форма изменения карточки складского учета

На данной форме администратор может добавлять новых пользователь, а также выставлять их уровень доступа (рисунок 2.2.9).

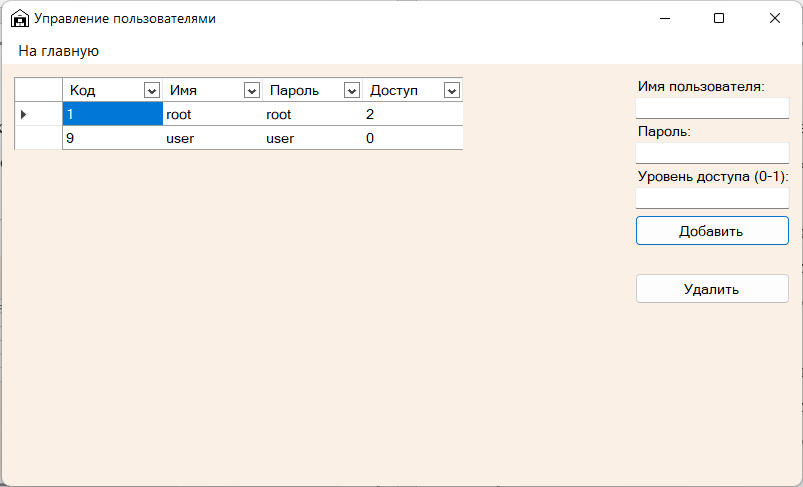


Рисунок 2.2.9 – Форма управления пользователями

На данной форме пользователь может добавлять новые справочные данные о клиентах (рисунок 2.2.10).

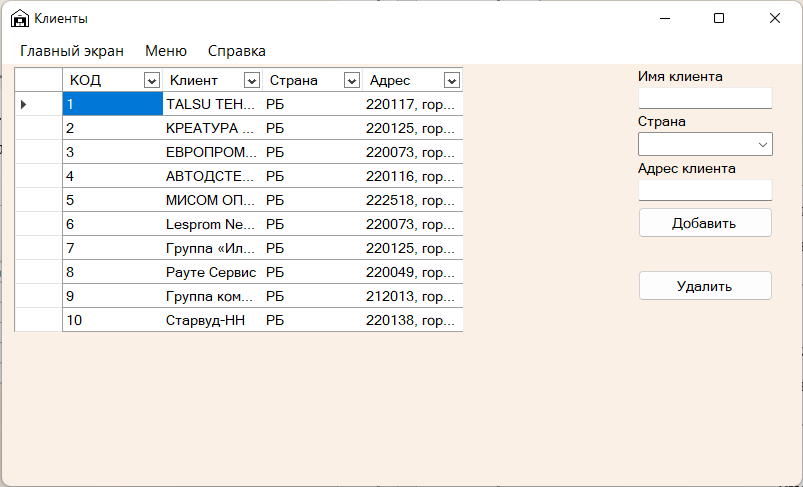


Рисунок 2.2.10 – Форма добавления клиентов

На данной форме пользователь может добавлять новые справочные данные о поставщиках (рисунок 2.2.11).

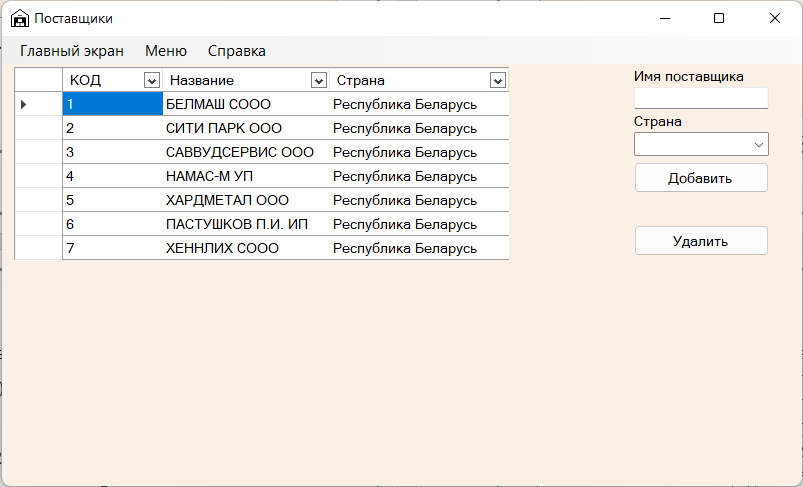


Рисунок 2.2.11 – Форма добавления поставщиков

На данной форме пользователь может добавлять новые справочные данные о товарах (рисунок 2.2.12).

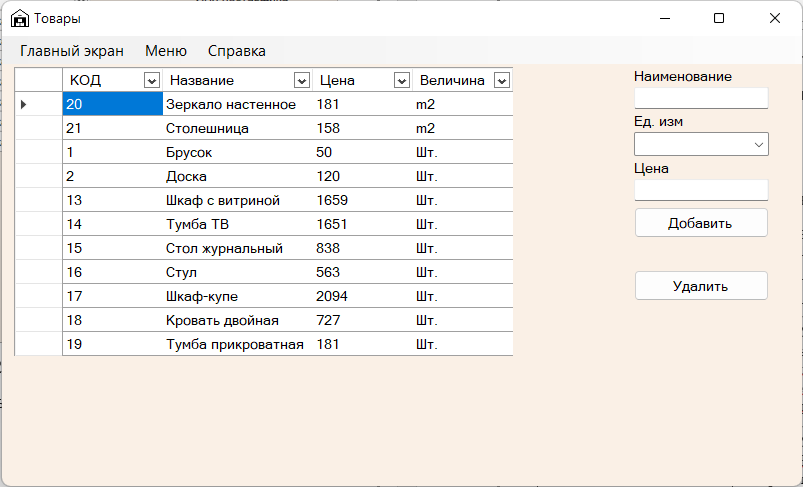


Рисунок 2.2.12 – Форма добавления товаров

На данной форме пользователь может добавлять новые справочные данные о сотрудниках (рисунок 2.2.13).

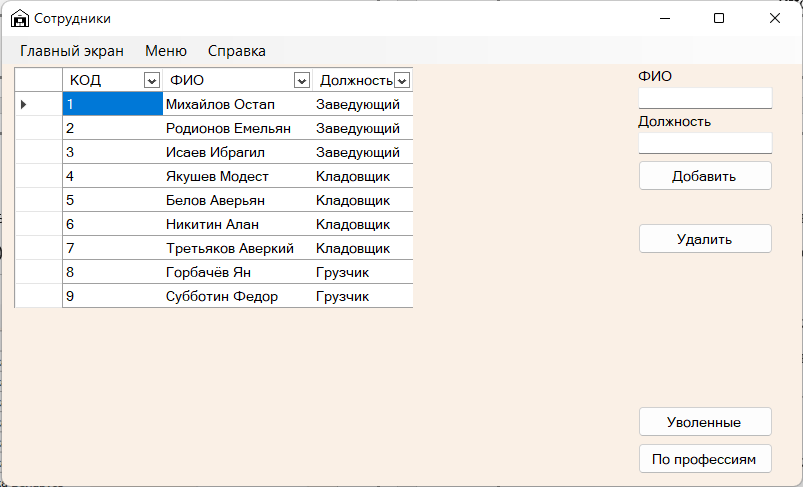


Рисунок 2.2.13 – Форма добавления сотрудников

На данной форме пользователь может добавлять новые справочные данные о странах (рисунок 2.2.14).

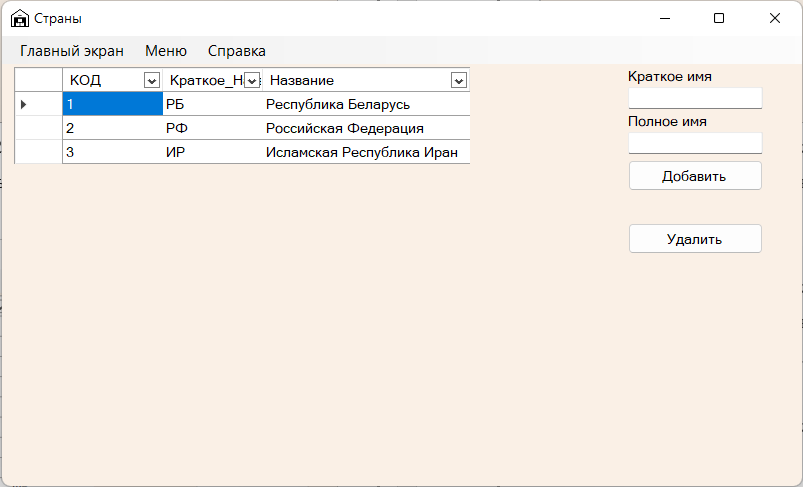


Рисунок 2.2.14 – Форма добавления стран

На данной форме пользователь может добавлять новые справочные данные о единицах измерения (рисунок 2.2.15).

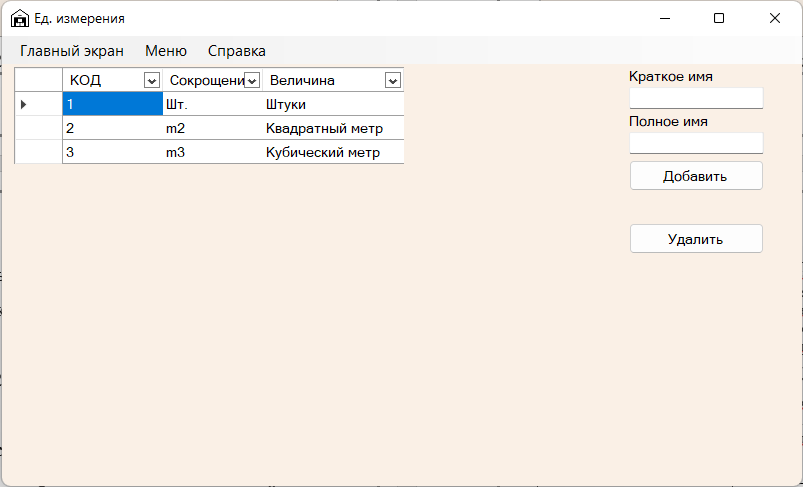


Рисунок 2.2.15 – Форма добавления единиц измерения

2.3 Описание запросов и представлений данных

Представление – это виртуальная таблица, содержимое которой определяется запросом. Как и настоящая таблица, представление состоит из совокупности именованных столбцов и строк данных. Пока представление не будет проиндексировано, оно не существует в базе данных как хранимая совокупность значений [7].

В данной программе используются несколько типов запросов на выборку, которые представлены в таблице 2.3.1.

Таблица 2.3.1 – Запросы

|  |  |
| --- | --- |
| Назначение запроса | Код запроса |
| 1 | 2 |
| Вывод таблицы остатки | CREATE VIEW `view\_balance` AS  select vi.id\_ksu as bid\_ksu, p.name\_produkt, vi.sum\_input\_produkt as input\_produkt, vr.sum\_released as released, (vi.sum\_input\_produkt - vr.sum\_released) as balance, vp.unit\_produkt, ((vi.sum\_input\_produkt - vr.sum\_released) \* p.unit\_price\_produkt) as total\_price  from produkt\_help AS p  JOIN view\_relesed AS vr |

Продолжение таблицы 2.3.1

|  |  |
| --- | --- |
| 1 | 2 |
|  | JOIN view\_input AS vi on vi.id\_produkt = vr.id\_produkt  JOIN view\_produkt\_help AS vp  JOIN ntn\_two AS n  JOIN ksu\_one AS k  inner join produkt\_help  on vi.id\_produkt = p.id\_produkt  group by vr.id\_produkt; |
| Вывод таблицы карточка складского учета | CREATE VIEW `view\_ksu\_two` AS  select vn.ntn\_date, vt.id\_input\_order, vn.id\_ntn, vt.name\_provider as aaa, vn.name\_client, vb.input\_produkt, vb.released, vb.balance, vb.total\_price, vb.bid\_ksu  from produkt\_help AS p  JOIN ksu\_one AS k  JOIN view\_produkt\_help AS vp  JOIN view\_ntn\_one AS vn  JOIN view\_ttn AS vt  JOIN view\_balance AS vb  GROUP BY vb.bid\_ksu; |
| Вывод таблицы требование-накладная | CREATE VIEW `view\_ntn\_two` AS  select id\_ntn, name\_produkt, id\_ksu, requested, released, fio\_worker  from ntn\_two  inner join produkt\_help  on ntn\_two.id\_produkt = produkt\_help.id\_produkt  inner join worker\_help  on ntn\_two.id\_worker = worker\_help.id\_worker; |
| Вывод таблицы товарно-транспортной накладной | CREATE VIEW `view\_ttn` AS  select id\_input\_order, date\_ttn, name\_produkt, input\_produkt, name\_provider, id\_ksu, fio\_worker  from ttn  inner join produkt\_help  on ttn.id\_produkt = produkt\_help.id\_produkt  inner join provider\_help  on ttn.id\_provider = provider\_help.id\_provider  inner join worker\_help  on ttn.id\_worker = worker\_help.id\_worker; |
| Вывод таблицы товары | CREATE VIEW `view\_produkt\_help` AS  select id\_produkt, name\_produkt, unit\_price\_produkt, unit\_produkt  from produkt\_help  inner join unit\_produkt  on produkt\_help.id\_unit\_produkt = unit\_produkt.id\_unit\_produkt; |
| Вывод таблицы клиенты | CREATE VIEW `view\_client\_help` AS  select id\_client, name\_client, country\_name\_min, address\_client  from client\_help  inner join country  on client\_help.id\_country = country.id\_country; |
| Вывод таблицы поставщики | CREATE VIEW `view\_provider\_help` AS  select id\_provider, name\_provider, country\_name\_max  from provider\_help  inner join country  on provider\_help.id\_country = country.id\_country; |

2.4 Описание триггеров и хранимых процедур

Триггер в MySQL — это определяемая пользователем SQL-команда, которая автоматически вызывается во время операций INSERT , DELETE или UPDATE [7].

В данной программе используются несколько триггеров, которые представлены в таблице 2.3.1.

Таблица 2.4.1 – Триггеры

|  |  |
| --- | --- |
| Назначение триггера | Код запроса |
| 1 | 2 |
| Каскадное добавление нового работника | CREATE DEFINER = CURRENT\_USER TRIGGER `mydb`.`worker\_help\_AFTER\_UPDATE` AFTER UPDATE ON `worker\_help` FOR EACH ROW  BEGIN  insert into worker (fio\_worker, post\_worker) values (old.fio\_worker, old.post\_worker);  END |
| Обновление таблицы всех работников | CREATE DEFINER = CURRENT\_USER TRIGGER `mydb`.`worker\_terminated\_AFTER\_INSERT` AFTER INSERT ON `worker\_terminated` FOR EACH ROW  BEGIN  insert into worker (fio\_worker, post\_worker) values (new.fio\_worker, new.post\_worker);  END |
| Занесение работника в уволенные | CREATE DEFINER = CURRENT\_USER TRIGGER `mydb`.`worker\_help\_AFTER\_DELETE` AFTER DELETE ON `worker\_help` FOR EACH ROW  BEGIN  insert into worker\_terminated (fio\_worker, post\_worker) values (old.fio\_worker, old.post\_worker);  END |

3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для непосредственной работы с программой ее необходимо установить. Перед началом установки рекомендуется закрыть все запущенные приложения для быстродействия системы. Запускаем установочный файл «.exe» и следуем указаниям мастера установки, после окончания установки программы появится на рабочем столе ярлык для запуска программы (Приложение В).

При запуске программы открывается форма входа, на которой ввести данные уже существующего пользователя получив данные для входа у администратора (рисунок 3.1).

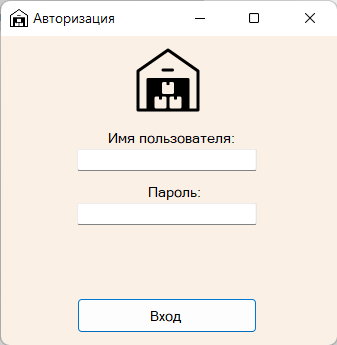


Рисунок 3.1 – Форма авторизации

После удачного входа в программу откроется основная форма программы, на которой можно выбрать интересующую форму с помощью системы меню или просмотреть остатки материалов на складе. Также можно просмотреть отображаемые данные на диаграммах и отчетах. При возникновении вопросов по работе с программой воспользуйтесь справкой (рисунок 3.2).

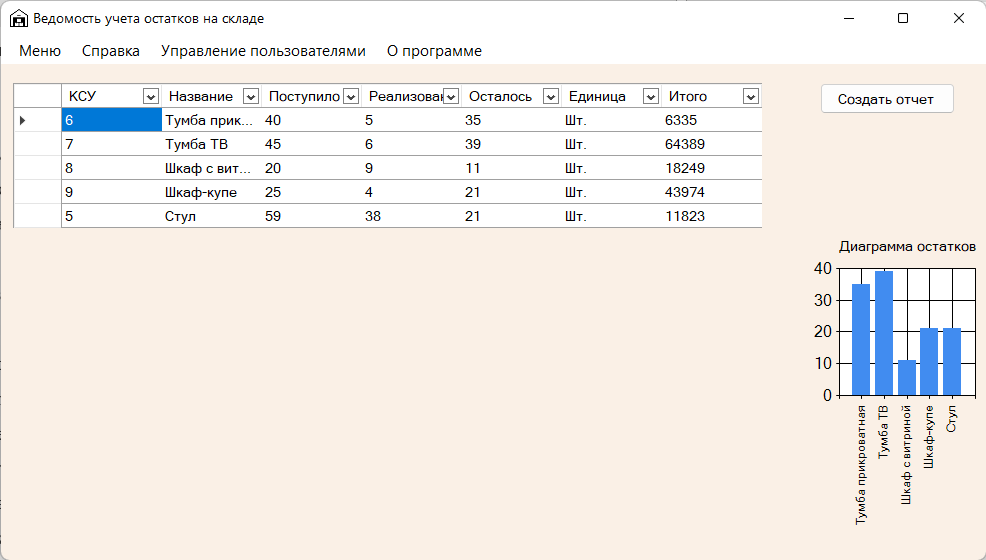


Рисунок 3.2 – Главная форма программы

Добавить новый справочный материал можно перейдя на форму «Клиенты» с помощью системы меню. Данные можно заполнять с помощью меню справа, а также пользоваться справочными данными для автоподстановки (рисунок 3.3).

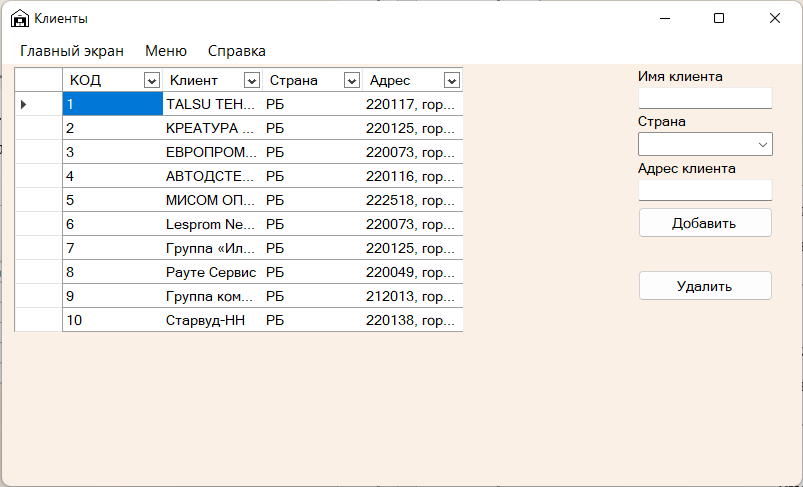


Рисунок 3.3 – Форма добавления клиентов

Добавить новый справочный материал можно перейдя на форму “Поставщики” с помощью системы меню. Данные можно заполнять с помощью меню справа, а также пользоваться справочными данными для автоподстановки (рисунок 3.4).

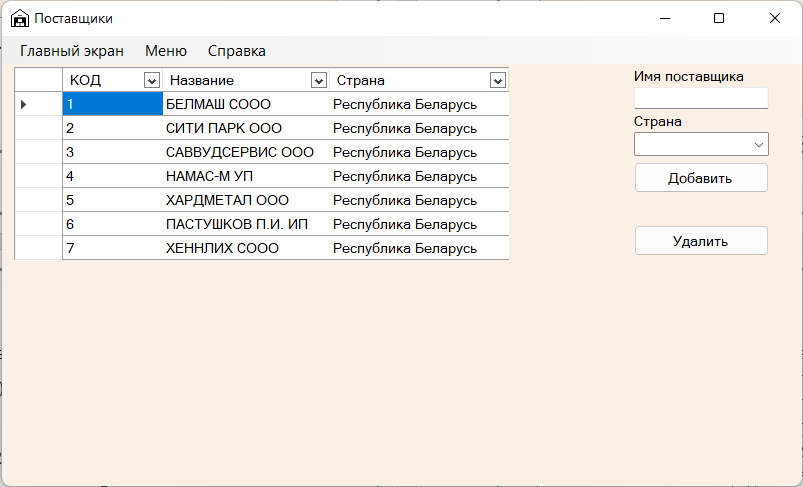


Рисунок 3.4 – Форма добавления поставщиков

Добавить новый справочный материал можно перейдя на форму “Товары” с помощью системы меню. Данные можно заполнять с помощью меню справа, а также пользоваться справочными данными для автоподстановки (рисунок 3.5).

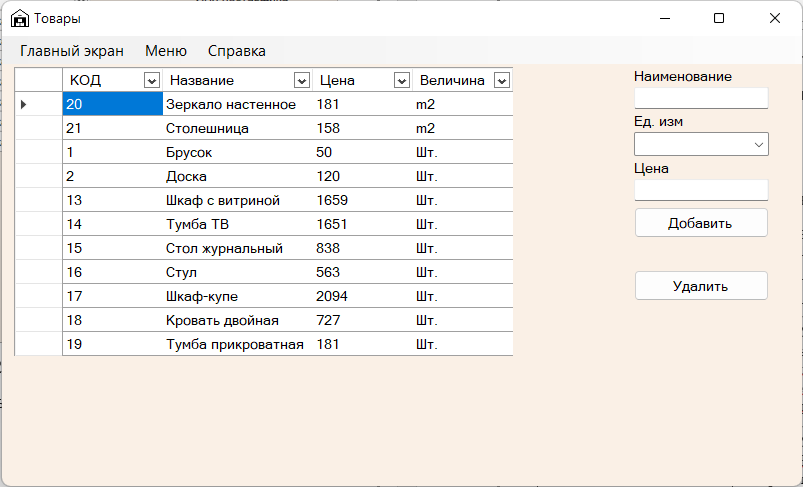


Рисунок 3.5 – Форма добавления товаров

Добавить новый справочный материал можно перейдя на форму “Сотрудники” с помощью системы меню. Данные можно заполнять с помощью меню справа (рисунок 3.6).

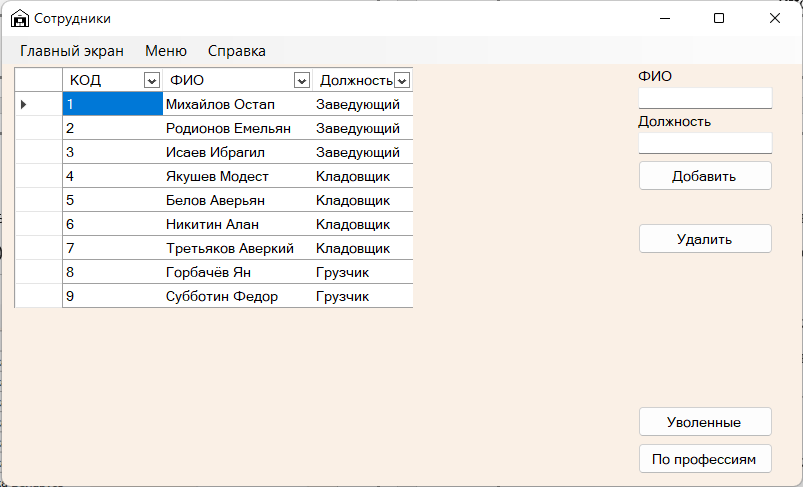


Рисунок 3.6 – Форма добавления сотрудников

Добавить новый справочный материал можно перейдя на форму “Страны” с помощью системы меню. Данные можно заполнять с помощью меню справа (рисунок 3.7).

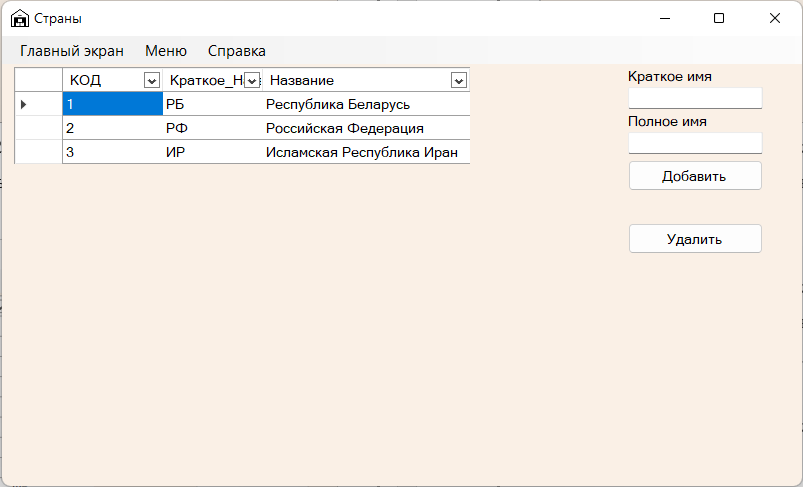


Рисунок 3.7 – Форма добавления стран

Добавить новый справочный материал можно перейдя на форму “Единицы измерения” с помощью системы меню. Данные можно заполнять с помощью меню справа (рисунок 3.8).

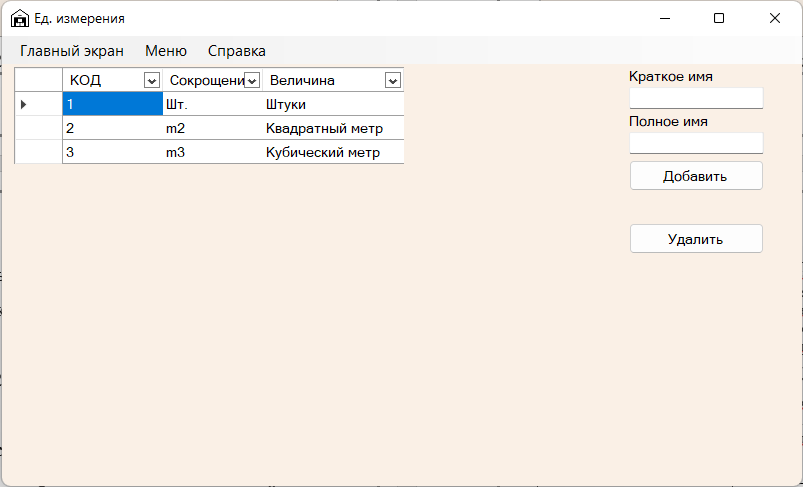
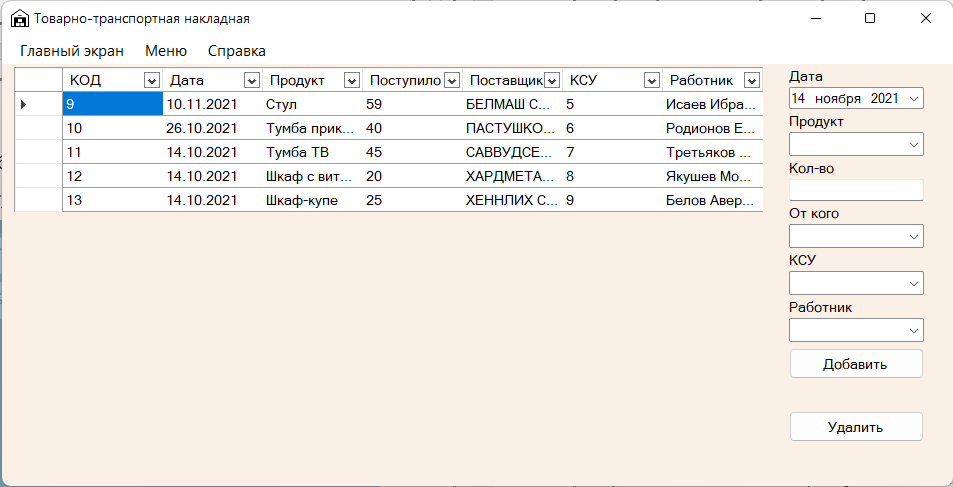


Рисунок 3.8 – Форма добавления единиц измерения

Просматривать проведенные документы по товарно-транспортной накладной, а также добавлять новые поставки пользователь может перейдя на форму “Товарно-транспортная накладная” (рисунок 3.9).

Рисунок 3.9 – Форма товарно-транспортных накладных

Увидеть все требования-накладные, а также найти интересующие данные пользователь может воспользоваться поиском. Который выводит шапку требование-накладной и соответствующие ей записи. Добавление данных предусмотрено нажатием на соответствующие кнопки напротив таблиц (рисунок 3.10).

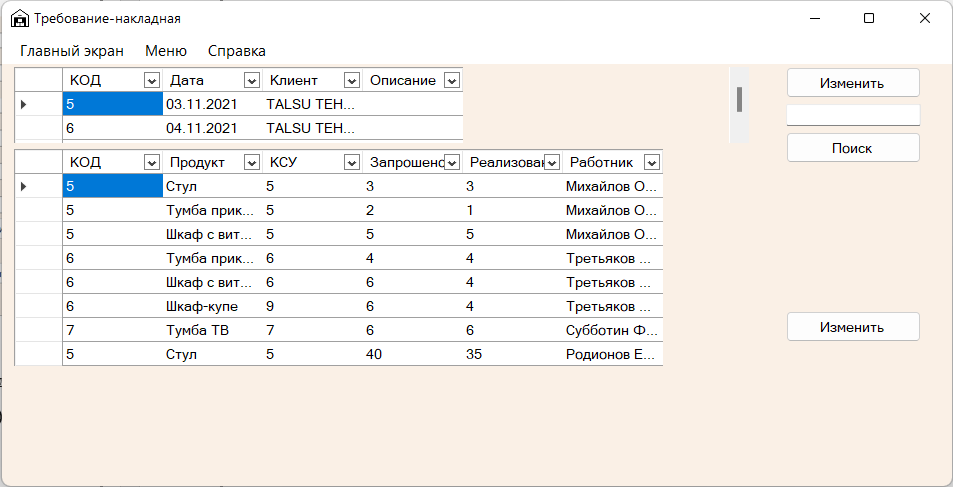


Рисунок 3.10 – Форма просмотра требование-накладных

Добавление требование-накладной происходит за два этапа. Первым этапом пользователь добавляет новый документ в соответствующем разделе (рисунок 3.11).

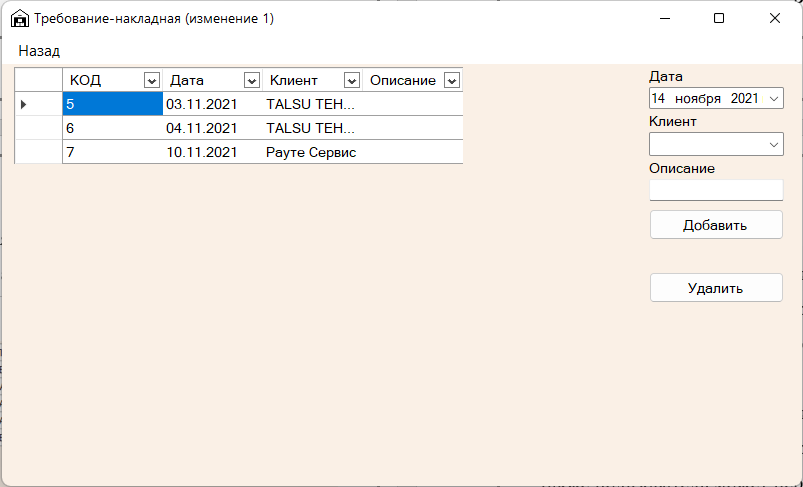


Рисунок 3.11 – Форма добавления требований-накладных

Добавление записей требование-накладной происходит вторым этапом. В котором пользователь добавляет материалы уже по имеющимся требование-накладным (рисунок 3.12).

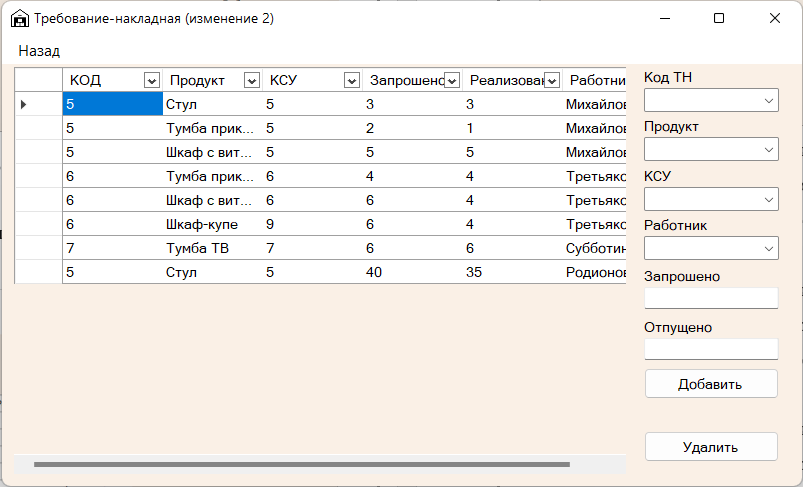


Рисунок 3.12 – Форма добавления материалов в требование-накладную

Просматривать движение материалов позволит форма “Карточка складского учета”. Добавить новые карточки можно нажав на кнопку “Изменить” (рисунок 3.13).

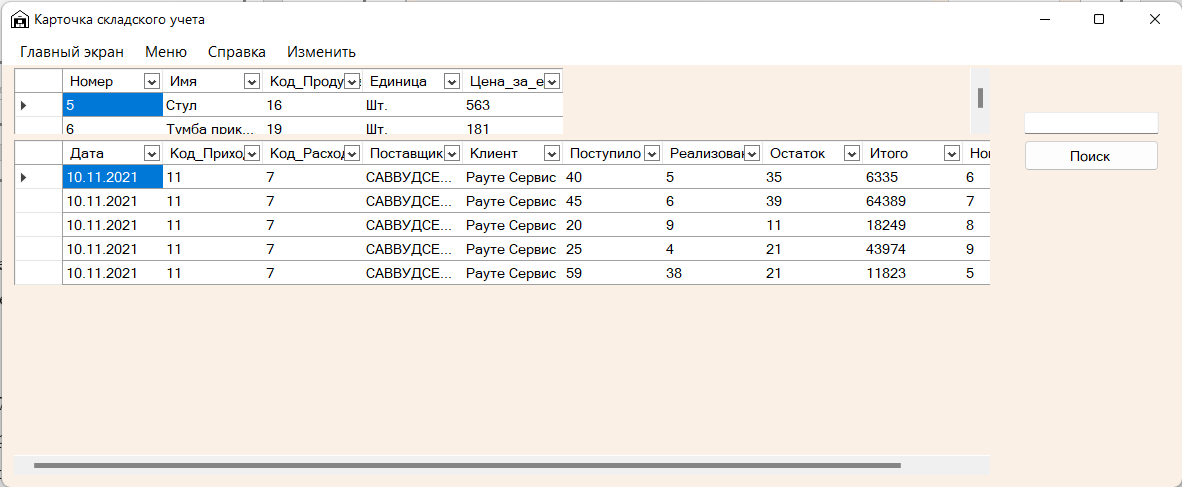


Рисунок 3.13 – Форма просмотра карточки складского учета

Изменять данные карточки складского учета можно на данной форме (рисунок 3.14).

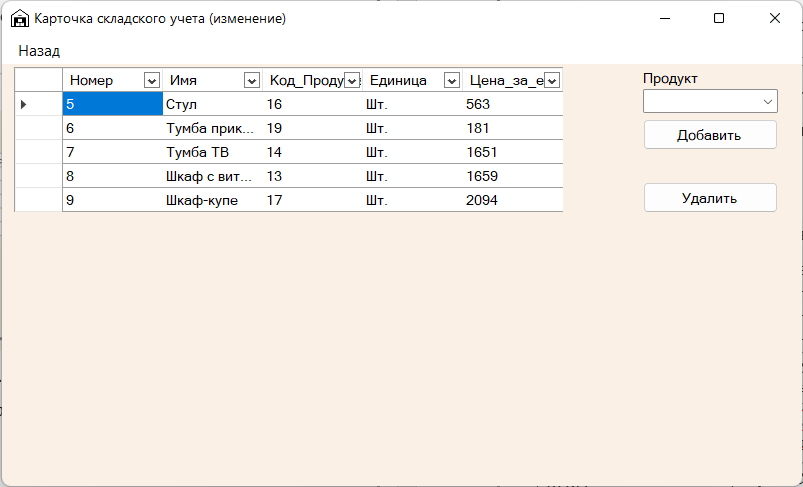


Рисунок 3.14 – Форма изменения карточки складского учета

4 СПРАВОЧНАЯ СИСТЕМА

Справочная система представлена отдельной формой. Вызвать справку можно нажат на кнопку «О программе», находящейся на главной форме программного модуля (рисунок 4.1).

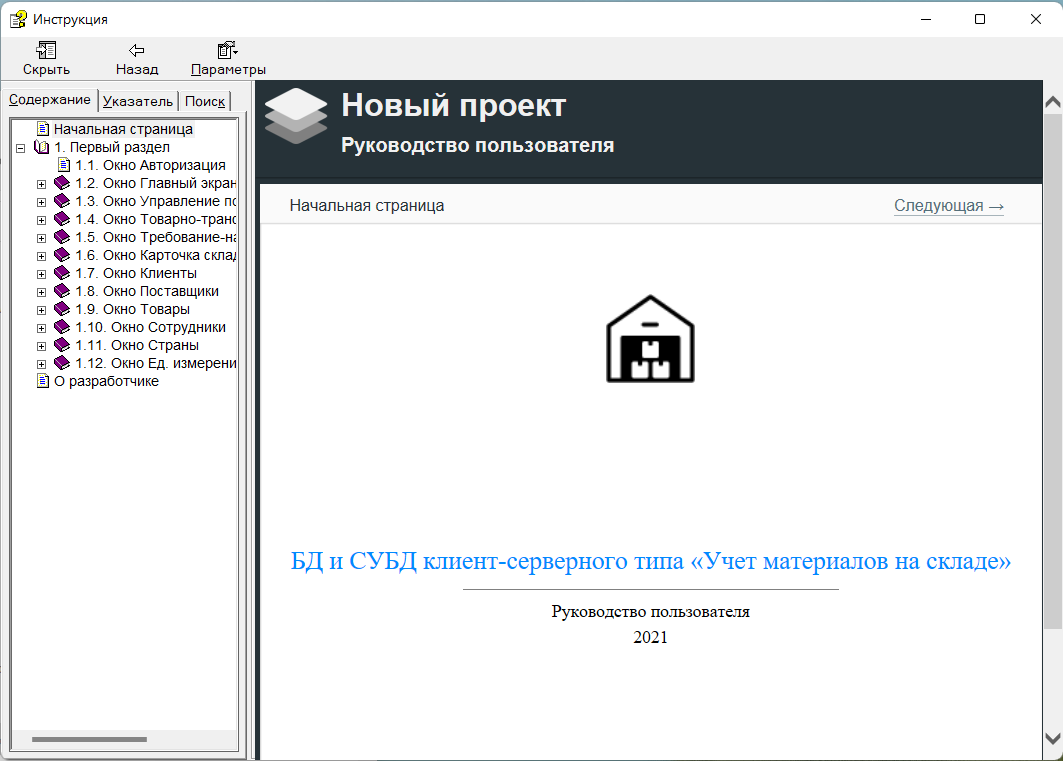


Рисунок 4.1 – Справка о программе

В справочной системе приведена краткая инструкция о программе.

5 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ

Существуют два принципа тестирования программы:

функциональное тестирование (тестирование «черного ящика»);

структурное тестирование (тестирование «белого ящика»).

При функциональном тестировании программа рассматривается как черный ящик. Целью тестирования ставится выяснение обстоятельств, в которых поведение программы не соответствует спецификации. Для обнаружения всех ошибок в программе необходимо выполнить исчерпывающее тестирование, то есть тестирование на всевозможных наборах данных.

Объектом испытаний является приложение клиент-серверной типа «Учет материалов на складе». Для проверки разработанного модуля методом «черного ящика» необходимо выполнить следующие тесты:

проверка загрузки приложения;

проверка работы программы;

проверка выхода из приложения.

Результаты проведенного тестирования программы приведены в таблице 5.1:

Таблица 5.1 Порядок выполнения тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Входные данные | Ожидаемый  результат | Фактический  результат | Результат  тестирования |
| 1 | 2 | 3 | 4 |
| ***Проверка загрузки приложения*** | | | |
| Запуск приложения | Загрузка приложения | Приложение загружено | Успешно |
| Авторизация | Вход в систему | Открылось главное окно | Успешно |
| Ввод пароля и логина администратора | Переход к форме «Администратор» | Открылась форма «Администратор» | Успешно |
| Открытие главной формы | Открытие главной формы | Форма открылась | Успешно |

Продолжение таблицы 5.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | | 3 | | 4 | |
| ***Работа программы*** | | | | | | |
| Открытие формы администратора | | Открытие формы администратора | | Форма открылась | | Успешно |
| Открытие формы «Карточка складского учета» | | Открытие формы карточка складского учета | | Форма открылась | | Успешно |
| Нажатие на кнопку «Добавить» | | Товар добавлен | | Товар добавился | | Успешно |
| Нажатие на кнопку «О приложении» | | Открытие справочного файла | | Справка открылась | | Успешно |
| Просмотр отчетов | | Открытие отчетов | | Программа открыла отчеты | | Успешно |
| Ввод даты | | Ввод текущей даты | | Дата не вывелась | | Неуспешно |
| Просмотр диаграмм | | Отображение диаграмм | | Диаграммы отобразились | | Успешно |
| Автоматическое обновление данных | | Данные обновлены | | Данные обновлены | | Успешно |
| ***Проверка завершения приложения*** | | | | | | |
| Завершение системы с использованием крестика меню | | Приложение  завершится | | Приложение  завершилось | | Успешно |

При тестировании программы была обнаружена ошибка, вводилась некорректная дата. Ошибка была устранена путем доработки программного кода, также было использовано регрессионное тестирование для повторной проверки программы на отсутствие ошибок.

ЗАКЛЮЧЕНИЕ

Актуальность темы проекта определяется потребностью в автоматизации системы складского учета. Автоматизация заключатся в более удобном процессе работы пользователя с бумагами бухгалтерского учета. Также актуальность темы заключается в простате использования электронного бухгалтерского помощника в виде клиент-серверной системы «Учет материалов на складе».

Цель курсового проекта: БД и СУБД клиент-серверного типа «Учет материалов на складе» была достигнута.

Были решены следующие задачи:

собрана необходимая информация;

продумана структура проекта (Приложение Б);

определены функции, которые будет выполнять программа (Приложение Г);

реализован рабочий проект (Приложение А);

отлажен и протестирован созданный проект;

разработана документация.

БД и СУБД клиент-серверного типа «Учет материалов на складе», в свою очередь, позволит:

оптимизировать рабочий процесс;

повысить скорость работы работника;

повысить качество работы;

увеличить качество работы;

предоставить комфортные условия работы со складом.

К недостаткам программы можно отнести:

предназначается только для Windows;

существуют конкурентоспособные программы.

Созданная программа имеет четкую структуру, качественное информативное содержание, а также интуитивно понятную навигацию.

В процессе проектирования были использованы следующие методы: анализ литературы, изучение и обобщение отечественной и зарубежной прак-тики, сравнение, моделирование, синтез, аналогия и обобщение.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

***Основная литература***

1. ГОСТ 19.104-78. Единая система программной документации. Основные надписи. – Введ. 81-07-01. – М.: Издательство стандартов, 2001.
2. ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом. – Введ 81-07-01. – М.: Изд-во стандартов, 2001.
3. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. – Введ. 81-07-01. – М.: Издательство стандартов, 2001.
4. ГОСТ 19.301-2000 ЕСПД. Руководство оператора. – М.: Издательство стандартов, 2001.
5. ГОСТ 19.402-2000 ЕСПД. Описание программы. – М.: Издательство стандартов, 2001.
6. ГОСТЬ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. – Введ. 01-01-92. – М.: Изд-во стандартов, 2001.

***Дополнительная литература***

1. Введение в MySQL [Электронный ресурс] – Режим доступа: https://metanit.com/sql/mysql/2.1.php Дата доступа: 18.10.2021.
2. Руководство по wpf [Электронный ресурс] – Режим доступа:

<https://metanit.com/sharp/wpf/> Дата доступа: 18.10.2021.

1. Подключение библиотек [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/tutorial/3.46.php> Дата доступа: 18.10.2021.
2. Руководство по C# [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/> Дата доступа: 18.10.2021.
3. О программировании [Электронный ресурс] – Режим доступа:

<https://metanit.com/sharp/> Дата доступа: 18.10.2021.

1. C# для начинающих [Электронный ресурс] – Режим доступа: <https://www.cyberforum.ru/> Дата доступа: 18.10.2021.
2. Платформа .NET Framework [Электронный ресурс] – Режим доступа: [https://professorweb.ru/](https://professorweb.ru/%20) Дата доступа: 18.10.2021.
3. Материально-производственные запасы (МПЗ) [Электронный ресурс] – Режим доступа: [https://www.audit-it.ru/terms/accounting](https://www.audit-it.ru/terms/accounting/materialno_proizvodstvennye_zapasy_mpz.html) Дата доступа: 18.10.2021

ПРИЛОЖЕНИЕ А

**Листинг программы**

Листинг 1 – Модуль main.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Sklad.Spravka;

using System.Diagnostics;

using FastReport;

using System.IO;

namespace Sklad

{

public partial class Main : Form

{

// Проверка авторизации

IRegistration \_autorithation;

public Main(IRegistration autorithation)

{

InitializeComponent();

\_autorithation = autorithation;

}

// Код позволяет загрузить данные в таблицу

private void Main\_Load(object sender, EventArgs e)

{

try

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

string script = "SELECT bid\_ksu as КСУ, name\_produkt as Название, input\_produkt as Поступило, released as Реализовано, balance as Осталось, unit\_produkt as Единица, total\_price as Итого FROM view\_balance;";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

connection.Close();

if (\_autorithation.primary == false)

{

упровлениеПользователямиToolStripMenuItem.Visible = false;

}

int ss = 0;

while (ss < advancedDataGridView1.Rows.Count)

{

var bal1 = advancedDataGridView1.Rows[ss].Cells[1].Value;

var bal2 = advancedDataGridView1.Rows[ss].Cells[4].Value;

chart1.Series[0].Points.AddXY(Convert.ToString(bal1), Convert.ToInt32(bal2));

ss++;

}

}

catch (FormatException)

{

MessageBox.Show("Неверный формат.");

}

catch (ArgumentOutOfRangeException)

{

MessageBox.Show("Значение вне границ.");

}

catch (MySql.Data.MySqlClient.MySqlException)

{

MessageBox.Show("Значение вне границ.");

}

catch (System.OverflowException)

{

MessageBox.Show("Используеться слишком большое число.");

}

}

// Переход на другие формы

private void клиентыToolStripMenuItem\_Click(object sender, EventArgs e)

{

client\_help form = new client\_help();

form.Visible = true;

this.Visible = false;

}

private void упровлениеПользователямиToolStripMenuItem\_Click(object sender, EventArgs e)

{

reg form = new reg();

form.Visible = true;

this.Visible = false;

}

private void advancedDataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

// Создание отчета

private void button1\_Click(object sender, EventArgs e)

{

Report report = new Report();

report.Load("report.frx");

designerControl1.Report = report;

designerControl1.Report.Show();

}

// Вызов справки

private void оПрограммеToolStripMenuItem\_Click(object sender, EventArgs e)

{

Process.Start("help.chm");

}

}

}

Листинг 2 – Модуль autorithation.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Sklad

{

public partial class autorithation : Form, IRegistration

{

// Подключение бд

public static MySqlConnection connection = new MySqlConnection("server=127.0.0.1; port=3306; username=root; password=1234; database=mydb");

DataTable datatable = new DataTable();

public bool primary { get; set; }

public void Con()

{

if (connection.State == System.Data.ConnectionState.Closed)

connection.Open();

}

public autorithation()

{

Con();

InitializeComponent();

}

// Проверка данных входа

private void button1\_Click(object sender, EventArgs e)

{

try

{

MySqlDataAdapter dataAdapter = new MySqlDataAdapter

($"Select \* from users where users\_name = '{textBox1.Text}' and users\_pass = '{textBox2.Text}'", connection);

dataAdapter.Fill(datatable);

if (datatable.Rows.Count == 1)

{

primary = Convert.ToBoolean(datatable.Rows[0].ItemArray[3]);

Main form = new Main(this);

form.Visible = true;

this.Visible = false;

}

else

{

MessageBox.Show("Неверный логин или пароль!");

}

}

catch (FormatException)

{

MessageBox.Show("Неверный формат.");

}

catch (ArgumentOutOfRangeException)

{

MessageBox.Show("Значение вне границ.");

}

catch (MySql.Data.MySqlClient.MySqlException)

{

MessageBox.Show("Значение вне границ.");

}

catch (System.OverflowException)

{

MessageBox.Show("Используеться слишком большое число.");

}

}

private void autorithation\_Load(object sender, EventArgs e)

{

}

}

}

Листинг 3 – Модуль ksu.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Sklad.Spravka;

namespace Sklad

{

public partial class ksu : Form, IRegistration

{

// Подключение бд

public bool primary { get { return true; } set { } }

MySqlConnection connection;

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

public ksu()

{

InitializeComponent();

connection = new MySqlConnection(connectionString);

}

// Код позволяет загрузить данные в таблицу

private void ksu\_Load(object sender, EventArgs e)

{

string script = "SELECT id\_ksu as Номер, name\_produkt as Имя, id\_produkt as Код\_Продукта, unit\_produkt as Единица, unit\_price\_produkt as Цена\_за\_ед FROM view\_ksu\_one;";

connection.Open();

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

connection.Close();

string script1 = "SELECT ntn\_date as Дата, id\_input\_order as Код\_Прихода, id\_ntn as Код\_Расхода, aaa as Поставщик, name\_client as Клиент, input\_produkt as Поступило, released as Реализовано, balance as Остаток, total\_price as Итого, bid\_ksu as Номер FROM view\_ksu\_two;";

connection.Open();

MySqlDataAdapter mySql\_dataAdapter1 = new MySqlDataAdapter(script1, connection);

DataTable table1 = new DataTable();

mySql\_dataAdapter1.Fill(table1);

advancedDataGridView2.DataSource = table1;

SelectProductsNames();

connection.Close();

}

private void SelectProductsNames()

{

}

// Поиск

private void button1\_Click(object sender, EventArgs e)

{

try

{

MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT id\_ksu as Номер, name\_produkt as Имя, id\_produkt as Код\_Продукта, unit\_produkt as Единица, unit\_price\_produkt as Цена\_за\_ед FROM view\_ksu\_one WHERE id\_ksu LIKE '%" + textBox1.Text + "%'", connection);

DataSet ds = new DataSet();

adapter.Fill(ds);

this.advancedDataGridView1.AutoGenerateColumns = true;

this.advancedDataGridView1.DataSource = ds.Tables[0];

connection.Close();

MySqlDataAdapter adapter2 = new MySqlDataAdapter("SELECT ntn\_date as Дата, id\_input\_order as Код\_Прихода, id\_ntn as Код\_Расхода, aaa as Поставщик, name\_client as Клиент, input\_produkt as Поступило, released as Реализовано, balance as Остаток, total\_price as Цена, bid\_ksu as Номер FROM view\_ksu\_two WHERE bid\_ksu LIKE '%" + textBox1.Text + "%'", connection);

DataSet ds2 = new DataSet();

adapter2.Fill(ds2);

this.advancedDataGridView2.AutoGenerateColumns = true;

this.advancedDataGridView2.DataSource = ds2.Tables[0];

connection.Close();

}

catch (FormatException)

{

MessageBox.Show("Неверный формат.");

}

catch (ArgumentOutOfRangeException)

{

MessageBox.Show("Значение вне границ.");

}

catch (MySql.Data.MySqlClient.MySqlException)

{

MessageBox.Show("Значение вне границ.");

}

catch (System.OverflowException)

{

MessageBox.Show("Используеться слишком большое число.");

}

}

private void button2\_Click(object sender, EventArgs e)

{

}

// Переход на другие формы

private void главныйЭкранToolStripMenuItem\_Click(object sender, EventArgs e)

{

Main form = new Main(this);

form.Visible = true;

this.Visible = false;

}

private void товарнотранспортнаяНакладнаяToolStripMenuItem\_Click(object sender, EventArgs e)

{

ttn form = new ttn();

form.Visible = true;

this.Visible = false;

}

}

}

Листинг 4 – Модуль ntn.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

using Sklad.Spravka;

namespace Sklad

{

public partial class ntn : Form, IRegistration

{

// Подключение бд

private DataGridViewRow \_advancedDataGridViewSelectedRow;

public bool primary { get { return true; } set { } }

MySqlConnection connection;

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

public ntn()

{

InitializeComponent();

connection = new MySqlConnection(connectionString);

}

// Переход на другие формы

private void главныйЭкранToolStripMenuItem\_Click(object sender, EventArgs e)

{

Main form = new Main(this);

form.Visible = true;

this.Visible = false;

}

private void товарнотранспортнаяНакладнаяToolStripMenuItem\_Click(object sender, EventArgs e)

{

ttn form = new ttn();

form.Visible = true;

this.Visible = false;

}

private void карточкаСкладскогоУчетаToolStripMenuItem\_Click(object sender, EventArgs e)

{

ksu form = new ksu();

form.Visible = true;

this.Visible = false;

}

private void клиентыToolStripMenuItem\_Click(object sender, EventArgs e)

{

client\_help form = new client\_help();

form.Visible = true;

this.Visible = false;

}

// Код позволяет загрузить данные в таблицу

private void ntn\_Load(object sender, EventArgs e)

{

connection.Open();

string script = "SELECT id\_ntn as КОД, ntn\_date as Дата, name\_client as Клиент, about\_ntn as Описание FROM view\_ntn\_one;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

script = "SELECT id\_ntn as КОД, name\_produkt as Продукт, id\_ksu as КСУ, requested as Запрошено, released as Реализовано, fio\_worker as Работник FROM view\_ntn\_two;";

mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView2.DataSource = table;

connection.Close();

}

// Переход на другие формы

private void button1\_Click(object sender, EventArgs e)

{

ntn\_ed form = new ntn\_ed();

form.Visible = true;

this.Visible = false;

}

private void button2\_Click(object sender, EventArgs e)

{

ntn\_ed2 form = new ntn\_ed2();

form.Visible = true;

this.Visible = false;

}

// Поиск

private void button3\_Click(object sender, EventArgs e)

{

MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT id\_ntn as КОД, ntn\_date as Дата, name\_client as Клиент, about\_ntn as Описание FROM view\_ntn\_one WHERE id\_ntn LIKE '%" + textBox1.Text + "%'", connection);

DataSet ds = new DataSet();

adapter.Fill(ds);

this.advancedDataGridView1.AutoGenerateColumns = true;

this.advancedDataGridView1.DataSource = ds.Tables[0];

connection.Close();

MySqlDataAdapter adapter2 = new MySqlDataAdapter("SELECT id\_ntn as КОД, name\_produkt as Продукт, id\_ksu as КСУ, requested as Запрошено, released as Реализовано, fio\_worker as Работник FROM view\_ntn\_two WHERE id\_ntn LIKE '%" + textBox1.Text + "%'", connection);

DataSet ds2 = new DataSet();

adapter2.Fill(ds2);

this.advancedDataGridView2.AutoGenerateColumns = true;

this.advancedDataGridView2.DataSource = ds2.Tables[0];

connection.Close();

}

}

}

Листинг 5 – Модуль ntn\_ed2.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Sklad

{

public partial class ntn\_ed2 : Form, IRegistration

{

// Подключение бд

private DataGridViewRow \_advancedDataGridViewSelectedRow;

public bool primary { get { return true; } set { } }

MySqlConnection connection;

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

public ntn\_ed2()

{

InitializeComponent();

connection = new MySqlConnection(connectionString);

}

// Переход на другие формы

private void назадToolStripMenuItem\_Click(object sender, EventArgs e)

{

ntn form = new ntn();

form.Visible = true;

this.Visible = false;

}

// Код позволяет загрузить данные в таблицу

private void ntn\_ed2\_Load(object sender, EventArgs e)

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

string script = "SELECT id\_ntn as КОД, name\_produkt as Продукт, id\_ksu as КСУ, requested as Запрошено, released as Реализовано, fio\_worker as Работник FROM view\_ntn\_two;";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

Select();

connection.Close();

}

// Добавление данных в бд

private void ProductAddingButton\_Click(object sender, EventArgs e)

{

connection.Open();

Insetrt();

connection.Close();

}

private void Insetrt()

{

try

{

string unitProductSelectionQuery = $"Select `id\_ntn` from `ntn\_one`" +

$" where `id\_ntn` = '{comboBox1.SelectedItem}'";

MySqlCommand selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex = Convert.ToInt32(selectionQuery.ExecuteScalar());

unitProductSelectionQuery = $"Select `id\_produkt` from `produkt\_help`" +

$" where `name\_produkt` = '{comboBox2.SelectedItem}'";

selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex2 = Convert.ToInt32(selectionQuery.ExecuteScalar());

unitProductSelectionQuery = $"Select `id\_ksu` from `ksu\_one`" +

$" where `id\_ksu` = '{comboBox3.SelectedItem}'";

selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex3 = Convert.ToInt32(selectionQuery.ExecuteScalar());

unitProductSelectionQuery = $"Select `id\_worker` from `worker\_help`" +

$" where `fio\_worker` = '{comboBox4.SelectedItem}'";

selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex4 = Convert.ToInt32(selectionQuery.ExecuteScalar());

string query = $"insert into `ntn\_two` (id\_ntn, id\_produkt, id\_ksu, id\_worker, requested, released) " +

$"values ({unitIndex}, {unitIndex2}, {unitIndex3}, {unitIndex4}, {Convert.ToSingle(textBox1.Text)}, {Convert.ToSingle(NewProductName.Text)});";

MySqlCommand insertProductCommand = new MySqlCommand(query, connection);

insertProductCommand.ExecuteNonQuery();

string script = "SELECT id\_ntn as КОД, name\_produkt as Продукт, id\_ksu as КСУ, requested as Запрошено, released as Реализовано, fio\_worker as Работник FROM view\_ntn\_two;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

}

catch (FormatException)

{

MessageBox.Show("Неверный формат.");

}

catch (ArgumentOutOfRangeException)

{

MessageBox.Show("Значение вне границ.");

}

catch (MySql.Data.MySqlClient.MySqlException)

{

MessageBox.Show("Значение вне границ.");

}

catch (System.OverflowException)

{

MessageBox.Show("Используеться слишком большое число.");

}

}

// Загрузка данных из бд в combobox

private new void Select()

{

connection.Open();

string query = "Select `id\_ntn` from `ntn\_one`;";

MySqlCommand nameSelectionQuery = new MySqlCommand(query, connection);

var reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox1.Items.Add(reader["id\_ntn"]);

}

connection.Close();

connection.Open();

query = "Select `name\_produkt` from `produkt\_help`;";

nameSelectionQuery = new MySqlCommand(query, connection);

reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox2.Items.Add(reader["name\_produkt"]);

}

connection.Close();

connection.Open();

query = "Select `id\_ksu` from `ksu\_one`;";

nameSelectionQuery = new MySqlCommand(query, connection);

reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox3.Items.Add(reader["id\_ksu"]);

}

connection.Close();

connection.Open();

query = "Select `fio\_worker` from `worker\_help`;";

nameSelectionQuery = new MySqlCommand(query, connection);

reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox4.Items.Add(reader["fio\_worker"]);

}

connection.Close();

}

// Удаление выделенной строки

private void DeleteProductButton\_Click(object sender, EventArgs e)

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

if (advancedDataGridView1.SelectedRows.Count > 0)

{

\_advancedDataGridViewSelectedRow = advancedDataGridView1.SelectedRows[0];

string name = \_advancedDataGridViewSelectedRow.Cells[3].Value.ToString();

string sql = string.Format($"Delete from ntn\_two where requested = '{name}';");

using (MySqlCommand cmdSel = new MySqlCommand(sql, connection))

{

try

{

cmdSel.ExecuteNonQuery();

advancedDataGridView1.Refresh();

}

catch (MySqlException ex)

{

Exception error = new Exception("Уже удалено", ex);

throw error;

}

}

}

string script = "SELECT id\_ntn as КОД, name\_produkt as Продукт, id\_ksu as КСУ, requested as Запрошено, released as Реализовано, fio\_worker as Работник FROM view\_ntn\_two;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

connection.Close();

}

}

}

Листинг 6 – Модуль ttn.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

using Sklad.Spravka;

namespace Sklad

{

public partial class ttn : Form, IRegistration

{

// Подключение бд

private DataGridViewRow \_advancedDataGridViewSelectedRow;

public bool primary { get { return true; } set { } }

MySqlConnection connection;

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

public ttn()

{

InitializeComponent();

connection = new MySqlConnection(connectionString);

}

// Переход на другие формы

private void главныйЭкранToolStripMenuItem\_Click(object sender, EventArgs e)

{

Main form = new Main(this);

form.Visible = true;

this.Visible = false;

}

private void требованиенакладнаяToolStripMenuItem\_Click(object sender, EventArgs e)

{

// Код позволяет загрузить данные в таблицу

private void ttn\_Load(object sender, EventArgs e)

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

string script = "SELECT id\_input\_order as КОД, date\_ttn as Дата, name\_produkt as Продукт, input\_produkt as Поступило, name\_provider as Поставщик, id\_ksu as КСУ, fio\_worker as Работник FROM view\_ttn;";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

Select();

connection.Close();

dateTimePicker1.CustomFormat = "YYYY-MM-DD";

dateTimePicker1.Value = DateTime.Today;

}

// Добавление данных в бд

private void ProductAddingButton\_Click(object sender, EventArgs e)

{

connection.Open();

Insetrt();

connection.Close();

}

private void Insetrt()

{

try

{

string unitProductSelectionQuery = $"Select `id\_produkt` from `produkt\_help`" +

$" where `name\_produkt` = '{comboBox1.SelectedItem}'";

MySqlCommand selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex = Convert.ToInt32(selectionQuery.ExecuteScalar());

unitProductSelectionQuery = $"Select `id\_provider` from `provider\_help`" +

$" where `name\_provider` = '{comboBox2.SelectedItem}'";

selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex2 = Convert.ToInt32(selectionQuery.ExecuteScalar());

unitProductSelectionQuery = $"Select `id\_ksu` from `ksu\_one`" +

$" where `id\_ksu` = '{comboBox3.SelectedItem}'";

selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex3 = Convert.ToInt32(selectionQuery.ExecuteScalar());

unitProductSelectionQuery = $"Select `id\_worker` from `worker\_help`" +

$" where `fio\_worker` = '{comboBox4.SelectedItem}'";

selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex4 = Convert.ToInt32(selectionQuery.ExecuteScalar());

string query = $"insert into `ttn` (date\_ttn, id\_produkt, input\_produkt, id\_provider, id\_ksu, id\_worker) " +

$"values ('{dateTimePicker1.Value.ToString("yyyy-MM-dd")}', {unitIndex}, {Convert.ToSingle(NewProductName.Text)}, {unitIndex2}, {unitIndex3}, {unitIndex4});";

MySqlCommand insertProductCommand = new MySqlCommand(query, connection);

insertProductCommand.ExecuteNonQuery();

string script = "SELECT id\_input\_order as КОД, date\_ttn as Дата, name\_produkt as Продукт, input\_produkt as Поступило, name\_provider as Поставщик, id\_ksu as КСУ, fio\_worker as Работник FROM view\_ttn;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

}

catch (FormatException)

{

MessageBox.Show("Неверный формат.");

}

catch (ArgumentOutOfRangeException)

{

MessageBox.Show("Значение вне границ.");

}

catch (MySql.Data.MySqlClient.MySqlException)

{

MessageBox.Show("Значение вне границ.");

}

catch (System.OverflowException)

{

MessageBox.Show("Используеться слишком большое число.");

}

}

// Загрузка данных из бд в combobox

private new void Select()

{

connection.Open();

string query = "Select `name\_produkt` from `produkt\_help`;";

MySqlCommand nameSelectionQuery = new MySqlCommand(query, connection);

var reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox1.Items.Add(reader["name\_produkt"]);

}

connection.Close();

connection.Open();

query = "Select `name\_provider` from `provider\_help`;";

nameSelectionQuery = new MySqlCommand(query, connection);

reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox2.Items.Add(reader["name\_provider"]);

}

connection.Close();

connection.Open();

query = "Select `id\_ksu` from `ksu\_one`;";

nameSelectionQuery = new MySqlCommand(query, connection);

reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox3.Items.Add(reader["id\_ksu"]);

}

connection.Close();

connection.Open();

query = "Select `fio\_worker` from `worker\_help`;";

nameSelectionQuery = new MySqlCommand(query, connection);

reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox4.Items.Add(reader["fio\_worker"]);

}

connection.Close();

}

// Удаление выделенной строки

private void DeleteProductButton\_Click(object sender, EventArgs e)

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

if (advancedDataGridView1.SelectedRows.Count > 0)

{

\_advancedDataGridViewSelectedRow = advancedDataGridView1.SelectedRows[0];

string name = \_advancedDataGridViewSelectedRow.Cells[0].Value.ToString();

string sql = string.Format($"Delete from ttn where id\_input\_order = '{name}';");

using (MySqlCommand cmdSel = new MySqlCommand(sql, connection))

{

try

{

cmdSel.ExecuteNonQuery();

advancedDataGridView1.Refresh();

}

catch (MySqlException ex)

{

Exception error = new Exception("Уже удалено", ex);

throw error;

}

}

}

string script = "SELECT id\_input\_order as КОД, date\_ttn as Дата, name\_produkt as Продукт, input\_produkt as Поступило, name\_provider as Поставщик, id\_ksu as КСУ, fio\_worker as Работник FROM view\_ttn;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

connection.Close();

}

}

}

Листинг 7 – Модуль client\_help.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Sklad.Spravka

{

public partial class client\_help : Form, IRegistration

{

private DataGridViewRow \_advancedDataGridViewSelectedRow;

// Подключение бд

public bool primary { get { return true; } set { } }

MySqlConnection connection;

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

public client\_help()

{

InitializeComponent();

connection = new MySqlConnection(connectionString);

}

// Код позволяет загрузить данные в таблицу

private void client\_help\_Load(object sender, EventArgs e)

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

string script = "SELECT id\_client as КОД, name\_client as Клиент, country\_name\_min as Страна, address\_client as Адрес FROM view\_client\_help;";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

Select();

connection.Close();

}

// Переход на другие формы

private void главныйЭкранToolStripMenuItem\_Click(object sender, EventArgs e)

{

Main form = new Main(this);

form.Visible = true;

this.Visible = false;

}

private void Insetrt()

{

try

{

string unitProductSelectionQuery = $"select `id\_country` from `country`" +

$" where `country\_name\_max` = '{comboBox1.SelectedItem}'";

MySqlCommand selectionQuery = new MySqlCommand(unitProductSelectionQuery, connection);

int unitIndex = Convert.ToInt32(selectionQuery.ExecuteScalar());

string query = $"insert into `client\_help` (name\_client, id\_country, address\_client) " +

$"values ('{NewProductName.Text}', {unitIndex}, '{NewProductPrice.Text}');";

MySqlCommand insertProductCommand = new MySqlCommand(query, connection);

insertProductCommand.ExecuteNonQuery();

string script = "SELECT id\_client as КОД, name\_client as Клиент, country\_name\_min as Страна, address\_client as Адрес FROM view\_client\_help;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

}

catch (FormatException)

{

MessageBox.Show("Неверный формат.");

}

catch (ArgumentOutOfRangeException)

{

MessageBox.Show("Значение вне границ.");

}

catch (MySql.Data.MySqlClient.MySqlException)

{

MessageBox.Show("Значение вне границ.");

}

catch (System.OverflowException)

{

MessageBox.Show("Используеться слишком большое число.");

}

}

// Pагрузка данных из бд в combobox

private new void Select()

{

connection.Open();

string query = "Select `country\_name\_max` from `country`;";

MySqlCommand nameSelectionQuery = new MySqlCommand(query, connection);

var reader = nameSelectionQuery.ExecuteReader();

while (reader.Read())

{

comboBox1.Items.Add(reader["country\_name\_max"]);

}

connection.Close();

}

// Удаление выделенной строки

private void DeleteProductButton\_Click(object sender, EventArgs e)

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

if (advancedDataGridView1.SelectedRows.Count > 0)

{

\_advancedDataGridViewSelectedRow = advancedDataGridView1.SelectedRows[0];

string name = \_advancedDataGridViewSelectedRow.Cells[0].Value.ToString();

string sql = string.Format($"Delete from client\_help where id\_client = '{name}';");

using (MySqlCommand cmdSel = new MySqlCommand(sql, connection))

{

try

{

cmdSel.ExecuteNonQuery();

advancedDataGridView1.Refresh();

}

catch (MySqlException ex)

{

Exception error = new Exception("Уже удалено", ex);

throw error;

}

}

}

string script = "SELECT id\_client as КОД, name\_client as Клиент, country\_name\_min as Страна, address\_client as Адрес FROM view\_client\_help;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

connection.Close();

}

}

}

Листинг 8 – Модуль country.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Sklad.Spravka

{

public partial class country : Form, IRegistration

{

// Подключение бд

private DataGridViewRow \_advancedDataGridViewSelectedRow;

public bool primary { get { return true; } set { } }

MySqlConnection connection;

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

public country()

{

InitializeComponent();

connection = new MySqlConnection(connectionString);

}

// Переход на другие формы

private void главныйЭкранToolStripMenuItem\_Click(object sender, EventArgs e)

{

Main form = new Main(this);

form.Visible = true;

this.Visible = false;

}

// Код позволяет загрузить данные в таблицу

private void country\_Load(object sender, EventArgs e)

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

string script = "SELECT id\_country as КОД, country\_name\_min as Краткое\_Название, country\_name\_max as Название FROM country;";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

Select();

connection.Close();

}

// Добавление данных в бд

private void ProductAddingButton\_Click(object sender, EventArgs e)

{

connection.Open();

Insetrt();

connection.Close();

}

private void Insetrt()

{

try

{

string query = $"insert into `country` (country\_name\_min, country\_name\_max) " +

$"values ('{NewProductName.Text}', '{NewProductPrice.Text}');";

MySqlCommand insertProductCommand = new MySqlCommand(query, connection);

insertProductCommand.ExecuteNonQuery();

string script = "SELECT id\_country as КОД, country\_name\_min as Краткое\_Название, country\_name\_max as Название FROM country;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

}

catch (FormatException)

{

MessageBox.Show("Неверный формат.");

}

catch (ArgumentOutOfRangeException)

{

MessageBox.Show("Значение вне границ.");

}

catch (MySql.Data.MySqlClient.MySqlException)

{

MessageBox.Show("Значение вне границ.");

}

catch (System.OverflowException)

{

MessageBox.Show("Используеться слишком большое число.");

}

}

// Удаление выделенной строки

private void DeleteProductButton\_Click(object sender, EventArgs e)

{

string connectionString = "server=127.0.0.1; port=3306; username=root; password=1234; database=mydb";

MySqlConnection connection = new MySqlConnection(connectionString);

connection.Open();

if (advancedDataGridView1.SelectedRows.Count > 0)

{

\_advancedDataGridViewSelectedRow = advancedDataGridView1.SelectedRows[0];

string name = \_advancedDataGridViewSelectedRow.Cells[0].Value.ToString();

string sql = string.Format($"Delete from country where id\_country = '{name}';");

using (MySqlCommand cmdSel = new MySqlCommand(sql, connection))

{

try

{

cmdSel.ExecuteNonQuery();

advancedDataGridView1.Refresh();

}

catch (MySqlException ex)

{

Exception error = new Exception("Уже удалено", ex);

throw error;

}

}

}

string script = "SELECT id\_country as КОД, country\_name\_min as Краткое\_Название, country\_name\_max as Название FROM country;";

MySqlDataAdapter mySql\_dataAdapter = new MySqlDataAdapter(script, connection);

DataTable table = new DataTable();

mySql\_dataAdapter.Fill(table);

advancedDataGridView1.DataSource = table;

connection.Close();

}

}

}

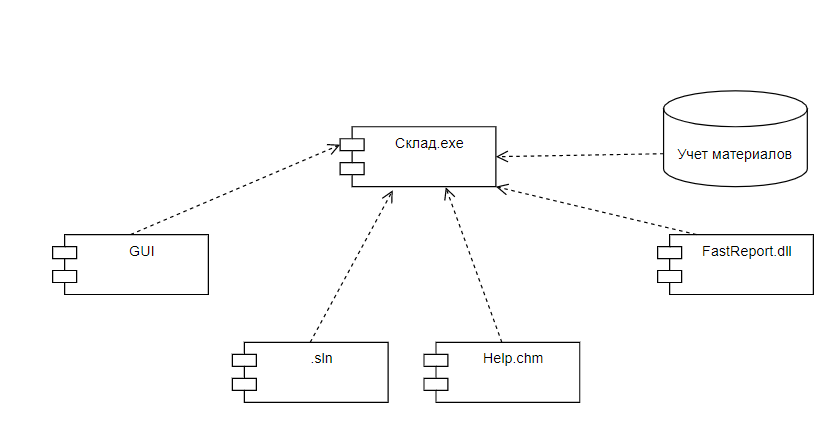
ПРИЛОЖНИЕ Б

**Структурная схема**



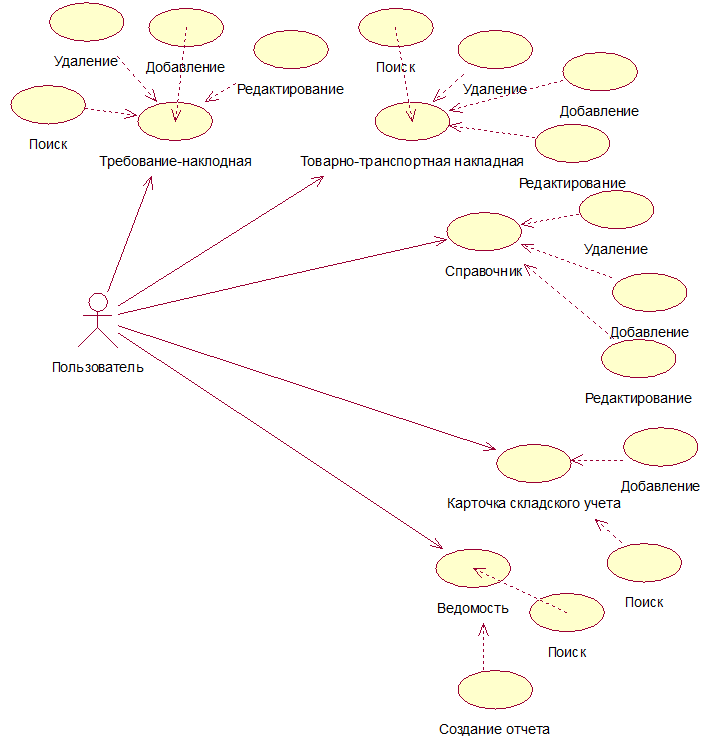
ПРИЛОЖНИЕ В

**Диаграмма компонентов**

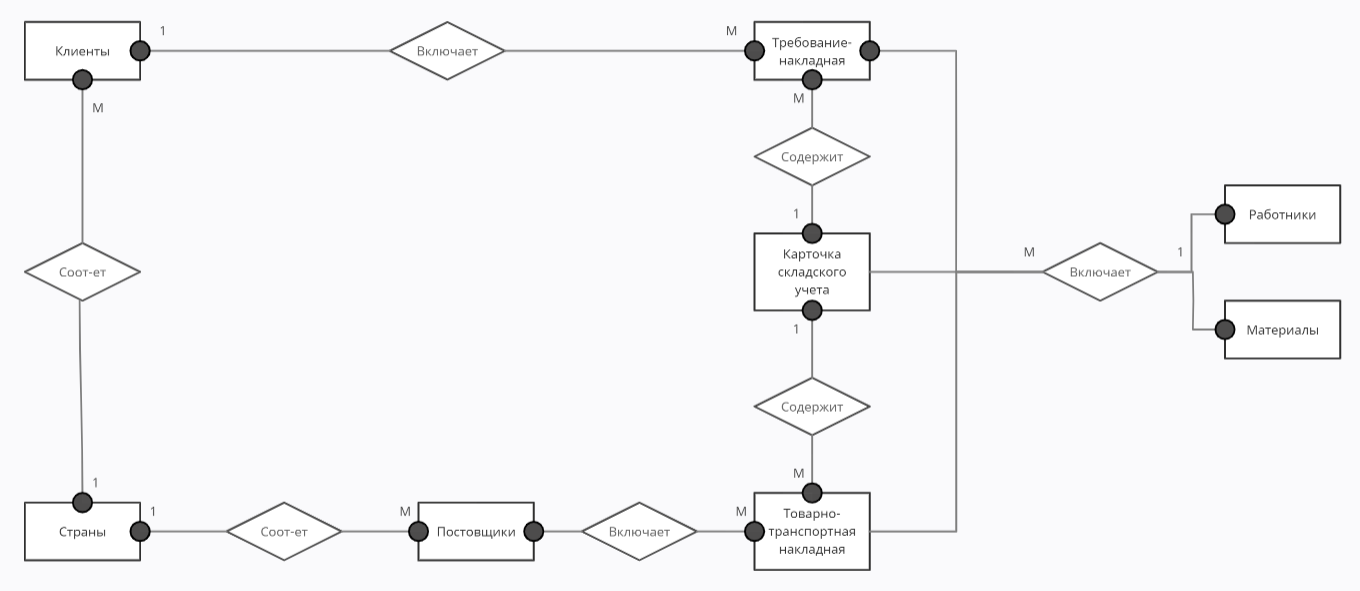


ПРИЛОЖЕНИЕ Г

**Диаграмма вариантов использования**



ПРИЛОЖЕНИЕ Д

**Диаграмма «сущность-связь»**